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

How to Check the Ordering Part Number

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

For More Information

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

About Cypress

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



MB91520 Series

32-bit Microcontroller

FR Family FR81S Hardware Manual

Doc. No. 002-05528 Rev. *B

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

© Cypress Semiconductor Corporation, 2011-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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

Preface

Thank you for your continued use of Cypress semiconductor products.

Read this manual and "MB91520 series Data Sheet" thoroughly before using products in the MB91520 series.

■ Purpose of this manual and intended readers

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

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. 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

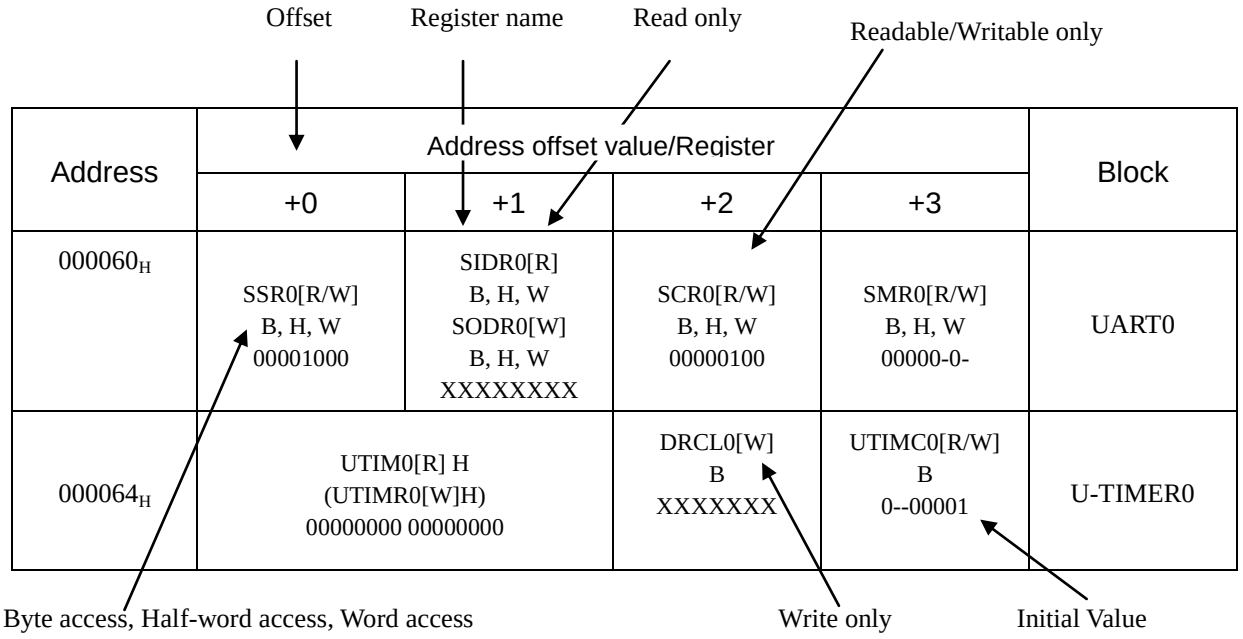
Primary Terms

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.
32-bit peripheral bus	A 32-bit width, low-speed internal bus. It connects to various types of peripherals.
16-bit peripheral bus (R-bus)	A 16-bit width, low-speed internal bus. It connects to various types of peripherals. The 32-bit width access to this bus is divided into 16 bits × 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.

Term	Explanation
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.
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 makes the clock frequency change slightly and oscillates it (= frequency modulation).
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)

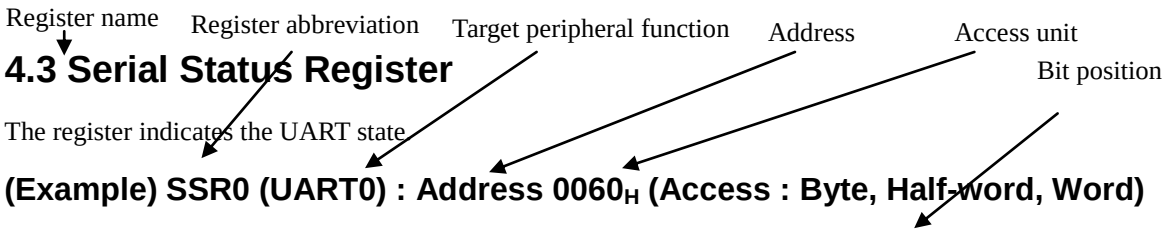
The following explains the address position during access.

- 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.)

Table of Contents

Chapter 1: Overview 42

1. Overview 43
2. Features 43
   2.1. FR81S CPU Core 43
   2.2. Peripheral Functions 44
3. Product Line-up 47
4. Function overview 59
5. Block Diagram 63
6. Memory Map 70
7. Pin Assignment 72
8. Device Package 78
9. Explanation of Pin Functions 85
   9.1. Pins of Each Function 99
10. I/O Circuit Types 116

Chapter 2: Handling The Device 120

1. Handling Precautions 121
   1.1. Precautions for Product Design 121
   1.2. Precautions for Package Mounting 122
   1.3. Precautions for Use Environment 123
2. Handling Device 124
3. Application Notes 126
   3.1. Function Switching of a Multiplexed Port 126
   3.2. Low-power Consumption Mode 126
   3.3. Notes When Writing Data in a Register that Includes the Status Flag 127

Chapter 3: CPU 128

1. Overview 129
2. Features 129
3. CPU Operating Description 130
   3.1. CPU Operating Status 130
      3.1.1. Reset State 131
      3.1.2. Normal Run State 131
      3.1.3. Low-power Consumption State 132
      3.1.4. Debug Run State 132
4. Pipeline Operation 132
5. Floating Point Operation Processing 133
6. Data Structure 133
7. Addressing 134
8. Programming Model 134

8.1. General-purpose Registers, Dedicated Registers, and Floating Point Registers 135
8.2. System Register 136
9. Reset and EIT Processing 136
9.1. Reset 136
9.2. EIT Processing 137
9.3. Vector Table 137
10. Memory Protection Function (MPU) 139
10.1. Overview 139
10.2. List of Registers 139
10.3. Description of Registers 140
10.3.1. MPU Control Register : MPUCR 141
10.3.2. Instruction Access Protection Violation Address Register : IPVAR 144
10.3.3. Instruction Access Protection Violation Status Register : IPVSR 144
10.3.4. Data Access Protection Violation Address Register :DPVAR 146
10.3.5. Data Access Protection Violation Status Register : DPVSR 146
10.3.6. Data Access Error Address Register : DEAR 148
10.3.7. Data Access Error Status Register : DESR 148
10.3.8. Protection Area Base Address Register 0 to 7 : PABR0 to PABR7 150
10.3.9. Protection Area Control Register 0 to 7 : PACR0 to PACR7 151
10.4. Operations of Memory Protection Function 154
10.4.1. Setting Up Memory Protection Areas 155
10.4.2. Instruction Access Protection Violation 155
10.4.3. Data Access Protection Violation 155
10.4.4. Data Access Errors 156
10.4.5. Memory Protection Operation by Delay Slot 156
10.4.6. DEAR and DESR Update 157
10.4.7. Notes 157

Chapter 4: Operation Mode 158

1. Overview 159
2. Features 159
3. Configuration 159
4. Register 160
4.1. Bus Mode Register : BMODR (Bus MODe Register) 160
5. Operation 160
5.1. MD0, MD1, P006 Pins Settings 161
5.2. Fetching the Operation Mode 161
5.3. Explanation of Each Operation Mode 163
5.3.1. User Mode 163
5.3.2. Serial Writer Mode 163

Chapter 5: Clock 167

1. Overview 168
2. Features 170
3. Configuration 171
4. Registers 175
4.1. Division Configuration Register 0 : DIVR0 (DIVision clock configuration Register 0) 177
4.2. Division Configuration Register 1 : DIVR1 (DIVision clock configuration Register 1) 178
4.3. Division Configuration Register 2 : DIVR2 (DIVision clock configuration Register 2) 179
4.4. Clock Source Selection Register : CSELR (Clock source SELection Register) 180
4.5. Clock Source Monitor Register : CMONR (Clock source MONitor Register) 182

4.6. Main Timer Control Register : MTMCR (Main clock TiMer Control Register) 184
4.7. Sub Timer Control Register : STMCR (Sub clock TiMer Control Register) 187
4.8. PLL Setting Register : PLLCR (PLL Configuration Register) 189
4.9. Clock Stabilization Selection Register : CSTBR (Clock STaBilization selection Register) 191
4.10. PLL Oscillation Timer Control Register : PTMCR (PLL clock osc TiMer Control Register) 193
4.11. PLL/SSCG Clock Selection Register : CCPSSELR (CCtl Pll/Sscg clock SELection Register) 194
4.12. PLL/SSCG Output Clock Division Setting Register : CCPSDIVR (CCtl Pll/Sscg clock DIVision Register) 195
4.13. PLL Feedback Division Setting Register : CCPLLFBR (CCtl PLL FB clock division Register) 197
4.14. SSCG Feedback Division Setting Register 0 : CCSSFBR0 (CCtl SScg FB clock division Register 0) 198
4.15. SSCG Feedback Division Setting Register 1 : CCSSFBR1 (CCtl SScg FB clock division Register 1) 199
4.16. SSCG Configuration Setting Register 0 : CCSSCCR0 (CCtl SSCg Config. Register 0) 200
4.17. SSCG Configuration Setting Register 1 : CCSSCCR1 (CCtl SSCg Config. Register 1) 202
4.18. Clock Gear Configuration Setting Register 0 : CCCGRCR0 (CCtl Clock GeaR Config. Register 0) 203
4.19. Clock Gear Configuration Setting Register 1 : CCCGRCR1 (CCtl Clock GeaR Config. Register 1) 204
4.20. Clock Gear Configuration Setting Register 2 : CCCGRCR2 (CCtl Clock GeaR Config. Register 2) 205
4.21. RTC/PMU Clock Selection Register : CCRTSELR (CCtl RTc pmu clock SELection Register) 206
4.22. PMU Clock Division Setting Register 0 : CCPMUCR0 (CCtl PMU Clock division Register 0) 208
4.23. PMU Clock Division Setting Register 1 : CCPMUCR1 (CCtl PMU Clock division Register 1) 209
4.24. Sync/Async Control Register : SACR 210
4.25. Peripheral Interface Clock Divider : PICD 211
5. Operation 212
5.1. Oscillation Control 212
5.1.1. Main Clock (MCLK) 212
5.1.2. Sub Clock (SBCLK) 213
5.1.3. PLL/SSCG Clock (PLLSSCLK) 213
5.1.4. Limitations when PLL/SSCG Clock is used 215
5.2. Oscillation Stabilization Wait 216
5.2.1. Conditions for Generating Stabilization Wait Time 216
5.2.2. Selecting Stabilization Wait Time 217
5.2.3. End of the Stabilization Wait Time 217
5.3. Selecting the Source Clock (SRCCLK) 218
5.3.1. Selecting the Source Clock at the Time of Initialization 218
5.3.2. Procedure of switching the source clock 218
5.4. Timer 223
5.4.1. Main Clock Oscillation Stabilization Wait Timer (Main Timer) 223
5.4.2. Sub Clock Oscillation Stabilization Wait Timer (Sub Timer) 224
5.4.3. PLL/SSCG Clock Oscillation Stabilization Wait timer (PLL Timer) 224
5.4.4. Setting 224
5.4.5. Procedure for Setting the Timer Interrupt 225
5.4.6. Timer Operations 226
5.4.7. Watch Mode and Timer Interrupt 226
5.5. Notes when Clocks Conflict 227
5.6. The Clock Gear Circuit 227
5.6.1. Procedure of Gear Up 227
5.6.2. Procedure of Gear Down 227

5.7. Operations during MDI Communications 228
5.8. About PMU clock (PMUCLK) 228

Chapter 6: Clock Reset State Transitions 231

1. Overview 232
2. Device States and Transitions 232
2.1. Diagram of State Transitions 232
2.2. Explanation of Each States 235
2.3. Priority of State Transition Requests 236
3. Device State and Regulator Mode Corresponding to those States 237

Chapter 7: Reset 239

1. Overview 240
2. Features 240
3. Configuration 240
4. Registers 245
4.1. Reset Source Register : RSTRR (ReSeT Result Register) 245
4.2. Reset Control Register : RSTCR (ReSeT Control Register) 248
4.3. CPU Abnormal Operation Register : CPUAR (CPU Abnormal operation Register) 249
4.4. PMU Status Register : PMUSTR (Power Management Unit STatus Register) 251
5. Operation 252
5.1. Reset Level 253
5.1.1. Initialize Reset (INIT) 253
5.1.2. Reset (RST) 253
5.2. Reset Factor 254
5.2.1. Power-on Reset 254
5.2.2. RSTX Pin Input 254
5.2.3. Watchdog Reset 0 254
5.2.4. Watchdog Reset 1 254
5.2.5. External Low-Voltage Detection Reset 255
5.2.6. Illegal Standby Mode Transition Detection Reset 255
5.2.7. Internal Low-Voltage Detection Reset 255
5.2.8. Flash Security Violation Reset 255
5.2.9. Software Reset (RSTCR:SRST) 256
5.2.10. Recovery from Standby (Power Shutdown) 256
5.3. Reset Acceptance 256
5.3.1. Generation of Reset Request 256
5.3.2. Acceptance of Reset Request 257
5.3.3. Reset Issue Delay Counter 257
5.3.4. Irregular Reset 257
5.4. Reset Issue 257
5.4.1. Super Initialize Reset (SINIT) 258
5.4.2. Initialize Reset (INIT) 260
5.4.3. Reset (RST) 261
5.5. Reset Sequence 262
5.5.1. Reset Cycle 264
5.5.2. Reset Release 264
5.5.3. Operating Mode Fix 264
5.5.4. Transition of Bus Control 264
5.5.5. Reset Vector Fetch 264
5.5.6. Reset and Forced Break 265

5.6. Notes 265

Chapter 8: DMA Controller (DMAC) 266

1. Overview 267
2. Features 267
3. Configuration 268
4. Registers 269
   - 4.1. DMA Control Register: DMACR (DMA Control Register) 272
   - 4.2. DMA Channel Control Register 0 to 15: DCCR0 to 15 (DMA Channel Control Register 0 to 15) 274
   - 4.3. DMA Channel Status Register 0 to 15 : DCSR0 to 15: (DMA Channel Status Register 0 to 15) 280
   - 4.4. DMA Transfer Count Register 0 to 15 : DTCR0 to 15: (DMA Transfer Count Register 0 to 15) ... 282
   - 4.5. DMA Transfer Source Register 0 to 15 : DSAR0 to 15: (DMA Source Address Register 0 to 15) . 283
   - 4.6. DMA Transfer Destination Register 0 to 15 : DDAR0 to 15 (DMA Destination Address Register 0 to 15) 284
   - 4.7. DMA Transfer Suppression NMI Flag Register : DNMIR (DMA-halt by NMI Register) 285
   - 4.8. DMA Transfer Suppression Level Register : DILVR (DMA-halt by Interrupt Level Register) 286
5. Operation 287
   - 5.1. Configuration 288
     - 5.1.1. Common Items for All Channels 288
     - 5.1.2. Separate Items for Each Channel 288
     - 5.1.3. Operations 292
6. DMA Usage Examples 306

Chapter 9: Generation And Clearing Of DMA Transfer Requests 308

1. Overview 309
2. Features 309
   - 2.1. Transfer Request Generation Setting 309
   - 2.2. Interrupt Clearing Setting 309
3. Configuration 310
4. Registers 310
   - 4.1. DMA Request Clear Register 0 : ICSEL0 (Interrupt Clear SELect register 0) 312
   - 4.2. DMA Request Clear Register 1 : ICSEL1 (Interrupt Clear SELect register 1) 313
   - 4.3. DMA Request Clear Register 2 : ICSEL2 (Interrupt Clear SELect register 2) 314
   - 4.4. DMA Request Clear Register 3 : ICSEL3 (Interrupt Clear SELect register 3) 314
   - 4.5. DMA Request Clear Register 5 : ICSEL5 (Interrupt Clear SELect register 5) 315
   - 4.6. DMA Request Clear Register 6 : ICSEL6 (Interrupt Clear SELect register 6) 316
   - 4.7. DMA Request Clear Register 7 : ICSEL7 (Interrupt Clear SELect register 7) 317
   - 4.8. DMA Request Clear Register 8 : ICSEL8 (Interrupt Clear SELect register 8) 318
   - 4.9. DMA Request Clear Register 9 : ICSEL9 (Interrupt Clear SELect register 9) 318
   - 4.10. DMA Request Clear Register 10 : ICSEL10 (Interrupt Clear SELect register 10) 319
   - 4.11. DMA Request Clear Register 11 : ICSEL11 (Interrupt Clear SELect register 11) 319
   - 4.12. DMA Request Clear Register 13 : ICSEL13 (Interrupt Clear SELect register 13) 320
   - 4.13. DMA Request Clear Register 14 : ICSEL14 (Interrupt Clear SELect register 14) 321
   - 4.14. DMA Request Clear Register 15 : ICSEL15 (Interrupt Clear SELect register 15) 321
   - 4.15. DMA Request Clear Register 16 : ICSEL16 (Interrupt Clear SELect register 16) 322
   - 4.16. DMA Request Clear Register 17 : ICSEL17 (Interrupt Clear SELect register 17) 323
   - 4.17. DMA Request Clear Register 18 : ICSEL18 (Interrupt Clear SELect register 18) 324
   - 4.18. DMA Request Clear Register 19 : ICSEL19 (Interrupt Clear SELect register 19) 325
   - 4.19. DMA Request Clear Register 20 : ICSEL20 (Interrupt Clear SELect register 20) 326
   - 4.20. DMA Request Clear Register 21 : ICSEL21 (Interrupt Clear SELect register 21) 326
   - 4.21. DMA Request Clear Register 22 : ICSEL22 (Interrupt Clear SELect register 22) 327

4.22. DMA Request Clear Register 23 : ICSEL23 (Interrupt Clear SELect register 23) 328
4.23. DMA Request Clear Register 24 : ICSEL24 (Interrupt Clear SELect register 24) 328
4.24. DMA Request Clear Register 25 : ICSEL25 (Interrupt Clear SELect register 25) 329
4.25. DMA Request Clear Register 26 : ICSEL26 (Interrupt Clear SELect register 26) 330
4.26. DMA Request Clear Register 27 : ICSEL27 (Interrupt Clear SELect register 27) 331
4.27. IO Transfer Request Setting Register 0 to 15 : IORR0 to 15 (IO triggered DMA Request Register for ch. 0 to 15) 331
5. Operation 333
5.1. Configuration 333
5.2. Notes 333

Chapter 10: FixedVector Function 334

1. Overview 335
2. Features 335
3. Configuration 335
4. Registers 335
5. Operation 335
5.1. Operation After Reset Released 336
5.2. Usage 336
6. Notes 336

Chapter 11: I/O Ports 337

1. Overview 338
2. Features 338
3. Configuration 339
4. Registers 339
4.1. Port Data Register 00 to 19 : PDR00 to 19 (Port Data Register 00 to 19) 342
4.2. Data Direction Register 00 to 19 : DDR00 to 19 (Data Direction Register 00 to 19) 343
4.3. Port Function Register 00 to 19 : PFR00 to 19 (Port Function Register 00 to 19) 344
4.4. Input Data Direct Register 00 to 19 : PDDR00 to 19 (Port Data Direct Register 00 to 19) 345
4.5. Port Pull-up/down Enable Register 00 to 19 : PPER00 to 19 (Port Pull-up/down Enable Register 00 to 19) 346
4.6. Extended Port Function Register 00 to 88 : EPFR00 to 88 (Extended Port Function Register 00 to 88) 347
4.6.1. Extended Port Function Register 00, 01, 56 : EPFR00, EPFR01, EPFR56 347
4.6.2. Extended Port Function Register 02 to 05, 57 to 60 : EPFR02 to 05, 57 to 60 349
4.6.3. Extended Port Function Register 06 to 09, 33 to 36, 61 to 64 : EPFR06 to 09, 33 to 36, 61 to 64 352
4.6.4. Extended Port Function Register 10 to 15, 45, 71 to 78 : EPFR10 to 15, 45, 71 to 78 356
4.6.5. Extended Port Function Register 79, 80 : EPFR79, 80 361
4.6.6. Extended Port Function Register 86 : EPFR86 362
4.6.7. Extended Port Function Register 26 : EPFR26 363
4.6.8. Extended Port Function Register 27 : EPFR27 364
4.6.9. Extended Port Function Register 28 : EPFR28 365
4.6.10. Extended Port Function Register 29, 81, 82 : EPFR29, 81, 82 366
4.6.11. Extended Port Function Register 83 : EPFR83 367
4.6.12. Extended Port Function Register 42 : EPFR42 368
4.6.13. Extended Port Function Register 43, 44 : EPFR43, 44 368
4.6.14. Extended Port Function Register 48 to 51 : EPFR48 to 51 369
4.6.15. Extended Port Function Register 65 to 70 : EPFR65 to 70 370
4.6.16. Extended Port Function Register 84, 85 : EPFR84, 85 372

4.6.17. Extended Port Function Register 87 : EPFR87 373
4.6.18. Extended Port Function Register 88 : EPFR88 374
4.7. Port Input Enable Register: PORTEN (PORT ENable register) 375
4.8. KEY CoDe Register : KEYCDR 375
5. Operation 377
5.1. Pin I/O Assignment 377
5.1.1. Peripheral I/O (Bidirectional) Pin Assignment 379
5.1.2. Peripheral Input Assignment 380
5.1.3. Peripheral Output Assignment 381
5.1.4. External Bus Assignment 382
5.1.5. Port Function (Input) Assignment 383
5.1.6. Port Function (Output) Assignment 384
5.1.7. A/D Converter Input Assignment 385
5.1.8. D/A converter output assignment 385
5.2. EPFR setting priority 386
5.3. Notes on Input I/O Relocation Setting 386
5.4. Noise Filter 386
5.5. Input blocked by GPORTEN 387
5.6. Notes on Pins with the A/D Converter Function 387
5.7. Setting when Using the Base Timer TIOA1 Pin 387
5.8. Key Code Register Function Settings 388
5.9. Operation at Wake Up from Power Shutdown 389
5.10. Notes on switching the I/O port function 389
5.11. Input blocked when specific peripheral functions are used 389

Chapter 12: Interrupt Control (Interrupt Controller) 390

1. Overview 391
2. Features 391
3. Configuration 392
4. Registers 393
4.1. Interrupt Control Registers 00 to 47 : ICR00 to ICR47 (Interrupt Control Register 00 to 47) 394
5. Operation 395
5.1. Setting 395
5.2. Starting 395
5.3. Determining Priorities 396
5.4. Recovering From Stop Mode 396
5.5. Recovering From Standby Mode (Power shutdown) 396

Chapter 13: External Interrupt Input 397

1. Overview 398
2. Features 398
3. Configuration 398
4. Registers 399
4.1. External Interrupt Factor Register 0/1 : EIRR0/EIRR1 (External Interrupt Request Register 0/1) . 400
4.2. External Interrupt Enable Register 0/1 : ENIR0/ENIR1 (ENable Interrupt request Register 0/1) ... 401
4.3. External Interrupt Request Level Register 0/1 : ELVR0/ELVR1 (External interrupt LeVel Register 0/1) 402
5. Operation 403
6. Setting 405
7. Q&A 406
8. Notes 407

Chapter 14: NMI Input 408

1. Overview 409
2. Features 409
3. Configuration 409
4. Register 409
5. Operation 410
6. Usage Example 411

Chapter 15: Delay Interrupt 412

1. Overview 413
2. Features 413
3. Configuration 413
4. Registers 413
5. Operation 414
6. Restrictions 414

Chapter 16: Interrupt Request Batch Read 415

1. Overview 416
2. Features 416
3. Configuration 416
4. Registers 417
   - 4.1. Interrupt Request Batch Read Register 0 upper-order : IRPR0H (Interrupt Request Peripheral Read register 0H) 418
   - 4.2. Interrupt Request Batch Read Register 0 lower-order : IRPR0L (Interrupt Request Peripheral Read register 0L) 419
   - 4.3. Interrupt Request Batch Read Register 1 upper-order : IRPR1H (Interrupt Request Peripheral Read register 1H) 420
   - 4.4. Interrupt Request Batch Read Register 1 lower-order : IRPR1L (Interrupt Request Peripheral Read register 1L) 420
   - 4.5. Interrupt Request Batch Read Register 3 upper-order : IRPR3H (Interrupt Request Peripheral Read register 3H) 421
   - 4.6. Interrupt Request Batch Read Register 3 lower-order : IRPR3L (Interrupt Request Peripheral Read register 3L) 422
   - 4.7. Interrupt Request Batch Read Register 4 upper-order : IRPR4H (Interrupt Request Peripheral Read register 4H) 422
   - 4.8. Interrupt Request Batch Read Register 4 lower-order : IRPR4L (Interrupt Request Peripheral Read register 4L) 423
   - 4.9. Interrupt Request Batch Read Register 5 upper-order : IRPR5H (Interrupt Request Peripheral Read register 5H) 424
   - 4.10. Interrupt Request Batch Read Register 5 lower-order : IRPR5L (Interrupt Request Peripheral Read register 5L) 424
   - 4.11. Interrupt Request Batch Read Register 6 upper-order : IRPR6H (Interrupt Request Peripheral Read register 6H) 425
   - 4.12. Interrupt Request Batch Read Register 6 lower-order : IRPR6L (Interrupt Request Peripheral Read register 6L) 425
   - 4.13. Interrupt Request Batch Read Register 7 upper-order : IRPR7H (Interrupt Request Peripheral Read register 7H) 426
   - 4.14. Interrupt Request Batch Read Register 7 lower-order : IRPR7L (Interrupt Request Peripheral Read register 7L) 427
   - 4.15. Interrupt Request Batch Read Register 8 upper-order : IRPR8H (Interrupt Request Peripheral Read register 8H) 427

4.16. Interrupt Request Batch Read Register 8 lower-order : IRPR8L (Interrupt Request Peripheral Read register 8L) 428
4.17. Interrupt Request Batch Read Register 9 upper-order : IRPR9H (Interrupt Request Peripheral Read register 9H) 428
4.18. Interrupt Request Batch Read Register 9 lower-order : IRPR9L (Interrupt Request Peripheral Read register 9L) 429
4.19. Interrupt Request Batch Read Register 10 upper-order : IRPR10H (Interrupt Request Peripheral Read register 10H) 429
4.20. Interrupt Request Batch Read Register 10 lower-order : IRPR10L (Interrupt Request Peripheral Read register 10L) 430
4.21. Interrupt Request Batch Read Register 11 upper-order : IRPR11H (Interrupt Request Peripheral Read register 11H) 430
4.22. Interrupt Request Batch Read Register 11 lower-order : IRPR11L (Interrupt Request Peripheral Read register 11L) 431
4.23. Interrupt Request Batch Read Register 12 upper-order : IRPR12H (Interrupt Request Peripheral Read register 12H) 431
4.24. Interrupt Request Batch Read Register 12 lower-order : IRPR12L (Interrupt Request Peripheral Read register 12L) 432
4.25. Interrupt Request Batch Read Register 13 upper-order : IRPR13H (Interrupt Request Peripheral Read register 13H) 432
4.26. Interrupt Request Batch Read Register 13 lower-order : IRPR13L (Interrupt Request Peripheral Read register 13L) 433
4.27. Interrupt Request Batch Read Register 14 upper-order : IRPR14H (Interrupt Request Peripheral Read register 14H) 433
4.28. Interrupt Request Batch Read Register 14 lower-order : IRPR14L (Interrupt Request Peripheral Read register 14L) 434
4.29. Interrupt Request Batch Read Register 15 upper-order : IRPR15H (Interrupt Request Peripheral Read register 15H) 435
4.30. Interrupt Request Batch Read Register 15 lower-order : IRPR15L (Interrupt Request Peripheral Read register 15L) 436
5. Operation 436

Chapter 17: PPG 437

1. Overview 438
2. Features 438
3. Configuration 443
4. Registers 444
4.1. PPG Control Status Register : PCN0 to PCN 47 455
4.2. PPG Cycle Setting Register : PCSR0 to PCSR47 459
4.3. PPG Duty Setting Register : PDUT0 to PDUT47 460
4.4. PPG Timer Register : PTMR0 to PTMR47 461
4.5. PPG Control Status Register2 : PCN200 to PCN247 462
4.6. Start Delay Value Setting Register : PSDR0 to PSDR47 464
4.7. Timing Point Capture Value Setting Register : PTPC0 to PTPC47 465
4.8. PPG Communication Mode High Format Cycle Setting Register : PHCSR0 to PHCSR3 466
4.9. PPG Communication Mode Low Format Cycle Setting Register : PLCSR0 to PLCSR3 467
4.10. PPG Communication Mode High Format Duty Setting Register : PHDUT0 to PHDUT3 468
4.11. PPG Communication Mode Low Format Duty Setting Register : PLDUT0 to PLDUT3 469
4.12. PPG Communication Mode Data Setting Register : PCMDDT0 to PCMDDT3 470
4.13. PPG Communication Mode Data Bit Length Setting Register : PCMDWD0 to PCMDWD3 470
4.14. GATE Function Control Register : GATEC0, GATEC2, GATEC4 472

4.15. General-purpose Trigger Selection Register : GTRS0 to GTRS23 473
4.16. General-purpose Trigger Setting Register : GTREN0 to GTREN2 475
5. Operation 477
5.1. PWM Operation (Normal Wave Form) 477
5.2. PWM Operation (Center Aligned Wave Form Selected) 479
5.3. One-shot Operation (Normal Wave Form Selected) 481
5.4. One-shot Operation (Center Aligned Wave Form Selected) 483
5.5. Restart Operation 485
5.6. GATE Operation 486
5.7. Start Delay Mode Operation (PWM Normal Wave Form Selected) 487
5.8. Timing Point Capture Mode Operation (PWM Normal Wave Form Selected) 489
5.9. PPG Communication Mode Operation 492
5.10. PPG Communication Activation 493
5.11. PPG Communication Operation 494
5.12. PPG Communication Forced Stop and Restart operation 498
5.13. PPG Output Pulse Polarity Selection 500
5.14. Interrupt 502
6. Notes 503

Chapter 18: Watchdog Timer 508

1. Overview 509
2. Features 509
2.1. Watchdog Timer 0 (Software Watchdog) 509
2.2. Watchdog Timer 1 (Hardware Watchdog) 510
3. Configuration 511
4. Registers 512
4.1. Watchdog Timer 0 Control Register : WDTCR0 (WatchDog Timer 0 Configuration Register) 512
4.2. Watchdog Timer 0 Clear Register : WDTCPR0 (WatchDog Timer Clear Pattern Register 0) 514
4.3. Watchdog Timer 0 Extended Configuration Register : WDTECR0 (Watchdog Timer Extended Configuration Register 0) 514
4.4. Watchdog Timer 1 Cycle information Register : WDTCR1 (WatchDog Timer Cycle information Register 1) 516
4.5. Watchdog Timer 1 Clear Register : WDTCPR1 (WatchDog Timer Clear Pattern Register 1) 516
5. Operation 517
5.1. Software Watchdog Function 517
5.1.1. Settings 517
5.1.2. Activation 518
5.1.3. Operation 518
5.2. Hardware Watchdog Function 519
5.2.1. Settings 519
5.2.2. Activation 519
5.2.3. Operation 520
6. Usage Example 520

Chapter 19: Base Timer 522

1. Overview 523
2. Features 523
2.1. 16/32-bit Reload Timer 523
2.2. 16-bit PWM Timer 524
2.3. 16/32-bit PWC Timer 525
2.4. 16-bit PPG Timer 526

3. Configuration 527
4. Registers 528
4.1. Common Registers 529
4.1.1. Timer Registers 0, 1 : BTxTMR (Base Timer 0/1 TiMer Register) 529
4.1.2. Timer Control Registers 0, 1 : BTxTMCR (Base Timer 0/1 TiMer Control Register) 530
4.1.3. I/O Selection Register : BTSEL01 (Base Timer SELect register ch.0 and ch.1) 536
4.1.4. Simultaneous Software Activation Register : BTSSSR (Base Timer Software Synchronous Start Register) 537
4.2. Registers for 16/32-bit Reload Timer 538
4.2.1. Status Control Registers 0, 1 : BTxSTC (Base Timer 0/1 STatus Control) 538
4.2.2. Cycle Setting Registers 0, 1 : BTxPCSR (Base Timer 0/1 Pulse Counter Start Register) 539
4.3. Registers for 16-bit PWM Timer 540
4.3.1. Status Control Registers 0, 1 : BTxSTC (Base Timer 0/1 STatus Control) 540
4.3.2. Cycle Setting Registers 0, 1 : BTxPCSR (Base Timer 0/1 Pulse Counter Start Register) 542
4.3.3. Duty Setting Registers 0, 1 : BTxPDUT (Base Timer 0/1 Pulse DuTy register) 543
4.4. Registers for 16-bit PPG Timer 544
4.4.1. Status Control Registers 0, 1 : BTxSTC (Base Timer 0/1 STatus Control) 544
4.4.2. L Width Setting Registers 0, 1 : BTxPRLL (Base Timer 0/1 Pulse Length of "L" register) 545
4.4.3. H Width Setting Registers 0, 1 : BTxPRLH (Base Timer 0/1 Pulse Length of "H" register) 546
4.5. 16/32-bit PWC Timer Register 547
4.5.1. Status Control Registers 0, 1 : BTxSTC (Base Timer 0/1 STatus Control) 547
4.5.2. Data Buffer Registers 0, 1 : BTxDTBF (Base Timer 0/1 DaTa BuFfer register) 548
5. Operation 549
5.1. Selection of Timer Function 549
5.2. I/O Allocation 550
5.3. 32-bit Mode Operation 552
5.3.1. 32-bit Mode Function 553
5.3.2. 32-bit Mode Setting 553
5.3.3. 32-bit Mode Operation 553
5.4. 16/32-bit Reload Timer Operation 554
5.4.1. Overview 555
5.4.2. Operation in Reload Mode 556
5.4.3. Operation in One-Shot Mode 558
5.4.4. 32-bit Timer Mode Operation 560
5.4.5. Interrupts 562
5.4.6. Precautions for Using this Device 563
5.5. 16-bit PWM Timer Operation 563
5.5.1. Overview 564
5.5.2. Operation in Reload Mode 565
5.5.3. Operation in One-Shot Mode 569
5.5.4. Interrupt 571
5.5.5. Precautions for Using this Device 572
5.6. 16-bit PPG Timer Operation 572
5.6.1. Overview 573
5.6.2. Pulse Width Calculation Method 574
5.6.3. Operation in Reload Mode 574
5.6.4. Operation in One-Shot Mode 579
5.6.5. Interrupts 582
5.6.6. Application Notes 582
5.7. 16/32-bit PWC Timer Operation 583
5.7.1. Overview 586

5.7.2. Operation during PWC Measurement 590
5.7.3. 32-bit Timer Mode Operation 593
5.7.4. Interrupt 595
5.7.5. Application Notes 596

Chapter 20: Reload Timer 597

1. Overview 598
2. Features 598
3. Configuration 600
4. Registers 601
4.1. Control Status Register : TMCSR (TiMer Control and Status Register) 603
4.2. 16-bit Timer Register : TMR (16bit TiMer Register) 607
4.3. 16-bit Timer Reload Register A, 16-bit Timer Reload Register B : TMRLRA, TMRLRB(16bit TiMer ReLoad Register A/B) 608
5. Operation 610
5.1. Setting 610
5.1.1. Count Source 611
5.1.2. Timer Underflow cycle 611
5.1.3. Trigger 612
5.1.4. Gate 612
5.1.5. Counter Operation Selection 613
5.1.6. TOUT Pin Level Setting 614
5.2. Operation Procedure 616
5.2.1. Activation 616
5.2.2. Retrigger 618
5.2.3. Underflow/Reload 620
5.2.4. Generation of Interrupt Requests 620
5.2.5. Concurrent Operation of Register Write and Timer Activation 621
5.3. Operations of Each Counter 621
5.3.1. Single One-shot Operation 621
5.3.2. Single Reload Operation 623
5.3.3. Dual One-shot Operation 625
5.3.4. Dual Reload Operation 627
5.3.5. Compare One-shot Operation 629
5.3.6. Compare Reload Operation 632
5.3.7. Capture Mode 635
5.4. Cascade Input 637
5.5. Priority of Concurrent Operations 638
6. Application Note 639
6.1. Single One-shot Timer 640
6.2. Reload Timer 642
6.3. PPG 645
6.4. PWM 648
6.5. PWC 651

Chapter 21: 32-Bit Free-Run Timer 653

1. Overview 654
2. Features 654
2.1. Functions of the 32-bit Free-run Timer 655
2.2. Functions of the Free-run Timer Selector 655
3. Configuration 655

3.1. Configuration Diagram of the 32-bit Free-run Timer 656
3.2. Configuration Diagram of the Free-run Timer Selector 657
4. Registers 658
4.1. Registers of the 32-bit Free-run Timer 659
4.1.1. Timer Control Register (Upper Bit) : TCCSH 659
4.1.2. Timer Control Register (Lower Bit) : TCCSL 660
4.1.3. Compare Clear Register : CPCLR 662
4.1.4. Timer Data Register : TCDT 663
4.2. Registers of the Free-run Timer Selector 663
4.2.1. Free-run Timer Selection Register : FRS 663
5. Operation 670
5.1. Operation of the 32-bit Free-run Timer 670
5.1.1. Count Operation 670
5.1.2. Counting Up 671
5.1.3. Timer Clear 672
5.1.4. Each Clear Operations of the Free-run Timer 672
5.1.5. Timer Interrupt 674
5.2. Operation of the 32-bit Free-run Timer Selector 674
6. Setting 675
7. Q&A 676
7.1. How to Select Internal Clock Dividers 676
7.2. How to Select the External Clock 677
7.3. How to Enable/Disable the Count Operation of the Free-run Timer 677
7.4. How to Clear the Free-run Timer 677
7.5. About Interrupt Related Registers 678
7.6. How to Enable Compare Clear Interrupt 678
7.7. How to Stop the Free-run Timer Operation 679
8. Sample Program 679
9. Notes 680

Chapter 22: 32-Bit Output Compare 681

1. Overview 682
2. Features 683
3. Configuration Diagram 685
4. Registers 686
4.1. Output Control Register (Upper Bit) : OCSH 687
4.2. Output Control Register (Lower Bit) : OCSL 689
4.3. Compare Register : OCCP 690
4.4. Output Level Control Register : OCLS 691
5. Operation 692
5.1. Output Compare Output (Independent Invert) CMOD = "0" 693
5.2. Output Compare Output (Coordinated Invert) CMOD = "1" 694
5.3. Output Compare Operation Timing 695
5.3.1. Compare Register Write 695
5.3.2. Compare Match, Interrupt 696
5.3.3. Pin Output 696
6. Setting 697
7. Q&A 698
7.1. How to Set the Compare Value 698
7.2. How to Set the Compare Mode (Example with OCU7) 698
7.3. How to Enable/Disable the Compare Operation (Example with OCU6, 7) 699

7.4. How to Set the Compare Pin Output Initial Level (Example with OCU6, 7) 699
7.5. How to Set the Compare Pin OCU6-OCU7 for Output 699
7.6. How to Clear the Free-run Timer 700
7.7. How to Enable the Compare Operation (Example with OCU6, 7) 700
7.8. Interrupt Related Register 700
7.9. Interrupt Type 701
7.10. How to Enable the Interrupt 701
7.11. Calculation Method for the Compare Value 701
7.11.1. Toggle Output Pulse 702
7.11.2. PWM Output 702
7.12. How to Set the Operation Mode 703
8. Sample Program 704
9. Notes 706

Chapter 23: 32-Bit Input Capture 707

1. Overview 708
2. Features 708
3. Configuration 710
4. Registers 711
4.1. Input Capture Data Register : IPCP 713
4.2. Input Capture Control Register : ICS 713
4.3. LIN SYNCH FIELD Switching Register : LSYNS 715
4.4. Cycle Measurement Data Register : MSCY 716
4.5. Cycle and Pulse Width Measurement Control Register (Upper bit) : MSCH 717
4.6. Cycle and Pulse Width Measurement Control Register (Lower bit) : MSCL 718
5. Operation 719
5.1. Capture and Interrupt Timings 719
5.2. Edge Detection Specifications for Input Capture And Their Operations 721
5.3. Cycle and Pulse Width Measurement Operation 722
6. Setting 726
7. Q&A 727
7.1. Effective Edge Polarity of External Input: Types and How to Select Them 727
7.2. How to Enable External Input Pins (ICU4 to ICU9) 727
7.3. About Interrupt Related Registers 728
7.4. About Interrupt Types 728
7.5. How to Enable Interrupt 728
7.6. How to Measure the Pulse Width of the Input Signal 729
7.7. How to Set the Operation Mode 729
8. Sample Program 730
9. Notes 731

Chapter 24: 16-Bit Free-Run Timer 732

1. Overview 733
2. Features 734
3. Configuration 735
4. Registers 738
4.1. Registers for the Free-run Timer Simultaneous Activation 739
4.1.1. Timer Synchronous Activation Register : (TCGS) 740
4.1.2. Timer Synchronous Activation Enable Register : TCGSE 741
4.2. Registers for the 16-bit Free-run Timer 742
4.2.1. Compare Clear Buffer Register : CPCLRB/ Compare Clear Register : CPCLR 742

4.2.2. Timer Data Register : TCDT0 to TCDT2 744
4.2.3. Timer State Control Register : TCCS0 to TCCS2 745
4.3. Register for the Free-run Timer Selector 752
4.3.1. Free-run Timer Selection Register : FRS 752
5. Operation 765
5.1. Interrupt for the 16-bit Free-run Timer 765
5.2. Operation of the 16-bit Free-run Timer 766
5.2.1. Timer Clear 766
5.2.2. Timer Mode 767
5.2.3. Compare Clear Buffer 768
5.2.4. Timer Interrupt 769
5.2.5. Interrupt Mask Function 770
5.2.6. Selected External Count Clock 772
5.3. Operation of the Free-run Timer Selector 773
5.4. Notes on Operating Specifications 775
5.4.1. Notes at Accessing the Buffer Registers 775
5.4.2. Notes on Using the 16-bit Free-run Timer 775
5.4.3. Notes on Using the Free-run Timer Selector 776

Chapter 25: 16-Bit Output Compare 777

1. Overview 778
2. Features 778
3. Configuration Diagram 779
4. Registers 779
4.1. 16-bit Output Compare Registers 780
4.1.1. Output Compare Buffer Registers (OCCPB0 to OCCPB5)/Output Compare Registers (OCCP0 to OCCP5) 780
4.1.2. Compare Control Register (OCS) 784
4.1.3. Compare Mode Control Register (OCMOD) 790
5. Operation 791
5.1. Interrupts for 16-bit Output Compare 791
5.2. Operation of 16-bit Output Compare 792
5.2.1. Operation of 16-bit Output Compare (Inverted Mode, MOD0= 0 in OCMOD01 Register) ... 792
5.2.2. Operation of 16-bit Output Compare (Set/Reset Mode, MOD0 = 1 in OCMOD01 Register) 796
5.2.3. 16-bit Output Compare Timing 797
5.2.4. Operation of 16-bit Output Compare and Free-run Timer 798
5.3. Notes on Using 16-bit Output Compare 805

Chapter 26: 16-Bit Input Capture 807

1. Overview 808
2. Features 808
3. Configuration 809
4. Registers 810
4.1. 16-bit Input Capture Registers 811
4.1.1. Input Capture Data Register : IPCP0 to IPCP3 811
4.1.2. Input Capture State Control Register : ICS 812
4.1.3. LIN SYNCH FIELD Switching Register : LSYNS 815
5. Operation 816
5.1. Interrupts for 16-bit Input Capture 816
5.2. Operation of 16-bit Input Capture 816
5.2.1. Operation of 16-bit Input Capture 817

5.2.2. 16-bit Input Capture Input Timing 818
5.3. Notes on Using the 16-bit Input Capture 818

Chapter 27: Up/Down Counter 819

1. Overview 820
2. Features 820
3. Configuration 821
4. Registers 823
4.1. Reload Compare Register (RCR0, RCR1) 824
4.2. Up/Down Count Register (UDCR0, UDCR1) 825
4.3. Counter Control Register (CCR0, CCR1) 827
4.4. Counter Status Register (CSR0, CSR1) 832
5. Interrupt 835
6. Operation and Setting Procedure Examples 836
6.1. Operation in Timer Mode 840
6.2. Operation in Up/down Count Mode 841
6.3. Operation in the Phase Difference Count Mode (Multiply-by-Two) 845
6.4. Operation in the Phase Difference Count Mode (Multiply-by-Four) 847

Chapter 28: Real-Time Clock(RTC) 849

1. Overview 850
2. Features 850
3. Configuration 851
4. Registers 852
4.1. RTC Control Register : WTCR 852
4.2. Sub-second Register : WTBR 856
4.3. Day/Hour/Minute/Second Register : WTDR/ WTHR/ WTMR/ WTSR 857
5. Operation 859
6. Setting 862
7. Q&A 863
7.1. How to Set the 0.5 Second Count Interval 863
7.2. How to Initialize the Real-time Clock 863
7.3. How to Set/Update Number of Days (Day) and Time (Hour/Minute/Second) 864
7.4. How to Start/Stop the Count of the Real-time Clock 864
7.5. How to Confirm That the Real-time Clock Is Running 864
7.6. How to Know the Number of Days and Time 864
7.7. How to Stop the Real-time Clock 865
7.8. How to Calibrate the Real-time Clock 865
7.9. Interrupt Related Registers 865
7.10. Interrupt Types and How to Select Them 866
7.11. How to Enable Interrupts 866
8. Sample Program 867
9. Notes 867

Chapter 29: RTC/WDT1 Calibration 869

1. Overview 870
2. Features 870
3. Configuration 871
4. Registers 872
4.1. Calibration Unit Control Register 0 : CUCR0 (Calibration Unit Control Register 0) 872
4.2. Sub Clock Timer Data Register : CUTD0 (Calibration Unit Timer Data register 0) 873

4.3. Main Oscillation Timer Result Register 0 : CUTR0 (Calibration Unit Timer Result register 0) 874
4.4. Calibration Unit Control Register 1 : CUCR1 (Calibration Unit Control Register 1) 875
4.5. CR Clock Timer Data Register : CUTD1 (Calibration Unit Timer Data register 1) 876
4.6. Main Oscillation Timer Result Register 1 : CUTR1 (Calibration Unit Timer Result register 1) 877
5. Operation 878
5.1. Real-Time Clock (RTC) Calibration 878
5.2. Measurement of Errors in CR Clock 878
5.3. Note 879

Chapter 30: Power Consumption Control 880

1. Overview 881
2. Features 881
3. Configuration 882
4. Registers 883
4.1. Standby Control Register: STBCR (STandBy mode Control Register) 884
4.2. PMU Control Register : PMUCTLR (Power Management Unit ConTroL Register) 885
4.3. Power on Timing Control Register : PWRTMCTL (PoWeR on TiMing ConTroL register) 887
4.4. PMU Interrupt Flag Register 0 : PMUINTF0 (Power Management Unit INTerrupt Flag0 register) 888
4.5. PMU Interrupt Flag Register 1 : PMUINTF1 (Power Management Unit INTerrupt Flag1 register) 889
4.6. PMU Interrupt Flag Register 2 : PMUINTF2 (Power Management Unit INTerrupt Flag2 register) 890
5. Operation 891
5.1. Clock Control 891
5.1.1. Division Setting 892
5.1.2. Stopping of Unused Clocks 892
5.2. List of Clocks Supplied in Low-power Consumption Mode 892
5.3. Sleep Mode 893
5.3.1. CPU Sleep Mode 893
5.3.2. Bus Sleep Mode 893
5.3.3. Configuration of Sleep Mode 894
5.3.4. Activation of Sleep Mode 894
5.3.5. Wake Up from the Sleep Mode 895
5.3.6. Effect of Sleep Mode 895
5.4. Standby Mode : Watch Mode 895
5.4.1. Configuration of Watch Mode 896
5.4.2. Activation of Watch Mode 896
5.4.3. Wake Up from the Watch Mode 896
5.4.4. Effect of Watch Mode 897
5.5. Standby Mode : Watch Mode with Power-shutdown 897
5.5.1. Configuration of Watch Mode with Power-shutdown 897
5.5.2. Activation of Watch Mode with Power-shutdown 898
5.5.3. Wake Up from the Watch Mode with Power-shutdown 900
5.5.4. Effect of Watch Mode with Power-shutdown 902
5.6. Standby Mode : Stop Mode 902
5.6.1. Configuration of Stop Mode 902
5.6.2. Activation of Stop Mode 902
5.6.3. Wake Up from the Stop Mode 903
5.6.4. Effect of Stop Mode 903
5.7. Standby Mode : Stop Mode with Power-shutdown 904

5.7.1. Configuration of Stop Mode with Power-shutdown 904
5.7.2. Activation of Stop Mode with Power-shutdown 904
5.7.3. Wake Up from the Stop Mode with Power-shutdown 907
5.7.4. Effect of Stop Mode with Power-shutdown 909
5.8. Stop State of Microcontroller 909
5.9. Transition to Illegal Standby Mode 909
5.10. Restrictions on Power-Shutdown and Normal Standby Control 910
6. Usage Example 913

Chapter 31: Low-Voltage Detection (Internal Low-Voltage Detection) 914

1. Overview 915
2. Features 915
3. Configuration 916
4. Registers 916
4.1. Internal Low-Voltage Detection Register : LVD (Low-Voltage Detect internal power fall register) 917
5. Operation 918
5.1. Internal Low-voltage Detection 918
6. Notes 919

Chapter 32: Low-Voltage Detection (External Low-Voltage Detection) 920

1. Overview 921
2. Features 921
3. Configuration 922
4. Registers 922
4.1. External Low-Voltage Detection Rise Detection Register : LVD5R (Low-Voltage Detect external 5v Rise register) 922
4.2. External Low-Voltage Detection Fall Detection Register : LVD5F (Low-Voltage Detect external 5v Fall register) 923
5. Operation 925
6. Notes 926

Chapter 33: Wild Register 927

1. Overview 928
2. Features 928
3. Configuration 928
4. Registers 929
4.1. Wild Register Data Enable Register : WREN (Wild Register ENable register) 930
4.2. Wild Register Address Register 00 to 15 : WRAR00 to 15 (Wild Register Address Register 00 to 15) 931
4.3. Wild Register Data Register 00 to 15 : WRDR00 to 15 (Wild Register Data Register 00 to 15) 932
5. Operation 932
6. Usage Example 933

Chapter 34: Clock Supervisor 934

1. Overview 935
2. Configuration 935
3. Register 937
3.1. Clock Supervisor Control Register : CSVCR(Clock SuperVisor Control Register) 937
4. Operation 939
4.1. Initial State 940
4.2. Stopping CR Oscillator and the Clock Supervisor Function 941

4.3. Re-enabling the Clock Supervisor 941
4.4. Sub Clock Mode 942
4.5. Stop Mode 942
4.6. Watch Mode 943
4.7. Checking the Reset Factor Using the Clock Supervisor 943
4.8. Return from CR Clock 944
4.9. Sub Clock Mode Enabled by Setting SCKS Bit 945

Chapter 35: Regulator Control 946

1. Overview 947
2. Features 947
3. Configuration 948
4. Register 949
   4.1. Regulator Output Voltage Select Register : REGSEL (REGulator output voltage SELect register) 949
5. Operation 950

Chapter 36: External Bus Interface 951

1. Overview 952
2. Features 952
3. Configuration 953
4. Registers 954
   4.1. CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3) 955
   4.2. CS Bus Setting Registers: ACR0 to ACR3 (Area Configuration Register 0-3) 958
   4.3. CS Wait Registers : AWR0 to AWR3 (Area Wait Register 0-3) 960
   4.4. External DMA Transfer Registers: DMAR0-3 (DMA transfer Register 0-3) 965
5. Operation 966
   5.1. External Pin Table 967
   5.2. External Bus Signal Protocol 969
      5.2.1. Address/Data Split Bus Read Protocol 969
      5.2.2. Address/Data split bus write protocol 971
      5.2.3. Address/data multiplexed bus read protocol 973
      5.2.4. Address/Data multiplexed bus write protocol 975
   5.3. Address Alignment 977
   5.4. Split Access 977
   5.5. Data Alignment 977
   5.6. Address Information 981
      5.6.1. Address information and output pins 981
      5.6.2. Address type 981
   5.7. Idle Cycle Insertion Function 982
   5.8. External Bus Output Signal Timing Settings 983
   5.9. RDY Pin Access Cycle Extension Function 988
   5.10. CS Setting Flow 989
   5.11. Example of Connecting to Asynchronous Memory 995
   5.12. Example of Connection to Little Endian Device 997

Chapter 37: Bus Performance Counters 998

1. Overview 999
2. Features 999
3. Configuration 1000
4. Registers 1001
   4.1. BPC-A Control Register : BPCCRA (Bus Performance Counter Control Register A) 1001

4.2. BPC-B Control Register : BPCCRB (Bus Performance Counter Control Register B) 1003
4.3. BPC-C Control Register : BPCCRC (Bus Performance Counter Control Register C) 1003
4.4. BPC-A Count Register : BPCTRA (Bus Performance CounTer Register A) 1004
4.5. BPC-B Count Register : BPCTRB (Bus Performance CounTer Register B) 1004
4.6. BPC-C Count Register : BPCTRC (Bus Performance CounTer Register C) 1005
5. Operation 1005
5.1. Setting 1005
5.2. Starting and Stopping 1007
5.3. Operation 1007
5.4. Measurement and Result Processing 1008

Chapter 38: CRC 1010

1. Overview 1011
2. Features 1011
3. Configuration 1011
4. Registers 1012
4.1. CRC Control Register : CRCCR (CRC Control Register) 1012
4.2. CRC Initial Value Register : CRCINIT (CRC Initial value register) 1013
4.3. CRC Input Data Register : CRCIN (CRC INput data register) 1014
4.4. CRC Register : CRCR (CRC Register) 1014
5. Operation 1015
5.1. CRC Definition 1015
5.2. Reset Operation 1015
5.3. Initialization 1016
5.4. Byte and Bit Orders 1016
5.5. CRC Calculation Sequence 1016
5.6. Examples 1017
5.6.1. Example 1 CRC16, Fixed Byte Input 1018
5.6.2. Example 2 CRC16, Mixture of Different Input Bit Widths 1020
5.6.3. Example 3 CRC32, Byte Order, Big-endian 1022
5.6.4. Example 4 CRC32, Byte Order, Little-endian 1023

Chapter 39: RAMECC 1024

1. Overview 1025
2. Features 1025
3. Configuration 1026
4. Registers 1028
4.1. Single-bit ECC Error Address Register XBS RAM : SEEARX 1029
4.2. Double-bit ECC Error Address Register XBS RAM : DEEARX 1030
4.3. ECC Error Control Register XBS RAM : EECSRX 1031
4.4. ECC False Error Generation Address Register XBS RAM : EFEARX 1032
4.5. ECC False Error Generation Control Register XBS RAM : EFECRX 1033
4.6. Single-bit ECC Error Address Register BACKUP-RAM : SEEARA 1035
4.7. Double-bit ECC Error Address Register BACKUP-RAM : DEEARA 1036
4.8. ECC Error Control Register BACKUP-RAM : EECSRA 1037
4.9. ECC False Error Generation Address Register BACKUP-RAM : EFEARA 1038
4.10. ECC False Error Generation Control Register BACKUP-RAM : EFECRA 1039
5. Operation 1041
5.1. RAMECC Function 1041
5.2. Interrupt-related Register 1042
5.3. Test Mode 1043

5.4. Note 1044

Chapter 40: Multi-Function Serial Interface 1045

1. Overview 1046
2. Features 1046
3. Configuration 1053
4. Registers 1054

4.1. Common Registers 1061
4.1.1. Serial Mode Register: SMR 1061
4.1.2. FIFO Control Register 1: FCR1 1064
4.1.3. FIFO Control Register 0: FCR0 1066
4.1.4. FIFO BYTE Register: FBYTE 1069
4.1.5. Transmission FIFO Interrupt Control Register: FTICR 1071
4.2. Registers for UART 1072
4.2.1. Serial Control Register: SCR 1072
4.2.2. Serial Status Register: SSR 1074
4.2.3. Extended Serial Control Register: ESCR 1076
4.2.4. Receive Data Register/Transmit Data Register: RDR/TDR 1077
4.2.5. Serial Aid Control Status Register: SACSR 1080
4.2.6. Serial TiMer Register: STMR 1082
4.2.7. Serial Timer Comparison Register: STMCR 1082
4.2.8. Baud rate Generator Register: BGR 1083
4.3. Registers for CSIO 1084
4.3.1. Serial Control Register: SCR 1084
4.3.2. Serial Status Register: SSR 1087
4.3.3. Extended Serial Control Register: ESCR 1089
4.3.4. Receive Data Register/Transmit Data Register: RDR/TDR 1092
4.3.5. Serial Aid Control Status Register: SACSR 1095
4.3.6. Serial Timer Register: STMR 1099
4.3.7. Serial Timer Compare Register: STMCR 1100
4.3.8. Serial Chip Select Control Status Register: SCSCR 1101
4.3.9. Serial Chip Select Timing Register: SCSTR3-0 1104
4.3.10. Serial Chip Select Format Register: SCSFR2-0 1108
4.3.11. Transfer BYTE register: TBYTE3-0 1117
4.3.12. Baud rate Generator Register: BGR 1118
4.4. Registers for LIN 1119
4.4.1. Serial Control Register: SCR 1119
4.4.2. Serial Status Register: SSR 1122
4.4.3. Extended Serial Control Register: ESCR 1125
4.4.4. Receive Data Register/Transmit Data Register: RDR/TDR 1127
4.4.5. Serial Aid Control Status Register: SACSR 1129
4.4.6. Serial Timer Register: STMR 1133
4.4.7. Serial Timer Compare Register: STMCR 1133
4.4.8. Sync Field Upper limit Register: SFUR 1134
4.4.9. Sync Field Lower limit Register: SFLR 1135
4.4.10. Baud Rate Generator Register: BGR 1136
4.4.11. LIN Assist Mode Status Register: LAMSR 1137
4.4.12. LIN Assist Mode Control Register: LAMCR 1139
4.4.13. LIN Assist Mode Interrupt Enable Register: LAMIER 1142
4.4.14. LIN Assist Mode Transmission/Reception ID register: LAMTID / LAMRID 1144
4.4.15. LIN Assist Mode Error Status Register: LAMESR 1145

4.4.16. LIN Assist Mode trouble Examination Register: LAMERT 1147
4.5. Registers for I²C 1150
4.5.1. I²C Bus Control Register: IBCR 1151
4.5.2. Serial Status Register: SSR 1156
4.5.3. I²C Bus Status Register: IBSR 1159
4.5.4. Receive Data Register/Transmit Data Register: RDR/TDR 1163
4.5.5. Serial Aid Control Status Register: SACSR 1165
4.5.6. Serial Timer Register: STMR 1167
4.5.7. Serial Timer Compare Register: STMCR 1167
4.5.8. 7-bit Slave Address Mask Register: ISMK 1168
4.5.9. 7-bit Slave Address Register: ISBA 1169
4.5.10. Baud rate Generator Register: BGR 1170
5. Operation of UART 1171
5.1. Interrupt of UART 1171
5.1.1. List of Interrupt of UART 1172
5.1.2. Reception Interrupts and Flag Setting Timing 1173
5.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing 1175
5.1.4. Transmission Interrupts and Flag Setting Timing 1177
5.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing 1178
5.1.6. Timing of Timer Interrupt Generation and Flag Setting 1179
5.2. Operation of UART 1179
5.2.1. Transmission/Reception Data Format 1180
5.2.2. Transmission Operation 1181
5.2.3. Reception Operation 1182
5.2.4. Clock Selection 1183
5.2.5. Start Bit Detection 1183
5.2.6. Stop Bit 1183
5.2.7. Error Detection 1184
5.2.8. Parity Bit 1184
5.2.9. Data Signaling Method 1185
5.2.10. Operation of Serial Timer 1185
5.2.11. Test Mode 1187
5.2.12. UART Baud Rate Selection/Setting 1188
5.3. Setup Procedure and Program Flow 1192
5.3.1. Operation Mode 0 (One-to-One Connection) 1192
5.3.2. Operation Mode 1 (One-to-N Connection) 1194
6. Operation of CSIO 1197
6.1. Interrupts of CSIO 1198
6.1.1. List of Interrupts of CSIO 1198
6.1.2. Reception Interrupts and Flag Setting Timing 1200
6.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing 1202
6.1.4. Transmission Interrupts and Flag Setting Timing 1204
6.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing 1205
6.1.6. Timing of Timer Interrupt and Flag Setting 1206
6.1.7. Timing of Chip Select Error Generation and Flag Setting 1206
6.2. Operation of CSIO 1208
6.2.1. Normal Transfer (I) 1208
6.2.2. Normal Transfer (II) 1217
6.2.3. SPI Transfer (I) 1226
6.2.4. SPI Transfer (II) 1236
6.2.5. Operation of Serial Timer 1247

6.2.6. Operation of Serial Chip Select 1250
6.2.7. Test Mode 1259
6.2.8. Baud Rate Generation 1260
6.3. Setup Procedure and Program Flow 1262
7. Operation of LIN Interface (v2.1) 1264
7.1. Interrupt of LIN Interface (v2.1) manual mode 1265
7.1.1. List of Interrupts of LIN Interface (v2.1) (manual mode) 1265
7.1.2. Reception Interrupts and Flag Setting Timing 1266
7.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing 1268
7.1.4. Transmission Interrupts and Flag Setting Timing 1269
7.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing 1271
7.1.6. Timer Interrupt and Flag Setting Timing 1271
7.1.7. Sync Field Detection Interrupts and Flag Setting Timing 1272
7.2. Interrupts in LIN Interface (v2.1) Assist Mode 1272
7.2.1. List of Interrupts of LIN Interface (v2.1) (assist mode) 1273
7.2.2. Reception Interrupts and Flag Setting Timing in Assist Mode 1275
7.2.3. Reception Interrupts and Flag Setting Timing when using Reception FIFO 1280
7.2.4. Transmission Interrupts and Flag Setting Timing 1281
7.2.5. Interrupts and Flag Setting Timing when using Transmission FIFO 1282
7.2.6. Timer Interrupts and Flag Setting Timing 1282
7.2.7. Status Interrupts and Flag Setting Timing in Assist Mode 1282
7.3. Operation of Serial Timer 1284
7.4. Test Mode 1286
7.4.1. Manual Mode 1287
7.4.2. Assist Mode 1288
7.5. Operation of LIN Interface (v2.1) 1292
7.5.1. Manual mode 1293
7.5.2. Assist Mode 1307
7.5.3. LIN Baud Rate Selection/Setting 1327
7.6. Setup Procedure and Program Flow 1327
7.6.1. Manual mode 1327
7.6.2. Assist mode 1331
8. Operation of I^2C 1336
8.1. Interrupts of I^2C 1336
8.1.1. List of Interrupts of I^2C Interface 1337
8.1.2. Timing of Timer Interrupt Generation and Flag Setting 1338
8.2. Operation for I^2C Interface Communication 1339
8.2.1. I^2C Bus Start Condition 1339
8.2.2. I^2C Bus Stop Condition 1339
8.2.3. I^2C Bus Repeated Start Condition 1340
8.2.4. I^2C Bus Error 1340
8.2.5. Serial Timer Operations 1341
8.2.6. Baud Rate Generation 1342
8.3. I^2C Master Mode 1343
8.3.1. Start Condition Generation 1344
8.3.2. Slave Address Output 1345
8.3.3. Acknowledge Reception by Transmitting First Byte 1348
8.3.4. Data Transmission by Master 1355
8.3.5. Data Reception by Master 1374
8.3.6. Arbitration Lost 1381
8.3.7. Wait of the Master Mode 1381

8.3.8. Repetition Start Condition Issue when DMA Mode Enabled (SSR:DMA=1) 1382
8.4. I²C Slave Mode 1382
8.4.1. Detection of Slave Address Matching 1383
8.4.2. Data Direction Bit 1384
8.4.3. Slave Mode Reception 1384
8.4.4. Slave Mode Transmission 1392
8.5. Example of I²C Flowchart 1393

Chapter 41: CAN 1400

1. Overview 1401
2. Features 1401
3. Configuration 1401
4. Registers 1402
4.1. Overview 1403
4.1.1. List of Base-addresses (Base-addr), External Pins and Buffer Size 1404
4.1.2. List of Overall Control Register 1404
4.1.3. List of Message Interface Register 1406
4.1.4. List of Message Handler Register 1409
4.2. Overall Control Registers 1411
4.2.1. CAN Control Register : CTRLR 1411
4.2.2. CAN Status Register : STATR 1415
4.2.3. CAN Error Counter : ERRCNT 1418
4.2.4. CAN Bit Timing Register : BTR 1419
4.2.5. CAN Interrupt Register : INTR 1420
4.2.6. CAN Test Register : TESTR 1421
4.2.7. CAN Prescaler Extension Register : BRPER 1423
4.3. Message Interface Register 1423
4.3.1. IFx Command Request Register : IFxCREQ 1424
4.3.2. IFx Command Mask Register (IFxCMSK) 1427
4.3.3. IFx Mask Registers 1, 2 : IFxMSK1, IFxMSK2 1431
4.3.4. IFx Arbitration Registers 1, 2 : IFxARB1, IFxARB2 1432
4.3.5. IFx Message Control Register : IFxMCTR 1434
4.3.6. IFx Data Registers A1, A2, B1, B2 : IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2 1435
4.4. Message Object 1436
4.4.1. Configuration of Message Object 1436
4.4.2. Functions of Message Object 1436
4.5. Message Handler Registers 1441
4.5.1. CAN Transmission Request Registers : TREQR1 to TREQR4 1441
4.5.2. CAN Data Update Registers : NEWDT1 to NEWDT4 1445
4.5.3. CAN Interrupt Pending Registers :INTPND1 to INTPND4 1448
4.5.4. CAN Message Valid Registers : MSGVAL1 to MSGVAL4 1450
5. Operation 1453
5.1. Message Object 1453
5.1.1. Message Object 1453
5.1.2. Data Transmission/Reception with Message RAM 1454
5.2. Message Transmission Operation 1455
5.2.1. Message Transmission 1455
5.2.2. Transmission Priority 1456
5.2.3. Transmission Message Object Setting 1456
5.2.4. Update of Transmission Message Object 1457
5.3. Message Reception Operation 1458

5.3.1. Reception Message Acceptance Filter 1458
5.3.2. Reception Priority 1458
5.3.3. Data Frame Reception 1458
5.3.4. Remote Frame 1459
5.3.5. Reception Message Object Setting 1459
5.3.6. Reception Message Processing 1460
5.4. FIFO Buffer Function 1460
5.4.1. Configuration of FIFO Buffer 1460
5.4.2. Message Reception by FIFO Buffer 1461
5.4.3. Reading from FIFO Buffer 1461
5.5. Interrupt Function 1463
5.6. Bit Timing and CAN System Clock (fsys) Generation 1464
5.7. Test Mode 1466
5.7.1. Test Mode Setting 1466
5.7.2. Silent Mode 1466
5.7.3. Loopback Mode 1467
5.7.4. Combination of Silent and Loopback Modes 1468
5.7.5. Basic Mode 1469
5.7.6. Software Control of the CAN_TX Pin 1470
5.8. Software Initialization 1470
5.9. CAN Wake Up Function 1471
6. Restrictions 1472
6.1. INIT bit 1472
6.1.1. Restrictions 1473
6.1.2. Bypass method 1474

Chapter 42: CAN Clock Prescaler 1475

1. Overview 1476
2. Features 1476
3. Configuration 1476
4. Registers 1477
4.1. CAN Prescaler Register : CANPRE 1477

Chapter 43: D/A Converter 1479

1. Overview 1480
2. Features 1480
3. Configuration 1481
4. Registers 1481
4.1. D/A Control Register : DACR 1482
4.2. D/A Data Register : DADR 1483
5. Operation 1483
6. Note 1484

Chapter 44: 12-Bit A/D Converter 1485

1. Overview 1486
2. Features 1486
2.1. Function of A/D Activation Compare 1486
2.2. Function of A/D Activation Arbitration 1488
2.3. Functions of 12-bit A/D Converter Control 1489
3. Configuration 1490
4. Registers 1492

4.1. Register of Analog Input Control 1501
4.1.1. Analog Input Enable Register : ADER 1501
4.2. Register of A/D Activation Compare 1503
4.2.1. A/D Software Activation Register: ADTSS0, ADTSS1 1504
4.2.2. A/D Software Activation Channel Select Register : ADTSE0, ADTSE1 1504
4.2.3. Compare Buffer Register / Compare Register : ADCOMPB0 to ADCOMPB47 / ADCOMP0 to ADCOMP47 1507
4.2.4. A/D Activation Trigger Control Status Register : ADTCS0 to ADTCS47 1508
4.2.5. A/D Data Register : ADTCD0 to ADTCD47 1513
4.2.6. A/D Activation Trigger Extend Control Register : ADTECS0 to ADTECS47 1514
4.2.7. Upper Bound Threshold Setting Register : ADRCUT0 to ADRCUT7 1517
4.2.8. Lower Bound Threshold Setting Register : ADRCLT0 to ADRCLT7 1519
4.2.9. Range Compare Control Status Register: ADRCCS0 to ADRCCS47 1520
4.2.10. Range Compare Threshold Over Flag Register : ADRCOT0, ADRCOT1 1523
4.2.11. Range Compare Flag Register : ADRCIF0, ADRCIF1 1525
4.2.12. Scan Conversion Control Status Register : ADSCANS0, ADSCANS1 1528
4.2.13. Activation Channel Conversion Count Setting Register : ADNCS0 to ADNCS23 1530
4.2.14. Data Protection Status Flag Register : ADPRTF0, ADPRTF1 1531
4.2.15. Activation Channel Conversion Completion Flag Register : ADEOCF0, ADEOCF1 1533
4.3. Register of 12-BIT A/D Converter Control 1535
4.3.1. A/D Control Status Register: ADCS0, ADCS1 1536
4.3.2. A/D Channel Status Register : ADCH 1536
4.3.3. A/D Mode Setting Register : ADMD 1538
4.3.4. A/D Sampling Time Setting Per Channel Register : ADSTPCS 1540
5. Operation 1542
5.1. Interrupt of A/D activation compare 1542
5.1.1. A/D conversion completion interrupt 1542
5.1.2. Scan conversion completion interrupt by conversion count specification 1543
5.1.3. Range comparison interrupt 1544
5.2. A/D activation compare operation 1544
5.2.1. A/D activation 1545
5.2.2. A/D activation enable 1545
5.2.3. Free-run timer input 1546
5.2.4. Analog channel select 1546
5.2.5. Software activation 1546
5.2.6. External trigger activation 1546
5.2.7. Reload timer activation 1547
5.2.8. Compare match activation 1547
5.2.9. PPG activation 1553
5.2.10. Activation request mode 1554
5.2.11. A/D conversion data 1554
5.2.12. Protection function 1555
5.2.13. Scan conversion mode 1555
5.2.14. High priority activation request operation of other activation channel during the scan conversion 1562
5.2.15. Forced termination of activation request 1563
5.2.16. Range comparison function 1564
5.3. A/D Activation Arbitration Operation 1569
5.3.1. A/D Activation Trigger Arbitration 1569
5.3.2. Analog Channel Select 1570
5.3.3. A/D Conversion Cancel Function 1570

5.4. 12-bit A/D Converter Operation 1570
5.4.1. Operation Timing 1571
5.4.2. Activation Factors 1571
5.4.3. A/D Conversion 1572
5.4.4. Re-activation 1572
5.4.5. A/D Conversion Cancel 1572
5.4.6. Analog Channel Select Control 1572
5.4.7. A/D Conversion Time 1572
5.4.8. A/D Conversion Completion and A/D Data Retrieval 1573
5.4.9. Power down 1574
6. Notes 1574

Chapter 45: Flash Memory 1576

1. Overview 1577
2. Features 1577
3. Configuration 1578
3.1. Block Diagram 1579
3.2. Sector Configuration Diagram 1580
3.3. Sector Number and Flash Macro Number Correspondence Chart 1584
4. Registers 1590
4.1. Flash Control Register : FCTLR (Flash ConTroL Register) 1590
4.2. Flash Status Register : FSTR (Flash STatus Register) 1592
4.3. Flash Interface Control Register : FLIFCTLR(Flash I/F ConTrol Register) 1594
4.4. FLash I/F Feature Extension Register 1: FLIFFER1 1595
4.5. FLash I/F Feature Extension Register 2: FLIFFER2 1596
5. Operation 1596
5.1. Access Mode Setting 1597
5.1.1. Configuring CPU-ROM Mode 1597
5.1.2. Configuring CPU Programming Mode 1597
5.2. Writing to Flash Memory by CPU 1597
5.3. Automatic Algorithm 1598
5.3.1. Command Sequence 1598
5.3.2. Automatic Algorithm Execution State 1601
5.4. Reset Command 1605
5.5. Write Command 1605
5.6. Chip Erase Command 1607
5.7. Sector Erase Command 1608
5.8. Sector Erase Suspend Command 1610
5.9. Security Function 1611
5.9.1. Flash Security On/Off Determination When Reset Released 1611
5.9.2. Flash Security Setting Method 1611
5.9.3. Unlocking Flash Security 1611
5.9.4. Flash Access Restrictions When Security is ON 1612
5.10. Notes on Using Flash Memory 1613

Chapter 46: WorkFlash Memory 1614

1. Overview 1615
2. Features 1615
3. Configuration 1616
3.1. Block Diagram 1616
3.2. Sector Configuration Diagram 1617

4. Registers 1617
4.1. WorkFlash Control Register : DFCTLR (WorkFlash ConTroL Register) 1618
4.2. WorkFlash Status Register : DFSTR (WorkFlash STatus Register) 1619
4.3. Flash Interface Control Register : FLIFCTLR (Flash I/F ConTroL Register) 1620
5. Operation 1621
5.1. Access Mode Setting 1622
5.1.1. Configuring CPU-ROM Mode below 1622
5.1.2. Configuring CPU Programming Mode 1622
5.2. Writing Flash Memory by CPU 1622
5.3. Automatic Algorithm 1623
5.3.1. Command Sequence 1623
5.3.2. Automatic Algorithm Execution State 1626
5.4. Reset Command 1630
5.5. Write Command 1630
5.6. Chip Erase Command 1632
5.7. Sector Erase Command 1633
5.8. Sector Erase Suspend Command 1635
5.9. Security Function 1636
5.9.1. Flash Security On/Off Determination When Reset Released 1636
5.9.2. Flash Security Setting Method 1636
5.9.3. Unlocking Flash Security 1637
5.9.4. Flash Access Restrictions When Security is ON 1637
5.10. Notes on Using WorkFlash Memory 1638

Chapter 47: On-Chip Debugger : OCD 1639

1. Overview 1640
2. Features 1640
3. Configuration 1642
3.1. DEBUG I/F Clock 1643
3.1.1. DEBUG I/F Main Clock : M_MCLK 1643
3.1.2. DEBUG I/F PLL Clock : M_PCLK 1644
4. Registers 1644
4.1. DBG Register 1644
4.1.1. DSU Control Register : DSUCR 1645
4.2. User IO Register 1645
4.2.1. User Event Register : UER 1646
4.2.2. High-Speed Communication Frequency Register : HSCFR 1646
4.2.3. Message BuffeR : MBR 1647
5. Operation 1648
5.1. OCDU Operating Mode 1648
5.1.1. Operating Mode 1649
5.1.2. Operating Mode Status Transition 1649
5.2. Overview of DEBUG I/F 1650
5.2.1. Chip Reset Sequence 1651
5.2.2. Security Function 1653
5.3. Specification Restrictions at Connection to OCD Tool of This Series 1653
5.3.1. Clock Setting 1654
5.3.2. Standby Mode 1654
5.3.3. Clock Reset State Transitions 1654
5.3.4. Summary of Specification Restrictions 1656
5.4. OCD-DSU ID Code and Mount Type Information on This Series 1661

Chapter 48: Waveform Generator 1663

1. Overview 1664
2. Features 1664
3. Configuration 1664
4. Registers 1666
   4.1. Registers for the Waveform Generator 1667
      4.1.1. 16-bit dead TIMeR Register (TMRR) 1667
      4.1.2. 16-bit Dead Timer State Control Register (DTSCR) 1668
      4.1.3. 16-bit Dead Timer reload Interrupt Register (DTIR) 1675
      4.1.4. 16-bit Dead Timer Minus Control Register (DTMNS) 1677
      4.1.5. Waveform Control Register 1/2 (SIGCR1, SIGCR2) 1679
      4.1.6. PPG Output Control Register (PICS) 1682
5. Operation 1683
   5.1. Interrupts for Waveform Generator 1684
   5.2. Operation of the Waveform Generator 1685
6. Notes 1704

Chapter 49: Bus Diagnosis Function 1706

1. Overview 1707
2. Features 1707
3. Configuration 1708
4. Registers 1710
   4.1. BUS DIaGnosis Status Register : BUSDIGSR 1711
   4.2. BUS diagnosis TeST Register : BUSTSTR0/1 1713
   4.3. BUS diagnosis ADdRess Register : BUSADR 1718
5. Operation 1719
   5.1. Error Detection 1719
   5.2. Test Function 1721
   5.3. Notes 1722
   5.4. Example of Operating Bus Diagnosis 1723

Chapter 50: RAM Diagnosis Function 1730

1. Overview 1731
2. Features 1731
3. Configuration 1732
4. Registers 1734
   4.1. TEST Error Address Register 0 XBS RAM : TEAR0X 1735
   4.2. TEST Error Address Register 1 XBS RAM : TEAR1X 1736
   4.3. TEST Error Address Register 2 XBS RAM : TEAR2X 1737
   4.4. TEST Start Address Register XBS RAM : TASARX 1739
   4.5. TEST End Address Register XBS RAM : TAEARX 1740
   4.6. TEST Diagnosis Function Register XBS RAM : TTCRX 1741
   4.7. TEST Initialization Function Register XBS RAM : TICRX 1743
   4.8. TEST Software Reset Generation Control Register XBS RAM : TSRCRX 1744
   4.9. TEST Fake Error Generation Control Register XBS RAM : TFECRX 1745
   4.10. TEST Key Code Control Register XBS RAM : TKCCRX 1746
   4.11. TEST Error Address Register 0 BACKUP-RAM : TEAR0A 1747
   4.12. TEST Error Address Register 1 BACKUP-RAM : TEAR1A 1748
   4.13. TEST Error Address Register 2 BACKUP-RAM : TEAR2A 1750
   4.14. TEST Start Address Register BACKUP-RAM : TASARA 1751
   4.15. TEST End Address Register BACKUP-RAM : TAEARA 1752

4.16. TEST Diagnosis Function Register BACKUP-RAM : TTCRA 1753
4.17. TEST Initialization Function Register BACKUP-RAM : TICRA 1756
4.18. TEST Software Reset Generation Control Register BACKUP-RAM : TSRCRA 1757
4.19. TEST Fake Error Generation Control Register BACKUP-RAM : TFECRA 1758
4.20. TEST Key Code Control Register BACKUP-RAM : TKCCRA 1759
5. Operation 1760
5.1. RAM Diagnosis 1760
5.2. RAM Initialization 1761
5.3. Interrupt-Related Register 1761
5.4. RAM Diagnosis Fake Error Generation Procedure 1762
5.5. Number of Required Cycles 1762
5.6. Note 1765

Chapter 51: Timing Protection Unit 1766

1. Overview 1767
2. Features 1767
3. Configuration 1767
4. Registers 1768
4.1. TPU Unlock Register : TPUUNLOCK 1769
4.2. TPU Lock Status Register : TPULST 1769
4.3. TPU Access Violation Status Register : TPUVST 1770
4.4. TPU Configuration Register : TPUCFG 1771
4.5. TPU Timer Interrupt Request Register : TPUTIR 1773
4.6. TPU Timer Status Register : TPUTST 1773
4.7. TPU Timer Interrupt Enable Register : TPUTIE 1774
4.8. TPU Module ID Register : TPUTMID 1775
4.9. TPU Timer Control Register 00 to 07 : TPUTCN00 to 07 1775
4.10. TPU Timer Control Register 10 to 17 : TPUTCN10 to 17 1777
4.11. TPU Timer Current Count Register 0 to 7 : TPUTCC0 to 7 1778
5. Operation 1779
5.1. TPU Control Register Access Protection 1779
5.2. Global Prescaler 1780
5.3. Interrupt Control 1780
5.4. Timer Operation 1781
5.5. Free-run Function 1782
5.6. Individual Prescaler Function 1782
5.7. Debug Support Function 1782
5.8. Operation Flow 1782

Chapter 52: Clock Monitor 1784

1. Overview 1785
2. Features 1785
3. Configuration 1785
4. Registers 1786
4.1. Clock Monitor Configuration Registers : CMCFG 1786
5. Operation 1788
6. Setting 1788
7. Q&A 1789
7.1. How to Configure the Output Pin (MONCLK) 1789
7.2. How to Select the Output Frequency 1789
7.3. How to Enable or Disable Clock Monitor Output 1790

7.4. How to Set the Clock Output Mark Level 1790
8. Notes 1790

Appendix 1792

A. I/O Map 1793
B. List of Interrupt Vector 1854
C. Pins Statuses in State of CPU 1878
D. Major Changes 1882

Revision History 2150

Document Revision History 2150

Chapter 1: Overview

This chapter explains the overview.

1. Overview
2. Features
3. Product Line-up
4. Function overview
5. Block Diagram
6. Memory Map
7. Pin Assignment
8. Device Package
9. Explanation of Pin Functions
10. I/O Circuit Types

Code : OVER-1v0-91520-3-E

1. Overview

This section explains overview of MB91520 series.

MB91520 series is Cypress 32-bit microcontroller for application control for automotives. The FR81S CPU that is compatible with the FR family is used.

2. Features

This section explains features of MB91520 series.

2.1. FR81S CPU Core

2.2. Peripheral Functions

2.1. FR81S CPU Core

FR81S CPU core is shown.

- 32-bit RISC, load/store architecture, 5-stage pipeline
- Maximum operating frequency: 80 MHz (Source oscillation = 4.0 MHz and 20 multiplied (PLL clock multiplication system))
- General-purpose register : 32-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

2.2. Peripheral Functions

Peripheral functions is shown.

- Clock generation (equipped with SSCG function)
  - Main oscillation (4MHz to 16MHz)
  - Sub oscillation (32kHz) or no sub oscillation
  - PLL multiplication rate : 1 to 20 times
  - Equipped with a 100kHz CR oscillator
- Built-in Program flash capacity

  MB91F522 : 256 + 64KB
  MB91F523 : 384 + 64KB
  MB91F524 : 512 + 64KB
  MB91F525 : 768 + 64KB
  MB91F526 : 1024 + 64KB

- Built-in Data flash (WorkFlash) 64KB
- Built-in RAM capacity
  - Main RAM

    MB91F522 : 48KB
    MB91F523 : 48KB
    MB91F524 : 64KB
    MB91F525 : 96KB
    MB91F526 : 128KB
  - Backup RAM 8KB
- General-purpose ports :

  MB91F52xB : 44(none sub oscillation), 42 (with sub oscillation)
  MB91F52xD : 56(none sub oscillation), 54 (with sub oscillation)
  MB91F52xF : 76(none sub oscillation), 74 (with sub oscillation)
  MB91F52xJ : 96(none sub oscillation), 94 (with sub oscillation)
  MB91F52xK : 120(none sub oscillation), 118 (with sub oscillation)
  MB91F52xL : 152(none sub oscillation), 150 (with sub oscillation)

  Included I^2C pseudo open drain ports : 16
- External bus interface
  - 22-bit address, 16-bit data
- DMA Controller
  - Up to 16 channels can be started simultaneously.
  - 2 transfer factors (Internal peripheral request and software)
- A/D converter (successive approximation type)

- 12-bit resolution : Max. 48 channels (32 channels +16 channels)
- Conversion time : 1.4µs

· D/A converter (R-2R type)
- 8-bit resolution : 2 channels

· External interrupt input: 8 channels × 2 unit Total:16 channels
· Level ("H" / "L"), or edge detection (rising or falling) supported
· Multi-function serial communication (built-in transmission/reception FIFO memory) : Max. 12 channels
5V tolerant input 4 channels (ch.6, ch.8, ch.9, ch.11) CMOS hysteresis input
< UART (Asynchronous serial interface) >
- Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
- Parity or no parity is selectable.
- Built-in dedicated baud rate generator
- The external clock can be used as the transfer clock
- Parity, frame, and overrun error detect functions provided
- DMA transfer support

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

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

< I^2C >
- 2 channels (ch.3, ch.4) Standard mode / Fast mode supported
- 6 channels (ch.5 to ch.8, ch.10, ch.11) Standard mode supported
- Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
- Standard mode (Max. 100kbps) / Fast mode (Max. 400kbps) supported
- DMA transfer supported (for transmission only)

· CAN : 3 channels
- Transfer speed : Up to 1Mbps
- 128-transmission/reception message buffering : 1 channel (ch.0),
64-transmission/reception message buffering : 2 channels (ch.1, ch.2)

· PPG : 16-bit × Max. 48 channels
- LED drive output 4 channels (ch.11 to ch.14)

· Reload timer : 16-bit × Max. 8 channels
· Free-run timer :
16-bit × 3 channels
32-bit × Max. 3 channels
· Input capture :

16-bit × 4 channels (linked to the free-run timer)
32-bit × Max. 6 channels (linked to the free-run timer)

- Output compare :
  16-bit × 6 channels (linked to the free-run timer)
  32-bit × Max. 6 channels (linked to the free-run timer)
- Wave generator : 6 channels
- U/D counter:
  - 8/16-bit up/down counter × 2 channels

- Real-time clock (RTC) (for day, hours, minutes, seconds)
  - Main oscillation / sub oscillation frequency can be selected for the operation clock.
- Calibration: A real-time clock (RTC) of the sub clock drive.
  - 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 sub oscillation (32kHz) (dual clock products) and main oscillation (4 MHz).
  - When abnormality is detected, it switches to the CR clock.
- For some devices, ON/OFF can be selected as the initial value.
- Base timer : Max. 2 channels
  - 16-bit timer
  - The timer mode is selected from PWM/PPG/PWC/reload.
  - As for the PWC function and the reload timer function, a pair of 16-bit timers can be used as one 32-bit timer in the cascaded mode.
- CRC generation
- Watchdog timer
  - Hardware watchdog
  - Software watchdog (An effective range of a clear counter can be set.)
- 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
- Low-voltage detection reset (External power supply and Internal power supply are independently observed.)
- For some devices, ON/OFF can be selected as the initial value for external power supply.
- Device Package : LQFP-176/144/120/100/80/64
- CMOS 90nm Technology
- Power supplies
  - 5V Power supply
  - The internal 1.2V is generated from 5V with the voltage step-down regulator.

3. Product Line-up

This section shows product line-up of MB91520 series.

Table 3-1 Product Line-up (64 pin)

	MB91F522B	MB91F523B	MB91F524B	MB91F525B	MB91F526B
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 KB + 64KB	384 KB + 64KB	512 KB + 64KB	768 KB + 64KB	1024 KB + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	None				
DMA Transfer	16 channels				
16-bit Base Timer	None				
Free-run Timer	16-bit × 3 channels, 32-bit × 1 channel				
Input capture	16-bit × 4 channels, 32-bit × 5 channels				
Output Compare	16-bit × 6 channels, 32-bit × 4 channels				
16-bit Reload Timer	7 channels				
PPG	16-bit × 21 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 13 channels (1 unit) 12-bit ×13 channels (1 unit)				
D/A (8-bit)	1 channel				
Multi-Function Serial	8 channels[*1]				
CAN	64msg × 2 channel / 128msg × 1 channel				
Hardware watchdog	Yes				

	MB91F522B	MB91F523B	MB91F524B	MB91F525B	MB91F526B
CRC generation	Yes				
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	44 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
NMI request function	Yes				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQD064				

*1: Only channel 5, channel 6 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-2 Product Line-up (80 pin)

	MB91F522D	MB91F523D	MB91F524D	MB91F525D	MB91F526D
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 + 64KB	384 + 64KB	512 + 64KB	768 + 64KB	1024 + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	None				
DMA Transfer	16 channels				
16-bit Base Timer	1 channel				
Free-run Timer	16-bit × 3 channels 32-bit × 2 channels				
Input capture	16-bit × 4 channels 32-bit × 5 channels				
Output Compare	16-bit × 6 channels 32-bit × 4 channels				
16-bit Reload Timer	7 channels				
PPG	16-bit × 27 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 16 channels (1 unit) 12-bit ×16 channels (1 unit)				
D/A (8-bit)	1 channel				
Multi-Function Serial	9 channels[*1]				
CAN	64msg × 2 channel / 128msg × 1 channel				
Hardware watchdog	Yes				
CRC generation	Yes				

	MB91F522D	MB91F523D	MB91F524D	MB91F525D	MB91F526D
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	56 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
NMI request function	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQH080				

*1: Only channel 5, channel 6 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-3 Product Line-up (100 pin)

	MB91F522F	MB91F523F	MB91F524F	MB91F525F	MB91F526F
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 + 64KB	384 + 64KB	512 + 64KB	768 + 64KB	1024 + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	None				
DMA Transfer	16 channels				
16-bit Base Timer	1 channel				
Free-run Timer	16-bit × 3 channels 32-bit × 3 channels				
Input capture	16-bit × 4 channels 32-bit × 6 channels				
Output Compare	16-bit × 6 channels 32-bit × 6 channels				
16-bit Reload Timer	8 channels				
PPG	16-bit × 34 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 21 channels (1 unit) 12-bit ×16 channels (1 unit)				
D/A (8-bit)	2 channels				
Multi-Function Serial	12 channels[*1]				
CAN	64msg × 2 channels / 128msg × 1 channel				
Hardware watchdog	Yes				
CRC generation	Yes				

	MB91F522F	MB91F523F	MB91F524F	MB91F525F	MB91F526F
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	76 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
NMI request function	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQI100				

*1: Only channel 5, channel 6, channel 7, channel 8 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-4 Product Line-up (120 pin)

	MB91F522J	MB91F523J	MB91F524J	MB91F525J	MB91F526J
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 + 64KB	384 + 64KB	512 + 64KB	768 + 64KB	1024 + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	None				
DMA Transfer	16 channels				
16-bit Base Timer	2 channels				
Free-run Time	16-bit × 3 channels 32-bit × 3 channels				
Input capture	16-bit × 4 channels 32-bit × 6 channels				
Output Compare	16-bit × 6 channels 32-bit × 6 channels				
16-bit Reload Timer	8 channels				
PPG	16-bit × 38 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 26 channels (1 unit) 12-bit ×16 channels (1 unit)				
D/A (8-bit)	2 channels				
Multi-Function Serial	12 channels[*1]				
CAN	64msg × 2 channels / 128msg × 1 channel				
Hardware watchdog	Yes				
CRC generation	Yes				

	MB91F522J	MB91F523J	MB91F524J	MB91F525J	MB91F526J
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	96 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
NMI request function	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQM120				

*1: Only channel 3 and channel 4 support the I^2C (fast mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-5 Product Line-up (144 pin)

	MB91F522K	MB91F523K	MB91F524K	MB91F525K	MB91F526K
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 + 64KB	384 + 64KB	512 + 64KB	768 + 64KB	1024 + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	Yes				
DMA Transfer	16 channels				
16-bit Base Timer	2 channels				
Free-run Timer	16-bit × 3 channels 32-bit × 3 channels				
Input capture	16-bit × 4 channels 32-bit × 6 channels				
Output Compare	16-bit × 6 channels 32-bit × 6 channels				
16-bit Reload Timer	8 channels				
PPG	16-bit × 44 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 32 channels (1 unit) 12-bit ×16 channels (1 unit)				
D/A (8-bit)	2 channels				
Multi-Function Serial	12 channels[*1]				
CAN	64msg × 2 channels / 128msg × 1 channel				
Hardware watchdog	Yes				
CRC generation	Yes				

	MB91F522K	MB91F523K	MB91F524K	MB91F525K	MB91F526K
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	120 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
NMI request function	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQS144 / LQN144				

*1: Only channel 3 and channel 4 support the I^2C (fast mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8, channel 10 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-6 Product Line-up (176 pin)

	MB91F522L	MB91F523L	MB91F524L	MB91F525L	MB91F526L
System Clock	On-chip PLL Clock multiple method				
Minimum instruction execution time	12.5ns (80MHz)				
FLASH Capacity (Program)	256 + 64KB	384 + 64KB	512 + 64KB	768 + 64KB	1024 + 64KB
FLASH Capacity (Data)	64KB				
RAM Capacity	48KB + 8KB		64KB + 8KB	96KB + 8KB	128KB + 8KB
External Bus I/F (22 address/16 data/4cs)	Yes				
DMA Transfer	16 channels				
16-bit Base Timer	2 channels				
Free-run Timer	16-bit × 3 channels 32-bit × 3 channels				
Input capture	16-bit × 4 channels 32-bit × 6 channels				
Output Compare	16-bit × 6 channels 32-bit × 6 channels				
16-bit Reload Timer	8 channels				
PPG	16-bit × 48 channels				
Up/down Counter	2 channels				
Clock Supervisor	Yes				
External interrupt	8 channels × 2 units				
A/D	12-bit × 32 channels (1 unit) 12-bit ×16 channels (1 unit)				
D/A (8-bit)	2 channels				
Multi-Function Serial	12 channels[*1]				
CAN	64msg × 2 channels / 128msg × 1 channel				
Hardware watchdog	Yes				
CRC generation	Yes				

	MB91F522L	MB91F523L	MB91F524L	MB91F525L	MB91F526L
Low-voltage detection reset	Yes				
Flash Security	Yes				
ECC Flash/WorkFlash	Yes				
ECC RAM	Yes				
Memory Protection Function (MPU)	Yes				
Floating-point arithmetic (FPU)	Yes				
Real Time Clock (RTC)	Yes				
General-purpose port (#GPIOs)	152 ports				
SSCG	Yes				
Sub clock	Yes				
CR oscillator	Yes				
NMI request function	Yes				
OCD(On-Chip Debug)	Yes				
TPU (Timing Protection Unit)	Yes				
Key Code Register	Yes				
Wave Generator	6 channels				
Operation guaranteed temperature (Ta)	-40°C to +125°C				
Power supply	2.7 V to 5.5 V*2				
Package	LQP176				

*1: Only channel 3 and channel 4 support the I^2C (fast mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8, channel 10 and channel 11 support the I^2C (standard mode).

*2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.

Table 3-7 Correspondence table of ON/OFF for initial values of clock supervisor and external low-voltage detection reset

Clock	Initial value of clock supervisor	Initial value of external low-voltage detection reset	Function
Single	ON	ON	S
		OFF	U
	OFF	ON	H
		OFF	K
Dual	ON	ON	W
		OFF	Y
	OFF	ON	J
		OFF	L

```
MB91F52Xxyz
          └─→Revision:B, C, D, E
            └─→Function:See Table 3-7
              └─→PKG Type:B 64 pin
                          D 80 pin
                          F 100 pin
                          J 120 pin
                          K 144 pin
                          L 176 pin
                └─→Memory Size:2 256KB
                               3 384KB
                               4 512KB
                               5 768KB
                               6 1MB
```

4. Function overview

This section shows function overview of MB91520 series.

Table 4-1 : 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 (Up to 16MHz can be input) Sub oscillation : 32kHz or None PLL multiplication rate: Up to 20 times of multiplication Built-in 100kHz CR oscillator
I/O ports	Each bit can be programmed for I/O or peripheral signals Pull-up can be set.
External bus Interface	22-bit address, 8/16-bit Data output

Function	Features
Internal bus interface	On-chip bus : 32-bit, maximum operating frequency : 80MHz
Peripheral bus interface	Maximum operating frequency : 40MHz 32-bit peripheral bus, or 16-bit peripheral bus (R-bus) Note: Both of them operate in the same frequency.
Flash interface	Wild register function provided. For small sector (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 (MB91F52xD, MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL) Any of four PWM/PPG/PWC/reload timer functions can be selected and used. A 32-bit timer can be used in 2 channels of cascade mode for the reload timer/PWC function.
Free-run timer	16-bit/32-bit up counter Free-run timer ch.0 to ch.2 : 16-bit Free-run timer ch.3 to ch.5 : 32-bit
Input capture	16-bit/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 free-run timer is latched and an interrupt request is generated. Cooperation with the free-run timer is as follows. Input capture ch.0 to ch.3 : 16-bit → Free-run timer ch.0 to ch.2 Input capture ch.4 to ch.9 : 32-bit → Free-run timer ch.3 to ch.5 Cooperation with LIN synch break/synch field is as follows. Input capture ch.0 → Multi-function serial ch.0 Input capture ch.1 → Multi-function serial ch.1 Input capture ch.2 → Multi-function serial ch.2 Input capture ch.3 → Multi-function serial ch.3 Input capture ch.4 → Multi-function serial ch.4 Input capture ch.5 → Multi-function serial ch.5 Input capture ch.6 → Multi-function serial ch.6 Input capture ch.7 → Multi-function serial ch.7 Input capture ch.8 → Multi-function serial ch.8, ch.9 Input capture ch.9 → Multi-function serial ch.10, ch.11 Built-in cycle/pulse width measurement function (only 32-bit supported)

Function	Features
Output compare	An interrupt signal is output during collating with the 16-bit/32-bit free-run timer. Cooperation with the free-run timer is as follows. Output compare ch.0 to ch.5 : 16-bit → Free-run timer ch.0 to ch.2 Output compare ch.6 to ch.11 : 32-bit → Free-run timer ch.3 to ch.5 Built-in compare level control function (only 32-bit supported) 16-bit output compare has no dedicated output pins. There is only the output through the wave generator.
Reload timer	16-bit reload timer operation (The toggle output or one-shot output can be selected) Event count function can be selected.
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.
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, $1/2^2$, $1/2^4$, $1/2^6$ Strat Delay of each channel can be set. It is possible to use it as activation trigger of A/D Converter. The cycle of the High format and the Low format and duty can be set.
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 2 units of resolution in 12-bit Able to sample the analog value from up to 48 channels input port Conversion time : 12-bit A/D Converter 1.4µs External trigger activation Can be activated by an internal timer (16-bit reload timer/compare match/PPG are used). Has the function of selecting the sampling time for each channel. Built-in range comparator Has the function of selecting the sampling time for each channel.

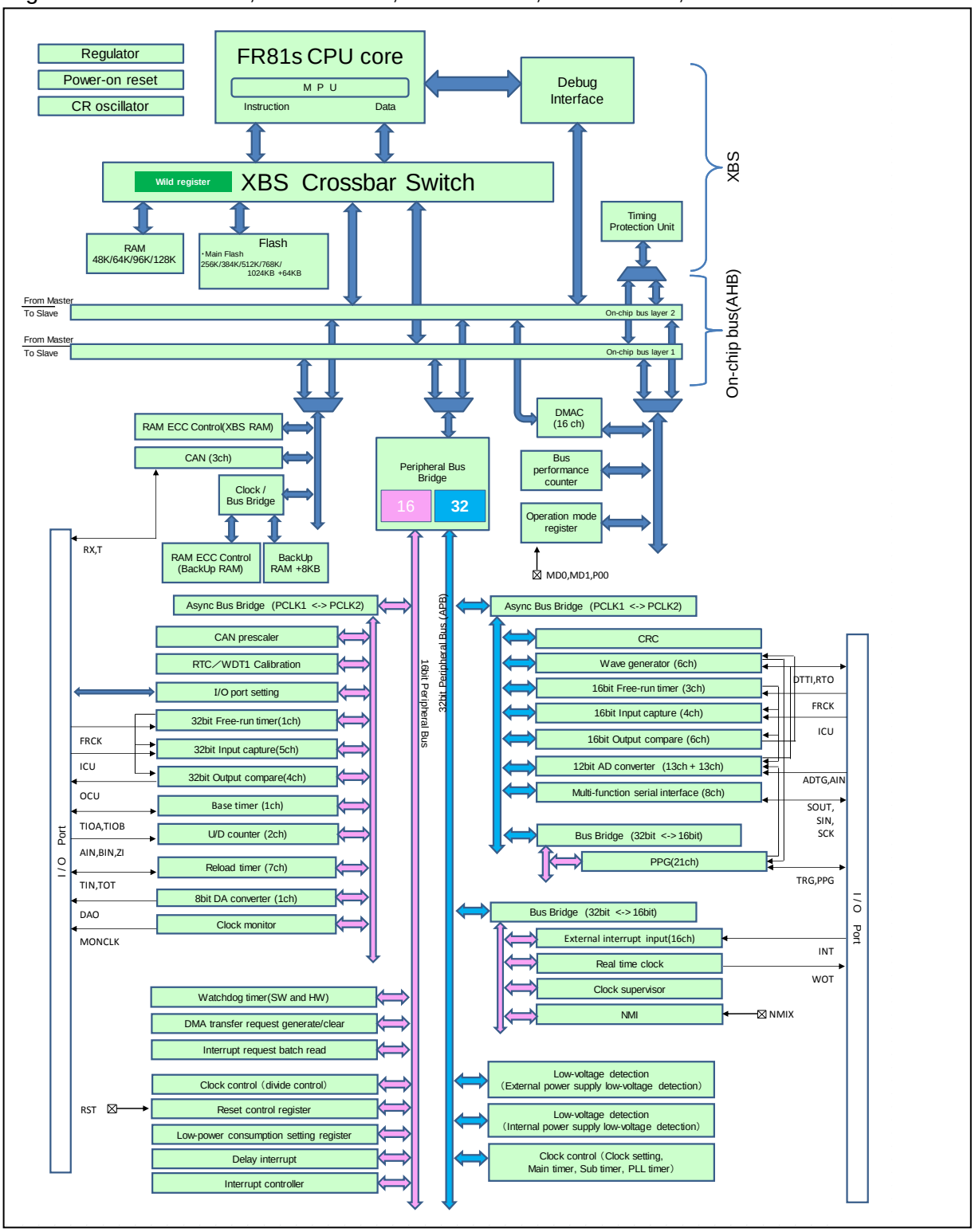
Function	Features
D/A Converter	Built-in D/A converter 2 channels of resolution in 8-bit
Multi-function serial	Any of UART/CSIO/LIN /I^2C-UART functions can be selected and used. Transmission FIFO memory 16-byte, and reception FIFO memory 16-byte provided Reception interrupt cause (3 types) - Reception error detection (parity, overrun, and frame error) - Detects FIFO'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 I^2C (ch.3, ch.4) 100kbps and 400kbps supported I^2C (ch.5 to ch.8, ch.10, ch.11) only 100kbps supported
Interrupt controller	Detects an interrupt request. Sets an interrupt level.
Interrupt request batch read	A generation of multiple interrupts from peripherals can be read by a series of registers.
CAN interface	CAN Specifications Version 2.0, Part A and Part B satisfied Up to 64 message buffers × 2channel, 128 message buffers × 1channels 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 CAN wakeup function CAN clock source can switch main clock/PLL clock.
U/D counter (2 channels)	8/16-bit up/counter × 2channels
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). The lower limit of the term of validity to clear can be set up to 16 ways.
Hardware watchdog	RC-based CPU operation detection counter Used against program overrun Period: 218ms to 655ms (usually 328ms, depending on the accuracy of the CR oscillation) Note that as shown above, a period of the CR oscillation clock varies widely due to the production process.

Function	Features
CRC generation	When data is sequentially written in the input registers, the CRC code is displayed in the result register.
External low-voltage detection reset	Reset/interrupt generation at external low-voltage detection When an external power-supply voltage falls below the detection voltage value, reset/interrupt is generated. The detecting voltage (2.8 V to 4.3 V) is possible to select 11 types.
Internal low-voltage detection reset	Reset generation at internal low-voltage detection Monitors 1.2V power supply and generates 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 are shown below. A/D converter external trigger 2 channels× Max. 2 divergences CAN (Max. 2 divergences for ch.0, Max. 2 divergences for ch.1) External interrupt (Max. 2 divergences for ch.1 to ch.4, ch.7, ch.9, ch.13, ch.14) Multi-function serial (Max. 2 divergences for ch.0 to ch.2, ch.10; Max. 3 divergences for ch.3 and ch.4) Note that the I^2C cannot be relocated. Serial chip select input (Max. 2 divergences for ch.1, ch.3, ch.4, ch.10) PPG (Max. 2 divergences for ch.0 to ch.5, ch.16, ch.17, ch.23 to ch.41, ch.43, ch.44) U/D counter 2 channels× 3 divergences Output compare (Max. 2 divergences for ch.6 to ch.11) Input capture (Max. 4 divergences for ch.0 to ch.3; Max. 3 divergences for ch.4; Max. 2 divergences for ch.5 to ch.9) Free-run timer (Max. 2 divergences for ch.1, ch.3, ch.4, ch.5) Base timer 2 channels× 2 divergences Reload timer (Max. 3 divergences for ch.0, ch.1, ch.2, ch.3; Max. 2 divergences for ch.2, ch.4, ch.5, ch.6, ch.7) Wave Generator (Max. 2 divergences for ch.0 to ch.5; Max. 3 divergences for DTTI input)
NMI request	Non-maskable interrupt signal that is entered from NMIX pin.
Debug interface	Built-in OCD

5. Block Diagram

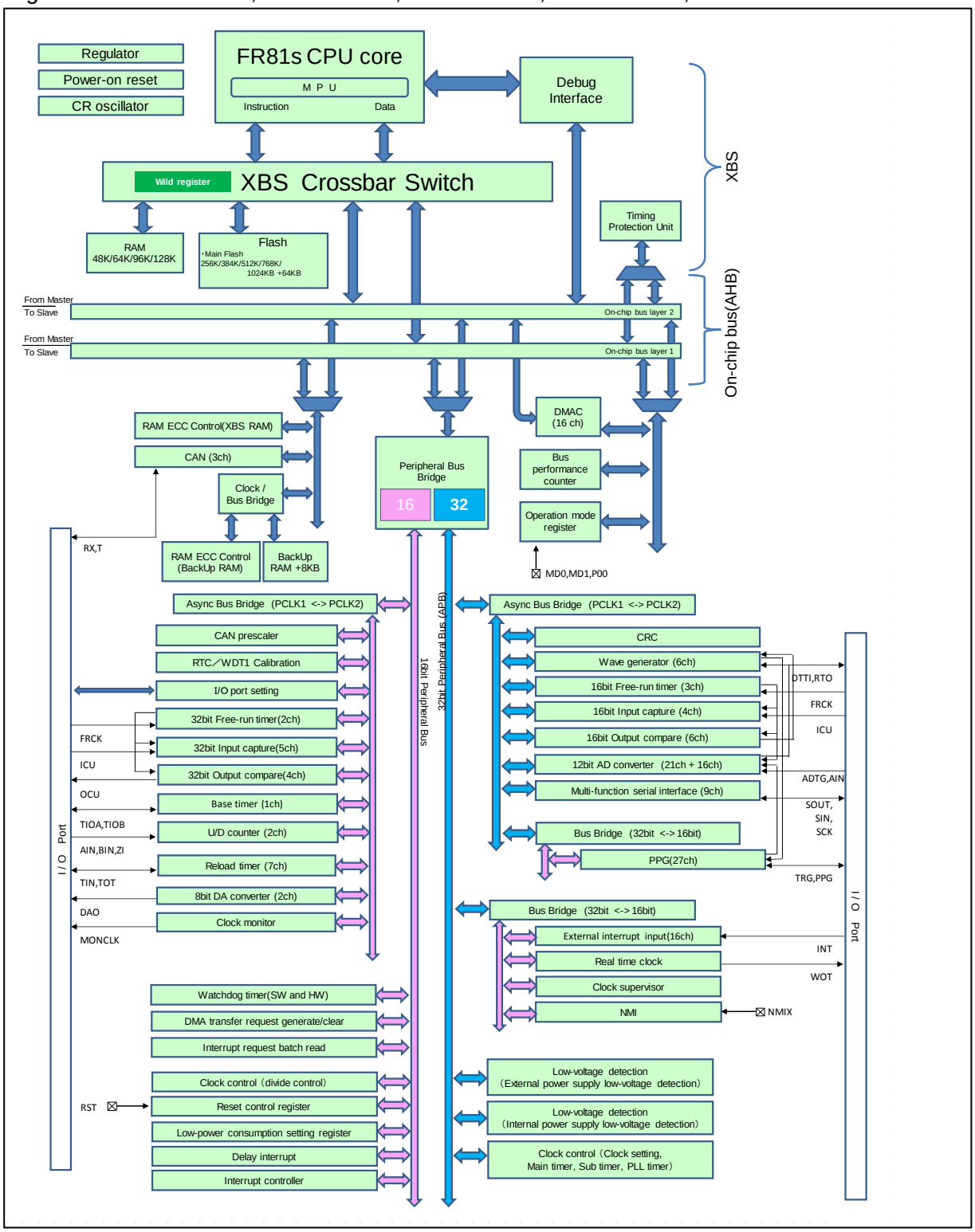
This section shows block diagram of MB91520 series.

Figure 5-1 MB91F522B, MB91F523B, MB91F524B, MB91F525B, MB91F526B



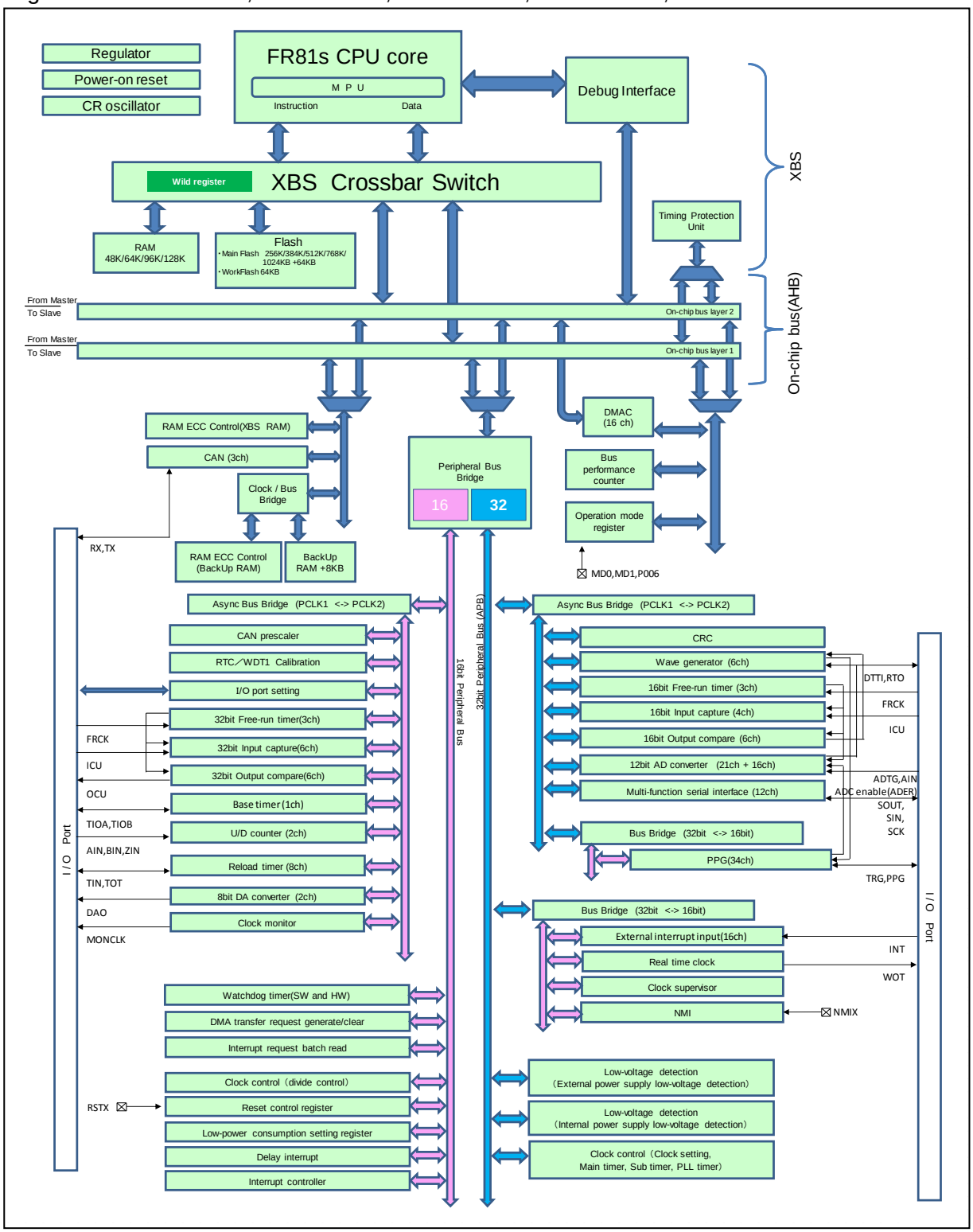
See "9.1 Pins of Each Function" for pins that can used by each function.

Figure 5-2 MB91F522D, MB91F523D, MB91F524D, MB91F525D, MB91F526D



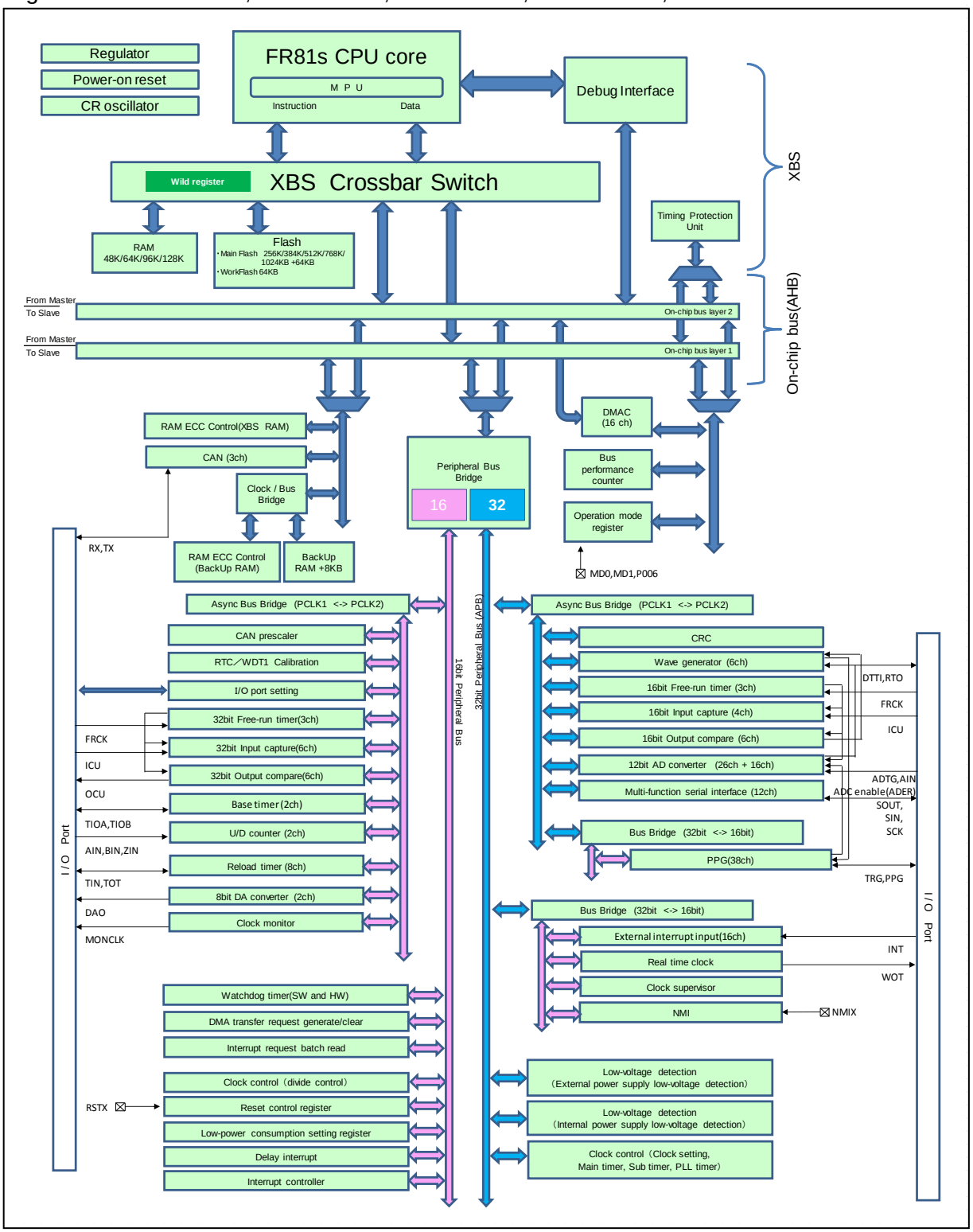
See "9.1 Pins of Each Function" for pins that can used by each function.

Figure 5-3 MB91F522F, MB91F523F, MB91F524F, MB91F525F, MB91F526F



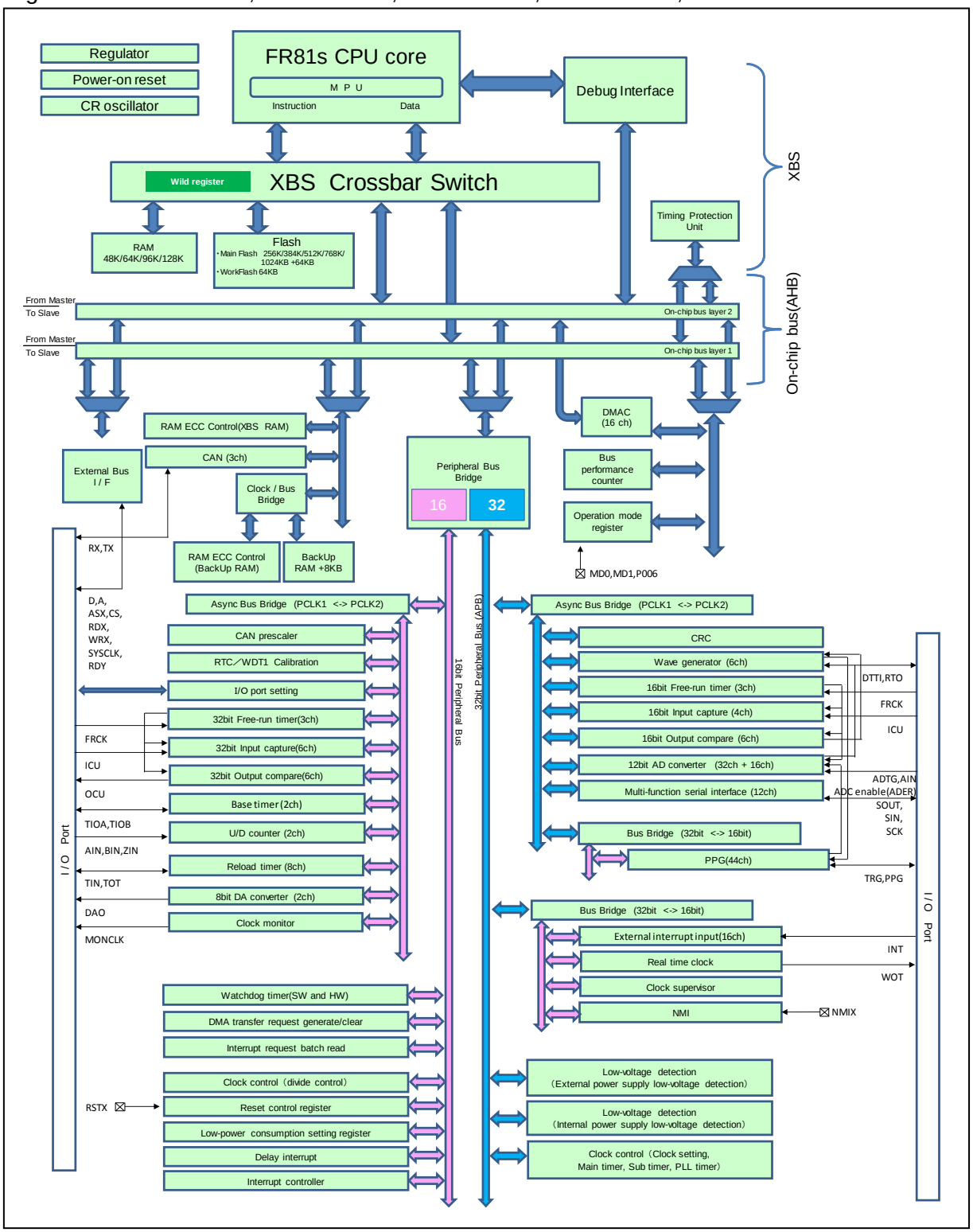
See "9.1 Pins of Each Function" for pins that can used by each function.

Figure 5-4 MB91F522J, MB91F523J, MB91F524J, MB91F525J, MB91F526J



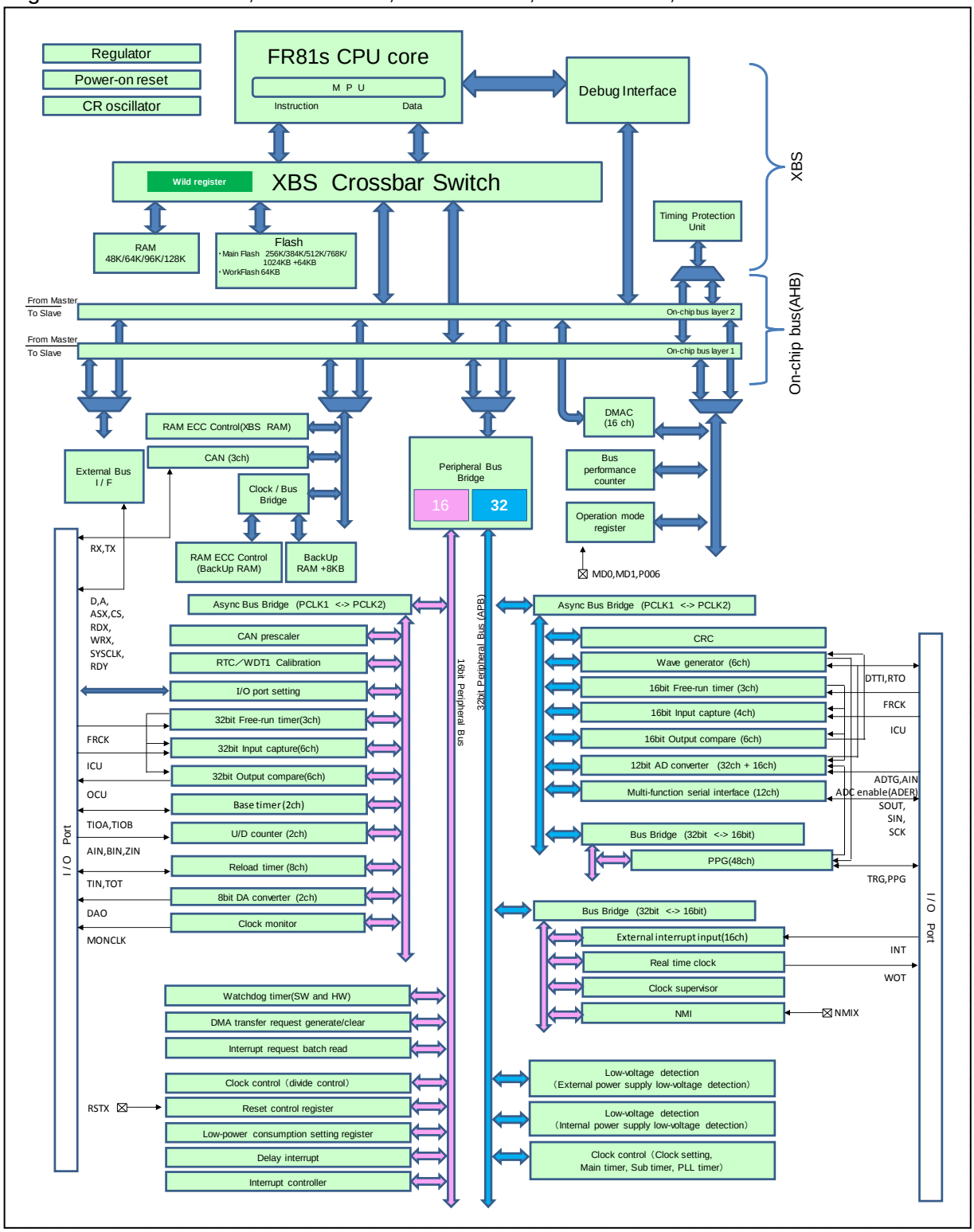
See "9.1 Pins of Each Function" for pins that can used by each function.

Figure 5-5 MB91F522K, MB91F523K, MB91F524K, MB91F525K, MB91F526K



See "9.1 Pins of Each Function" for pins that can used by each function.

Figure 5-6 MB91F522L, MB91F523L, MB91F524L, MB91F525L, MB91F526L



See "9.1 Pins of Each Function" for pins that can used by each function.

6. Memory Map

This section shows memory map of MB91520 series.

Figure 6-1 Memory Map MB91F522, MB91F523, MB91F524

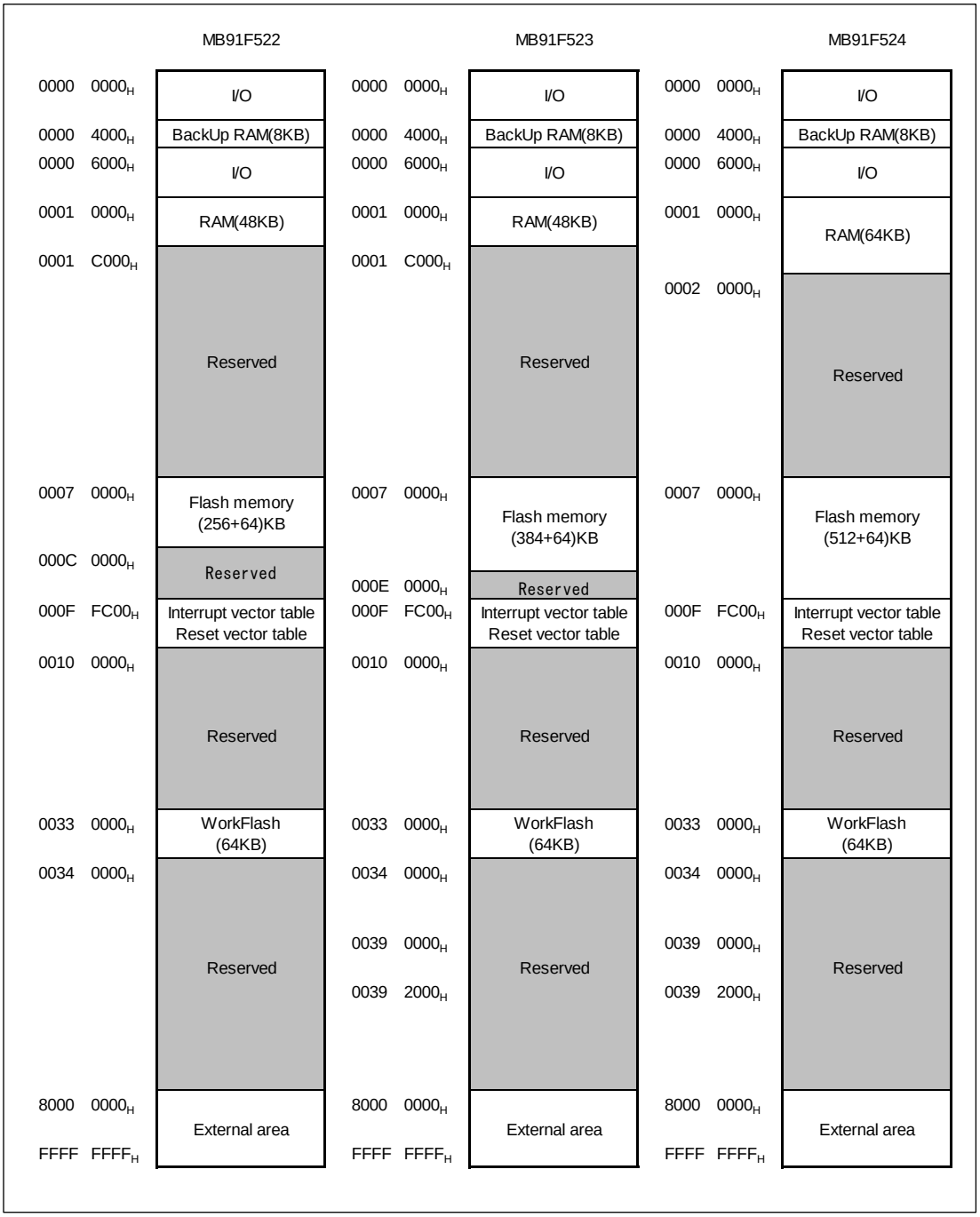
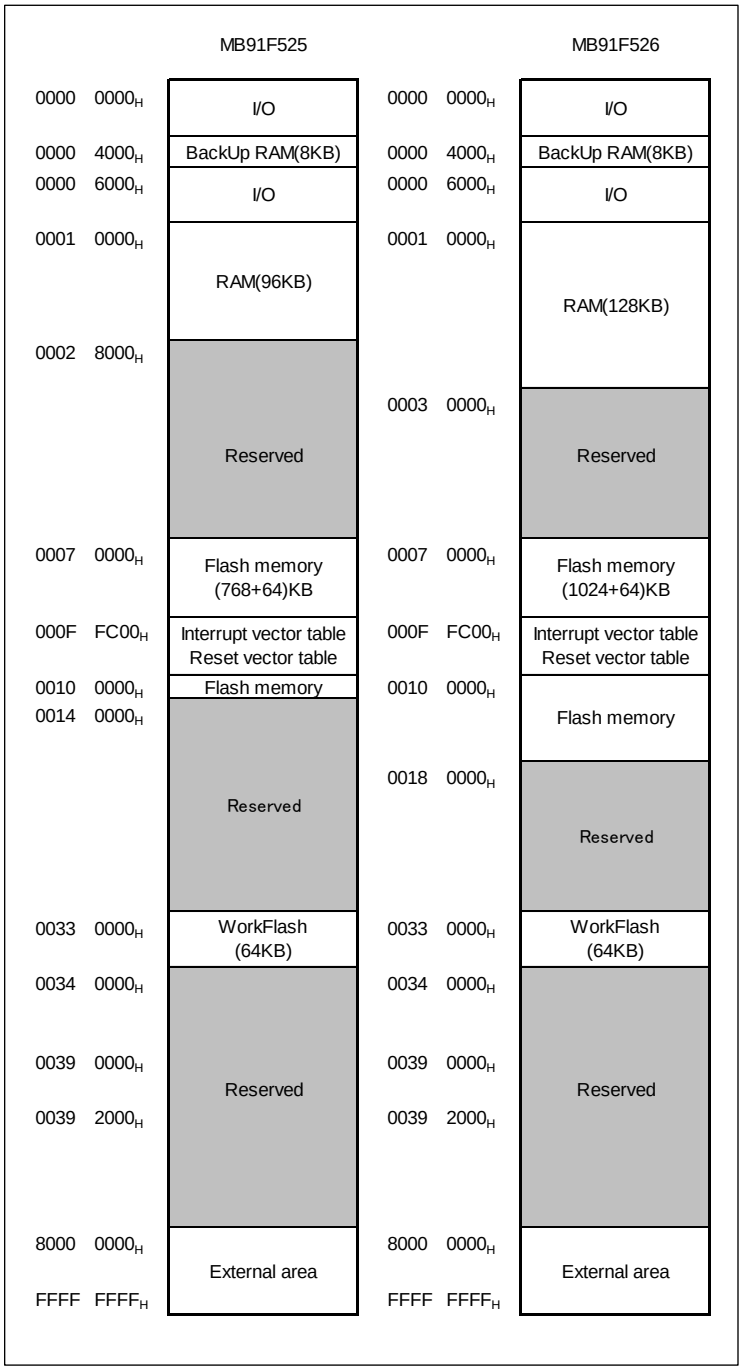


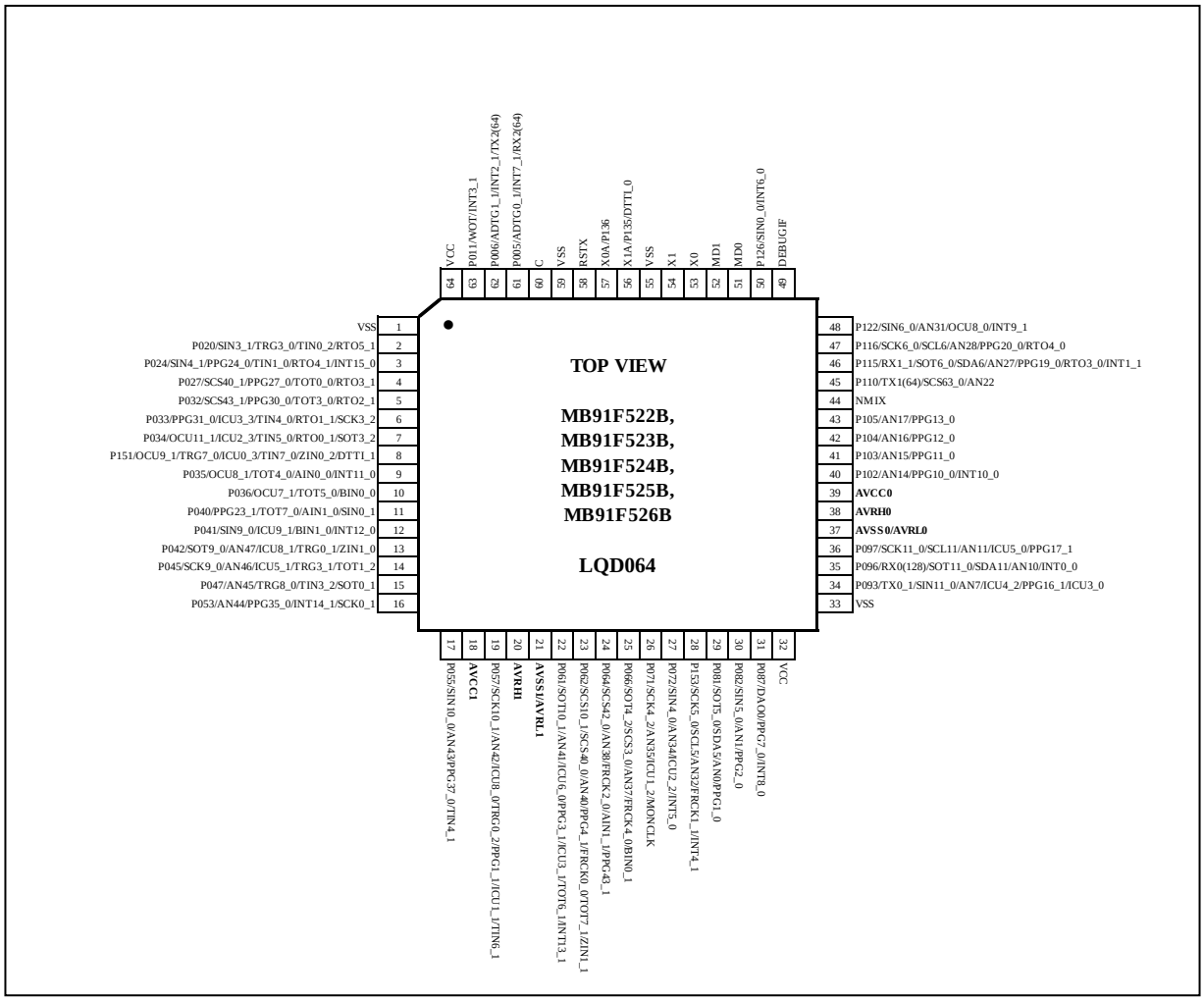
Figure 6-2 Memory Map MB91F525, MB91F526



7. Pin Assignment

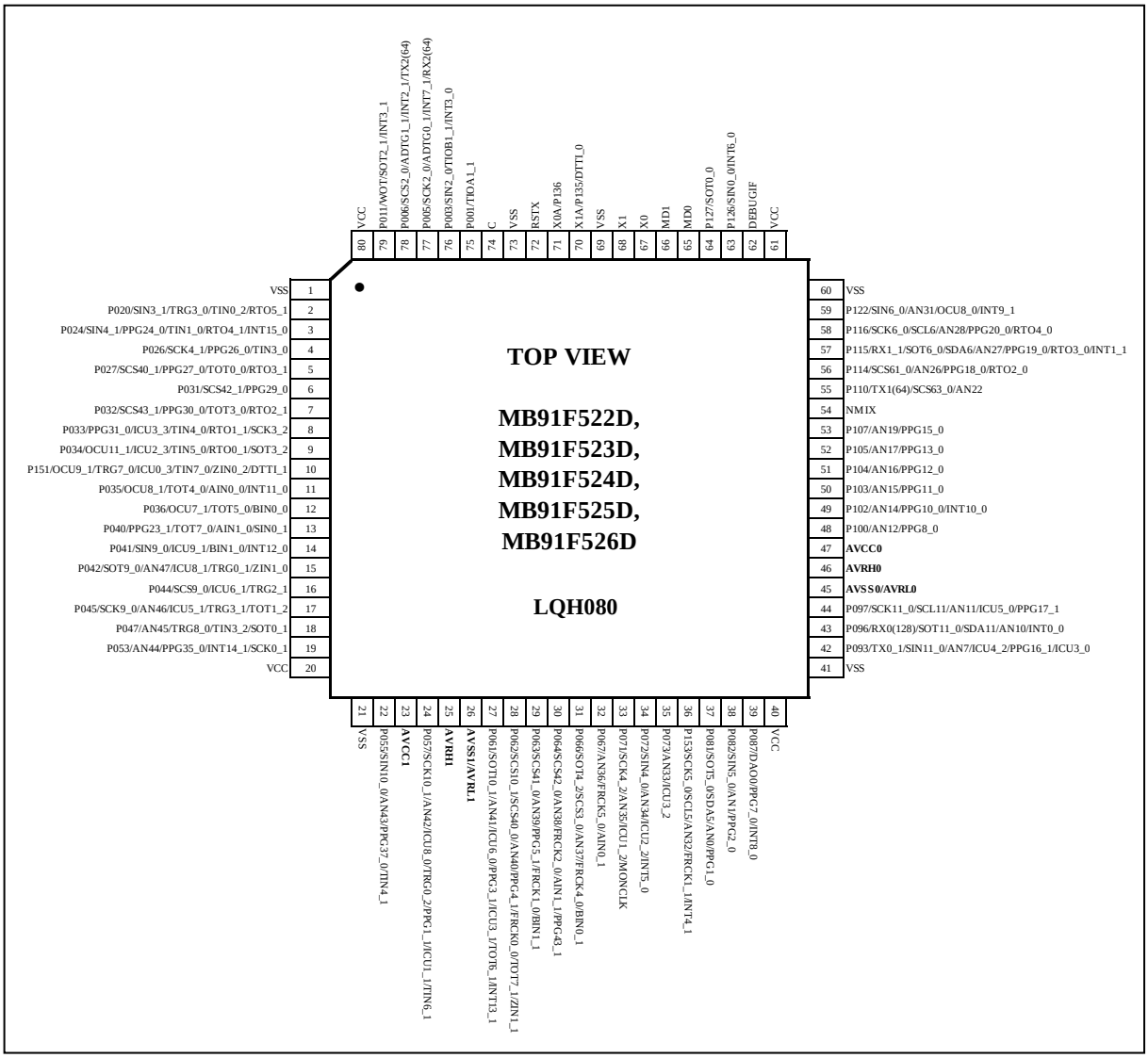
This section shows pin assignment of MB91520 series.

Figure 7-1 Pin Assignment MB91F522B, MB91F523B, MB91F524B, MB91F525B, MB91F526B



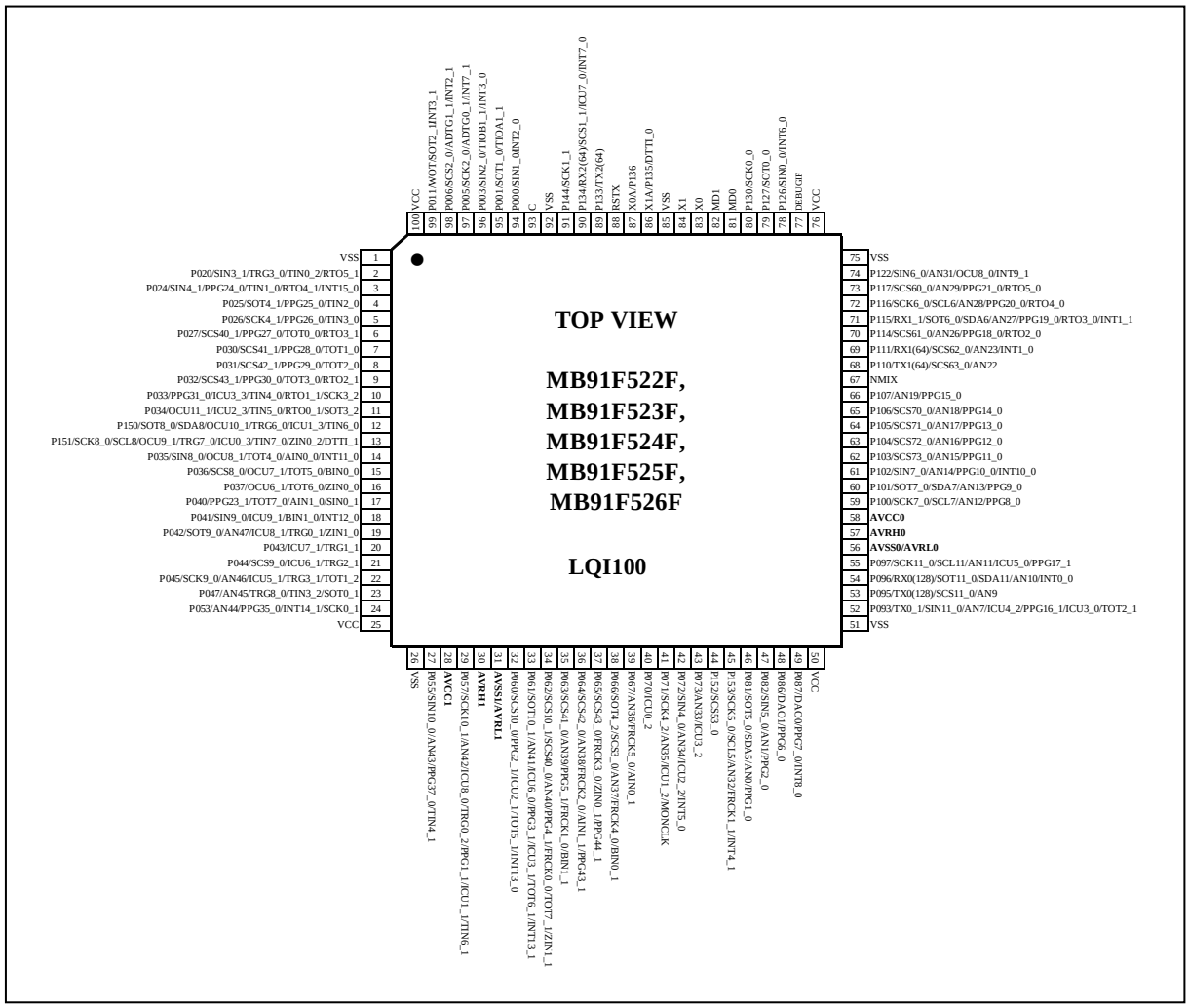
* In a single clock product, pin 56 and pin 57 are the general-purpose ports.

Figure 7-2 Pin Assignment MB91F522D, MB91F523D, MB91F524D, MB91F525D, MB91F526D



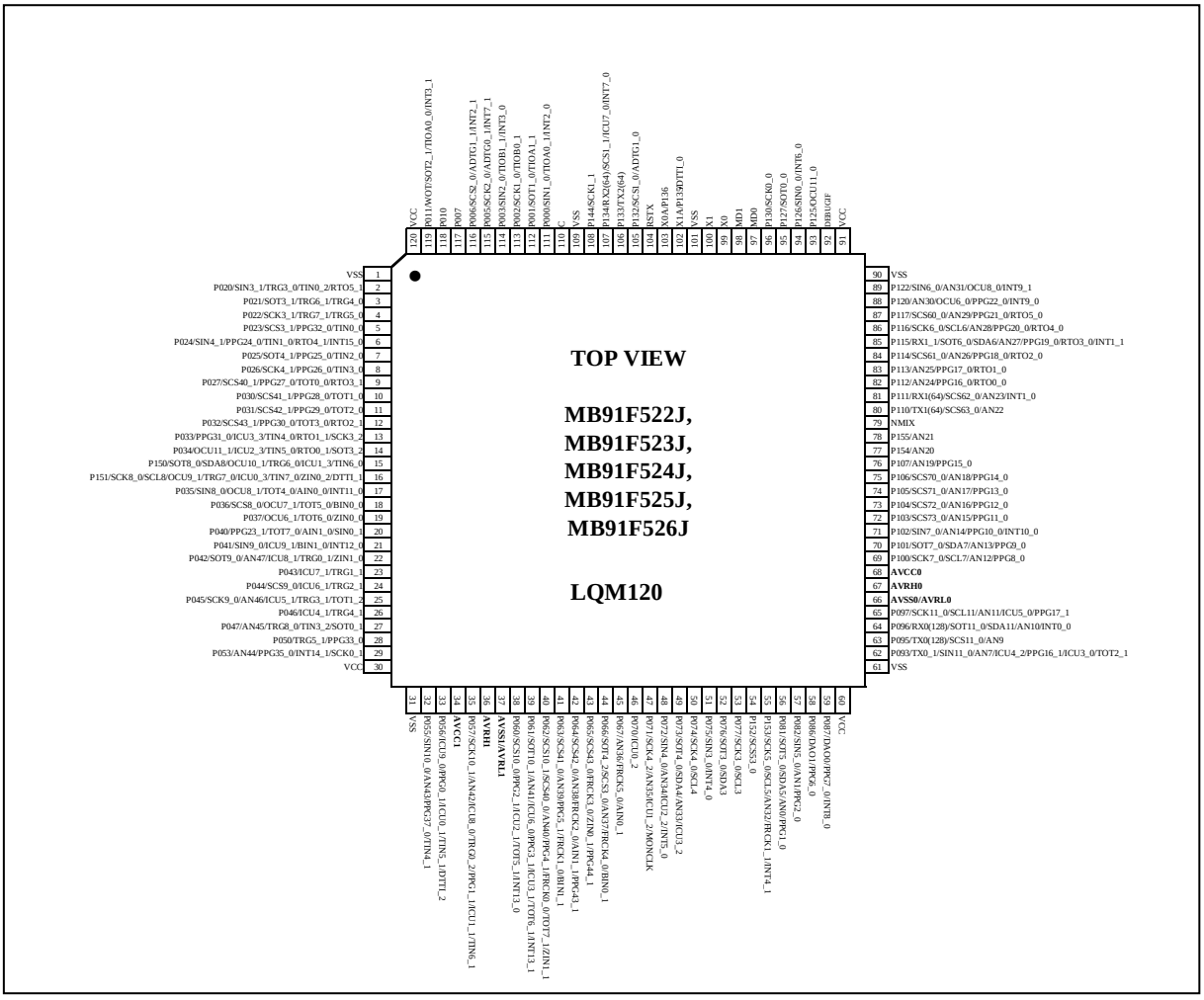
* In a single clock product, pin 70 and pin 71 are the general-purpose ports.

Figure 7-3 Pin Assignment MB91F522F, MB91F523F, MB91F524F, MB91F525F, MB91F526F



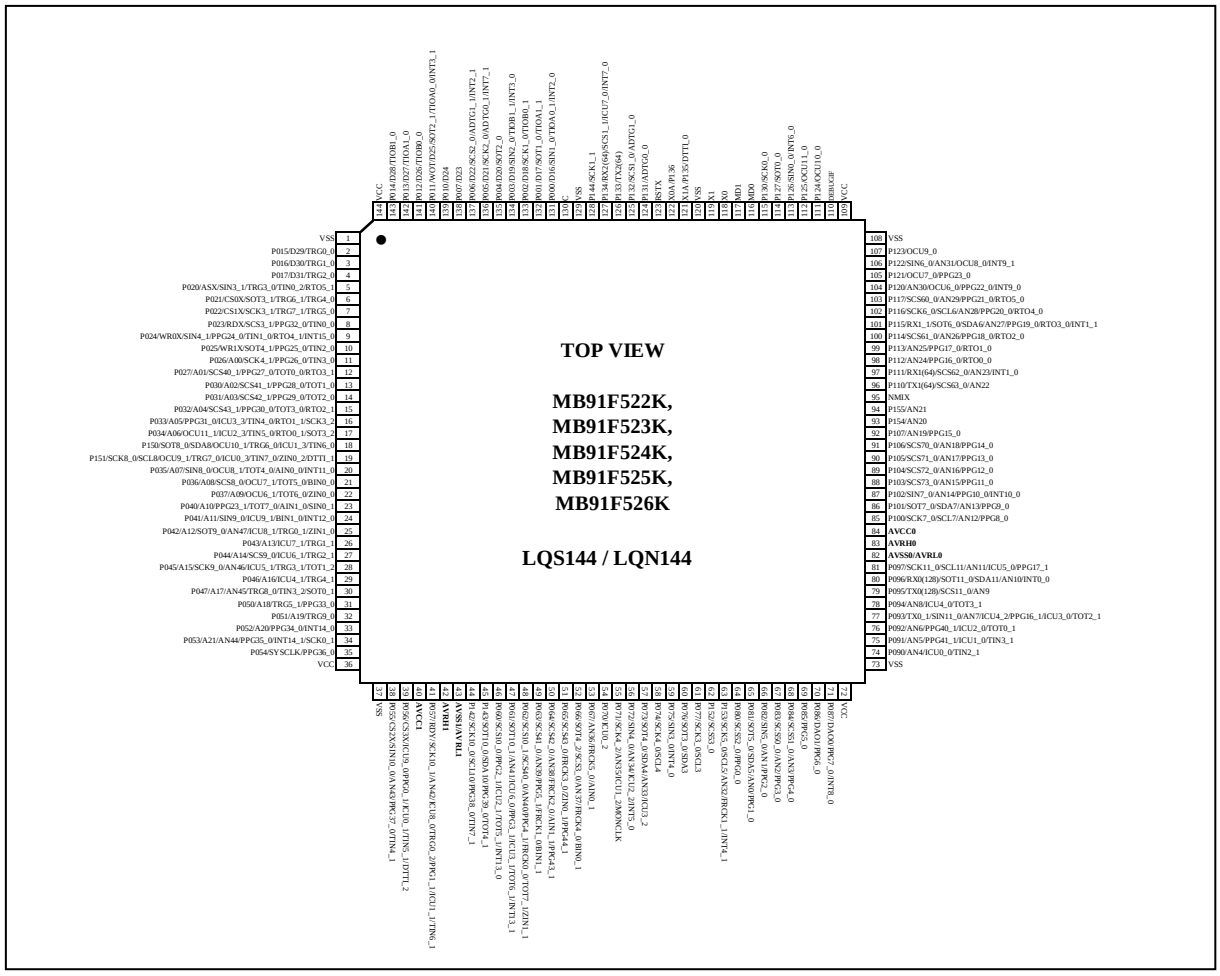
* In a single clock product, pin 86 and pin 87 are the general-purpose ports.

Figure 7-4 Pin Assignment MB91F522J, MB91F523J, MB91F524J, MB91F525J, MB91F526J



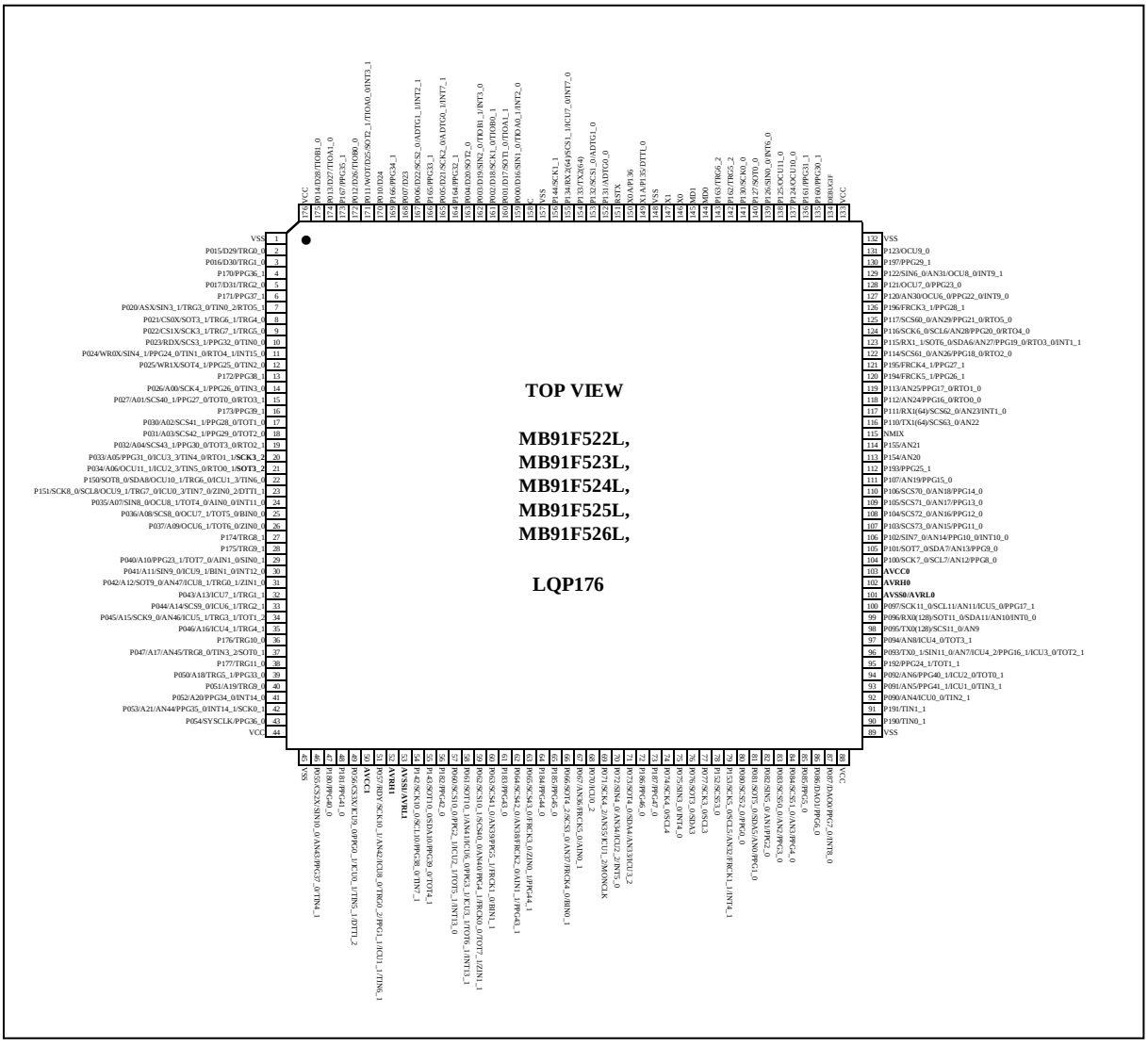
* In a single clock product, pin 102 and pin 103 are the general-purpose ports.

Figure 7-5 Pin Assignment MB91F522K, MB91F523K, MB91F524K, MB91F525K, MB91F526K



* In a single clock product, pin 121 and pin 122 are the general-purpose ports.

Figure 7-6 Pin Assignment MB91F522L, MB91F523L, MB91F524L, MB91F525L, MB91F526L



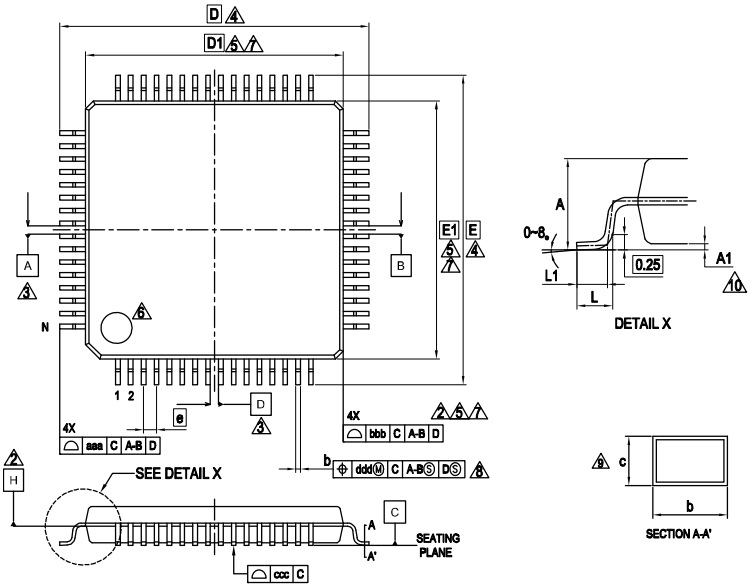
* In a single clock product, pin 149 and pin 150 are the general-purpose ports.

8. Device Package

This section explains device package of MB91520 series.

Figure 8-1 LQD064 External Dimensions

LQD064 , 64 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQD64		
SYMBOL	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.00	—	0.20
b	0.15	0.20	0.25
c	0.09	—	0.20
D	12.00 BSC.		
D1	10.00 BSC.		
e	0.50 BSC		
E	12.00 BSC.		
E1	10.00 BSC.		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.08
N	64		

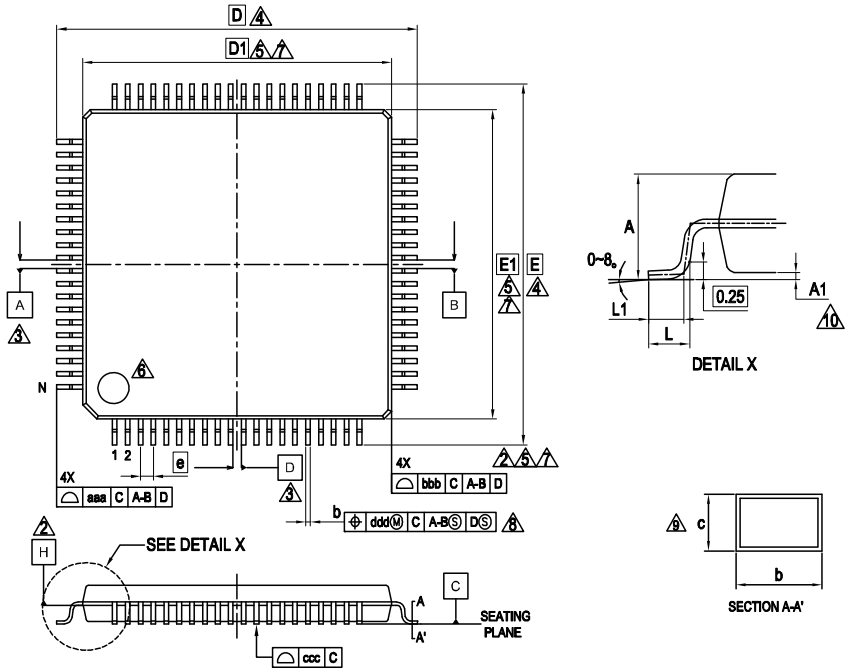
NOTES

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.

Rev. A

Figure 8-2 LQH080 External Dimensions

LQH080 , 80 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQH080		
SYMBOL	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.05	—	0.15
b	0.15	0.20	0.25
c	0.09	—	0.20
D	14.00 BSC.		
D1	12.00 BSC.		
e	0.50 BSC		
E	14.00 BSC.		
E1	12.00 BSC.		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.08
N	80		

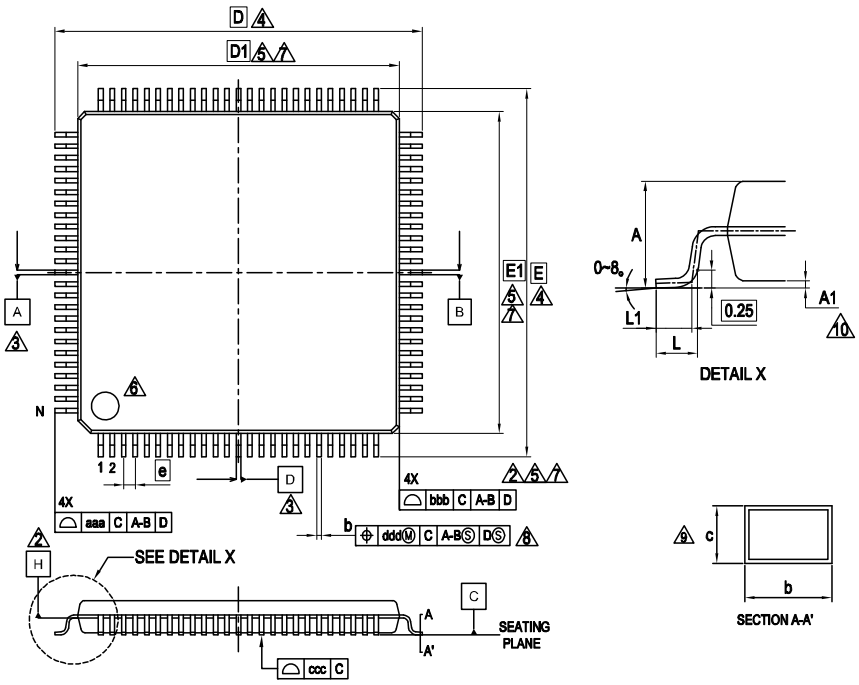
NOTES

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.

Rev. A

Figure 8-3 LQI100 External Dimensions

LQI100 , 100 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQI100		
SYMBOL	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.00	—	0.20
b	0.15	0.20	0.25
c	0.09	—	0.20
D	16.00 BSC.		
D1	14.00 BSC.		
e	0.50 BSC		
E	16.00 BSC.		
E1	14.00 BSC.		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.08
N	100		

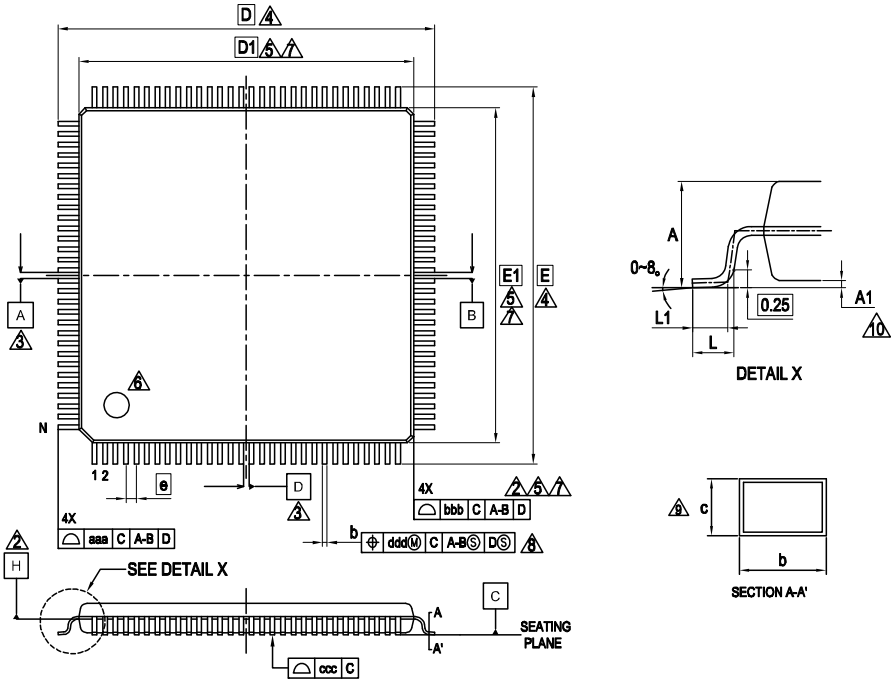
NOTES

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.

Rev. A

Figure 8-4 LQM120 External Dimensions

LQM120 , 120 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQM120		
SYMBOL	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.05	—	0.15
b	0.17	0.22	0.27
c	0.115	—	0.195
D	18.00 BSC.		
D1	16.00 BSC.		
e	0.50 BSC		
E	18.00 BSC.		
E1	16.00 BSC.		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.08
N	120		

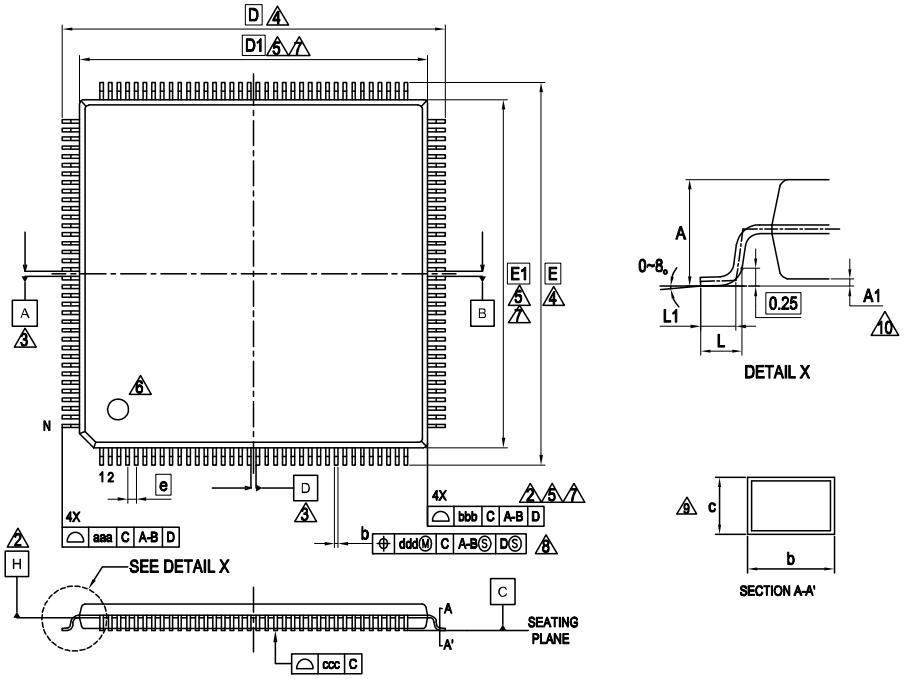
NOTES

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.

Rev. A

Figure 8-5 LQS144 External Dimensions

LQS144 , 144 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQS144		
SYMBOL	MIN.	NOM.	MAX.
A	——	——	1.70
A1	0.06	——	0.26
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
aaa	——	——	0.20
bbb	——	——	0.10
ccc	——	——	0.08
ddd	——	——	0.08
N	144		

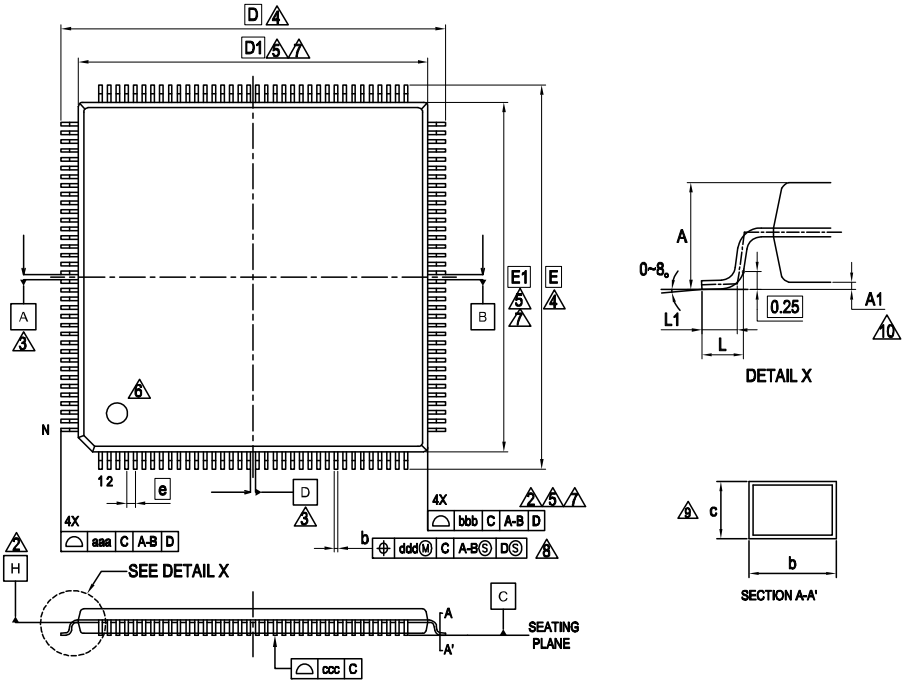
NOTES

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.

Rev. A

Figure 8-6 LQN144 External Dimensions

LQN144 , 144 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQN144		
SYMBOL	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
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.07
N	144		

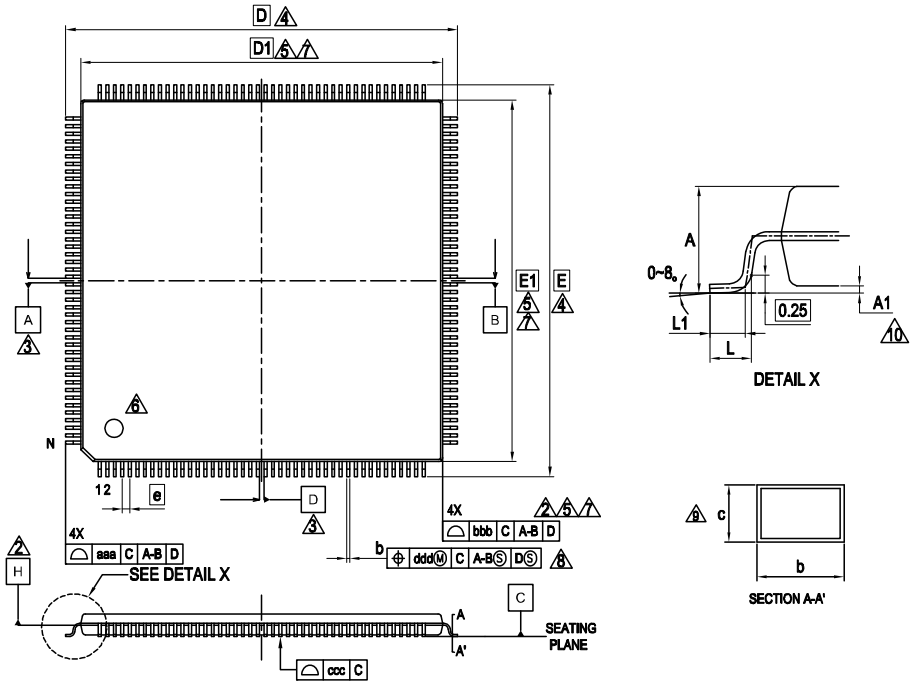
NOTES

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. 0°
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.

Rev. A

Figure 8-7 LQP176 External Dimensions

LQP176 , 176 Lead Plastic Low Profile Quad Flat Package



PACKAGE	LQP176		
SYMBOL	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.00	—	0.20
b	0.17	0.22	0.27
c	0.09	—	0.20
D	26.00 BSC.		
D1	24.00 BSC.		
e	0.50 BSC		
E	26.00 BSC.		
E1	24.00 BSC.		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
aaa	—	—	0.20
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.08
N	176		

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.

Rev. A

9. Explanation of Pin Functions

The pin function list of MB91520 series is shown below.

Table 9-1 List of Pin Functions

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	-	2	2	P015 D29 TRG0_0	- - -	A	General-purpose I/O port External Bus data bit29 I/O pin(0) PPG trigger 0 input pin(0)
-	-	-	-	3	3	P016 D30 TRG1_0	- - -	A	General-purpose I/O port External Bus data bit30 I/O pin(0) PPG trigger 1 input pin(0)
-	-	-	-	-	4	P170 PPG36_1	- -	A	General-purpose I/O port PPG ch.36 output pin(1)
-	-	-	-	4	5	P017 D31 TRG2_0	- - -	A	General-purpose I/O port External Bus data bit31 I/O pin(0) PPG trigger 2 input pin(0)
-	-	-	-	-	6	P171 PPG37_1	- -	A	General-purpose I/O port PPG ch.37 output pin(1)
2 *1	2 *1	2 *1	2 *1	5	7	P020 ASX *2, *3, *4, *5 SIN3_1 TRG3_0 TIN0_2 RTO5_1	- - - - - -	F	General-purpose I/O port External Bus address strobe output pin Multi-function serial ch.3 serial data input pin(1) PPG trigger 3 input pin(0) Reload timer ch.0 event input pin(2) Waveform generator ch.5 output pin(1)
-	-	-	3 *1	6	8	P021 CS0X *5 SOT3_1 TRG6_1 TRG4_0	- - - - -	A	General-purpose I/O port External Bus chip select 0 output pin Multi-function serial ch.3 serial data output pin(1) PPG trigger 6 input pin(1) PPG trigger 4 input pin(0)
-	-	-	4 *1	7	9	P022 CS1X *5 SCK3_1 TRG7_1 TRG5_0	- - - - -	F	General-purpose I/O port External Bus chip select 1 output pin Multi-function serial ch.3 clock I/O pin(1) PPG trigger 7 input pin(1) PPG trigger 5 input pin(0)
-	-	-	5 *1	8	10	P023 RDX *5 SCS3_1 PPG32_0 TIN0_0	- - - - -	A	General-purpose I/O port External Bus read strobe output pin Serial chip select 3 output pin(1) PPG ch.32 output pin(0) Reload timer ch.0 event input pin(0)
3 *1	3 *1	3 *1	6 *1	9	11	P024 WR0X *2, *3, *4, *5 SIN4_1 PPG24_0 TIN1_0 RTO4_1 INT15_0	- - - - - - -	F	General-purpose I/O port External Bus write strobe 0 output pin Multi-function serial ch.4 serial data input pin(1) PPG ch.24 output pin(0) Reload timer ch.1 event input pin(0) Waveform generator ch.4 output pin(1) INT15 external interrupt input pin(0)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	4 *1	7 *1	10	12	P025 WR1X *4, *5 SOT4_1 PPG25_0 TIN2_0	- - - - -	A	General-purpose I/O port External Bus write strobe 1 output pin Multi-function serial ch.4 serial data output pin(1) PPG ch.25 output pin(0) Reload timer ch.2 event input pin(0)
-	-	-	-	-	13	P172 PPG38_1	- -	A	General-purpose I/O port PPG ch.38 output pin(1)
-	4 *1	5 *1	8 *1	11	14	P026 A00 *3, *4, *5 SCK4_1 PPG26_0 TIN3_0	- - - - -	F	General-purpose I/O port External Bus address bit0 output pin(0) Multi-function serial ch.4 clock I/O pin(1) PPG ch.26 output pin(0) Reload timer ch.3 event input pin(0)
4 *1	5 *1	6 *1	9 *1	12	15	P027 A01 *2, *3, *4, *5 SCS40_1 PPG27_0 TOT0_0 RTO3_1	- - - - - -	A	General-purpose I/O port External Bus address bit1 output pin(0) Serial chip select 40 I/O pin(1) PPG ch.27 output pin(0) Reload timer ch.0 output pin(0) Waveform generator ch.3 output pin(1)
-	-	-	-	-	16	P173 PPG39_1	- -	A	General-purpose I/O port PPG ch.39 output pin(1)
-	-	7 *1	10 *1	13	17	P030 A02 *4, *5 SCS41_1 PPG28_0 TOT1_0	- - - - -	A	General-purpose I/O port External Bus address bit2 output pin(0) Serial chip select 41 output pin(1) PPG ch.28 output pin(0) Reload timer ch.1 output pin(0)
-	6 *1	8 *1	11 *1	14	18	P031 A03 *3, *4, *5 SCS42_1 PPG29_0 TOT2_0 *3	- - - - -	A	General-purpose I/O port External Bus address bit3 output pin(0) Serial chip select 42 output pin(1) PPG ch.29 output pin(0) Reload timer ch.2 output pin(0)
5 *1	7 *1	9 *1	12 *1	15	19	P032 A04 *2, *3, *4, *5 SCS43_1 PPG30_0 TOT3_0 RTO2_1	- - - - - -	A	General-purpose I/O port External Bus address bit4 output pin(0) Serial chip select 43 output pin(1) PPG ch.30 output pin(0) Reload timer ch.3 output pin(0) Waveform generator ch.2 output pin(1)
6 *1	8 *1	10 *1	13 *1	16	20	P033 A05 *2, *3, *4, *5 PPG31_0 ICU3_3 TIN4_0 RTO1_1 SCK3_2	- - - - - - -	A	General-purpose I/O port External Bus address bit5 output pin(0) PPG ch.31 output pin(0) Input capture ch.3 input pin(3) Reload timer ch.4 event input pin(0) Waveform generator ch.1 output pin(1) Multi-function serial ch.3 clock I/O pin(2)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
7 *1	9 *1	11 *1	14 *1	17	21	P034 A06 *2, *3, *4, *5 OCU11_1 ICU2_3 TIN5_0 RTO0_1 SOT3_2	- - - - - - -	A	General-purpose I/O port External Bus address bit6 output pin(0) Output compare ch.11 output pin(1) Input capture ch.2 input pin(3) Reload timer ch.5 event input pin(0) Waveform generator ch.0 output pin(1) Multi-function serial ch.3 serial data output pin(2)
-	-	12	15	18	22	P150 SOT8_0/SDA8 OCU10_1 TRG6_0 ICU1_3 TIN6_0	- - - - - -	F	General-purpose I/O port Multi-function serial ch.8 serial data output pin(0)/ I^2C bus serial data I/O pin Output compare ch.10 output pin(1) PPG trigger 6 input pin(0) Input capture ch.1 input pin(3) Reload timer ch.6 event input pin(0)
8 *1	10 *1	13	16	19	23	P151 SCK8_0/SCL8 *2, *3 OCU9_1 TRG7_0 ICU0_3 TIN7_0 ZIN0_2 DTTI_1	- - - - - - - -	F	General-purpose I/O port Multi-function serial ch.8 clock I/O pin(0)/ I^2C bus serial clock I/O pin Output compare ch.9 output pin(1) PPG trigger 7 input pin(0) Input capture ch.0 input pin(3) Reload timer ch.7 event input pin(0) U/D counter ch.0 ZIN input pin(2) Waveform generator ch.0-ch.5 input pin(1)
9 *1	11 *1	14 *1	17 *1	20	24	P035 A07 *2, *3, *4, *5 SIN8_0 *2, *3 OCU8_1 TOT4_0 AIN0_0 INT11_0	- - - - - - -	I	General-purpose I/O port External Bus address bit7 output pin Multi-function serial ch.8 serial data input pin(0) Output compare ch.8 output pin(1) Reload timer ch.4 output pin(0) U/D counter ch.0 AIN input pin(0) INT11 external interrupt input pin(0)
10 *1	12 *1	15 *1	18 *1	21	25	P036 A08 *2, *3, *4, *5 SCS8_0 *2, *3 OCU7_1 TOT5_0 BIN0_0	- - - - - -	A	General-purpose I/O port External Bus address bit8 output pin(0) Serial chip select 8 I/O pin(0) Output compare ch.7 output pin(1) Reload timer ch.5 output pin(0) U/D counter ch.0 BIN input pin(0)
-	-	16 *1	19 *1	22	26	P037 A09 *4, *5 OCU6_1 TOT6_0 ZIN0_0	- - - - -	A	General-purpose I/O port External Bus address bit9 output pin(0) Output compare ch.6 output pin(1) Reload timer ch.6 output pin(0) U/D counter ch.0 ZIN input pin(0)
-	-	-	-	-	27	P174 TRG8_1	- -	A	General-purpose I/O port PPG trigger 8 input pin(1)
-	-	-	-	-	28	P175 TRG9_1	- -	A	General-purpose I/O port PPG trigger 9 input pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
11 *1	13 *1	17 *1	20 *1	23	29	P040 A10 *2, *3, *4, *5 PPG23_1 TOT7_0 AIN1_0 SIN0_1	- - - - - -	A	General-purpose I/O port External Bus address bit10 output pin(0) PPG ch.23 output pin(1) Reload timer ch.7 output pin(0) U/D counter ch.1 AIN input pin(0) Multi-function serial ch.0 serial data input pin(1)
12 *1	14 *1	18 *1	21 *1	24	30	P041 A11 *2, *3, *4, *5 SIN9_0 ICU9_1 BIN1_0 INT12_0	- - - - - -	I	General-purpose I/O port External Bus address bit11 output pin(0) Multi-function serial ch.9 serial data input pin(0) Input capture ch.9 input pin(1) U/D counter ch.1 BIN input pin(0) INT12 external interrupt input pin(0)
13 *1	15 *1	19 *1	22 *1	25	31	P042 A12 *2, *3, *4, *5 SOT9_0 AN47 ICU8_1 TRG0_1 ZIN1_0	- - - - - - -	B	General-purpose I/O port External Bus address bit12 output pin Multi-function serial ch.9 serial data output pin(0) ADC analog 47 input pin Input capture ch.8 input pin(1) PPG trigger 0 input pin(1) U/D counter ch.1 ZIN input pin(0)
-	-	20 *1	23 *1	26	32	P043 A13 *4, *5 ICU7_1 TRG1_1	- - - -	A	General-purpose I/O port External Bus address bit13 output pin(0) Input capture ch.7 input pin(1) PPG trigger 1 input pin(1)
-	16 *1	21 *1	24 *1	27	33	P044 A14 *3, *4, *5 SCS9_0 ICU6_1 TRG2_1	- - - - -	A	General-purpose I/O port External Bus address bit14 output pin(0) Serial chip select 9 I/O pin(0) Input capture ch.6 input pin(1) PPG trigger 2 input pin(1)
14 *1	17 *1	22 *1	25 *1	28	34	P045 A15 *2, *3, *4, *5 SCK9_0 AN46 ICU5_1 TRG3_1 TOT1_2	- - - - - - -	G	General-purpose I/O port External Bus address bit15 output pin(0) Multi-function serial ch.9 clock I/O pin(0) ADC analog 46 input pin Input capture ch.5 input pin(1) PPG trigger 3 input pin(1) Reload timer ch.1 output pin(2)
-	-	-	26 *1	29	35	P046 A16 *5 ICU4_1 TRG4_1	- - - -	A	General-purpose I/O port External Bus address bit16 output pin(0) Input capture ch.4 input pin(1) PPG trigger 4 input pin(1)
-	-	-	-	-	36	P176 TRG10_0	- -	A	General-purpose I/O port PPG trigger 10 input pin(0)
15 *1	18 *1	23 *1	27 *1	30	37	P047 A17 *2, *3, *4, *5 AN45 TRG8_0 TIN3_2 SOT0_1	- - - - - -	B	General-purpose I/O port External Bus address bit17 output pin(0) ADC analog 45 input pin PPG trigger 8 input pin(0) Reload timer ch.3 event input pin(2) Multi-function serial ch.0 serial data output pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	-	-	38	P177 TRG11_0	- -	A	General-purpose I/O port PPG trigger 11 input pin(0)
-	-	-	28 *1	31	39	P050 A18 *5 TRG5_1 PPG33_0	- - - -	A	General-purpose I/O port External Bus address bit18 output pin PPG trigger 5 input pin(1) PPG ch.33 output pin(0)
-	-	-	-	32	40	P051 A19 TRG9_0	- - -	A	General-purpose I/O port External Bus address bit19 output pin PPG trigger 9 input pin(0)
-	-	-	-	33	41	P052 A20 PPG34_0 INT14_0	- - - -	A	General-purpose I/O port External Bus address bit20 output pin PPG ch.34 output pin(0) INT14 external interrupt input pin(0)
16 *1	19 *1	24 *1	29 *1	34	42	P053 A21 *2, *3, *4, *5 AN44 PPG35_0 INT14_1 SCK0_1	- - - - - -	B	General-purpose I/O port External Bus address bit21 output pin ADC analog 44 input pin PPG ch.35 output pin(0) INT14 external interrupt input pin(1) Multi-function serial ch.0 clock I/O pin(1)
-	-	-	-	35	43	P054 SYSCLK PPG36_0	- - -	A	General-purpose I/O port External Bus system clock output pin PPG ch.36 output pin(0)
17 *1	22 *1	27 *1	32 *1	38	46	P055 CS2X *2, *3, *4, *5 SIN10_0 AN43 PPG37_0 TIN4_1	- - - - - -	G	General-purpose I/O port External Bus chip select 2 output pin Multi-function serial ch.10 serial data input pin(0) ADC analog 43 input pin PPG ch.37 output pin(0) Reload timer ch.4 event input pin(1)
-	-	-	-	-	47	P180 PPG40_0	- -	A	General-purpose I/O port PPG ch.40 output pin(0)
-	-	-	-	-	48	P181 PPG41_0	- -	A	General-purpose I/O port PPG ch.41 output pin(0)
-	-	-	33 *1	39	49	P056 CS3X *5 ICU9_0 PPG0_1 ICU0_1 TIN5_1 DTTI_2	- - - - - - -	A	General-purpose I/O port External Bus chip select 3 output pin Input capture ch.9 input pin(0) PPG ch.0 output pin(1) Input capture ch.0 input pin(1) Reload timer ch.5 event input pin(1) Waveform generator ch.0 to ch.5 input pin(2)
19 *1	24 *1	29 *1	35 *1	41	51	P057 RDY *2, *3, *4, *5 SCK10_1 AN42 ICU8_0 TRG0_2 PPG1_1 ICU1_1 TIN6_1	- - - - - - - - -	G	General-purpose I/O port External Bus RDY input pin(0) Multi-function serial ch.10 clock I/O pin(1) ADC analog 42 input pin Input capture ch.8 input pin(0) PPG trigger 0 input pin(2) PPG ch.1 output pin(1) Input capture ch.1 input pin(1) Reload timer ch.6 event input pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	-	44	54	P142 SCK10_0/SCL10 PPG38_0 TIN7_1	- - - -	F	General-purpose I/O port Multi-function serial ch.10 clock I/O pin(0)/ I^2C bus serial clock I/O pin PPG ch.38 output pin(0) Reload timer ch.7 event input pin(1)
-	-	-	-	45	55	P143 SOT10_0/SDA10 PPG39_0 TOT4_1	- - - -	F	General-purpose I/O port Multi-function serial ch.10 serial data output pin(0)/ I^2C bus serial data I/O pin PPG ch.39 output pin(0) Reload timer ch.4 output pin(1)
-	-	-	-	-	56	P182 PPG42_0	- -	A	General-purpose I/O port PPG ch.42 output pin(0)
-	-	32	38	46	57	P060 SCS10_0 PPG2_1 ICU2_1 TOT5_1 INT13_0	- - - - - -	A	General-purpose I/O port Serial chip select 10 I/O pin(0) PPG ch.2 output pin(1) Input capture ch.2 input pin(1) Reload timer ch.5 output pin(1) INT13 external interrupt input pin(0)
22	27	33	39	47	58	P061 SOT10_1 AN41 ICU6_0 PPG3_1 ICU3_1 TOT6_1 INT13_1	- - - - - - - -	B	General-purpose I/O port Multi-function serial ch.10 serial data output pin(1) ADC analog 41 input pin Input capture ch.6 input pin(0) PPG ch.3 output pin(1) Input capture ch.3 input pin(1) Reload timer ch.6 output pin(1) INT13 external interrupt input pin(1)
23	28	34	40	48	59	P062 SCS10_1 SCS40_0 AN40 PPG4_1 FRCK0_0 TOT7_1 ZIN1_1	- - - - - - - -	B	General-purpose I/O port Serial chip select 10 I/O pin(1) Serial chip select 40 I/O pin(0) ADC analog 40 input pin PPG ch.4 output pin(1) Free-run timer 0 clock input pin(0) Reload timer ch.7 output pin(1) U/D counter ch.1 ZIN input pin(1)
-	29	35	41	49	60	P063 SCS41_0 AN39 PPG5_1 FRCK1_0 BIN1_1	- - - - - -	B	General-purpose I/O port Serial chip select 41 output pin(0) ADC analog 39 input pin PPG ch.5 output pin(1) Free-run timer 1 clock input pin(0) U/D counter ch.1 BIN input pin(1)
-	-	-	-	-	61	P183 PPG43_0	- -	A	General-purpose I/O port PPG ch.43 output pin(0)
24	30	36	42	50	62	P064 SCS42_0 AN38 FRCK2_0 AIN1_1 PPG43_1	- - - - - -	B	General-purpose I/O port Serial chip select 42 output pin(0) ADC analog 38 input pin Free-run timer 2 clock input pin(0) U/D counter ch.1 AIN input pin(1) PPG ch.43 output pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	37	43	51	63	P065 SCS43_0 FRCK3_0 ZIN0_1 PPG44_1	- - - - -	A	General-purpose I/O port Serial chip select 43 output pin(0) Free-run timer 3 clock input pin(0) U/D counter ch.0 ZIN input pin(1) PPG ch.44 output pin(1)
-	-	-	-	-	64	P184 PPG44_0	- -	A	General-purpose I/O port PPG ch.44 output pin(0)
-	-	-	-	-	65	P185 PPG45_0	- -	A	General-purpose I/O port PPG ch.45 output pin(0)
25	31	38	44	52	66	P066 SOT4_2 SCS3_0 AN37 FRCK4_0 BIN0_1	- - - - - -	B	General-purpose I/O port Multi-function serial ch.4 serial data output pin(2) Serial chip select 3 I/O pin(0) ADC analog 37 input pin Free-run timer 4 clock input pin(0) U/D counter ch.0 BIN input pin(1)
-	32	39	45	53	67	P067 AN36 FRCK5_0 AIN0_1	- - - -	B	General-purpose I/O port ADC analog 36 input pin Free-run timer 5 clock input pin(0) U/D counter ch.0 AIN input pin(1)
-	-	40	46	54	68	P070 ICU0_2	- -	A	General-purpose I/O port Input capture ch.0 input pin(2)
26	33	41	47	55	69	P071 SCK4_2 AN35 ICU1_2 MONCLK	- - - - -	G	General-purpose I/O port Multi-function serial ch.4 clock I/O pin(2) ADC analog 35 input pin Input capture ch.1 input pin(2) clock monitor output pin
27	34	42	48	56	70	P072 SIN4_0 AN34 ICU2_2 INT5_0	- - - - -	G	General-purpose I/O port Multi-function serial ch.4 serial data input pin(0) ADC analog 34 input pin Input capture ch.2 input pin(2) INT5 external interrupt input pin(0)
-	35 *1	43 *1	49	57	71	P073 SOT4_0/SDA4 *3, *4 AN33 ICU3_2	- - - -	D	General-purpose I/O port Multi-function serial ch.4 serial data output pin(0)/ I^2C bus serial data I/O pin ADC analog 33 input pin Input capture ch.3 input pin(2)
-	-	-	-	-	72	P186 PPG46_0	- -	A	General-purpose I/O port PPG ch.46 output pin(0)
-	-	-	-	-	73	P187 PPG47_0	- -	A	General-purpose I/O port PPG ch.47 output pin(0)
-	-	-	50	58	74	P074 SCK4_0/SCL4	- -	E	General-purpose I/O port Multi-function serial ch.4 clock I/O pin(0)/ I^2C bus serial clock I/O pin
-	-	-	51	59	75	P075 SIN3_0 INT4_0	- - -	F	General-purpose I/O port Multi-function serial ch.3 serial data input pin(0) INT4 external interrupt input pin(0)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	52	60	76	P076 SOT3_0/SDA3	- -	E	General-purpose I/O port Multi-function serial ch.3 serial data output pin(0)/ I^2C bus serial data I/O pin
-	-	-	53	61	77	P077 SCK3_0/SCL3	- -	E	General-purpose I/O port Multi-function serial ch.3 clock I/O pin(0)/ I^2C bus serial clock I/O pin
-	-	44	54	62	78	P152 SCS53_0	- -	A	General-purpose I/O port Serial chip select 53 output pin(0)
28	36	45	55	63	79	P153 SCK5_0/SCL5 AN32 FRCK1_1 INT4_1	- - - - -	G	General-purpose I/O port Multi-function serial ch.5 clock I/O pin(0)/ I^2C bus serial clock I/O pin ADC analog 32 input pin Free-run timer 1 clock input pin(1) INT4 external interrupt input pin(1)
-	-	-	-	64	80	P080 SCS52_0 PPG0_0	- - -	A	General-purpose I/O port Serial chip select 52 output pin(0) PPG ch.0 output pin(0)
29	37	46	56	65	81	P081 SOT5_0/SDA5 AN0 PPG1_0	- - - -	G	General-purpose I/O port Multi-function serial ch.5 serial data output pin(0)/ I^2C bus serial data I/O pin ADC analog 0 input pin PPG ch.1 output pin(0)
30	38	47	57	66	82	P082 SIN5_0 AN1 PPG2_0	- - - -	G	General-purpose I/O port Multi-function serial ch.5 serial data input pin(0) ADC analog 1 input pin PPG ch.2 output pin(0)
-	-	-	-	67	83	P083 SCS50_0 AN2 PPG3_0	- - - -	B	General-purpose I/O port Serial chip select 50 I/O pin(0) ADC analog 2 input pin PPG ch.3 output pin(0)
-	-	-	-	68	84	P084 SCS51_0 AN3 PPG4_0	- - - -	B	General-purpose I/O port Serial chip select 51 output pin(0) ADC analog 3 input pin PPG ch.4 output pin(0)
-	-	-	-	69	85	P085 PPG5_0	- -	A	General-purpose I/O port PPG ch.5 output pin(0)
-	-	48	58	70	86	P086 DAO1 PPG6_0	- - -	C	General-purpose I/O port DAC analog 1 output pin PPG ch.6 output pin(0)
31	39	49	59	71	87	P087 DAO0 PPG7_0 INT8_0	- - - -	C	General-purpose I/O port DAC analog 0 output pin PPG ch.7 output pin(0) INT8 external interrupt input pin(0)
-	-	-	-	-	90	P190 TIN0_1	- -	A	General-purpose I/O port Reload timer ch.0 event input pin(1)
-	-	-	-	-	91	P191 TIN1_1	- -	A	General-purpose I/O port Reload timer ch.1 event input pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	-	74	92	P090 AN4 ICU0_0 TIN2_1	- - - -	B	General-purpose I/O port ADC analog 4 input pin Input capture ch.0 input pin(0) Reload timer ch.2 event input pin(1)
-	-	-	-	75	93	P091 AN5 PPG41_1 ICU1_0 TIN3_1	- - - - -	B	General-purpose I/O port ADC analog 5 input pin PPG ch.41 output pin(1) Input capture ch.1 input pin(0) Reload timer ch.3 event input pin(1)
-	-	-	-	76	94	P092 AN6 PPG40_1 ICU2_0 TOT0_1	- - - - -	B	General-purpose I/O port ADC analog 6 input pin PPG ch.40 output pin(1) Input capture ch.2 input pin(0) Reload timer ch.0 output pin(1)
-	-	-	-	-	95	P192 PPG24_1 TOT1_1	- - -	A	General-purpose I/O port PPG ch.24 output pin(1) Reload timer ch.1 output pin(1)
34 *1	42 *1	52	62	77	96	P093 TX0_1 SIN11_0 AN7 ICU4_2 PPG16_1 ICU3_0 TOT2_1 *2, *3	- - - - - - - -	J	General-purpose I/O port CAN transmission data 0 output pin(1) Multi-function serial ch.11 serial data input pin(0) ADC analog 7 input pin Input capture ch.4 input pin(2) PPG ch.16 output pin(1) Input capture ch.3 input pin(0) Reload timer ch.2 output pin(1)
-	-	-	-	78	97	P094 AN8 ICU4_0 TOT3_1	- - - -	B	General-purpose I/O port ADC analog 8 input pin Input capture ch.4 input pin(0) Reload timer ch.3 output pin(1)
-	-	53	63	79	98	P095 TX0(128) SCS11_0 AN9	- - - -	B	General-purpose I/O port CAN transmission data 0 output pin Serial chip select 11 I/O pin(0) ADC analog 9 input pin
35	43	54	64	80	99	P096 RX0(128) SOT11_0/SDA11 AN10 INT0_0	- - - - -	G	General-purpose I/O port CAN reception data 0 input pin Multi-function serial ch.11 serial data output pin(0)/ I^2C bus serial data I/O pin ADC analog 10 input pin INT0 external interrupt input pin(0)
36	44	55	65	81	100	P097 SCK11_0/SCL11 AN11 ICU5_0 PPG17_1	- - - - -	G	General-purpose I/O port Multi-function serial ch.11 clock I/O pin(0)/ I^2C bus serial clock I/O pin ADC analog 11 input pin Input capture ch.5 input pin(0) PPG ch.17 output pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	48 *1	59	69	85	104	P100 SCK7_0/SCL7 *3 AN12 PPG8_0	- - - -	G	General-purpose I/O port Multi-function serial ch.7 clock I/O pin(0)/ I^2C bus serial clock I/O pin ADC analog 12 input pin PPG ch.8 output pin(0)
-	-	60	70	86	105	P101 SOT7_0/SDA7 AN13 PPG9_0	- - - -	G	General-purpose I/O port Multi-function serial ch.7 serial data output pin(0)/ I^2C bus serial data I/O pin ADC analog 13 input pin PPG ch.9 output pin(0)
40 *1	49 *1	61	71	87	106	P102 SIN7_0 *2, *3 AN14 PPG10_0 INT10_0	- - - - -	G	General-purpose I/O port Multi-function serial ch.7 serial data input pin(0) ADC analog 14 input pin PPG ch.10 output pin(0) INT10 external interrupt input pin(0)
41 *1	50 *1	62	72	88	107	P103 SCS73_0 *2, *3 AN15 PPG11_0	- - - -	H	General-purpose I/O port Serial chip select 73 output pin(0) ADC analog 15 input pin PPG ch.11 output pin(0)
42 *1	51 *1	63	73	89	108	P104 SCS72_0 *2, *3 AN16 PPG12_0	- - - -	H	General-purpose I/O port Serial chip select 72 output pin(0) ADC analog 16 input pin PPG ch.12 output pin(0)
43 *1	52 *1	64	74	90	109	P105 SCS71_0 *2, *3 AN17 PPG13_0	- - - -	H	General-purpose I/O port Serial chip select 71 output pin(0) ADC analog 17 input pin PPG ch.13 output pin(0)
-	-	65	75	91	110	P106 SCS70_0 AN18 PPG14_0	- - - -	H	General-purpose I/O port Serial chip select 70 I/O pin(0) ADC analog 18 input pin PPG ch.14 output pin(0)
-	53	66	76	92	111	P107 AN19 PPG15_0	- - -	B	General-purpose I/O port ADC analog 19 input pin PPG ch.15 output pin(0)
-	-	-	-	-	112	P193 PPG25_1	- -	A	General-purpose I/O port PPG ch.25 output pin(1)
-	-	-	77	93	113	P154 AN20	- -	B	General-purpose I/O port ADC analog 20 input pin
-	-	-	78	94	114	P155 AN21	- -	B	General-purpose I/O port ADC analog 21 input pin
44	54	67	79	95	115	NMIX	N	M	Non-maskable interrupt input pin
45	55	68	80	96	116	P110 TX1(64) SCS63_0 AN22	- - - -	B	General-purpose I/O port CAN transmission data 1 output pin Serial chip select 63 output pin(0) ADC analog 22 input pin

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	69	81	97	117	P111 RX1(64) SCS62_0 AN23 INT1_0	- - - - -	G	General-purpose I/O port CAN reception data 1 input pin Serial chip select 62 output pin(0) ADC analog 23 input pin INT1 external interrupt input pin(0)
-	-	-	82	98	118	P112 AN24 PPG16_0 RTO0_0	- - - -	B	General-purpose I/O port ADC analog 24 input pin PPG ch.16 output pin(0) Waveform generator ch.0 output pin(0)
-	-	-	83	99	119	P113 AN25 PPG17_0 RTO1_0	- - - -	B	General-purpose I/O port ADC analog 25 input pin PPG ch.17 output pin(0) Waveform generator ch.1 output pin(0)
-	-	-	-	-	120	P194 FRCK5_1 PPG26_1	- - -	A	General-purpose I/O port Free-run timer 5 clock input pin(1) PPG ch.26 output pin(1)
-	-	-	-	-	121	P195 FRCK4_1 PPG27_1	- - -	A	General-purpose I/O port Free-run timer 4 clock input pin(1) PPG ch.27 output pin(1)
-	56	70	84	100	122	P114 SCS61_0 AN26 PPG18_0 RTO2_0	- - - - -	B	General-purpose I/O port Serial chip select 61 output pin(0) ADC analog 26 input pin PPG ch.18 output pin(0) Waveform generator ch.2 output pin(0)
46	57	71	85	101	123	P115 RX1_1 SOT6_0/SDA6 AN27 PPG19_0 RTO3_0 INT1_1	- - - - - - -	G	General-purpose I/O port CAN reception data 1 input pin(1) Multi-function serial ch.6 serial data output pin(0)/ I^2C bus serial data I/O pin ADC analog 27 input pin PPG ch.19 output pin(0) Waveform generator ch.3 output pin(0) INT1 external interrupt input pin(1)
47	58	72	86	102	124	P116 SCK6_0/SCL6 AN28 PPG20_0 RTO4_0	- - - - -	G	General-purpose I/O port Multi-function serial ch.6 clock I/O pin(0)/ I^2C bus serial clock I/O pin ADC analog 28 input pin PPG ch.20 output pin(0) Waveform generator ch.4 output pin(0)
-	-	73	87	103	125	P117 SCS60_0 AN29 PPG21_0 RTO5_0	- - - - -	B	General-purpose I/O port Serial chip select 60 I/O pin(0) ADC analog 29 input pin PPG ch.21 output pin(0) Waveform generator ch.5 output pin(0)
-	-	-	-	-	126	P196 FRCK3_1 PPG28_1	- - -	A	General-purpose I/O port Free-run timer 3 clock input pin(1) PPG ch.28 output pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	88	104	127	P120 AN30 OCU6_0 PPG22_0 INT9_0	- - - - -	B	General-purpose I/O port ADC analog 30 input pin Output compare ch.6 output pin(0) PPG ch.22 output pin(0) INT9 external interrupt input pin(0)
-	-	-	-	105	128	P121 OCU7_0 PPG23_0	- - -	A	General-purpose I/O port Output compare ch.7 output pin(0) PPG ch.23 output pin(0)
48	59	74	89	106	129	P122 SIN6_0 AN31 OCU8_0 INT9_1	- - - - -	J	General-purpose I/O port Multi-function serial ch.6 serial data input pin(0) ADC analog 31 input pin Output compare ch.8 output pin(0) INT9 external interrupt input pin(1)
-	-	-	-	-	130	P197 PPG29_1	- -	A	General-purpose I/O port PPG ch.29 output pin(1)
-	-	-	-	107	131	P123 OCU9_0	- -	A	General-purpose I/O port Output compare ch.9 output pin(0)
49	62	77	92	110	134	DEBUGIF	-	L	MDI I/O pin for debug (OCD)
-	-	-	-	-	135	P160 PPG30_1	- -	A	General-purpose I/O port PPG ch.30 output pin(1)
-	-	-	-	-	136	P161 PPG31_1	- -	A	General-purpose I/O port PPG ch.31 output pin(1)
-	-	-	-	111	137	P124 OCU10_0	- -	A	General-purpose I/O port Output compare ch.10 output pin(0)
-	-	-	93	112	138	P125 OCU11_0	- -	A	General-purpose I/O port Output compare ch.11 output pin(0)
50	63	78	94	113	139	P126 SIN0_0 INT6_0	- - -	F	General-purpose I/O port Multi-function serial ch.0 serial data input pin(0) INT6 external interrupt input pin(0)
-	64	79	95	114	140	P127 SOT0_0	- -	A	General-purpose I/O port Multi-function serial ch.0 serial data output pin(0)
-	-	80	96	115	141	P130 SCK0_0	- -	F	General-purpose I/O port Multi-function serial ch.0 clock I/O pin(0)
-	-	-	-	-	142	P162 TRG5_2	- -	A	General-purpose I/O port PPG trigger 5 input pin(2)
-	-	-	-	-	143	P163 TRG6_2	- -	A	General-purpose I/O port PPG trigger 6 input pin(2)
51	65	81	97	116	144	MD0	-	K	Mode pin 0
52	66	82	98	117	145	MD1	-	K	Mode pin 1
53	67	83	99	118	146	X0	-	N	Main clock oscillation input pin
54	68	84	100	119	147	X1	-	N	Main clock oscillation output pin
56	70	86	102	121	149	P135 DTTI_0	- -	A	General-purpose I/O port Waveform generator ch.0 to ch.5 input pin(0)
						X1A	-	O	Sub clock oscillation output pin
57	71	87	103	122	150	P136	-	A	General-purpose I/O port
						X0A	-	O	Sub clock oscillation input pin
58	72	88	104	123	151	RSTX	N	M	External reset input pin

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
-	-	-	-	124	152	P131 ADTG0_0	- -	A	General-purpose I/O port A/D converter external trigger input pin 0(0)
-	-	-	105	125	153	P132 SCS1_0 ADTG1_0	- - -	A	General-purpose I/O port Serial chip select 1 I/O pin(0) A/D converter external trigger input pin 1(0)
-	-	89	106	126	154	P133 TX2(64)	- -	A	General-purpose I/O port CAN transmission data 2 output pin
-	-	90	107	127	155	P134 RX2(64) SCS1_1 ICU7_0 INT7_0	- - - - -	F	General-purpose I/O port CAN reception data 2 input pin Serial chip select 1 I/O pin(1) Input capture ch.7 input pin(0) INT7 external interrupt input pin(0)
-	-	91	108	128	156	P144 SCK1_1	- -	F	General-purpose I/O port Multi-function serial ch.1 clock I/O pin(1)
-	-	94 *1	111 *1	131	159	P000 D16 *4, *5 SIN1_0 TIOA0_1 *4 INT2_0	- - - - -	F	General-purpose I/O port External Bus data bit16 I/O pin(0) Multi-function serial ch.1 serial data input pin(0) Base timer ch.0 TIOA output pin(1) INT2 external interrupt input pin(0)
-	75 *1	95 *1	112 *1	132	160	P001 D17 *3, *4, *5 SOT1_0 *3 TIOA1_1	- - - -	A	General-purpose I/O port External Bus data bit17 I/O pin Multi-function serial ch.1 serial data output pin(0) Base timer ch.1 TIOA I/O pin(1)
-	-	-	113 *1	133	161	P002 D18 *5 SCK1_0 TIOB0_1	- - - -	F	General-purpose I/O port External Bus data bit18 I/O pin Multi-function serial ch.1 clock I/O pin(0) Base timer ch.0 TIOB input pin(1)
-	76 *1	96 *1	114 *1	134	162	P003 D19 *3, *4, *5 SIN2_0 TIOB1_1 INT3_0	- - - - -	F	General-purpose I/O port External Bus data bit19 I/O pin Multi-function serial ch.2 serial data input pin(0) Base timer ch.1 TIOB input pin(1) INT3 external interrupt input pin(0)
-	-	-	-	135	163	P004 D20 SOT2_0	- - -	A	General-purpose I/O port External Bus data bit20 I/O pin(0) Multi-function serial ch.2 serial data output pin(0)
-	-	-	-	-	164	P164 PPG32_1	- -	A	General-purpose I/O port PPG ch.32 output pin(1)
61 *1	77 *1	97 *1	115 *1	136 *1	165 *1	P005 D21 *2, *3, *4, *5 SCK2_0 *2 ADTG0_1 INT7_1 RX2(64) *4, *5, *6, *7	- - - - - -	F	General-purpose I/O port External Bus data bit21 I/O pin(0) Multi-function serial ch.2 clock I/O pin(0) A/D converter external trigger input pin 0(1) INT7 external interrupt input pin(1) CAN reception data 2 input pin
-	-	-	-	-	166	P165 PPG33_1	- -	A	General-purpose I/O port PPG ch.33 output pin(1)

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
62 *1	78 *1	98 *1	116 *1	137 *1	167 *1	P006 D22 *2, *3, *4, *5 SCS2_0 *2 ADTG1_1 INT2_1 TX2(64) *4, *5, *6, *7	- - - - -	A	General-purpose I/O port External Bus data bit22 I/O pin(0) Serial chip select 2 I/O pin(0) A/D converter external trigger input pin 1(1) INT2 external interrupt input pin(1) CAN transmission data 2 output pin
-	-	-	117 *1	138	168	P007 D23 *5	- -	A	General-purpose I/O port External Bus data bit23 I/O pin
-	-	-	-	-	169	P166 PPG34_1	- -	A	General-purpose I/O port PPG ch.34 output pin(1)
-	-	-	118 *1	139	170	P010 D24 *5	- -	A	General-purpose I/O port External Bus data bit24 I/O pin
63 *1	79 *1	99 *1	119 *1	140	171	P011 WOT D25 *2, *3, *4, *5 SOT2_1 *2 TIOA0_0 *2, *3, *4 INT3_1	- - - - - -	A	General-purpose I/O port RTC output pin External Bus data bit25 I/O pin Multi-function serial ch.2 serial data output pin(1) Base timer ch.0 TIOA output pin(0) INT3 external interrupt input pin(1)
-	-	-	-	141	172	P012 D26 TIOB0_0	- - -	A	General-purpose I/O port External Bus data bit26 I/O pin Base timer ch.0 TIOB input pin(0)
-	-	-	-	-	173	P167 PPG35_1	- -	A	General-purpose I/O port PPG ch.35 output pin(1)
-	-	-	-	142	174	P013 D27 TIOA1_0	- - -	A	General-purpose I/O port External Bus data bit27 I/O pin Base timer ch.1 TIOA I/O pin(0)
-	-	-	-	143	175	P014 D28 TIOB1_0	- - -	A	General-purpose I/O port External Bus data bit28 I/O pin Base timer ch.1 TIOB input pin(0)
18	23	28	34	40	50	AVCC1	-	-	A/D, D/A converter unit1 analog power supply pin
39	47	58	68	84	103	AVCC0	-	-	A/D, D/A converter unit0 analog power supply pin
20	25	30	36	42	52	AVRH1	-	-	A/D converter unit1 upper limit reference voltage pin
38	46	57	67	83	102	AVRH0	-	-	A/D converter unit0 upper limit reference voltage pin
21	26	31	37	43	53	AVSS1/AVRL1	-	-	A/D, D/A converter unit1 GND pin / A/D converter unit1 lower limit reference voltage pin
37	45	56	66	82	101	AVSS0/AVRL0	-	-	A/D, D/A converter unit0 GND pin / A/D converter unit0 lower limit reference voltage pin
60	74	93	110	130	158	C	-	-	External capacity connection output pin
32 64	20 40 61 80	25 50 76 100	30 60 91 120	36 72 109 144	44 88 133 176	VCC	-	-	+5.0v power supply pin

Pin Number						Pin Name	Polarity	I/O circuit form	Function (Please refer to the chapter of "I/O port" for the switch.)
64	80	100	120	144	176				
	1	1	1	1	1				
1	21	26	31	37	45				
33	41	51	61	73	89				
55	60	75	90	108	132	VSS	-	-	GND pin
59	69	85	101	120	148				
	73	92	109	129	157				

*1: There is a restriction of pin functions. See "Pin Name" of this table.
*2: not supported in 64pin
*3: not supported in 80pin
*4: not supported in 100pin
*5: not supported in 120pin
*6: not supported in 144pin
*7: not supported in 176pin

9.1. Pins of Each Function

Pins of each function are shown below.

Table 9-2 Pins of A/D Converter (ch.0 to ch.47)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
A/D converter external trigger input pin0(0)	ADTG0_0	Yes	-	-	-	-	124	152
A/D converter external trigger input pin0(1)	ADTG0_1	Yes	61	77	97	115	136	165
A/D converter external trigger input pin1(0)	ADTG1_0	Yes	-	-	-	105	125	153
A/D converter external trigger input pin1(1)	ADTG1_1	Yes	62	78	98	116	137	167
ADC analog 0 input pin	AN0	No	29	37	46	56	65	81
ADC analog 1 input pin	AN1	No	30	38	47	57	66	82
ADC analog 2 input pin	AN2	No	-	-	-	-	67	83
ADC analog 3 input pin	AN3	No	-	-	-	-	68	84
ADC analog 4 input pin	AN4	No	-	-	-	-	74	92
ADC analog 5 input pin	AN5	No	-	-	-	-	75	93
ADC analog 6 input pin	AN6	No	-	-	-	-	76	94
ADC analog 7 input pin	AN7	No	34	42	52	62	77	96
ADC analog 8 input pin	AN8	No	-	-	-	-	78	97
ADC analog 9 input pin	AN9	No	-	-	53	63	79	98
ADC analog 10 input pin	AN10	No	35	43	54	64	80	99
ADC analog 11 input pin	AN11	No	36	44	55	65	81	100
ADC analog 12 input pin	AN12	No	-	48	59	69	85	104
ADC analog 13 input pin	AN13	No	-	-	60	70	86	105
ADC analog 14 input pin	AN14	No	40	49	61	71	87	106

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
ADC analog 15 input pin	AN15	No	41	50	62	72	88	107
ADC analog 16 input pin	AN16	No	42	51	63	73	89	108
ADC analog 17 input pin	AN17	No	43	52	64	74	90	109
ADC analog 18 input pin	AN18	No	-	-	65	75	91	110
ADC analog 19 input pin	AN19	No	-	53	66	76	92	111
ADC analog 20 input pin	AN20	No	-	-	-	77	93	113
ADC analog 21 input pin	AN21	No	-	-	-	78	94	114
ADC analog 22 input pin	AN22	No	45	55	68	80	96	116
ADC analog 23 input pin	AN23	No	-	-	69	81	97	117
ADC analog 24 input pin	AN24	No	-	-	-	82	98	118
ADC analog 25 input pin	AN25	No	-	-	-	83	99	119
ADC analog 26 input pin	AN26	No	-	56	70	84	100	122
ADC analog 27 input pin	AN27	No	46	57	71	85	101	123
ADC analog 28 input pin	AN28	No	47	58	72	86	102	124
ADC analog 29 input pin	AN29	No	-	-	73	87	103	125
ADC analog 30 input pin	AN30	No	-	-	-	88	104	127
ADC analog 31 input pin	AN31	No	48	59	74	89	106	129
ADC analog 32 input pin	AN32	No	28	36	45	55	63	79
ADC analog 33 input pin	AN33	No	-	35	43	49	57	71
ADC analog 34 input pin	AN34	No	27	34	42	48	56	70
ADC analog 35 input pin	AN35	No	26	33	41	47	55	69
ADC analog 36 input pin	AN36	No	-	32	39	45	53	67
ADC analog 37 input pin	AN37	No	25	31	38	44	52	66
ADC analog 38 input pin	AN38	No	24	30	36	42	50	62
ADC analog 39 input pin	AN39	No	-	29	35	41	49	60
ADC analog 40 input pin	AN40	No	23	28	34	40	48	59
ADC analog 41 input pin	AN41	No	22	27	33	39	47	58
ADC analog 42 input pin	AN42	No	19	24	29	35	41	51
ADC analog 43 input pin	AN43	No	17	22	27	32	38	46
ADC analog 44 input pin	AN44	No	16	19	24	29	34	42
ADC analog 45 input pin	AN45	No	15	18	23	27	30	37
ADC analog 46 input pin	AN46	No	14	17	22	25	28	34
ADC analog 47 input pin	AN47	No	13	15	19	22	25	31
A/D, D/A converter unit1 analog power supply pin	AVCC1	-	18	23	28	34	40	50
A/D, D/A converter unit0 analog power supply pin	AVCC0	-	39	47	58	68	84	103
A/D converter unit1 upper limit reference voltage pin	AVRH1	-	20	25	30	36	42	52
A/D converter unit0 upper limit reference voltage pin	AVRH0	-	38	46	57	67	83	102
A/D, D/A converter unit1 GND pin / A/D converter unit1 lower limit reference voltage pin	AVSS1/AVRL1	-	21	26	31	37	43	53
A/D, D/A converter unit0 GND pin / A/D converter unit0 lower limit reference voltage pin	AVSS0/AVRL0	-	37	45	56	66	82	101

Table 9-3 Pins of CAN (ch.0 to ch.2)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
CAN reception data 0 input pin	RX0(128)	No	35	43	54	64	80	99
CAN transmission data 0 output pin	TX0(128)	-	-	-	53	63	79	98
CAN transmission data 0 output pin(1)	TX0_1	-	34	42	52	62	77	96
CAN reception data 1 input pin	RX1(64)	No	-	-	69	81	97	117
CAN reception data 1 input pin(1)	RX1_1	No	46	57	71	85	101	123
CAN transmission data 1 output pin	TX1(64)	-	45	55	68	80	96	116
CAN reception data 2 input pin	RX2(64)	No	61	77	90	107	127	155
CAN transmission data 2 output pin	TX2(64)	-	62	89	89	106	126	154

Table 9-4 Pins of D/A Converter (ch.0, ch.1)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
DAC analog 0 output pin	DAO0	-	31	39	49	59	71	87
DAC analog 1 output pin	DAO1	-	-	-	48	58	70	86

Table 9-5 Pins of External Interrupt Input

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
INT0 external interrupt input pin(0)	INT0_0	Yes	35	43	54	64	80	99
INT1 external interrupt input pin(0)	INT1_0	Yes	-	-	69	81	97	117
INT1 external interrupt input pin(1)	INT1_1	Yes	46	57	71	85	101	123
INT2 external interrupt input pin(0)	INT2_0	Yes	-	-	94	111	131	159
INT2 external interrupt input pin(1)	INT2_1	Yes	62	78	98	116	137	167
INT3 external interrupt input pin(0)	INT3_0	Yes	-	76	96	114	134	162
INT3 external interrupt input pin(1)	INT3_1	Yes	63	79	99	119	140	171
INT4 external interrupt input pin(0)	INT4_0	Yes	-	-	-	51	59	75
INT4 external interrupt input pin(1)	INT4_1	Yes	28	36	45	55	63	79
INT5 external interrupt input pin(0)	INT5_0	Yes	27	34	42	48	56	70
INT6 external interrupt input pin(0)	INT6_0	Yes	50	63	78	94	113	139
INT7 external interrupt input pin(0)	INT7_0	Yes	-	-	90	107	127	155
INT7 external interrupt input pin(1)	INT7_1	Yes	61	77	97	115	136	165
INT8 external interrupt input pin(0)	INT8_0	Yes	31	39	49	59	71	87
INT9 external interrupt input pin(0)	INT9_0	Yes	-	-	-	88	104	127
INT9 external interrupt input pin(1)	INT9_1	Yes	48	59	74	89	106	129
INT10 external interrupt input pin(0)	INT10_0	Yes	40	49	61	71	87	106
INT11 external interrupt input pin(0)	INT11_0	Yes	9	11	14	17	20	24
INT12 external interrupt input pin(0)	INT12_0	Yes	12	14	18	21	24	30
INT13 external interrupt input pin(0)	INT13_0	Yes	-	-	32	38	46	57
INT13 external interrupt input pin(1)	INT13_1	Yes	22	27	33	39	47	58
INT14 external interrupt input pin(0)	INT14_0	Yes	-	-	-	-	33	41
INT14 external interrupt input pin(1)	INT14_1	Yes	16	19	24	29	34	42
INT15 external interrupt input pin(0)	INT15_0	Yes	3	3	3	6	9	11

Table 9-6 Pins of Multi-function Serial Interface (ch.0 to ch.11)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
MFS ch.0 clock I/O pin(0)	SCK0_0	No	-	-	80	96	115	141
MFS ch.0 clock I/O pin(1)	SCK0_1	No	16	19	24	29	34	42
MFS ch.0 serial data output pin(0)	SOT0_0	-	-	64	79	95	114	140
MFS ch.0 serial data output pin(1)	SOT0_1	-	15	18	23	27	30	37
MFS ch.0 serial data input pin(0)	SIN0_0	No	50	63	78	94	113	139
MFS ch.0 serial data input pin(1)	SIN0_1	No	11	13	17	20	23	29
MFS ch.1 clock I/O pin(0)	SCK1_0	No	-	-	-	113	133	161
MFS ch.1 clock I/O pin(1)	SCK1_1	No	-	-	91	108	128	156
MFS ch.1 serial data output pin(0)	SOT1_0	-	-	-	95	112	132	160
MFS ch.1 serial data input pin(0)	SIN1_0	No	-	-	94	111	131	159
MFS ch.2 clock I/O pin(0)	SCK2_0	No	-	77	97	115	136	165
MFS ch.2 serial data output pin(0)	SOT2_0	-	-	-	-	-	135	163
MFS ch.2 serial data output pin(1)	SOT2_1	-	-	79	99	119	140	171
MFS ch.2 serial data input pin(0)	SIN2_0	No	-	76	96	114	134	162
MFS ch.3 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK3_0/ SCL3	No	-	-	-	53	61	77
MFS ch.3 clock I/O pin(1)	SCK3_1	No	-	-	-	4	7	9
MFS ch.3 clock I/O pin(2)	SCK3_2	No	6	8	10	13	16	20
MFS ch.3 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT3_0/ SDA3	No	-	-	-	52	60	76
MFS ch.3 serial data output pin(1)	SOT3_1	-	-	-	-	3	6	8
MFS ch.3 serial data output pin(2)	SOT3_2	-	7	9	11	14	17	21
MFS ch.3 serial data input pin(0)	SIN3_0	No	-	-	-	51	59	75
MFS ch.3 serial data input pin(1)	SIN3_1	No	2	2	2	2	5	7
MFS ch.4 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK4_0/ SCL4	No	-	-	-	50	58	74
MFS ch.4 clock I/O pin(1)	SCK4_1	No	-	4	5	8	11	14
MFS ch.4 clock I/O pin(2)	SCK4_2	No	26	33	41	47	55	69
MFS ch.4 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT4_0/ SDA4	-	-	-	-	49	57	71
MFS ch.4 serial data output pin(1)	SOT4_1	-	-	-	4	7	10	12
MFS ch.4 serial data output pin(2)	SOT4_2	-	25	31	38	44	52	66
MFS ch.4 serial data input pin(0)	SIN4_0	No	27	34	42	48	56	70
MFS ch.4 serial data input pin(1)	SIN4_1	No	3	3	3	6	9	11
MFS ch.5 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK5_0/ SCL5	No	28	36	45	55	63	79
MFS ch.5 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT5_0/ SDA5	-	29	37	46	56	65	81
MFS ch.5 serial data input pin(0)	SIN5_0	No	30	38	47	57	66	82
MFS ch.6 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK6_0/ SCL6	No	47	58	72	86	102	124
MFS ch.6 serial data output pin(0) / I^2C bus serial data I/O pin	SOT6_0/ SDA6	-	46	57	71	85	101	123
MFS ch.6 serial data input pin(0)	SIN6_0	No	48	59	74	89	106	129
MFS ch.7 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK7_0/ SCL7	No	-	-	59	69	85	104

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
MFS ch.7 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT7_0/ SDA7	-	-	-	60	70	86	105
MFS ch.7 serial data input pin(0)	SIN7_0	No	-	-	61	71	87	106
MFS ch.8 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK8_0/ SCL8	No	-	-	13	16	19	23
MFS ch.8 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT8_0/ SDA8	-	-	-	12	15	18	22
MFS ch.8 serial data input pin(0)	SIN8_0	No	-	-	14	17	20	24
MFS ch.9 clock I/O pin(0)	SCK9_0	No	14	17	22	25	28	34
MFS ch.9 serial data output pin(0)	SOT9_0	-	13	15	19	22	25	31
MFS ch.9 serial data input pin(0)	SIN9_0	No	12	14	18	21	24	30
MFS ch.10 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK10_0/ SCL10	No	-	-	-	-	44	54
MFS ch.10 clock I/O pin(1)	SCK10_1	No	19	24	29	35	41	51
MFS ch.10 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT10_0/ SDA10	-	-	-	-	-	45	55
MFS ch.10 serial data output pin(1)	SOT10_1	-	22	27	33	39	47	58
MFS ch.10 serial data input pin(0)	SIN10_0	No	17	22	27	32	38	46
MFS ch.11 clock I/O pin(0)/ I^2C bus serial clock I/O pin	SCK11_0/ SCL11	No	36	44	55	65	81	100
MFS ch.11 serial data output pin(0)/ I^2C bus serial data I/O pin	SOT11_0/ SDA11	-	35	43	54	64	80	99
MFS ch.11 serial data input pin(0)	SIN11_0	No	34	42	52	62	77	96
MFS ch.1 serial chip select I/O pin(0)	SCS1_0	No	-	-	-	105	125	153
MFS ch.1 serial chip select I/O pin(1)	SCS1_1	No	-	-	90	107	127	155
MFS ch.2 serial chip select I/O pin(0)	SCS2_0	No	-	78	98	116	137	167
MFS ch.3 serial chip select I/O pin(0)	SCS3_0	No	25	31	38	44	52	66
MFS ch.3 serial chip select I/O pin(1)	SCS3_1	No	-	-	-	5	8	10
MFS ch.40 serial chip select I/O pin(0)	SCS40_0	No	23	28	34	40	48	59
MFS ch.40 serial chip select I/O pin(1)	SCS40_1	No	4	5	6	9	12	15
MFS ch.41 serial chip select output pin(0)	SCS41_0	-	-	29	35	41	49	60
MFS ch.41 serial chip select output pin(1)	SCS41_1	-	-	-	7	10	13	17
MFS ch.42 serial chip select output pin(0)	SCS42_0	-	24	30	36	42	50	62
MFS ch.42 serial chip select output pin(1)	SCS42_1	-	-	6	8	11	14	18
MFS ch.43 serial chip select output pin(0)	SCS43_0	-	-	-	37	43	51	63
MFS ch.43 serial chip select output pin(1)	SCS43_1	-	5	7	9	12	15	19
MFS ch.50 serial chip select I/O pin(0)	SCS50_0	No	-	-	-	-	67	83
MFS ch.51 serial chip select output pin(0)	SCS51_0	-	-	-	-	-	68	84
MFS ch.52 serial chip select output pin(0)	SCS52_0	-	-	-	-	-	64	80
MFS ch.53 serial chip select output pin(0)	SCS53_0	-	-	-	44	54	62	78
MFS ch.60 serial chip select I/O pin(0)	SCS60_0	No	-	-	73	87	103	125
MFS ch.61 serial chip select output pin(0)	SCS61_0	-	-	56	70	84	100	122
MFS ch.62 serial chip select output pin(0)	SCS62_0	-	-	-	69	81	97	117
MFS ch.63 serial chip select output pin(0)	SCS63_0	-	45	55	68	80	96	116
MFS ch.70 serial chip select I/O pin(0)	SCS70_0	No	-	-	65	75	91	110
MFS ch.71 serial chip select output pin(0)	SCS71_0	-	-	-	64	74	90	109
MFS ch.72 serial chip select output pin(0)	SCS72_0	-	-	-	63	73	89	108
MFS ch.73 serial chip select output pin(0)	SCS73_0	-	-	-	62	72	88	107

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
MFS ch.8 serial chip select I/O pin(0)	SCS8_0	No	-	-	15	18	21	25
MFS ch.9 serial chip select I/O pin(0)	SCS9_0	No	-	16	21	24	27	33
MFS ch.10 serial chip select I/O pin(0)	SCS10_0	No	-	-	32	38	46	57
MFS ch.10 serial chip select I/O pin(1)	SCS10_1	No	23	28	34	40	48	59
MFS ch.11 serial chip select I/O pin(0)	SCS11_0	No	-	-	53	63	79	98

Table 9-7 Pins of PPG (ch.0 to ch.47)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
PPG ch.0 output pin(0)	PPG0_0	-	-	-	-	-	64	80
PPG ch.0 output pin(1)	PPG0_1	-	-	-	-	33	39	49
PPG ch.1 output pin(0)	PPG1_0	-	29	37	46	56	65	81
PPG ch.1 output pin(1)	PPG1_1	-	19	24	29	35	41	51
PPG ch.2 output pin(0)	PPG2_0	-	30	38	47	57	66	82
PPG ch.2 output pin(1)	PPG2_1	-	-	-	32	38	46	57
PPG ch.3 output pin(0)	PPG3_0	-	-	-	-	-	67	83
PPG ch.3 output pin(1)	PPG3_1	-	22	27	33	39	47	58
PPG ch.4 output pin(0)	PPG4_0	-	-	-	-	-	68	84
PPG ch.4 output pin(1)	PPG4_1	-	23	28	34	40	48	59
PPG ch.5 output pin(0)	PPG5_0	-	-	-	-	-	69	85
PPG ch.5 output pin(1)	PPG5_1	-	-	29	35	41	49	60
PPG ch.6 output pin(0)	PPG6_0	-	-	-	48	58	70	86
PPG ch.7 output pin(0)	PPG7_0	-	31	39	49	59	71	87
PPG ch.8 output pin(0)	PPG8_0	-	-	48	59	69	85	104
PPG ch.9 output pin(0)	PPG9_0	-	-	-	60	70	86	105
PPG ch.10 output pin(0)	PPG10_0	-	40	49	61	71	87	106
PPG ch.11 output pin(0)	PPG11_0	-	41	50	62	72	88	107
PPG ch.12 output pin(0)	PPG12_0	-	42	51	63	73	89	108
PPG ch.13 output pin(0)	PPG13_0	-	43	52	64	74	90	109
PPG ch.14 output pin(0)	PPG14_0	-	-	-	65	75	91	110
PPG ch.15 output pin(0)	PPG15_0	-	-	53	66	76	92	111
PPG ch.16 output pin(0)	PPG16_0	-	-	-	-	82	98	118
PPG ch.16 output pin(1)	PPG16_1	-	34	42	52	62	77	96
PPG ch.17 output pin(0)	PPG17_0	-	-	-	-	83	99	119
PPG ch.17 output pin(1)	PPG17_1	-	36	44	55	65	81	100
PPG ch.18 output pin(0)	PPG18_0	-	-	56	70	84	100	122
PPG ch.19 output pin(0)	PPG19_0	-	46	57	71	85	101	123
PPG ch.20 output pin(0)	PPG20_0	-	47	58	72	86	102	124
PPG ch.21 output pin(0)	PPG21_0	-	-	-	73	87	103	125
PPG ch.22 output pin(0)	PPG22_0	-	-	-	-	88	104	127
PPG ch.23 output pin(0)	PPG23_0	-	-	-	-	-	105	128
PPG ch.23 output pin(1)	PPG23_1	-	11	13	17	20	23	29
PPG ch.24 output pin(0)	PPG24_0	-	3	3	3	6	9	11
PPG ch.24 output pin(1)	PPG24_1	-	-	-	-	-	-	95

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
PPG ch.25 output pin(0)	PPG25_0	-	-	-	4	7	10	12
PPG ch.25 output pin(1)	PPG25_1	-	-	-	-	-	-	112
PPG ch.26 output pin(0)	PPG26_0	-	-	4	5	8	11	14
PPG ch.26 output pin(1)	PPG26_1	-	-	-	-	-	-	120
PPG ch.27 output pin(0)	PPG27_0	-	4	5	6	9	12	15
PPG ch.27 output pin(1)	PPG27_1	-	-	-	-	-	-	121
PPG ch.28 output pin(0)	PPG28_0	-	-	-	7	10	13	17
PPG ch.28 output pin(1)	PPG28_1	-	-	-	-	-	-	126
PPG ch.29 output pin(0)	PPG29_0	-	-	6	8	11	14	18
PPG ch.29 output pin(1)	PPG29_1	-	-	-	-	-	-	130
PPG ch.30 output pin(0)	PPG30_0	-	5	7	9	12	15	19
PPG ch.30 output pin(1)	PPG30_1	-	-	-	-	-	-	135
PPG ch.31 output pin(0)	PPG31_0	-	6	8	10	13	16	20
PPG ch.31 output pin(1)	PPG31_1	-	-	-	-	-	-	136
PPG ch.32 output pin(0)	PPG32_0	-	-	-	-	5	8	10
PPG ch.32 output pin(1)	PPG32_1	-	-	-	-	-	-	164
PPG ch.33 output pin(0)	PPG33_0	-	-	-	-	28	31	39
PPG ch.33 output pin(1)	PPG33_1	-	-	-	-	-	-	166
PPG ch.34 output pin(0)	PPG34_0	-	-	-	-	-	33	41
PPG ch.34 output pin(1)	PPG34_1	-	-	-	-	-	-	169
PPG ch.35 output pin(0)	PPG35_0	-	16	19	24	29	34	42
PPG ch.35 output pin(1)	PPG35_1	-	-	-	-	-	-	173
PPG ch.36 output pin(0)	PPG36_0	-	-	-	-	-	35	43
PPG ch.36 output pin(1)	PPG36_1	-	-	-	-	-	-	4
PPG ch.37 output pin(0)	PPG37_0	-	17	22	27	32	38	46
PPG ch.37 output pin(1)	PPG37_1	-	-	-	-	-	-	6
PPG ch.38 output pin(0)	PPG38_0	-	-	-	-	-	44	54
PPG ch.38 output pin(1)	PPG38_1	-	-	-	-	-	-	13
PPG ch.39 output pin(0)	PPG39_0	-	-	-	-	-	45	55
PPG ch.39 output pin(1)	PPG39_1	-	-	-	-	-	-	16
PPG ch.40 output pin(0)	PPG40_0	-	-	-	-	-	-	47
PPG ch.40 output pin(1)	PPG40_1	-	-	-	-	-	76	94
PPG ch.41 output pin(0)	PPG41_0	-	-	-	-	-	-	48
PPG ch.41 output pin(1)	PPG41_1	-	-	-	-	-	75	93
PPG ch.42 output pin(0)	PPG42_0	-	-	-	-	-	-	56
PPG ch.43 output pin(0)	PPG43_0	-	-	-	-	-	-	61
PPG ch.43 output pin(1)	PPG43_1	-	24	30	36	42	50	62
PPG ch.44 output pin(0)	PPG44_0	-	-	-	-	-	-	64
PPG ch.44 output pin(1)	PPG44_1	-	-	-	37	43	51	63
PPG ch.45 output pin(0)	PPG45_0	-	-	-	-	-	-	65
PPG ch.46 output pin(0)	PPG46_0	-	-	-	-	-	-	72
PPG ch.47 output pin(0)	PPG47_0	-	-	-	-	-	-	73

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
PPG trigger 0 input pin(0)	TRG0_0	Yes	-	-	-	-	2	2
PPG trigger 0 input pin(1)	TRG0_1	Yes	13	15	19	22	25	31
PPG trigger 0 input pin(2)	TRG0_2	Yes	19	24	29	35	41	51
PPG trigger 1 input pin(0)	TRG1_0	Yes	-	-	-	-	3	3
PPG trigger 1 input pin(1)	TRG1_1	Yes	-	-	20	23	26	32
PPG trigger 2 input pin(0)	TRG2_0	Yes	-	-	-	-	4	5
PPG trigger 2 input pin(1)	TRG2_1	Yes	-	16	21	24	27	33
PPG trigger 3 input pin(0)	TRG3_0	Yes	2	2	2	2	5	7
PPG trigger 3 input pin(1)	TRG3_1	Yes	14	17	22	25	28	34
PPG trigger 4 input pin(0)	TRG4_0	Yes	-	-	-	3	6	8
PPG trigger 4 input pin(1)	TRG4_1	Yes	-	-	-	26	29	35
PPG trigger 5 input pin(0)	TRG5_0	Yes	-	-	-	4	7	9
PPG trigger 5 input pin(1)	TRG5_1	Yes	-	-	-	28	31	39
PPG trigger 5 input pin(2)	TRG5_2	Yes	-	-	-	-	-	142
PPG trigger 6 input pin(0)	TRG6_0	Yes	-	-	12	15	18	22
PPG trigger 6 input pin(1)	TRG6_1	Yes	-	-	-	3	6	8
PPG trigger 6 input pin(2)	TRG6_2	Yes	-	-	-	-	-	143
PPG trigger 7 input pin(0)	TRG7_0	Yes	8	10	13	16	19	23
PPG trigger 7 input pin(1)	TRG7_1	Yes	-	-	-	4	7	9
PPG trigger 8 input pin(0)	TRG8_0	Yes	15	18	23	27	30	37
PPG trigger 8 input pin(1)	TRG8_1	Yes	-	-	-	-	-	27
PPG trigger 9 input pin(0)	TRG9_0	Yes	-	-	-	-	32	40
PPG trigger 9 input pin(1)	TRG9_1	Yes	-	-	-	-	-	28
PPG trigger 10 input pin(0)	TRG10_0	Yes	-	-	-	-	-	36
PPG trigger 11 input pin(0)	TRG11_0	Yes	-	-	-	-	-	38

Table 9-8 Pin of RTC

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
RTC output signal pin	WOT	-	63	79	99	119	140	171

Table 9-9 Pins of Up/down Counter

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
U/D counter ch.0 AIN input pin(0)	AIN0_0	Yes	9	11	14	17	20	24
U/D counter ch.0 AIN input pin(1)	AIN0_1	Yes	-	32	39	45	53	67
U/D counter ch.0 BIN input pin(0)	BIN0_0	Yes	10	12	15	18	21	25
U/D counter ch.0 BIN input pin(1)	BIN0_1	Yes	25	31	38	44	52	66
U/D counter ch.0 ZIN input pin(0)	ZIN0_0	Yes	-	-	16	19	22	26
U/D counter ch.0 ZIN input pin(1)	ZIN0_1	Yes	-	-	37	43	51	63
U/D counter ch.0 ZIN input pin(2)	ZIN0_2	Yes	8	10	13	16	19	23
U/D counter ch.1 AIN input pin(0)	AIN1_0	Yes	11	13	17	20	23	29
U/D counter ch.1 AIN input pin(1)	AIN1_1	Yes	24	30	36	42	50	62
U/D counter ch.1 BIN input pin(0)	BIN1_0	Yes	12	14	18	21	24	30

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
U/D counter ch.1 BIN input pin(1)	BIN1_1	Yes	-	29	35	41	49	60
U/D counter ch.1 ZIN input pin(0)	ZIN1_0	Yes	13	15	19	22	25	31
U/D counter ch.1 ZIN input pin(1)	ZIN1_1	Yes	23	28	34	40	48	59

Table 9-10 Pins of Output Compare (ch.0 to ch.5: 16bit, ch.6 to ch.11: 32bit)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Output compare ch.6 output pin(0)	OCU6_0	-	-	-	-	88	104	127
Output compare ch.6 output pin(1)	OCU6_1	-	-	-	16	19	22	26
Output compare ch.7 output pin(0)	OCU7_0	-	-	-	-	-	105	128
Output compare ch.7 output pin(1)	OCU7_1	-	10	12	15	18	21	25
Output compare ch.8 output pin(0)	OCU8_0	-	48	59	74	89	106	129
Output compare ch.8 output pin(1)	OCU8_1	-	9	11	14	17	20	24
Output compare ch.9 output pin(0)	OCU9_0	-	-	-	-	-	107	131
Output compare ch.9 output pin(1)	OCU9_1	-	8	10	13	16	19	23
Output compare ch.10 output pin(0)	OCU10_0	-	-	-	-	-	111	137
Output compare ch.10 output pin(1)	OCU10_1	-	-	-	12	15	18	22
Output compare ch.11 output pin(0)	OCU11_0	-	-	-	-	93	112	138
Output compare ch.11 output pin(1)	OCU11_1	-	7	9	11	14	17	21

*:16-bit output compare has no dedicated output pins. There is only the output through the wave generator.

Table 9-11 Pins of Input Capture (ch.0 to ch.3: 16bit, ch.4 to ch.9: 32bit)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Input capture ch.0 input pin(0)	ICU0_0	Yes	-	-	-	-	74	92
Input capture ch.0 input pin(1)	ICU0_1	Yes	-	-	-	33	39	49
Input capture ch.0 input pin(2)	ICU0_2	Yes	-	-	40	46	54	68
Input capture ch.0 input pin(3)	ICU0_3	Yes	8	10	13	16	19	23
Input capture ch.1 input pin(0)	ICU1_0	Yes	-	-	-	-	75	93
Input capture ch.1 input pin(1)	ICU1_1	Yes	19	24	29	35	41	51
Input capture ch.1 input pin(2)	ICU1_2	Yes	26	33	41	47	55	69
Input capture ch.1 input pin(3)	ICU1_3	Yes	-	-	12	15	18	22
Input capture ch.2 input pin(0)	ICU2_0	Yes	-	-	-	-	76	94
Input capture ch.2 input pin(1)	ICU2_1	Yes	-	-	32	38	46	57
Input capture ch.2 input pin(2)	ICU2_2	Yes	27	34	42	48	56	70
Input capture ch.2 input pin(3)	ICU2_3	Yes	7	9	11	14	17	21
Input capture ch.3 input pin(0)	ICU3_0	Yes	34	42	52	62	77	96
Input capture ch.3 input pin(1)	ICU3_1	Yes	22	27	33	39	47	58
Input capture ch.3 input pin(2)	ICU3_2	Yes	-	35	43	49	57	71
Input capture ch.3 input pin(3)	ICU3_3	Yes	6	8	10	13	16	20
Input capture ch.4 input pin(0)	ICU4_0	Yes	-	-	-	-	78	97
Input capture ch.4 input pin(1)	ICU4_1	Yes	-	-	-	26	29	35
Input capture ch.4 input pin(2)	ICU4_2	Yes	34	42	52	62	77	96

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Input capture ch.5 input pin(0)	ICU5_0	Yes	36	44	55	65	81	100
Input capture ch.5 input pin(1)	ICU5_1	Yes	14	17	22	25	28	34
Input capture ch.6 input pin(0)	ICU6_0	Yes	22	27	33	39	47	58
Input capture ch.6 input pin(1)	ICU6_1	Yes	-	16	21	24	27	33
Input capture ch.7 input pin(0)	ICU7_0	Yes	-	-	90	107	127	155
Input capture ch.7 input pin(1)	ICU7_1	Yes	-	-	20	23	26	32
Input capture ch.8 input pin(0)	ICU8_0	Yes	19	24	29	35	41	51
Input capture ch.8 input pin(1)	ICU8_1	Yes	13	15	19	22	25	31
Input capture ch.9 input pin(0)	ICU9_0	Yes	-	-	-	33	39	49
Input capture ch.9 input pin(1)	ICU9_1	Yes	12	14	18	21	24	30

Table 9-12 Pins of Free-run Timer (ch.0 to ch.2: 16bit, ch.3 to ch.5: 32bit)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Free-run timer 0 clock input pin(0)	FRCK0_0	Yes	23	28	34	40	48	59
Free-run timer 1 clock input pin(0)	FRCK1_0	Yes	-	29	35	41	49	60
Free-run timer 1 clock input pin(1)	FRCK1_1	Yes	28	36	45	55	63	79
Free-run timer 2 clock input pin(0)	FRCK2_0	Yes	24	30	36	42	50	62
Free-run timer 3 clock input pin(0)	FRCK3_0	Yes	-	-	37	43	51	63
Free-run timer 3 clock input pin(1)	FRCK3_1	Yes	-	-	-	-	-	126
Free-run timer 4 clock input pin(0)	FRCK4_0	Yes	25	31	38	44	52	66
Free-run timer 4 clock input pin(1)	FRCK4_1	Yes	-	-	-	-	-	121
Free-run timer 5 clock input pin(0)	FRCK5_0	Yes	-	32	39	45	53	67
Free-run timer 5 clock input pin(1)	FRCK5_1	Yes	-	-	-	-	-	120

Table 9-13 Pins of Base Timer (ch.0, ch.1)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Base timer ch.0 TIOA output pin(0)	TIOA0_0	-	-	-	-	119	140	171
Base timer ch.0 TIOA output pin(1)	TIOA0_1	-	-	-	-	111	131	159
Base timer ch.0 TIOB input pin(0)	TIOB0_0	Yes	-	-	-	-	141	172
Base timer ch.0 TIOB input pin(1)	TIOB0_1	Yes	-	-	-	113	133	161
Base timer ch.1 TIOA I/O pin(0)	TIOA1_0	Yes	-	-	-	-	142	174
Base timer ch.1 TIOA I/O pin(1)	TIOA1_1	Yes	-	75	95	112	132	160
Base timer ch.1 TIOB input pin(0)	TIOB1_0	Yes	-	-	-	-	143	175
Base timer ch.1 TIOB input pin(1)	TIOB1_1	Yes	-	76	96	114	134	162

Table 9-14 Pins of Reload Timer (ch.0 to ch.7)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Reload timer ch.0 output pin(0)	TOT0_0	-	4	5	6	9	12	15
Reload timer ch.0 output pin(1)	TOT0_1	-	-	-	-	-	76	94
Reload timer ch.0 event input pin(0)	TIN0_0	Yes	-	-	-	5	8	10
Reload timer ch.0 event input pin(1)	TIN0_1	Yes	-	-	-	-	-	90
Reload timer ch.0 event input pin(2)	TIN0_2	Yes	2	2	2	2	5	7

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Reload timer ch.1 output pin(0)	TOT1_0	-	-	-	7	10	13	17
Reload timer ch.1 output pin(1)	TOT1_1	-	-	-	-	-	-	95
Reload timer ch.1 output pin(2)	TOT1_2	-	14	17	22	25	28	34
Reload timer ch.1 event input pin(0)	TIN1_0	Yes	3	3	3	6	9	11
Reload timer ch.1 event input pin(1)	TIN1_1	Yes	-	-	-	-	-	91
Reload timer ch.2 output pin(0)	TOT2_0	-	-	-	8	11	14	18
Reload timer ch.2 output pin(1)	TOT2_1	-	-	-	52	62	77	96
Reload timer ch.2 event input pin(0)	TIN2_0	Yes	-	-	4	7	10	12
Reload timer ch.2 event input pin(1)	TIN2_1	Yes	-	-	-	-	74	92
Reload timer ch.3 output pin(0)	TOT3_0	-	5	7	9	12	15	19
Reload timer ch.3 output pin(1)	TOT3_1	-	-	-	-	-	78	97
Reload timer ch.3 event input pin(0)	TIN3_0	Yes	-	4	5	8	11	14
Reload timer ch.3 event input pin(1)	TIN3_1	Yes	-	-	-	-	75	93
Reload timer ch.3 event input pin(2)	TIN3_2	Yes	15	18	23	27	30	37
Reload timer ch.4 output pin(0)	TOT4_0	-	9	11	14	17	20	24
Reload timer ch.4 output pin(1)	TOT4_1	-	-	-	-	-	45	55
Reload timer ch.4 event input pin(0)	TIN4_0	Yes	6	8	10	13	16	20
Reload timer ch.4 event input pin(1)	TIN4_1	Yes	17	22	27	32	38	46
Reload timer ch.5 output pin(0)	TOT5_0	-	10	12	15	18	21	25
Reload timer ch.5 output pin(1)	TOT5_1	-	-	-	32	38	46	57
Reload timer ch.5 event input pin(0)	TIN5_0	Yes	7	9	11	14	17	21
Reload timer ch.5 event input pin(1)	TIN5_1	Yes	-	-	-	33	39	49
Reload timer ch.6 output pin(0)	TOT6_0	-	-	-	16	19	22	26
Reload timer ch.6 output pin(1)	TOT6_1	-	22	27	33	39	47	58
Reload timer ch.6 event input pin(0)	TIN6_0	Yes	-	-	12	15	18	22
Reload timer ch.6 event input pin(1)	TIN6_1	Yes	19	24	29	35	41	51
Reload timer ch.7 output pin(0)	TOT7_0	-	11	13	17	20	23	29
Reload timer ch.7 output pin(1)	TOT7_1	-	23	28	34	40	48	59
Reload timer ch.7 event input pin(0)	TIN7_0	Yes	8	10	13	16	19	23
Reload timer ch.7 event input pin(1)	TIN7_1	Yes	-	-	-	-	44	54

Table 9-15 Pins of External Bus Interface

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
External bus address bit0 output pin	A00	-	-	-	-	-	11	14
External bus address bit1 output pin	A01	-	-	-	-	-	12	15
External bus address bit2 output pin	A02	-	-	-	-	-	13	17
External bus address bit3 output pin	A03	-	-	-	-	-	14	18
External bus address bit4 output pin	A04	-	-	-	-	-	15	19
External bus address bit5 output pin	A05	-	-	-	-	-	16	20
External bus address bit6 output pin	A06	-	-	-	-	-	17	21
External bus address bit7 output pin	A07	-	-	-	-	-	20	24
External bus address bit8 output pin	A08	-	-	-	-	-	21	25

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
External bus address bit9 output pin	A09	-	-	-	-	-	22	26
External bus address bit10 output pin	A10	-	-	-	-	-	23	29
External bus address bit11 output pin	A11	-	-	-	-	-	24	30
External bus address bit12 output pin	A12	-	-	-	-	-	25	31
External bus address bit13 output pin	A13	-	-	-	-	-	26	32
External bus address bit14 output pin	A14	-	-	-	-	-	27	33
External bus address bit15 output pin	A15	-	-	-	-	-	28	34
External bus address bit16 output pin	A16	-	-	-	-	-	29	35
External bus address bit17 output pin	A17	-	-	-	-	-	30	37
External bus address bit18 output pin	A18	-	-	-	-	-	31	39
External bus address bit19 output pin	A19	-	-	-	-	-	32	40
External bus address bit20 output pin	A20	-	-	-	-	-	33	41
External bus address bit21 output pin	A21	-	-	-	-	-	34	42
External bus address strobe output pin	ASX	-	-	-	-	-	5	7
External bus system clock output pin	SYSCLK	-	-	-	-	-	35	43
External bus chip select 0 output pin	CS0X	-	-	-	-	-	6	8
External bus chip select 1 output pin	CS1X	-	-	-	-	-	7	9
External bus chip select 2 output pin(0)	CS2X	-	-	-	-	-	38	46
External bus chip select 3 output pin(0)	CS3X	-	-	-	-	-	39	49
External bus data bit16 I/O pin	D16	No	-	-	-	-	131	159
External bus data bit17 I/O pin	D17	No	-	-	-	-	132	160
External bus data bit18 I/O pin	D18	No	-	-	-	-	133	161
External bus data bit19 I/O pin	D19	No	-	-	-	-	134	162
External bus data bit20 I/O pin	D20	No	-	-	-	-	135	163
External bus data bit21 I/O pin	D21	No	-	-	-	-	136	165
External bus data bit22 I/O pin	D22	No	-	-	-	-	137	167
External bus data bit23 I/O pin	D23	No	-	-	-	-	138	168
External bus data bit24 I/O pin	D24	No	-	-	-	-	139	170
External bus data bit25 I/O pin	D25	No	-	-	-	-	140	171
External bus data bit26 I/O pin	D26	No	-	-	-	-	141	172
External bus data bit27 I/O pin	D27	No	-	-	-	-	142	174
External bus data bit28 I/O pin	D28	No	-	-	-	-	143	175
External bus data bit29 I/O pin	D29	No	-	-	-	-	2	2
External bus data bit30 I/O pin	D30	No	-	-	-	-	3	3
External bus data bit31 I/O pin	D31	No	-	-	-	-	4	5
External bus write strobe 0 output pin	WR0X	-	-	-	-	-	9	11
External bus write strobe 1 output pin	WR1X	-	-	-	-	-	10	12
External bus read strobe output pin	RDX	-	-	-	-	-	8	10
External bus RDY input pin(0)	RDY	No	-	-	-	-	41	51

Table 9-16 Pins of Waveform Generator (ch.0 to ch.5)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Waveform generator ch.0-ch.5 input pin(0) (unavailable with the sub oscillation)	DTTI_0	Yes	56	70	86	102	121	149
Waveform generator ch.0-ch.5 input pin(1)	DTTI_1	Yes	8	10	13	16	19	23
Waveform generator ch.0-ch.5 input pin(2)	DTTI_2	Yes	-	-	-	33	39	49
Waveform generator ch.0 output pin(0)	RTO0_0	-	-	-	-	82	98	118
Waveform generator ch.0 output pin(1)	RTO0_1	-	7	9	11	14	17	21
Waveform generator ch.1 output pin(0)	RTO1_0	-	-	-	-	83	99	119
Waveform generator ch.1 output pin(1)	RTO1_1	-	6	8	10	13	16	20
Waveform generator ch.2 output pin(0)	RTO2_0	-	-	56	70	84	100	122
Waveform generator ch.2 output pin(1)	RTO2_1	-	5	7	9	12	15	19
Waveform generator ch.3 output pin(0)	RTO3_0	-	46	57	71	85	101	123
Waveform generator ch.3 output pin(1)	RTO3_1	-	4	5	6	9	12	15
Waveform generator ch.4 output pin(0)	RTO4_0	-	47	58	72	86	102	124
Waveform generator ch.4 output pin(1)	RTO4_1	-	3	3	3	6	9	11
Waveform generator ch.5 output pin(0)	RTO5_0	-	-	-	73	87	103	125
Waveform generator ch.5 output pin(1)	RTO5_1	-	2	2	2	2	5	7

Table 9-17 Pin of Clock Monitor

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
Clock monitor output pin	MONCLK	-	26	33	41	47	55	69

Table 9-18 Pins of Port Function (General-Purpose I/O)

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
General-purpose I/O port	P000	Yes	-	-	94	111	131	159
General-purpose I/O port	P001	Yes	-	75	95	112	132	160
General-purpose I/O port	P002	Yes	-	-	-	113	133	161
General-purpose I/O port	P003	Yes	-	76	96	114	134	162
General-purpose I/O port	P004	Yes	-	-	-	-	135	163
General-purpose I/O port	P005	Yes	61	77	97	115	136	165
General-purpose I/O port	P006	Yes	62	78	98	116	137	167
General-purpose I/O port	P007	Yes	-	-	-	117	138	168
General-purpose I/O port	P010	Yes	-	-	-	118	139	170
General-purpose I/O port	P011	Yes	63	79	99	119	140	171
General-purpose I/O port	P012	Yes	-	-	-	-	141	172
General-purpose I/O port	P013	Yes	-	-	-	-	142	174
General-purpose I/O port	P014	Yes	-	-	-	-	143	175
General-purpose I/O port	P015	Yes	-	-	-	-	2	2
General-purpose I/O port	P016	Yes	-	-	-	-	3	3
General-purpose I/O port	P017	Yes	-	-	-	-	4	5
General-purpose I/O port	P020	Yes	2	2	2	2	5	7
General-purpose I/O port	P021	Yes	-	-	-	3	6	8

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
General-purpose I/O port	P022	Yes	-	-	-	4	7	9
General-purpose I/O port	P023	Yes	-	-	-	5	8	10
General-purpose I/O port	P024	Yes	3	3	3	6	9	11
General-purpose I/O port	P025	Yes	-	-	4	7	10	12
General-purpose I/O port	P026	Yes	-	4	5	8	11	14
General-purpose I/O port	P027	Yes	4	5	6	9	12	15
General-purpose I/O port	P030	Yes	-	-	7	10	13	17
General-purpose I/O port	P031	Yes	-	6	8	11	14	18
General-purpose I/O port	P032	Yes	5	7	9	12	15	19
General-purpose I/O port	P033	Yes	6	8	10	13	16	20
General-purpose I/O port	P034	Yes	7	9	11	14	17	21
General-purpose I/O port	P035	Yes	9	11	14	17	20	24
General-purpose I/O port	P036	Yes	10	12	15	18	21	25
General-purpose I/O port	P037	Yes	-	-	16	19	22	26
General-purpose I/O port	P040	Yes	11	13	17	20	23	29
General-purpose I/O port	P041	Yes	12	14	18	21	24	30
General-purpose I/O port	P042	Yes	13	15	19	22	25	31
General-purpose I/O port	P043	Yes	-	-	20	23	26	32
General-purpose I/O port	P044	Yes	-	16	21	24	27	33
General-purpose I/O port	P045	Yes	14	17	22	25	28	34
General-purpose I/O port	P046	Yes	-	-	-	26	29	35
General-purpose I/O port	P047	Yes	15	18	23	27	30	37
General-purpose I/O port	P050	Yes	-	-	-	28	31	39
General-purpose I/O port	P051	Yes	-	-	-	-	32	40
General-purpose I/O port	P052	Yes	-	-	-	-	33	41
General-purpose I/O port	P053	Yes	16	19	24	29	34	42
General-purpose I/O port	P054	Yes	-	-	-	-	35	43
General-purpose I/O port	P055	Yes	17	22	27	32	38	46
General-purpose I/O port	P056	Yes	-	-	-	33	39	49
General-purpose I/O port	P057	Yes	19	24	29	35	41	51
General-purpose I/O port	P060	Yes	-	-	32	38	46	57
General-purpose I/O port	P061	Yes	22	27	33	39	47	58
General-purpose I/O port	P062	Yes	23	28	34	40	48	59
General-purpose I/O port	P063	Yes	-	29	35	41	49	60
General-purpose I/O port	P064	Yes	24	30	36	42	50	62
General-purpose I/O port	P065	Yes	-	-	37	43	51	63
General-purpose I/O port	P066	Yes	25	31	38	44	52	66
General-purpose I/O port	P067	Yes	-	32	39	45	53	67
General-purpose I/O port	P070	Yes	-	-	40	46	54	68
General-purpose I/O port	P071	Yes	26	33	41	47	55	69
General-purpose I/O port	P072	Yes	27	34	42	48	56	70
General-purpose I/O port	P073	Yes	-	35	43	49	57	71

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
General-purpose I/O port	P074	Yes	-	-	-	50	58	74
General-purpose I/O port	P075	Yes	-	-	-	51	59	75
General-purpose I/O port	P076	Yes	-	-	-	52	60	76
General-purpose I/O port	P077	Yes	-	-	-	53	61	77
General-purpose I/O port	P080	Yes	-	-	-	-	64	80
General-purpose I/O port	P081	Yes	29	37	46	56	65	81
General-purpose I/O port	P082	Yes	30	38	47	57	66	82
General-purpose I/O port	P083	Yes	-	-	-	-	67	83
General-purpose I/O port	P084	Yes	-	-	-	-	68	84
General-purpose I/O port	P085	Yes	-	-	-	-	69	85
General-purpose I/O port	P086	Yes	-	-	48	58	70	86
General-purpose I/O port	P087	Yes	31	39	49	59	71	87
General-purpose I/O port	P090	Yes	-	-	-	-	74	92
General-purpose I/O port	P091	Yes	-	-	-	-	75	93
General-purpose I/O port	P092	Yes	-	-	-	-	76	94
General-purpose I/O port	P093	Yes	34	42	52	62	77	96
General-purpose I/O port	P094	Yes	-	-	-	-	78	97
General-purpose I/O port	P095	Yes	-	-	53	63	79	98
General-purpose I/O port	P096	Yes	35	43	54	64	80	99
General-purpose I/O port	P097	Yes	36	44	55	65	81	100
General-purpose I/O port	P100	Yes	-	48	59	69	85	104
General-purpose I/O port	P101	Yes	-	-	60	70	86	105
General-purpose I/O port	P102	Yes	40	49	61	71	87	106
General-purpose I/O port	P103	Yes	41	50	62	72	88	107
General-purpose I/O port	P104	Yes	42	51	63	73	89	108
General-purpose I/O port	P105	Yes	43	52	64	74	90	109
General-purpose I/O port	P106	Yes	-	-	65	75	91	110
General-purpose I/O port	P107	Yes	-	53	66	76	92	111
General-purpose I/O port	P110	Yes	45	55	68	80	96	116
General-purpose I/O port	P111	Yes	-	-	69	81	97	117
General-purpose I/O port	P112	Yes	-	-	-	82	98	118
General-purpose I/O port	P113	Yes	-	-	-	83	99	119
General-purpose I/O port	P114	Yes	-	56	70	84	100	122
General-purpose I/O port	P115	Yes	46	57	71	85	101	123
General-purpose I/O port	P116	Yes	47	58	72	86	102	124
General-purpose I/O port	P117	Yes	-	-	73	87	103	125
General-purpose I/O port	P120	Yes	-	-	-	88	104	127
General-purpose I/O port	P121	Yes	-	-	-	-	105	128
General-purpose I/O port	P122	Yes	48	59	74	89	106	129
General-purpose I/O port	P123	Yes	-	-	-	-	107	131
General-purpose I/O port	P124	Yes	-	-	-	-	111	137

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
General-purpose I/O port	P125	Yes	-	-	-	93	112	138
General-purpose I/O port	P126	Yes	50	63	78	94	113	139
General-purpose I/O port	P127	Yes	-	64	79	95	114	140
General-purpose I/O port	P130	Yes	-	-	80	96	115	141
General-purpose I/O port	P131	Yes	-	-	-	-	124	152
General-purpose I/O port	P132	Yes	-	-	-	105	125	153
General-purpose I/O port	P133	Yes	-	-	89	106	126	154
General-purpose I/O port	P134	Yes	-	-	90	107	127	155
General-purpose I/O port (without sub oscillation only)	P135	Yes	56	70	86	102	121	149
General-purpose I/O port (without sub oscillation only)	P136	Yes	57	71	87	103	122	150
General-purpose I/O port	P142	Yes	-	-	-	-	44	54
General-purpose I/O port	P143	Yes	-	-	-	-	45	55
General-purpose I/O port	P144	Yes	-	-	91	108	128	156
General-purpose I/O port	P150	Yes	-	-	12	15	18	22
General-purpose I/O port	P151	Yes	8	10	13	16	19	23
General-purpose I/O port	P152	Yes	-	-	44	54	62	78
General-purpose I/O port	P153	Yes	28	36	45	55	63	79
General-purpose I/O port	P154	Yes	-	-	-	77	93	113
General-purpose I/O port	P155	Yes	-	-	-	78	94	114
General-purpose I/O port	P160	Yes	-	-	-	-	-	135
General-purpose I/O port	P161	Yes	-	-	-	-	-	136
General-purpose I/O port	P162	Yes	-	-	-	-	-	142
General-purpose I/O port	P163	Yes	-	-	-	-	-	143
General-purpose I/O port	P164	Yes	-	-	-	-	-	164
General-purpose I/O port	P165	Yes	-	-	-	-	-	166
General-purpose I/O port	P166	Yes	-	-	-	-	-	169
General-purpose I/O port	P167	Yes	-	-	-	-	-	173
General-purpose I/O port	P170	Yes	-	-	-	-	-	4
General-purpose I/O port	P171	Yes	-	-	-	-	-	6
General-purpose I/O port	P172	Yes	-	-	-	-	-	13
General-purpose I/O port	P173	Yes	-	-	-	-	-	16
General-purpose I/O port	P174	Yes	-	-	-	-	-	27
General-purpose I/O port	P175	Yes	-	-	-	-	-	28
General-purpose I/O port	P176	Yes	-	-	-	-	-	36
General-purpose I/O port	P177	Yes	-	-	-	-	-	38
General-purpose I/O port	P180	Yes	-	-	-	-	-	47
General-purpose I/O port	P181	Yes	-	-	-	-	-	48
General-purpose I/O port	P182	Yes	-	-	-	-	-	56
General-purpose I/O port	P183	Yes	-	-	-	-	-	61
General-purpose I/O port	P184	Yes	-	-	-	-	-	64

Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
General-purpose I/O port	P185	Yes	-	-	-	-	-	65
General-purpose I/O port	P186	Yes	-	-	-	-	-	72
General-purpose I/O port	P187	Yes	-	-	-	-	-	73
General-purpose I/O port	P190	Yes	-	-	-	-	-	90
General-purpose I/O port	P191	Yes	-	-	-	-	-	91
General-purpose I/O port	P192	Yes	-	-	-	-	-	95
General-purpose I/O port	P193	Yes	-	-	-	-	-	112
General-purpose I/O port	P194	Yes	-	-	-	-	-	120
General-purpose I/O port	P195	Yes	-	-	-	-	-	121
General-purpose I/O port	P196	Yes	-	-	-	-	-	126
General-purpose I/O port	P197	Yes	-	-	-	-	-	130

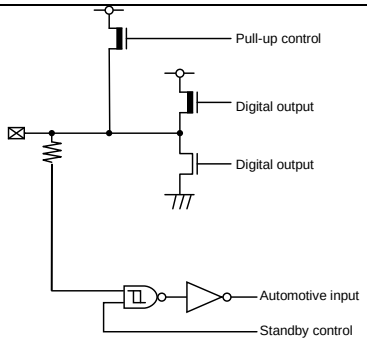
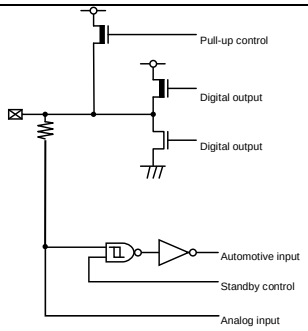
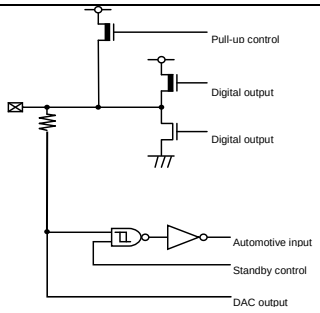
Table 9-19 Other Pins

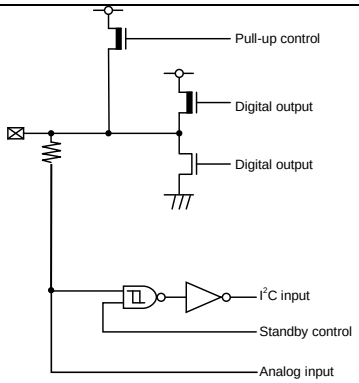
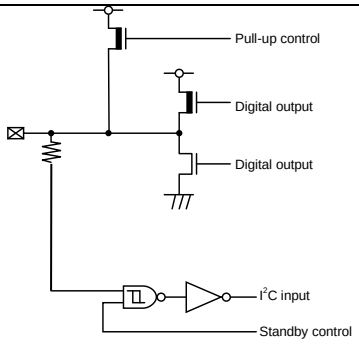
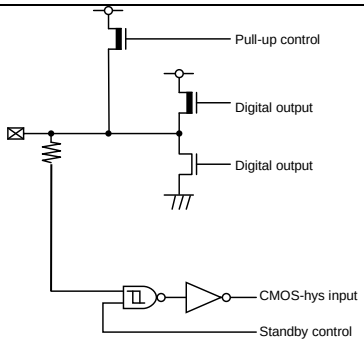
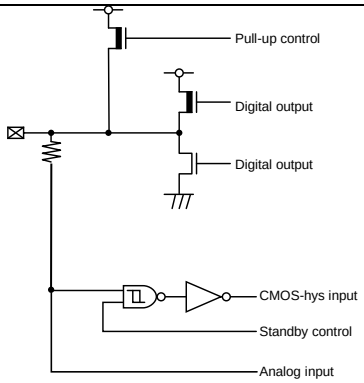
Function	Pin Name	Noise Filter	Pin Number					
			64	80	100	120	144	176
+5.0v power supply pin	VCC	-	32	20	25	30	36	44
		-	64	40	50	60	72	88
		-	-	61	76	91	109	133
		-	-	80	100	120	144	176
GND pin	VSS	-	1	1	1	1	1	1
		-	33	21	26	31	37	45
		-	55	41	51	61	73	89
		-	59	60	75	90	108	132
		-	-	69	85	101	120	148
		-	-	73	92	109	129	157
External capacity connection output pin	C	-	60	74	93	110	130	158
Main clock oscillation input pin	X0	Yes	53	67	83	99	118	146
Main clock oscillation output pin	X1	-	54	68	84	100	119	147
Sub clock oscillation input pin	X0A	Yes	57	71	87	103	122	150
Sub clock oscillation output pin	X1A	-	56	70	86	102	121	149
Mode pin 0	MD0	-	51	65	81	97	116	144
Mode pin 1	MD1	-	52	66	82	98	117	145
Interrupt input pin without mask	NMIX	Yes	44	54	67	79	95	115
DEBUGIF I/O pin for debug (OCD)	DEBUGIF	Yes	49	62	77	92	110	134
External reset input pin	RSTX	Yes	58	72	88	104	123	151

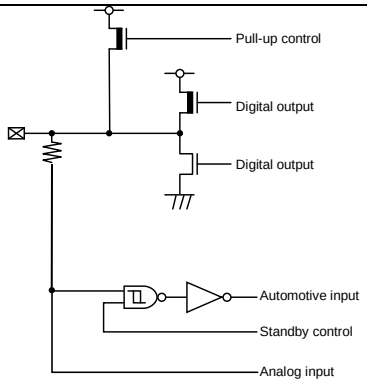
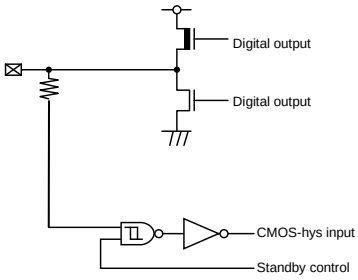
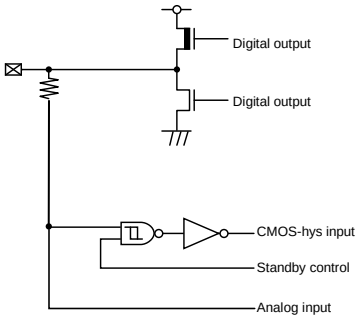
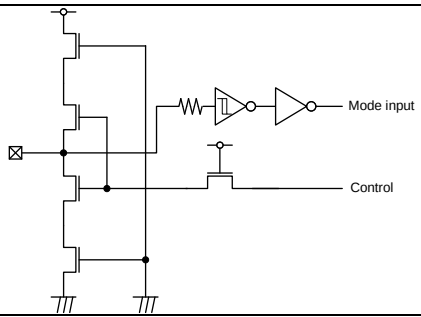
10. I/O Circuit Types

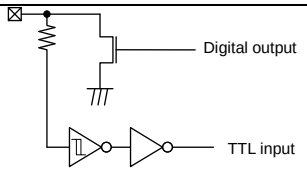
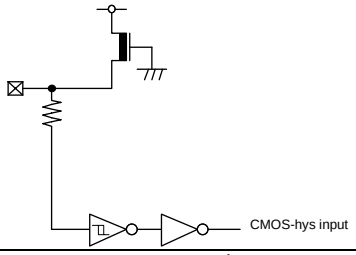
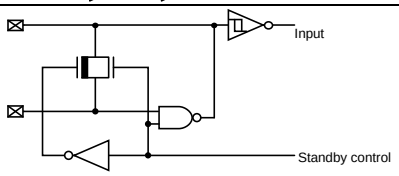
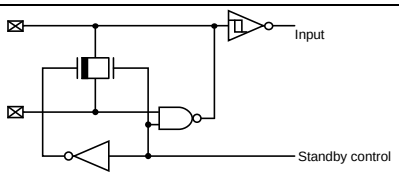
This section shows I/O Circuit Types.

Figure 10-1 I/O Circuit Types

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

Type	Circuit	Remarks
D		- I2C Analog input, General-purpose I/O port - Output 3mA - Pull-up resistor control 50kΩ - I2C hysteresis input
E		- I2C, General-purpose I/O port - Output 3mA - Pull-up resistor control 50kΩ - I2C hysteresis input
F		- General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - CMOS hysteresis input
G		- Analog input, General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - CMOS hysteresis input

Type	Circuit	Remarks
H		- Analog input, General-purpose I/O port - Output 12mA - Pull-up resistor control 50kΩ - Automotive input
I		- General-purpose I/O port (5V tolerant) - Output 4mA - CMOS hysteresis input
J		- Analog input, General-purpose I/O port (5V tolerant) - Output 4mA - CMOS hysteresis input
K		- Mode I/O - CMOS hysteresis input

Type	Circuit	Remarks
L		- Open-drain I/O - Output 25mA (Nch open-drain) - TTL input
M		- CMOS hysteresis input - Pull-up resistor 50kΩ
N		- Main oscillation I/O
O		- Sub oscillation I/O

Chapter 2: Handling The Device

This chapter explains the notes on using this series.

1. Handling Precautions
2. Handling Device
3. Application Notes

Code : HANDLING-1v1-91520-2-E

1. Handling Precautions

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

1.1. Precautions for Product Design

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

Absolute Maximum Ratings

Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of certain established limits, called absolute maximum 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.

1.2. Precautions for Package Mounting

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

Lead Insertion Type

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

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

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

Surface Mount Type

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

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

Lead-Free Packaging

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

Storage of Semiconductor Devices

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

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

Baking

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

Condition: 125 °C /24 h

Static Electricity

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

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

1.3. Precautions for Use Environment

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

For reliable performance, do the following:

(1) Humidity
Prolonged use in high humidity can lead to leakage in devices as well as printed circuit boards. If high humidity levels are anticipated, consider anti-humidity processing.
(2) Discharge of Static Electricity
When high-voltage charges exist close to semiconductor devices, discharges can cause abnormal operation. In such cases, use anti-static measures or processing to prevent discharges.

(3) Corrosive Gases, Dust, or Oil
Exposure to corrosive gases or contact with dust or oil may lead to chemical reactions that will adversely affect the device. If you use devices in such conditions, consider ways to prevent such exposure or to protect the devices.
(4) Radiation, Including Cosmic Radiation
Most devices are not designed for environments involving exposure to radiation or cosmic radiation. Users should provide shielding as appropriate.
(5) Smoke, Flame
CAUTION: Plastic molded devices are flammable, and therefore should not be used near combustible substances. If devices begin to smoke or burn, there is danger of the release of toxic gases.

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

2. Handling Device

This section explains the handling device.

■ Notes on Handling Device

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

● For latch-up prevention

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

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

In the correct power-on sequence in the microcontroller, turn on the digital power supply (Vcc) and analog power supplies (AVcc, AVRH), simultaneously. Or, turn on the digital power supply (Vcc5), and then turn on analog power supplies (AVcc, AVRH).

● Treatment of unused pins

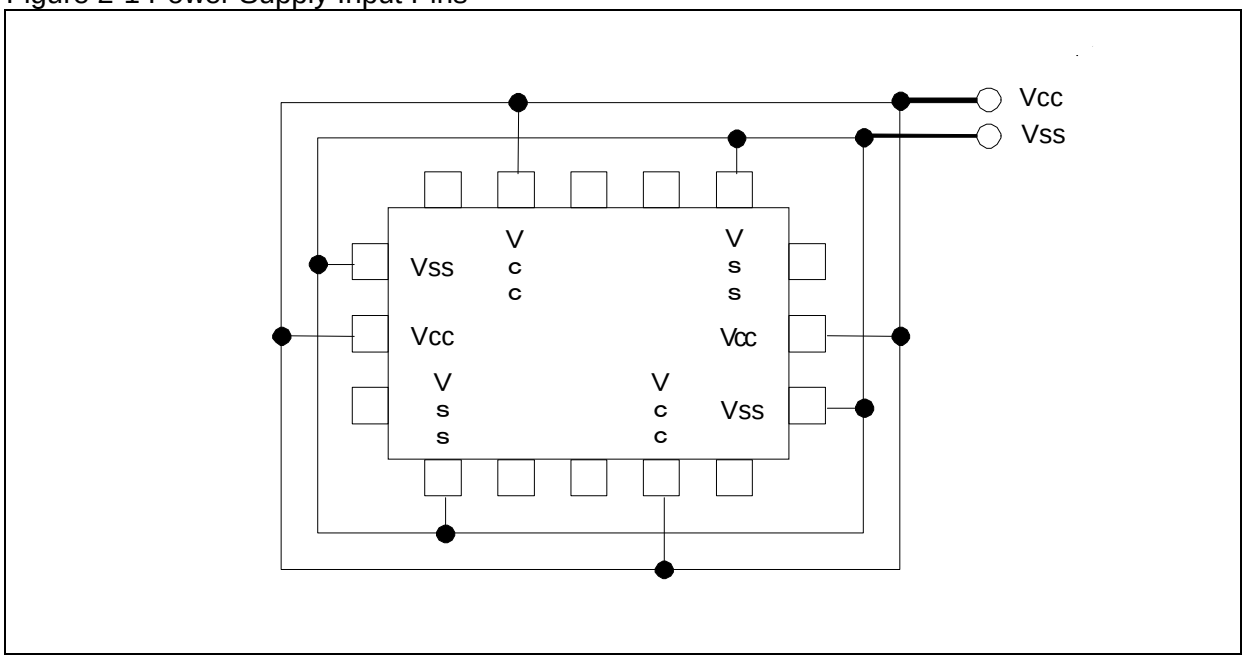
If unused input pins are left open, they may cause a permanent damage to the device due to malfunction or latch-up. Connect a 2kΩ resistor or more to each of unused pins for pull-up or pill-down processing.

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 prevent latch-up or other malfunctions by interconnecting VCC or VSS pins that should be kept at the same potential when the drive has multiple VCC or VSS pins. Be sure to connect all of these pins to an external power supply and ground to reduce unwanted radiation, prevent strobe signal malfunctioning due to a raised ground level, be in compliance with the total output current standard, etc. As shown in Figure 2-1, all Vss power supply pins must be treated in the same 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.

We recommend using a ceramic capacitor with a capacitance exceeding that of the C pin as a bypass capacitor between the VCC and VSS pins, in areas close to this device.

● 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 with ground circuits.

● Mode pins (MD1, MD0)

Connect the 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, the voltage rising must be monotonic increasing during power-on.

● Note during PLL clock operation

If the oscillator is disconnected or input stopped when the PLL clock has been selected, this clock may continue to operate at the free running frequency of the self oscillator circuit built in the PLL clock. This operation is not guaranteed.

● Treatment of A/D converter power supply pins

Connect the pins to have AVcc=AVRH=Vcc and AVss/AVRL=Vss even if the A/D converter is not used.

- **External clock is not supported**
  None of the external direct clock input can be used 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 (Vcc) first, and then turn on the A/D converter power supplies (AVcc, AVRH, AVRL) and analog inputs (AN0 to AN47). Also, turn off the A/D converter power supplies and analog inputs first, and then turn off the digital power supply (Vcc). When the AVRH 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 C pin**
  This device contains a voltage step-down circuit. A capacitor must always be connected to the C pin to assure the internal stabilization of the device. For the standard values, see the "Recommended Operating Conditions" of the latest data sheet.

Note:

For the detailed specifications of operating voltages, see the latest data sheet.

3. Application Notes

This section explains application notes.

3.1 Function Switching of a Multiplexed Port

3.2 Low-power Consumption Mode

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

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 multiplexed pin is also used as an external bus, its function is switched by the external bus setting. For details, see "CHAPTER: I/O PORTS".

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

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

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

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

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

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

Note:

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

Chapter 3: CPU



This chapter explains the CPU.

1. Overview
2. Features
3. CPU Operating Description
4. Pipeline Operation
5. Floating Point Operation Processing
6. Data Structure
7. Addressing
8. Programming Model
9. Reset and EIT Processing
10. Memory Protection Function (MPU)

Code : FR81S10-1v1-91520-2-E

1. Overview

This section explains the overview of the CPU.

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

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

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

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
- FPU, instruction access, and data access exception functions
  - FPU exceptions
  - Instruction access protection violation exception
  - Data access protection violation exception
  - Illegal instruction exception (changed from undefined instruction exception)
  - Data access error exception
  - Non-existent FPU exception

■ Memory Protection Function (MPU)

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

  - Read permitted / forbidden
  - Write permitted / forbidden
- The following attributes can be specified for each area
  - Bufferable/Non-Bufferable
- Access privileges and attributes can be specified for unset areas
- On protection violation, an instruction access protection violation exception or data access protection violation exception occurs

■ Floating Point Operations

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

3. CPU Operating Description

This section explains the operation of the CPU.

3.1 CPU Operating Status

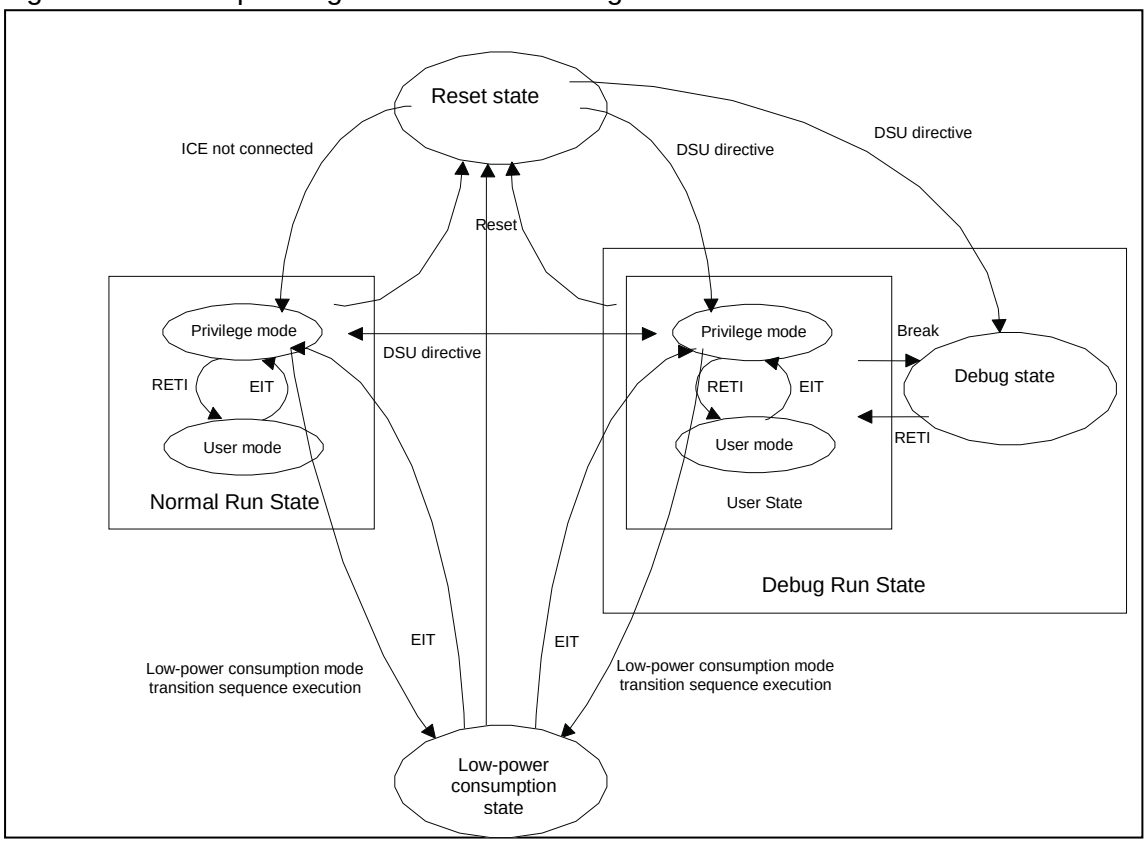
3.1. CPU Operating Status

The CPU operating status is shown below.

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

The operating state transitions are shown below.

Figure 3-1 CPU Operating State Transition Diagram



3.1.1. Reset State

The reset state is shown below.

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

3.1.2. Normal Run State

The normal run state is shown below.

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

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

3.1.3. Low-power Consumption State

The low-power consumption state is shown below.

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

3.1.4. Debug Run State

The debug run state is shown below.

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

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

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".

5. Floating Point Operation Processing

The floating point operation processing for the CPU is shown.

This series incorporates FPU.

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

6. Data Structure

This section explains the data structure of the CPU.

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

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

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

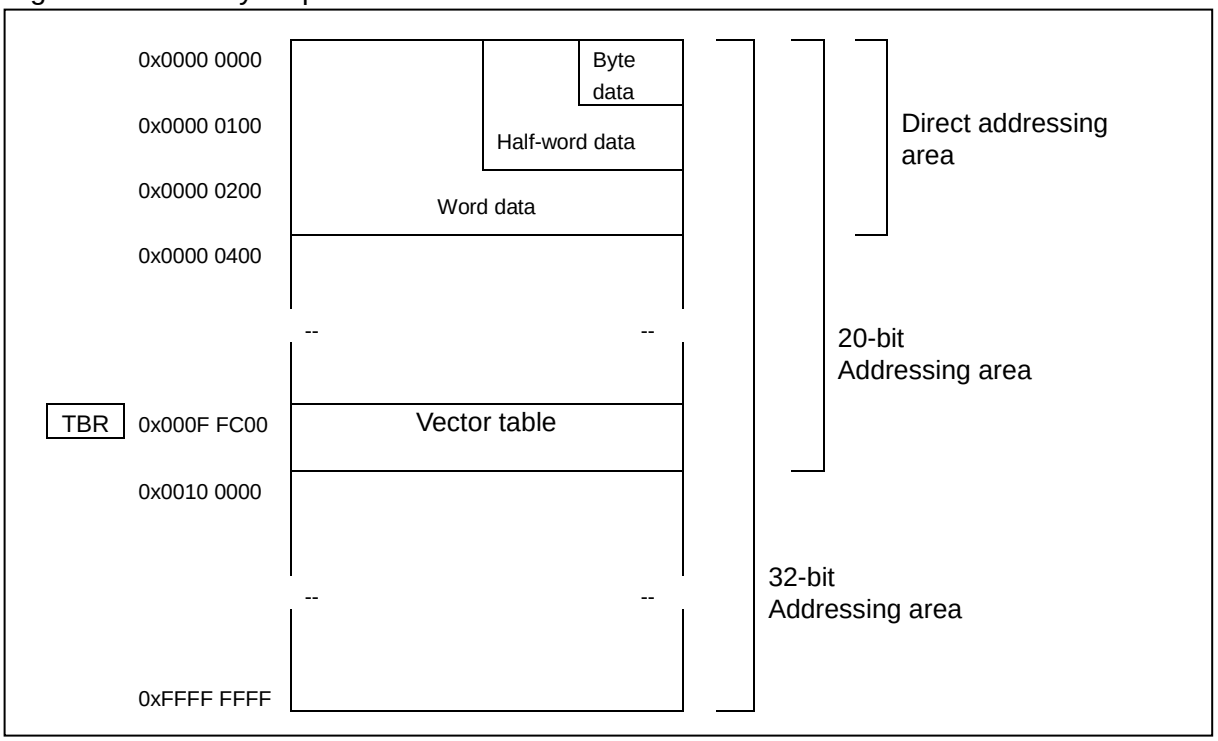
7. Addressing

This section explains addressing of the CPU.

A memory space is 32-bit linear.

The CPU manages the address space by bytes. Specify a value of 32-bit for the address on the address space to access from the CPU. Figure 7-1 shows the address space.

Figure 7-1 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".

8. Programming Model

This section explains the programming model of the CPU.

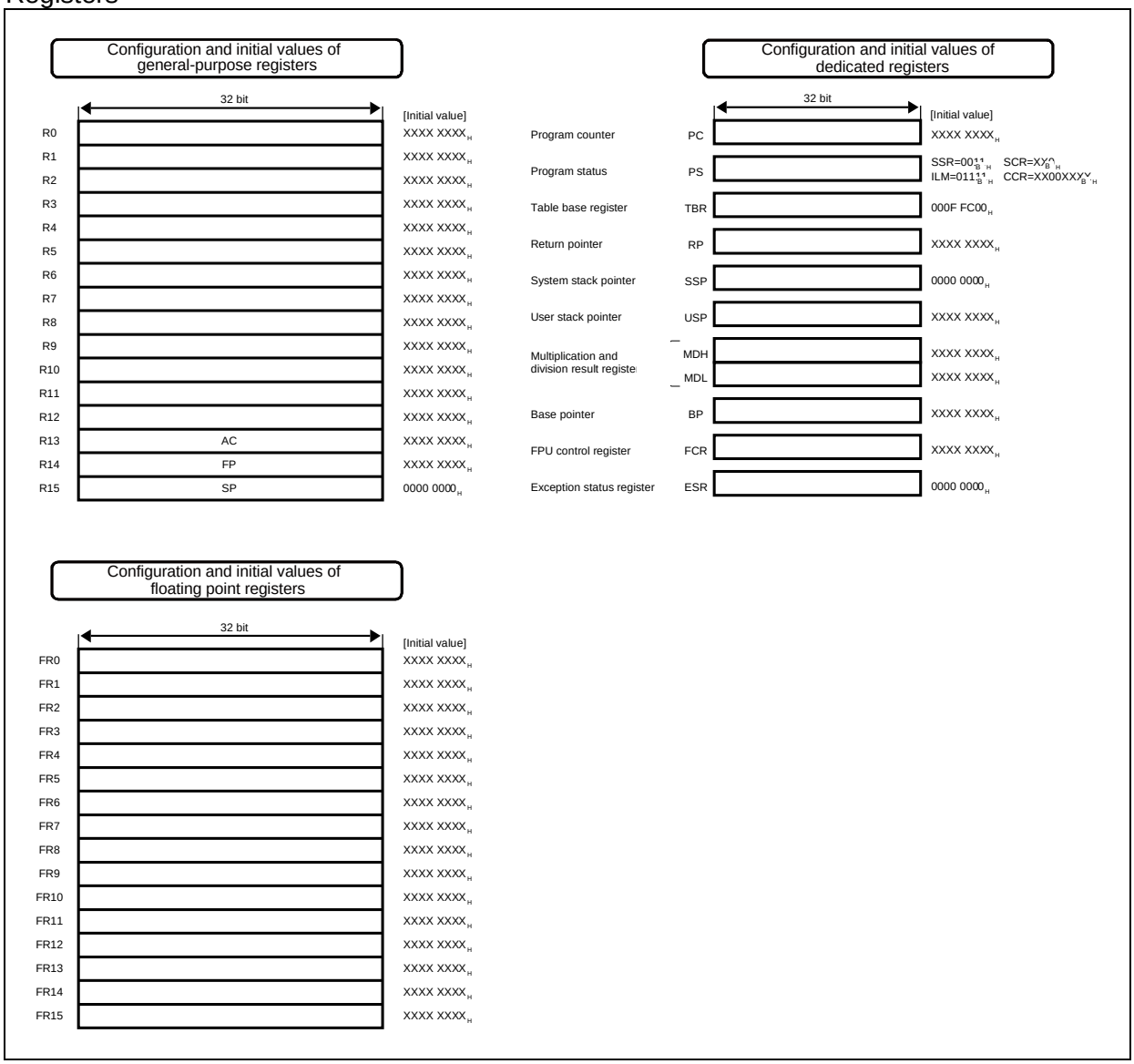
The CPU of FR81 has general-purpose registers, dedicated registers, and floating point registers. Besides these registers, the FR81 core has address-mapped system registers.

8.1. General-purpose Registers, Dedicated Registers, and Floating Point Registers

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

Figure 8-1 shows the initial values for this series. For details of each register, see "FR Family FR81 32-bit Microcontroller Programming Manual".

Figure 8-1 Initial Values of General-purpose Registers, Dedicated Registers, and Floating Point Registers



8.2. System Register

The system register is shown below.

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

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

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

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

9. Reset and EIT Processing

This section explains reset and EIT processing.

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

For details, see "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

9.1. Reset

The reset is shown below.

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

Note:

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

9.2. EIT Processing

The EIT processing is shown below.

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

9.3. Vector Table

The vector table is shown.

Table 9-1 Vector Table

Interruption Factor	Interrupt Vector Number		Interrupt Level	Offset	Address at TBR Initial Value
	Decimal	Hexadecimal			
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

Interruption Factor	Interrupt Vector Number		Interrupt Level	Offset	Address at TBR Initial Value
	Decimal	Hexadecimal			
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
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

10. Memory Protection Function (MPU)

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

10.1 Overview

10.2 List of Registers

10.3 Description of Registers

10.4 Operations of Memory Protection Function

10.1. Overview

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

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

- Eight protection areas can be specified that are shared by instructions and data
- The protection area with the highest priority is area 0, with the priority decreasing for areas 1, 2, 3, etc. (The areas can overlap)
- Areas are specified by a page address and a page size
  - Page size: Can be specified in units of 2^n bytes from 16 bytes
  - Page address: Misaligned 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

10.2. List of Registers

The list of registers is shown.

Table 10-1 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

10.3. Description of Registers

Registers are shown.

10.3.1 MPU Control Register : MPUCR

エラー! 参照元が見つかりません。 エラー! 参照元が見つかりません。

10.3.3 Instruction Access Protection Violation Status Register : IPVSR

10.3.4 Data Access Protection Violation Address Register :DPVAR

10.3.5 Data Access Protection Violation Status Register : DPVSR

10.3.6 Data Access Error Address Register : DEAR

10.3.7 Data Access Error Status Register : DESR

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

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

10.3.1. MPU Control Register : MPUCR

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

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

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

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

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

[bit15] PIE (Privilege Mode Instruction Fetch Enable)

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

PIE	Access to default area
0	Instruction fetch not permitted in privilege mode (Initial value)

1	Instruction fetch permitted in privilege mode

[bit14] PRE (Privilege Mode Read Access Enable)

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

PRE	Access to default area
0	Read access not permitted in privilege mode (Initial value)
1	Read access permitted in privilege mode

[bit13] PWE (Privilege Mode Write Access Enable)

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

PWE	Access to default area
0	Write access not permitted in privilege mode (Initial value)
1	Write access permitted in privilege mode

[bit12] UIE (User Mode Instruction Fetch Enable)

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

UIE	Access to default area
0	Instruction Fetch not enable 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 (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

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 "10.4.2. Instruction Access Protection Violation" and "10.4.7. Notes".

■ IPVAR : Address 0318_H (Access: Word)

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

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

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

Note:

Using this register is prohibited.

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 "10.4.2. Instruction Access Protection Violation" and "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.

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

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

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

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

10.3.6. Data Access Error Address Register : DEAR

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

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

■ DEAR : Address 0328_H (Access : Word)

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

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

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

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

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	bit30	•	•	•	bit10	Bit9	bit8
	PABR[31:8]							
Initial value	X	X	•	•	•	X	X	X
Attribute	R/W	R/W	•	•	•	R/W	R/W	R/W

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

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

These registers point to the base address of the protection area. The area from the address specified here to the size specified by the protection area control registers (PACR0 to 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.

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	Bufferable 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_B	Reserved
00001_B	Reserved
00010_B	Reserved
00011_B	16B
00100_B	32B
00101_B	64B
00110_B	128B
00111_B	256B
01000_B	512B
01001_B	1KB
01010_B	2KB
01011_B	4KB
01100_B	8KB
01101_B	16KB
01110_B	32KB
01111_B	64KB
10000_B	128KB
10001_B	256KB
10010_B	512KB
10011_B	1MB
10100_B	2MB
10101_B	4MB

ASZ[4:0]	Size of the specified protectorate area
10110_B	8MB
10111_B	16MB
11000_B	32MB
11001_B	64MB
11010_B	128MB
11011_B	256MB
11100_B	512MB
11101_B	1GB
11110_B	2GB
11111_B	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

10.4. Operations of Memory Protection Function

The memory protection function is shown below.

10.4.1 Setting Up Memory Protection Areas

10.4.2 Instruction Access Protection Violation

10.4.3 Data Access Protection Violation

10.4.4 Data Access Errors

10.4.5 Memory Protection Operation by Delay Slot

10.4.6 DEAR and DESR Update

10.4.7 Notes

10.4.1. Setting Up Memory Protection Areas

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

The memory protection function is configured by settings whether instructions, data reads, and data writes are permitted or forbidden in privilege mode and user mode for a maximum of 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.

10.4.2. Instruction Access Protection Violation

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

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

10.4.3. Data Access Protection Violation

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

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

If a data access protection violation occurs during the execution of an instruction that performs multiple data accesses, the data accesses that had executed up until the violation occurred are not 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.

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.

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.

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.

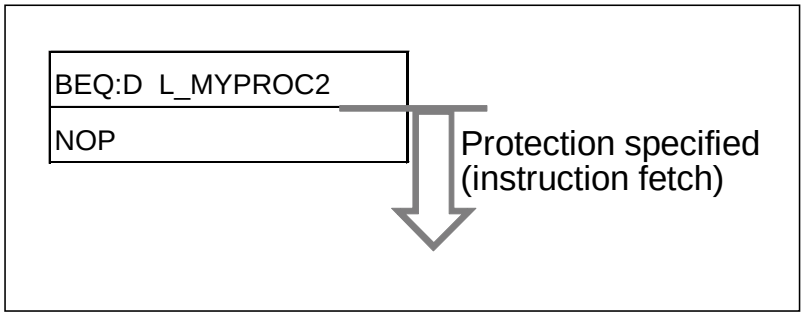
- System register access in user mode (illegal instruction exception)
- Bus error in buffer prohibition area access (illegal instruction exception)
- Bus error in buffer permission area access (data access error interrupt)

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.

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 "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.



Chapter 4: Operation Mode

This chapter explains the operation mode.

1. Overview
2. Features
3. Configuration
4. Register
5. Operation

Code : BMODED-2v0-91520-2-E

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.

2. Features

This section explains features of the operation mode.

This device supports the following operation modes.

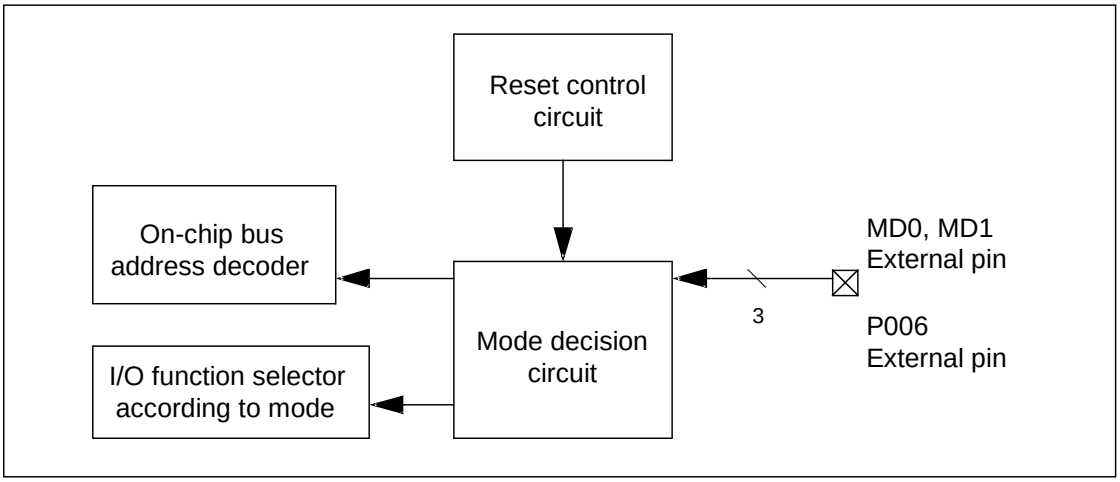
- User mode
  - The external bus interface can be used for MB91F52xL and MB91F52xK.
  - The program starts from the built-in FLASH.

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

3. Configuration

This section explains the configuration of the operation mode.

Figure 3-1 Block Diagram



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

4.1. Bus Mode Register : BMODR (Bus MODe Register)

The bit configuration of the bus mode register is shown.

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

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

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

*: It depends on operation mode.

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

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

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

5. Operation

This section explains operations of the operation mode.

5.1 MD0, MD1, P006 Pins Settings

5.2 Fetching the Operation Mode

5.3 Explanation of Each Operation Mode

5.1. MD0, MD1, P006 Pins Settings

MD0, MD1 and P006 pins settings are shown.

Table 5-1 Pin Settings

Operation mode	MD1	MD0	P006
User mode	0	1	-
Serial writer mode	1	0	1

Settings other than those shown in the table are prohibited.

Table 5-2 Correspondence of P006 Pin and Package

Package Type	Pin Number	Port Name
LQP176	167	P006
LQS144 / LQN144	137	P006
LQM120	116	P006
LQI100	98	P006
LQH080	78	P006
LQD064	62	P006

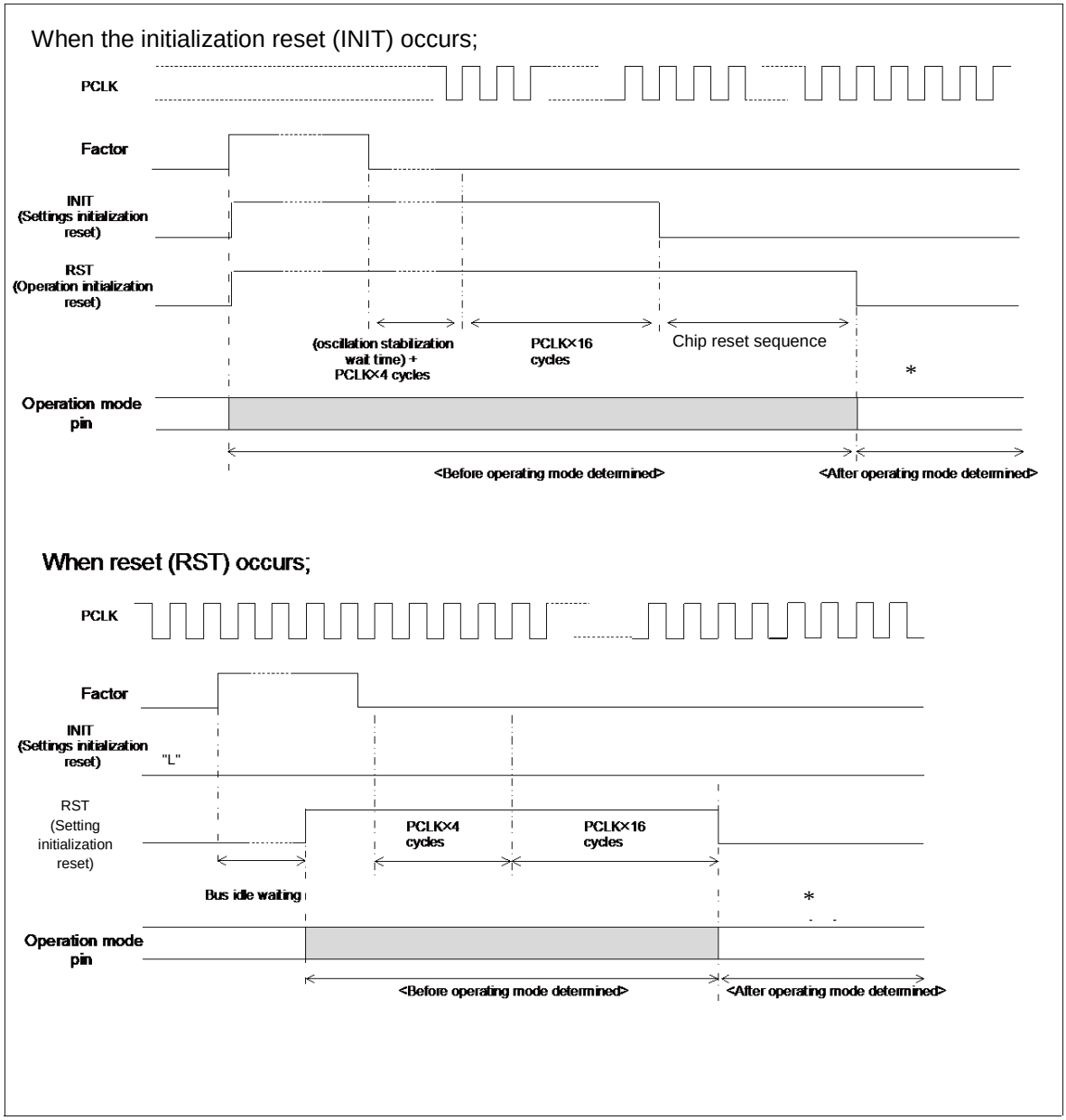
5.2. Fetching the Operation Mode

The fetching the operation mode is shown.

The operation mode is fetched by sampling the RST (Reset). During the time when an RST is issued and when it is released, the MD0, MD1 and P006 pin inputs must be determined. In User mode, the P006 pin does not need to be determined.

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

Figure 5-1 Operation Mode Fetch Timing Chart



* : Continue fixing MD0 and MD1 pins even after operating mode determined.
Note: When in serial writer mode, the P006 pin needs not be fixed after operating mode determined.

5.3. Explanation of Each Operation Mode

The each operation mode is shown.

The following details each operation mode.

5.3.1 User Mode

5.3.2 Serial Writer Mode

5.3.1. User Mode

The user mode is shown.

An external bus pin is reset immediately when a reset is entered for the external reset pin. For details, see "Pin Status Table" in "APPENDIX".

5.3.2. Serial Writer Mode

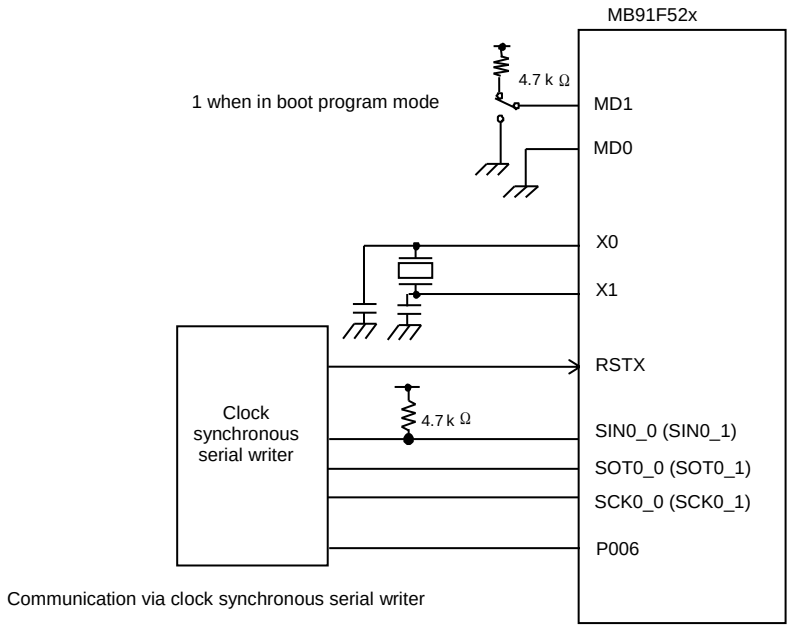
The serial writer mode is shown.

Table 5-3 Setting Pins in Serial Writer Mode

Pin number						Pin name	Serial writer mode		
64	80	100	120	144	176		Pin device/connection destination (for on-board programming)	Input level	Output level
51	65	81	97	116	144	MD0	GND	Hys.	-
52	66	82	98	117	145	MD1	VCC (Pull-up)	Hys.	-
58	72	88	104	123	151	RSTX	Reset input	Hys.	-
53	67	83	99	118	146	X0	Oscillation pin	-	-
54	68	84	100	119	147	X1	Oscillation pin	-	-
62	78	98	116	137	167	P006	Serial writer mode is started by adding pull-up resistor to external and, after releasing reset, setting level "H".	Auto	CMOS
-	-	78	94	113	139	P126 SIN0_0	Setting the input of this pin to "H" until the start of communication enables clock asynchronous communication mode, while setting it to "L" enables clock synchronous communication mode. Serial writer mode starts, and at the point at which communication starts, this pin is used as a UART serial data input pin.	Hys.	CMOS
11	13	-	-	-	-	P040 SIN0_1			
-	-	79	95	114	140	P127 SOT0_0	Serial writer mode starts and, at the point at which communication starts, this pin becomes a serial data output pin.	Auto	CMOS

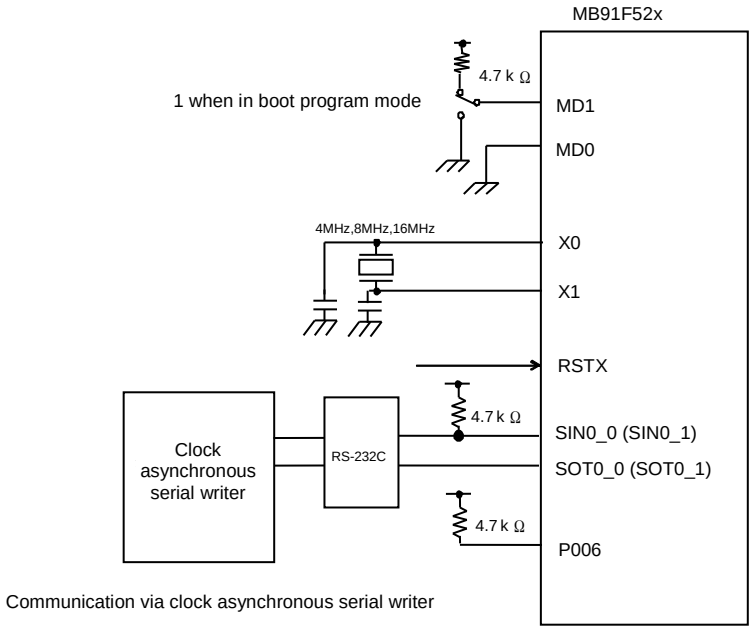
Pin number						Pin name	Serial writer mode		
64	80	100	120	144	176		Pin device/connection destination (for on-board programming)	Input level	Output level
15	18	-	-	-	-	P047 SOT0_1			
-	-	80	96	115	141	P130 SCK0_0	When the communication mode is clock synchronous communication mode, this pin becomes the serial clock input/output pin.	Hys.	CMOS
16	19	-	-	-	-	P053 SCK0_1			
32, 64	20, 40, 61, 80	25, 50, 76, 100	30, 60, 91, 120	36, 72, 109, 144	44, 88, 133, 176	VCC	+5.0v power supply	-	-
1, 33, 55, 59	1, 21, 41, 60, 69, 73	1, 26, 51, 75, 85, 92	1, 31, 61, 90, 101, 109	1, 37, 73, 108, 120, 129	1, 45, 89, 132, 148, 157	VSS	GND	-	-
18, 39	23, 47	28, 58	34, 68	40, 84	50, 103	AVCC1 AVCC0	VCC	-	-
21, 37	26, 45	30, 56	37, 66	43, 82	53, 101	AVSS1 AVRL1 AVSS0 AVRL0	VSS	-	-
20, 38	25, 46	30, 57	36, 67	42, 83	52, 102	AVRH1 AVRH0	VCC	-	-

Figure 5-2 Example of Connection for Serial Writer Mode Using Clock Synchronous Serial Writer



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

Figure 5-3 Example of Connection for Serial Writer Mode Using Clock Asynchronous Serial Writer



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

Chapter 5: Clock

This chapter explains the clock.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : BG4ACCTL-1v1-91520-2-E

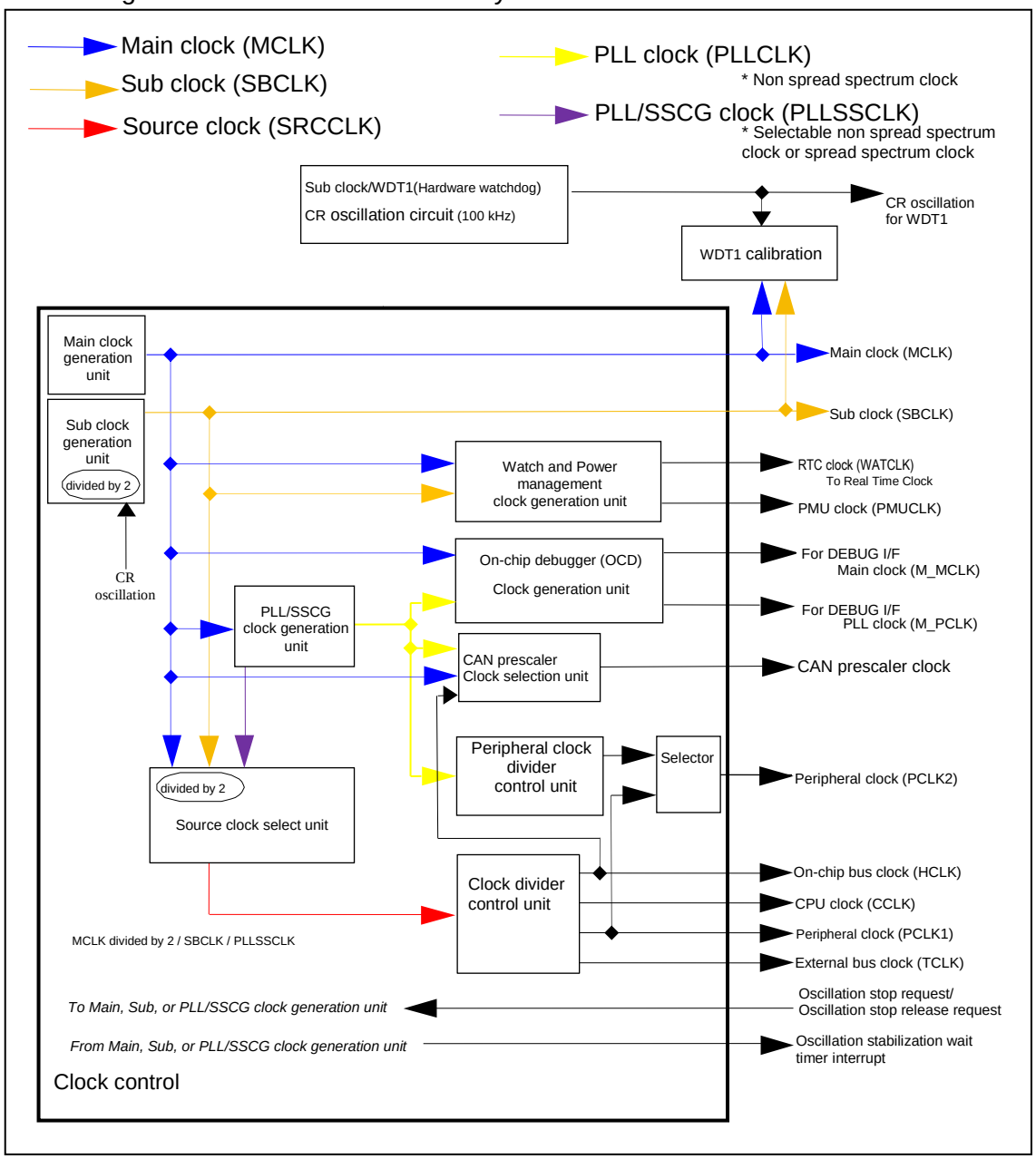
1. Overview

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 that can be used as sub-clock.

- 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 1-1 Diagram of the Clock Generation System



2. Features

This section explains features of the clock.

- 2 system on-chip oscillators are implemented.
- The main clock (MCLK) is multiplied by on-chip PLL/SSCG.
- Each clock has been forced not to supply by using the timer until it becomes stabilized (oscillation stabilization wait timer).
- Oscillation stabilization wait end interrupt can be generated.
- Main clock oscillation stabilization wait timer (main timer) 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.
- In a single clock product, the CR oscillation clock can be used as a sub-clock source. Refer to the "CHAPTER: CLOCK SUPERVISOR" for the selection method.
- Generates the clock for CAN prescaler. The clock can be selected from PLL clock (PLLCLK) [non spread spectrum clock] and main clock (MCLK). When PLL stops when PLL clock is selected, on-chip bus clock (HCLK) is used.
- For the noise decrement, the SSCG clock [spread spectrum clock] can be selected as CPU and a clock of the resource.

Note:

If main timer or sub timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.

3. Configuration

This section explains the configuration of the clock.

Figure 3-1 Connection Diagram of Clock (1)-1 Main Clock Generation Unit

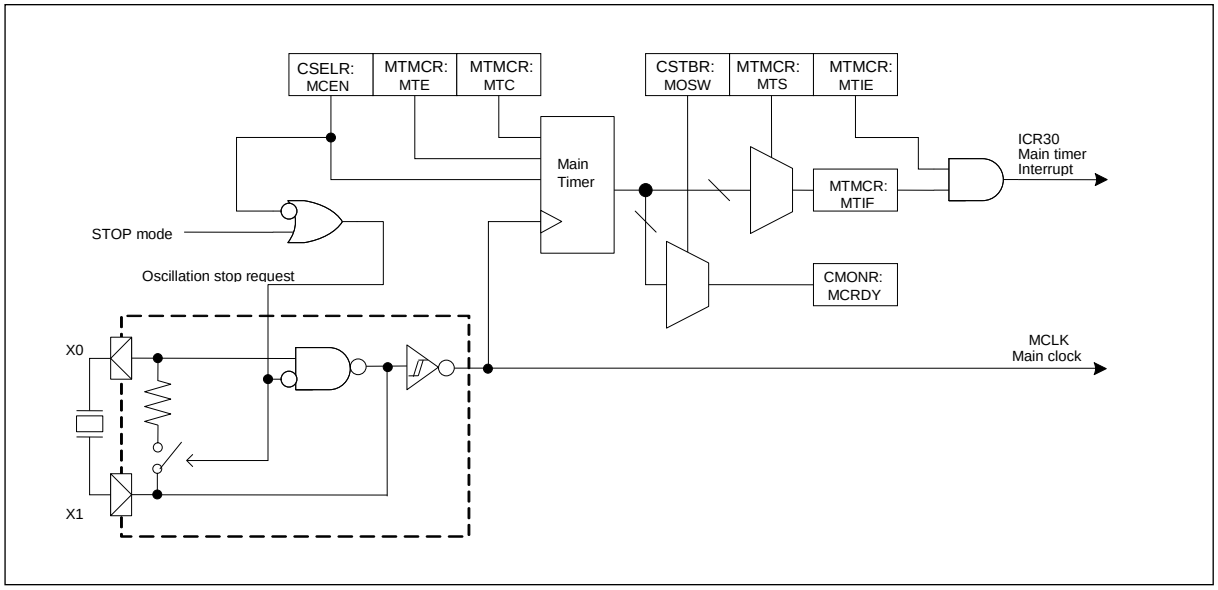


Figure 3-2 Connection Diagram of Clock (1)-2 Sub Clock Generation Unit

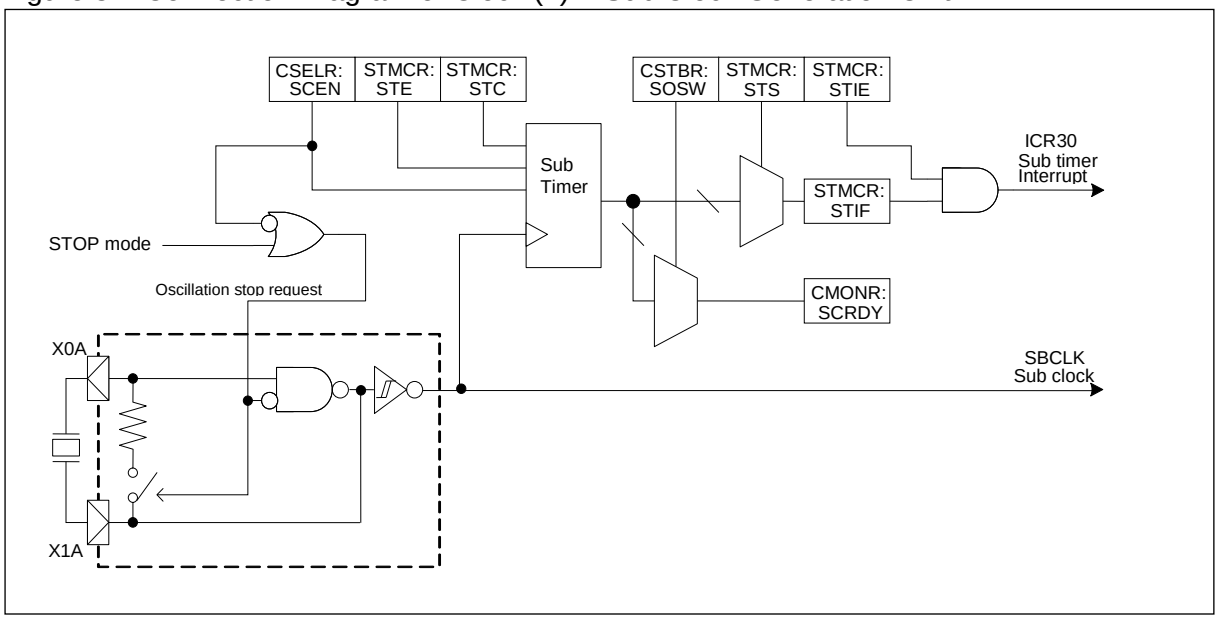


Figure 3-3 Connection Diagram of Clock (1)-3 PLL/SSCG Clock Generation Unit

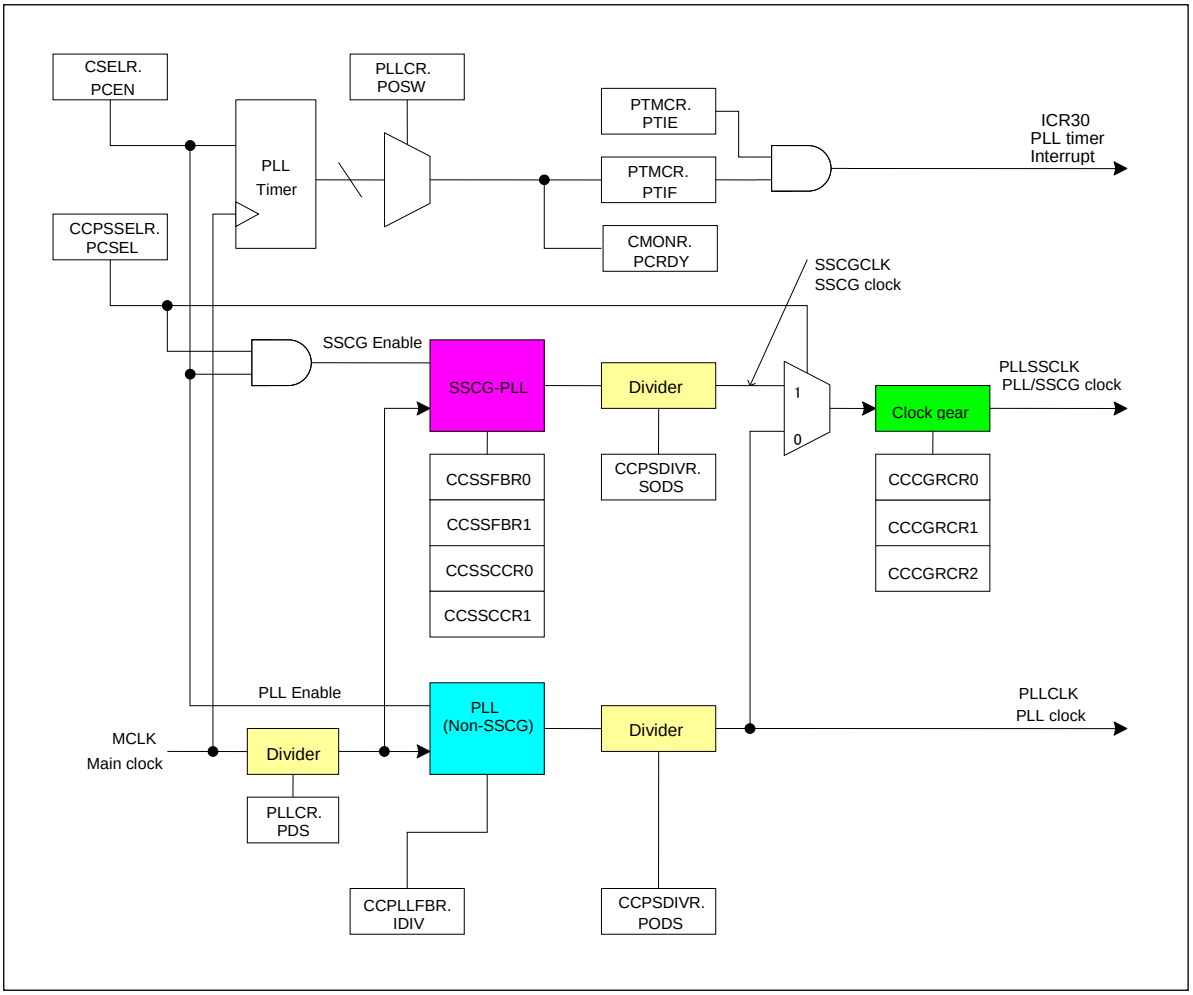


Figure 3-4 Connection Diagram of Clock (2) Source Clock Selection Unit

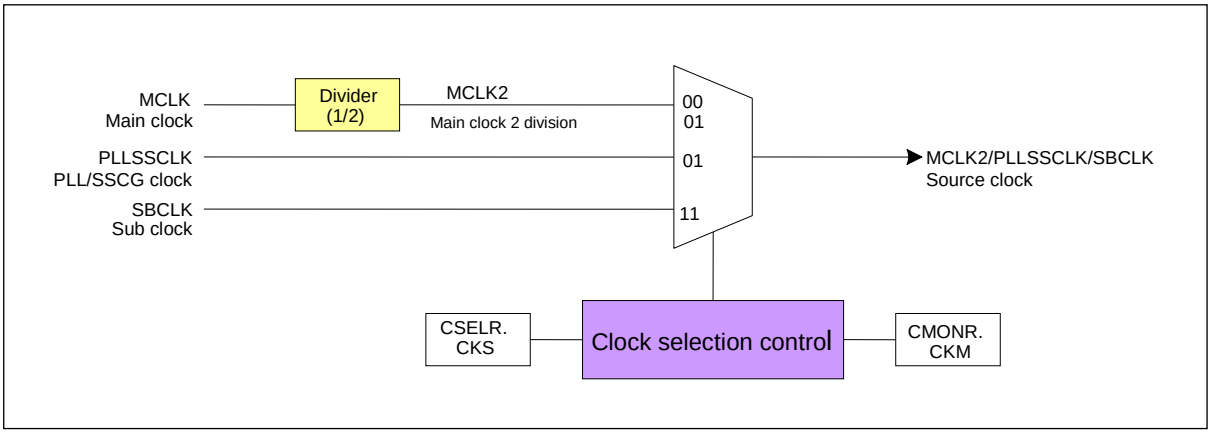


Figure 3-5 Connection Diagram of Clock (3) Divider Control

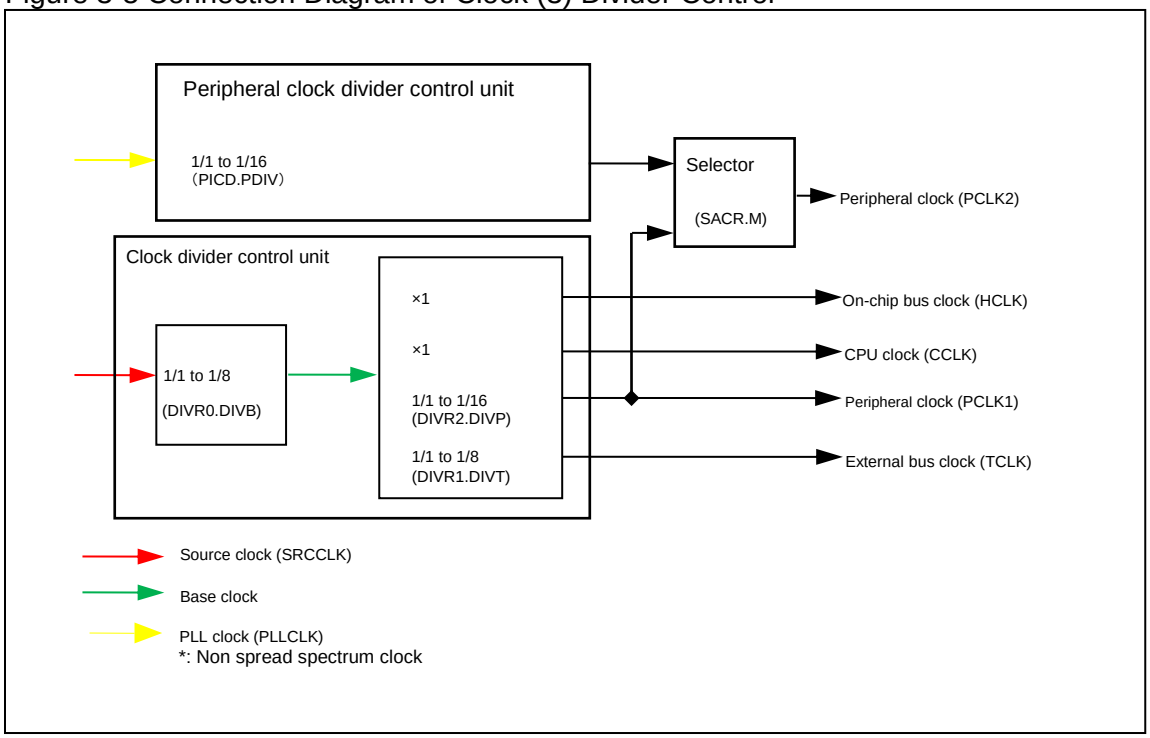


Figure 3-6 Connection Diagram of Clock (4) CAN Prescaler Clock Generation

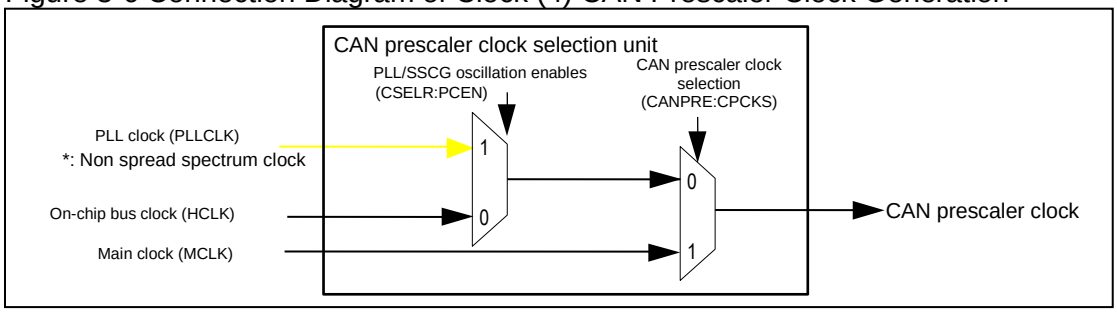


Figure 3-7 Connection Diagram of Clock (5) Watch/Power Management Clock Generation

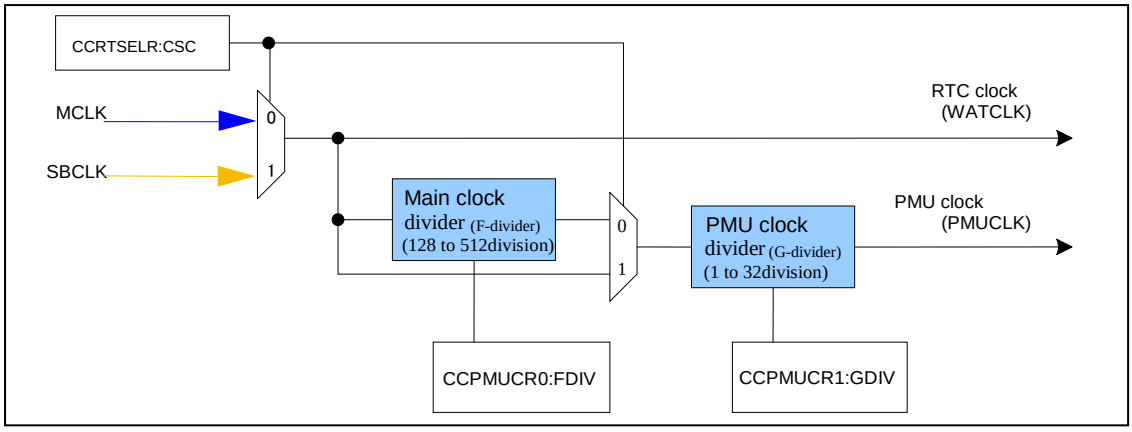


Figure 3-8 Diagram of the Clock System

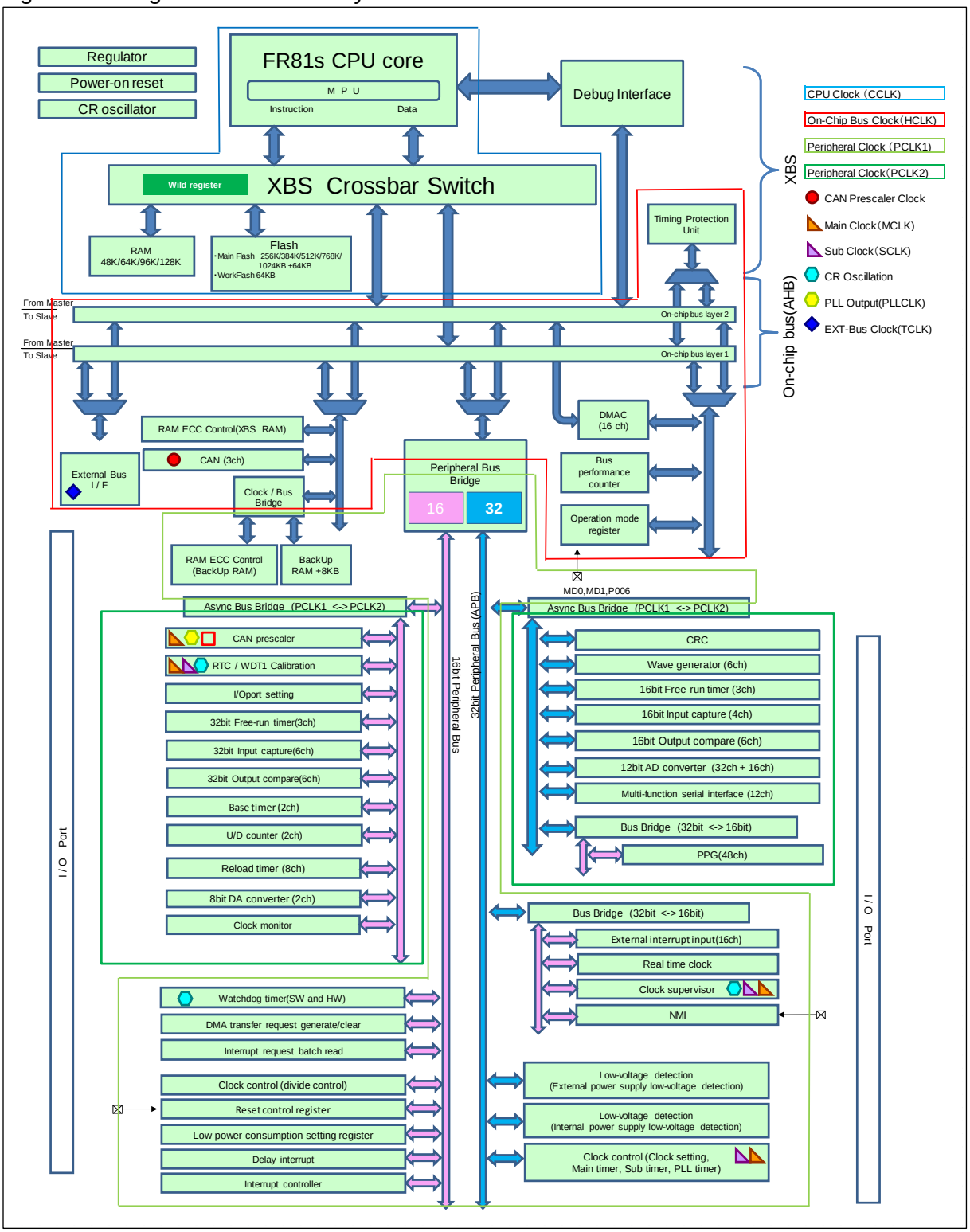


Table 3-1 List of functions that use PCLK1/PCLK2

Functions that use PCLK1	Functions that use PCLK2
RAM ECC Control (BackUp RAM)	CAN prescaler
BackUp RAM	RTC/WDT1 Calibration
Watchdog timer(SW and HW)	I/O port setting
DMA transfer request generate/clear	32bit Free-run timer (3ch)
Interrupt request batch read	32bit Input capture (6ch)
Clock control (divide control)	32bit Output compare (6ch)
Reset control register	Base timer (2ch)
Low-power consumption setting register	U/D counter (2ch)
Delay interrupt	Reload timer (8ch)
Interrupt controller	8bit DA converter (2ch)
External interrupt input (16ch)	Clock monitor
Real time clock	CRC
Clock supervisor	Wave generator (6ch)
NMI	16bit Free-run timer (3ch)
Low-voltage detection (External power supply low-voltage detection)	16bit Input capture (4ch)
Low-voltage detection (Internal power supply low-voltage detection)	16bit Output compare (6ch)
Clock control (Clock setting, Main timer, Sub timer, PLL timer)	12bit AD converter (32ch + 16ch)
-	Multi-function serial interface (12ch)
-	PPG (48ch)

4. Registers

This section explains registers of the clock.

Table 4-1 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

Address	Registers				Register function
	+0	+1	+2	+3	
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 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 Configuration Register 0 PMU Clock Division Configuration 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

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 Base clock) : Base clock division setting

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

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

[bit4 to bit0] (Reserved)

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 0489_H (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 STоP) : 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	No 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	No 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)

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 (PCLK) from the base clock.

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

Note:

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

[bit3 to bit0] (Reserved)

4.4. Clock Source Selection Register : CSELR (Clock source SELection Register)

The bit configuration of the division selection register 0 is shown.

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

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

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

*: This bit is initialized to "0". But this bit is not initialized by the return from the watch mode (power-shutdown).

Note:

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

[bit7] 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.
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".

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

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

PCEN	Oscillation control for PLL/SSCG clock (PLLSSCLK)
0	Stop oscillation (Initial value)
1	Oscillate

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

Note:

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

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

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

MCEN	Oscillation control for main clock
0	Stop oscillation
1	Oscillate (Initial value)

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

Note:

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

[bit4 to bit2] (Reserved)

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

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

CKS	Source selection
00	Division of the main clock (MCLK) by 2(Initial value)

CKS	Source selection
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.

4.5. Clock Source Monitor Register : CMONR (Clock source MONitor Register)

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

This register displays the status of each clock source and the selected source clock (SRCCLK).

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

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

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

*: This bit is initialized to "0". But this bit is not initialized by the return from the watch mode (power-shutdown).

Note:

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

[bit7] SCRDY (Sub Clock ReaDY) : 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.

You cannot select a sub clock (SBCLK) as the source clock 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) : PLL clock ready

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

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

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

Note:

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

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

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

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

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

Note:

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

[bit4 to bit2] (Reserved)

[bit1, bit0] CKM[1:0] (ClocK Monitor) : Source clock display
These bits show the source clock (SRCCLK) currently selected.

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

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 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.

An internal reset is issued at the return from standby mode (power-shutdown), and the main timer interrupt flag is not set.

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

This bit controls interrupts by main timer overflow as follows.

MTIE	Main timer interrupt
0	Interrupt disabled (Initial value)
1	Interrupt enabled (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	Read
0	Does nothing.	Operating normally
1	Clear the main timer.	Clearing the main timer

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

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

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

MTE	Main timer operation
0	Operation disabled (Initial value)
1	Operation enabled

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

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

[bit3 to bit0] MTS[3:0] (Main clock Timer interval 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 4 MHz
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 MTS[3] always reads "1".

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

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

This flag indicates that an overflow happens in the interval for which the sub timer has selected.
If this bit is set when the STIE bit is "1", 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.

An internal reset is issued at the return from standby mode (power-shutdown), and the sub timer interrupt flag is not set.

[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	Read
0	Does nothing.	Operating normally
1	Clear the sub timer.	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 enable

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 32 kHz	At CR clock selected
000	2^8× sub clock cycle	8[ms]	5.12[ms]
001	2^9× sub clock cycle	16[ms]	10.24[ms]
010	2^{10}× sub clock cycle	32[ms]	20.48[ms]
011	2^{11}× sub clock cycle	64[ms]	40.96[ms]
100	2^{12}× sub clock cycle	128[ms]	81.92[ms]
101	2^{13}× sub clock cycle	0.256[s]	163.84[ms]
110	2^{14}× sub clock cycle	0.512[s]	327.68[ms]
111	2^{15}× sub clock cycle (Initial value)	1.024[s]	655.36[ms]

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

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

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

This register configures the multiplication rate in the PLL/SSCG clock oscillation circuit generating the PLL/SSCG clock (PLLSSCLK) from the main clock (MCLK).

When PLL/SSCG clock oscillation is allowed (CSELR.PCEN=1), writing to this register has no effect.

[bit15, bit14] (Reserved)
Always write "0".

[bit13] (Reserved)

[bit12 to bit8] (Reserved)
Always write "0".

[bit7 to bit4] POSW[3:0] (Pll clock OSc Wait) : PLL oscillation stabilization wait selection
These bits select the oscillation stabilization wait time for the PLL/SSCG clock (PLLSSCLK) as follows.

POSW[3:0]	PLL/SSCG clock oscillation stabilization wait time	At 4 MHz	At 8 MHz
1000	2^9 × 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 lock up time wait time specification in this product is 200[µs]. Reserve the 200[µs] wait time or more by either of the following methods.

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

[bit3 to bit0] PDS[3:0] (Pll input clock Divider Selection) : 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 "5.1.3 PLL/SSCG Clock (PLLSSCLK)" for configuration samples.

A set value is limited. See "5.1.4 Limitations when PLL/SSCG Clock is used" when you set it.

4.9. Clock Stabilization Selection Register : CSTBR (Clock STaBilization selection Register)

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 32 kHz	At CR clock selected
000	2^8 × sub clock cycle (Initial value)	8[ms]	5.12[ms]
001	2^9 × sub clock cycle	16[ms]	10.24[ms]
010	2^{10} × sub clock cycle	32[ms]	20.48[ms]
011	2^{11} × sub clock cycle	64[ms]	40.96[ms]
100	2^{12} × sub clock cycle	128[ms]	81.92[ms]
101	2^{13} × sub clock cycle	0.256[s]	163.84[ms]
110	2^{14} × sub clock cycle	0.512[s]	327.68[ms]
111	2^{15} × sub clock cycle	1.024[s]	655.36[ms]

[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]

MOSW[3:0]	Main clock oscillation stabilization wait time	At 4 MHz
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.

The period of the failure detection cycle is as follow.

2^{12} ×CR Oscillation time = approx. 40.96 ms

4.10. PLL Oscillation Timer Control Register : PTMCR (PLL clock osc TiMer Control Register)

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

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

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

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

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

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

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

Writing "1" to this bit is ignored.

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 oscillation stabilization wait timer interrupt enable

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

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

[bit5 to bit0] (Reserved)

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.

This register selects which to use, PLL or SSCG.

It 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	Select PLL
1	Select SSCG

Note:

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

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.

This register sets the ratio of dividing frequency of the PLL/SSCG clock.

It can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

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

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

[bit7] (Reserved)

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

These bits set the ratio of dividing frequency of the PLL clock.

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:

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

[bit3] (Reserved)

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

These bits set the ratio of the dividing frequency of the SSCG clock.

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:

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

A set value is limited. See "5.1.4 Limitations when PLL/SSCG Clock is used" when you set it.

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.

This register sets the multiple ratio of PLL.

It can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

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

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

[bit7] (Reserved)

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

These bits set the PLL multiple ratio.

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 "5.1.4 Limitations when PLL/SSCG Clock is used" when you set it.

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.

This register sets the multiple ratio N 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

These bits set the SSCG multiple ratio N.

NDIV[5:0]	Dividing frequency ratio setting
000000	Setting is prohibited
000001	2
000010	3
…	……
111101	62
111110	63
111111	Setting is prohibited

A set value is limited. See "5.1.4 Limitations when PLL/SSCG Clock is used" when you set it.

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.

This register 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

These bits set the SSCG multiple ratio P.

PDIV[4:0]	Dividing frequency ratio setting
00000	1
00001	2
00010	3
…	……
11101	30
11110	31
11111	Setting is prohibited

A set value is limited. See "5.1.4 Limitations when PLL/SSCG Clock is used" when you set it.

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.

This register sets various settings of SSCG.

It can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0")

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

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

[bit7 to bit5] (Reserved)

[bit4] (Reserved)

Writing has no effect on operation.

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

These bits set the spread spectrum modulation frequency of SSCG.

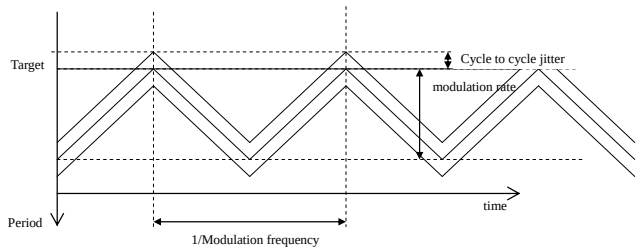
SFREQ[1:0]	Modulation frequency
00	1/1024
01	1/2048
1x	1/4096

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

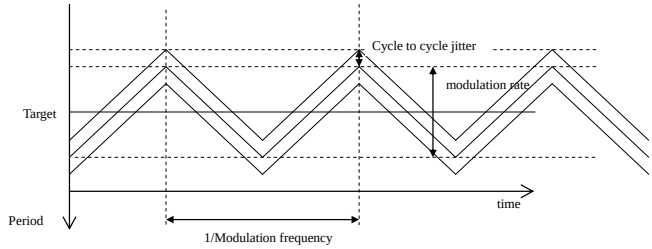
This bit sets the spread spectrum modulation mode of SSCG.

SMODE	Modulation mode
0	Down Spread
1	Center Spread

- **Down Spread**



- **Center Spread**



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

This bit enables spread spectrum of SSCG.

SSEN	Spread spectrum enable
0	Spread spectrum disabled
1	Spread spectrum enabled

Note:

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

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.

This register sets various settings of SSCG.

It 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) : Spread spectrum modulation rate selection

These bits set the spread spectrum modulation rate of SSCG.

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

[bit12 to bit10] (Reserved)

Writing to these bits has no effect.

[bit9 to bit0] (Reserved)

Always write "0" to these bits.

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.

This register sets various settings of clock gear.

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

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

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

These bits display the status of Clock gear.

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

[bit5 to bit2] (Reserved)

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

Writing "1" to this bit starts the operation of clock gear.

The operation of clock gear depends on the value of the GRSTS bits. (Gear up or gear down)

When GRSTS=00

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

When GRSTS=01/11

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

When GRSTS=10

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

Notes:

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

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

This bit enables the operation of clock gear.

GREN	Operation
0	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").

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.

This register sets various settings of clock gear.

It 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) : 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) : Clock gear start step number selection

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

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

Note:

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

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

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

This register sets various settings of clock gear.

It 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) : 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

4.21. RTC/PMU Clock Selection Register : CCRTSELR (CCtl RTc pmu clock SELection Register)

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

This register selects the RTC/PMU clock source.

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

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

*: 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): Clock source selection status monitor

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

CST	Monitor
0	Completion of clock switch
1	During clock switch

Note:

Normally, switch completes by main clock × about 3 cycles + sub clock × about 3 cycles.

[bit6 to bit1] (Reserved)

[bit0] CSC (Clock SourCe selection) : Clock source selection

Selects clock of RTC/PMU.

CSC	Clock source
0	Main oscillation clock
1	Sub oscillation clock

Notes:

- The CSC register can be rewritten only when SCRDY=1 and MCRDY=1.
- 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 register. 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 register is "0" (the status of the completion of switching).

- [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
- The CSC bit and the CST bit are 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 and the CST bit are not initialized. Initialize these bits 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.
- [MB91F52xxxD]
- The CSC bit and the CST bit are 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 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 and the CST bit are not initialized. Initialize these bits 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.

4.22. PMU Clock Division Setting Register 0 : CCPMUCR0 (CCtl PMU Clock division Register 0)

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

This register sets the clock dividing frequency of the PMU.

■ 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): 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, bit0] FDIV[1:0] (F-DIVide ratio setting): F-divide ratio setting

These bits set the division rate of F-divider. The clock equal to or less than 32 kHz must be provided with PMU. When CCRTSELR.CSC=0 (selection of main clock), this bit is set to be equal to or less than 32 kHz 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), the F-division rate becomes undivided in spite of the value of this bit.

4.23. PMU Clock Division Setting Register 1 : CCPMUCR1 (CCtl PMU Clock division Register 1)

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

This register sets the clock dividing frequency of the PMU.

■ 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): 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) : G-divide ratio setting

These bits set the division rate of G-divider. The period of the PMU clock must be equal to or greater 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	No divide (Initial value)
00001	2
00010	3
…	……
11101	30

GDIV[4:0]	Division rate
11110	31
11111	32

Note:

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

4.24. Sync/Async Control Register : SACR

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

This register selects the peripheral clock.

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

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

[bit7 to bit1] (Reserved)

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

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

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

4.25. Peripheral Interface Clock Divider : PICD

The bit configuration of peripheral interface clock divider is shown.

This register sets the dividing frequency of the peripheral clock.

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

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

[bit7 to bit4] (Reserved)

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

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

PDIV[3:0]	PLL clock (PLLCLK) [non spread spectrum clock] → PCLK2 division ratio
0000	No divide
0001	2 division
0010	3 division
0011	4 division (initial value)
0100	5 division
0101	6 division
0110	7 division
0111	8 division
1000	9 division
1001	10 division
1010	11 division
1011	12 division
1100	13 division
1101	14 division
1110	15 division
1111	16 division

Note:

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

5. Operation

This section explains operations of clock.

5.1. Oscillation Control

5.2. Oscillation Stabilization Wait

5.3. Selecting the Source Clock (SRCCLK)

5.4. Timer

5.5. Notes when Clocks Conflict

5.6. The Clock Gear Circuit

5.7. Operations during MDI Communications

5.8. About PMU clock (PMUCLK)

5.1. Oscillation Control

This section explains oscillation control.

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) is selected as the source clock 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.

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 and "0" is set to CSELR.SCEN bit.
- When the clock is used as a port because the clock is used for sub oscillation and port (without sub oscillation).

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.

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 5-1 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 and "0" is set to CSELR.PCEN.

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

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

(PLL/SSCG setting in Microcontroller unit)

- PLL/SSCG input clock frequency = (main oscillation frequency) / (PLLCR.PDS[3:0] division ratio)
- PLL multiplication rate = (CCPLLFBR.IDIV[6:0] FB input division ratio)
  SSCG multiplication rate = (CCSSFBR0.NDIV[5:0] FB input division ratio) × (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 5-1 PLL Peripheral Block Diagram

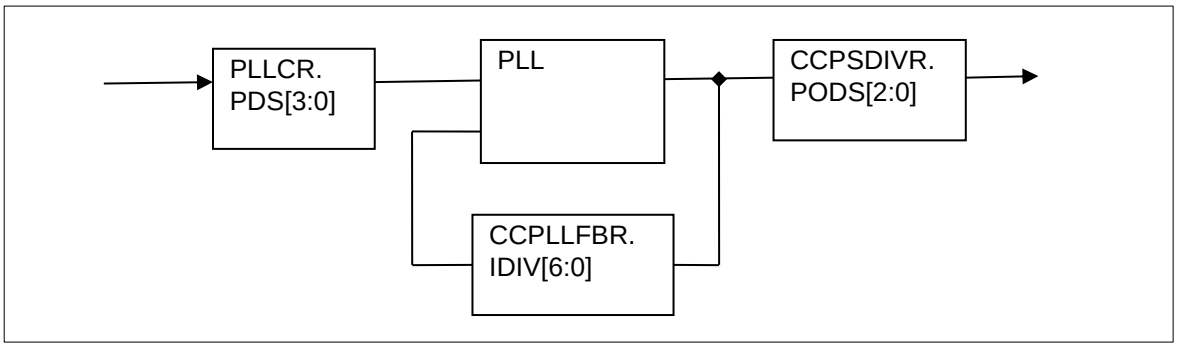
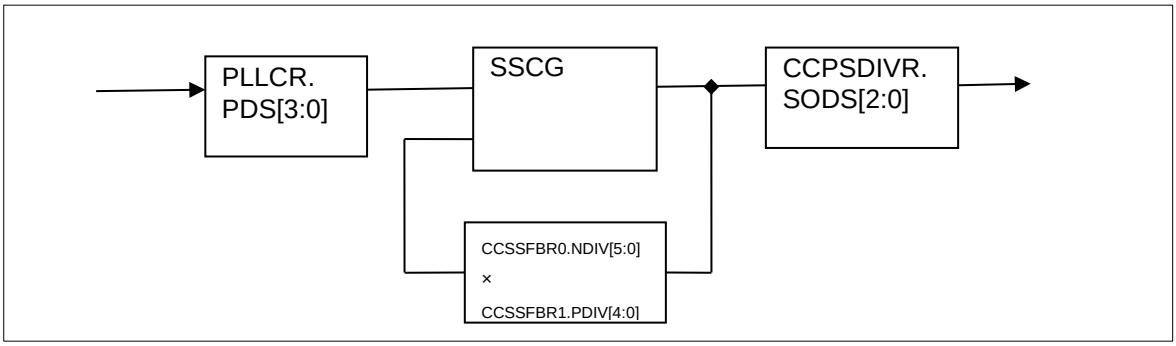


Figure 5-2 SSCG Peripheral Block Diagram



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

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

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 in Microcontroller unit :
  - 200 MHz ≤ PLL macro oscillation clock frequency ≤ 320 MHz
  - 200 MHz ≤ SSCG macro oscillation clock frequency ≤ 320 MHz (Down Speed)

5.1.4. Limitations when PLL/SSCG Clock is used

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

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

Clock Control PLL Clock Frequency

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

Note:

Set PLLCR or CCPSDIVR and CCPLLFBR so as not to exceed frequency (max). The frequency (max) is 80 MHz.

Microcontroller Unit Clock Control SSCG Clock Frequency

Frequency (max)	FCTLR: FAW	CCPSSELR: PCSEL	CCSSCCR0: SSEN	CCSSCCR0: SMODE	CCSSCCR1: RATESEL	Remarks
72 MHz	00	1	1	0	000 to 110	Down Spread
72 MHz	00	1	1	1	000	Center Spread (0.5%)
72 MHz	00	1	1	1	010	Center Spread (1%)
72 MHz	00	1	1	1	011	Center Spread (2%)
71 MHz	00	1	1	1	100	Center Spread (3%)
71 MHz	00	1	1	1	101	Center Spread (4%)
70 MHz	00	1	1	1	110	Center Spread (5%)
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). The frequency (max) is 80 MHz.

Relation between Modulation Rate and Division Ratio when SSCG is Used

CCSSCCR1:RATESEL[2:0]		CCSSFBR0:NDIV[5:0]		
Modulation rate	Set value	Range of division ratio	Set value lower limit	Set value upper limit
0.50%	00x	8 - 60	7_H	$3B_H$
1.00%	010	8 - 60	7_H	$3B_H$
2.00%	011	8 - 48	7_H	$2F_H$
3.00%	100	8 - 31	7_H	$1E_H$
4.00%	101	8 - 23	7_H	16_H
5.00%	110	8 - 18	7_H	11_H

5.2. Oscillation Stabilization Wait

Oscillation stabilization wait is shown.

This section describes oscillation stabilization wait for each clock input.

5.2.1. Conditions for Generating Stabilization Wait Time

Conditions for generating stabilization wait time are shown.

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

The main (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.

5.2.2. 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 2^{16}× main clock period
- Sub clock : CSTBR.SOSW[2:0] bit 2^{8}× sub clock period

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

The PLL/SSCG clock lock wait time is always specified by the initial value because PLLCR.POSW[3:0] is initialized by reset (INIT or RST). Except that case, the PLL/SSCG clock lock wait time can be changed by setting to PLLCR.POSW[3:0]. Set "1" to CSELR.PCEN after setting to PLLCR.POSW[3:0]. For details, see the explanation of POSW in "4.8 PLL Setting Register : PLLCR (PLL Configuration Register)".

The sub 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].

5.2.3. End of the Stabilization Wait Time

The end of the stabilization wait time is shown.

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

	Display of the clock oscillation stabilization wait status	Display of 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"

5.3. 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.

5.3.1. Selecting the Source Clock at the Time of Initialization

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

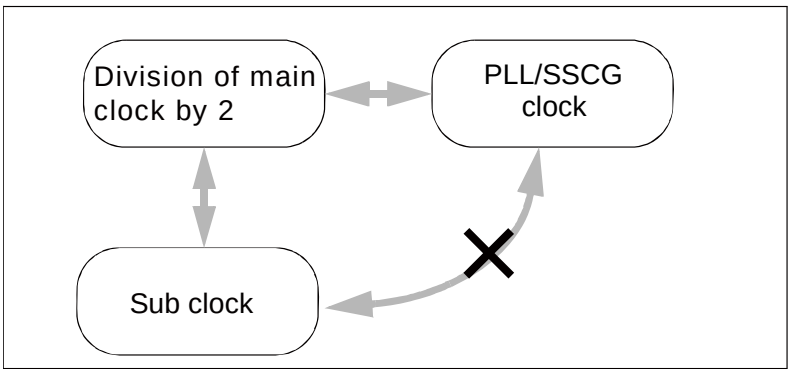
After reset (RST) the main clock (MCLK) divided by 2 is selected as the source clock. After program operation the source clock can be changed by setting CSELR.CKS[1:0].

5.3.2. 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 enable bit (CSELR.xCEN) is held, even though the source clock is switched.

Figure 5-3 Procedure of Switching the Source Clock



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

   ↓

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

↓

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

↓

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

↓

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

↓

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

↓

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

↓

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 to 10)

↓

The clock gear begins (CCCGRCR0.GRSTR=1)

↓

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

↓

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

2. PLL/SSCG clock→the main clock divided by 2
   While selecting PLL/SSCG clock as the source clock (CMONR.CKM[1:0]=10)

↓

Clock gear begins (CCCGRCR0.GRSTR=1)

↓

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

↓

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

↓

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

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)

   ↓

   In a single clock product, selecting the CR clock as a sub-clock source

(CSVCR.SCKS=1)

   ↓

   The sub clock oscillation begins (SCEN=0 to 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 to 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 clock oscillation begins (MCEN=0 to 1)

↓

The main clock oscillation stabilization wait loop (loop until 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 to 01)

↓

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

↓

In a single clock product, returning the sub clock source setting

The sub clock oscillation stops (CSELR:SCEN=1 to 0)

↓

The CR clock is not selected as a sub-clock (CSVCR:SCKS=1 to 0)

Figure 5-4 Example of PLL/SSCG Mode Setting Main → PLL/SSCG

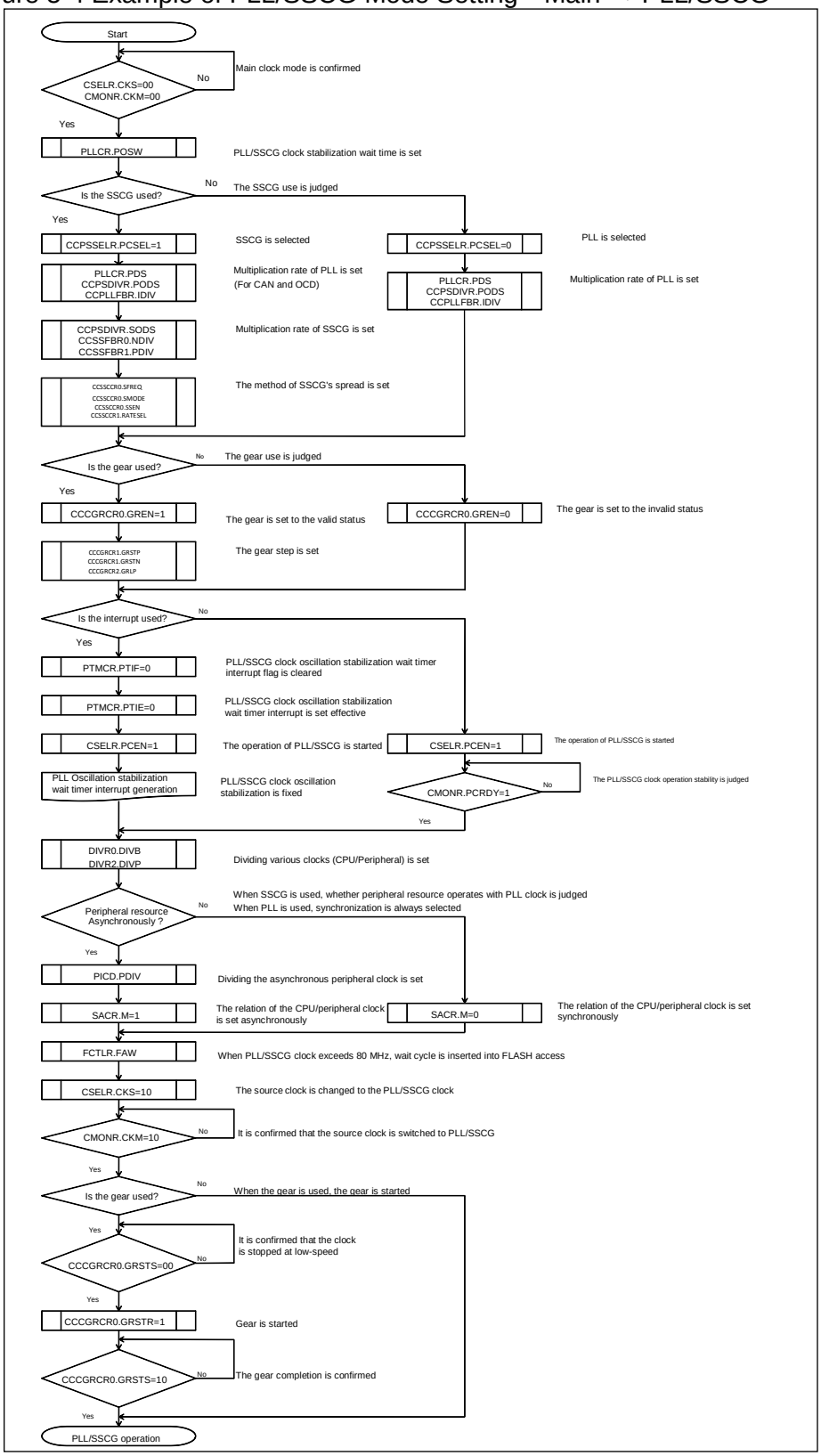
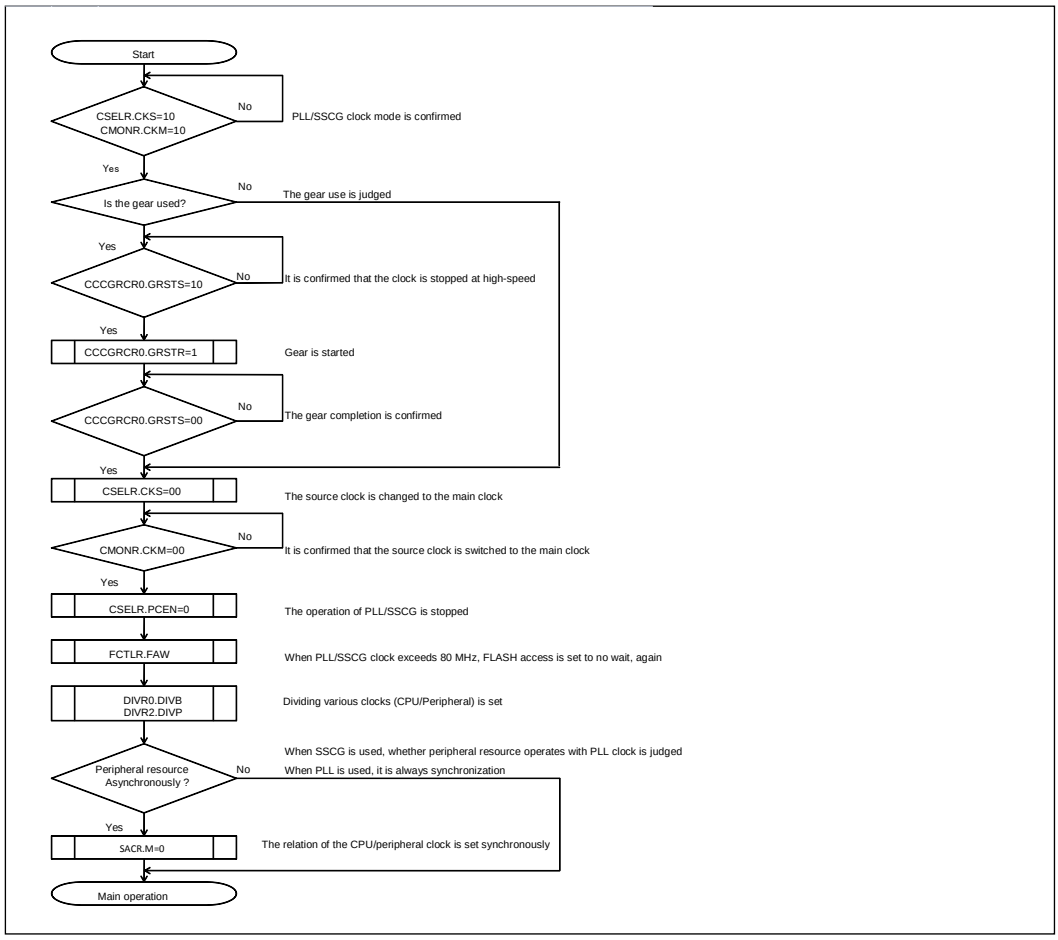


Figure 5-5 Example of PLL/SSCG Mode Setting PLL/SSCG → Main



5.4. Timer

The timer is shown.

5.4.1. Main Clock Oscillation Stabilization Wait Timer (Main Timer)

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

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

Note:

If main timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.

5.4.2. Sub Clock Oscillation Stabilization Wait Timer (Sub Timer)

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

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

Note:

If sub timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown

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

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

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 to equal to or greater than PCLK X 5 clock. When the period of the timer interrupt is set to the extremely short time, the interrupt factor may not be set.

5.4.5. 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

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

5.4.6. Timer Operations

Timer operations are shown.

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

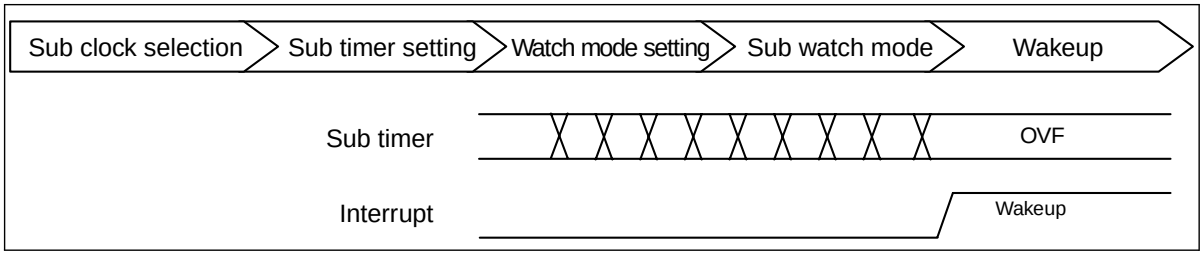
While STMCR.STE=1, the sub timer counts up by the sub clock (SBCLK). If the timer overflows by the period which is selected by STMCR.STS[2:0], STMCR.STIF is "1".

5.4.7. 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 5-6 Wake-up from the Watch Mode



Note:

If main/sub timer or real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.

5.5. Notes when Clocks Conflict

Notes when clocks conflict is shown.

Note 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 in duplicate.

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.

5.6.1. Procedure of Gear Up

The procedure of gear up is shown.

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

5.6.2. Procedure of Gear Down

The procedure of gear down is shown.

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

5.7. Operations during MDI Communications

Operations during MDI communications are shown.

The main oscillation is controlled so as not to be stopped during MDI communications even if the stop mode is transited to.
Moreover, during MDI high speed communication, the main oscillation is controlled so that the PLL reference clock is supplied even if CSELR.PCEN is cleared. The value of the register related to PLL is maintained and not updated.
However, when software sets PLLCR.PCEN=0, the value of the register related to PLL can be freely updated (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.
(Note): The registers related to PLL are as follows.

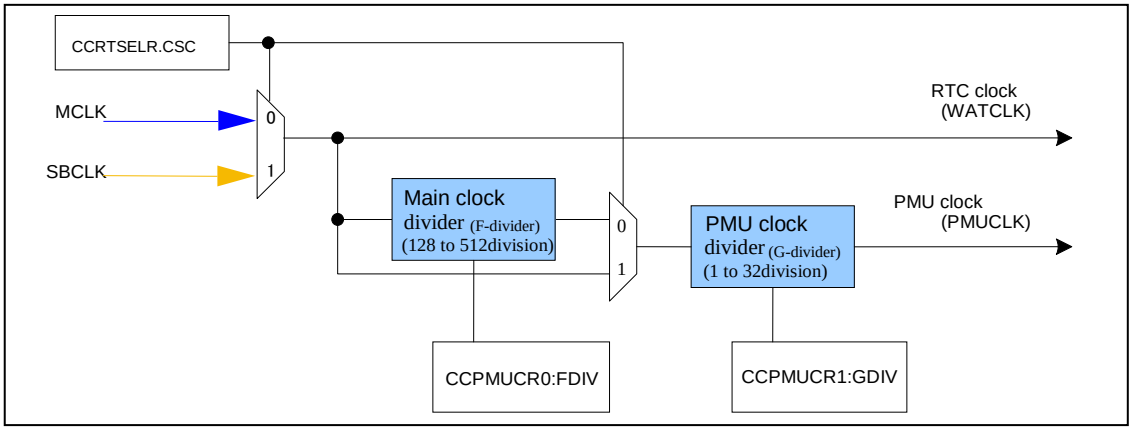
- CCPSDIVR.PODS
- CCPLLFBR.IDIV
- PLLCR.PDS

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 5-7 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) CCRTSELR:CSC must be set to select active clock that is under oscillation.
(2) F-divider must be set so that the PMU clock frequency become 32 kHz or less.
(3) G-divider must be set so that PMU clock frequency become 1/4 or less of the peripheral clock frequency (PCLK1).

The following explains each specification limitation.

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

(2) Set F-divider so that the PMU clock frequency becomes 32 kHz or less.
The PMU clock is used to control the power switch. Specify a frequency of 32 kHz or less for reasons of stabilization at times of rising pressure such as 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. When CCRTSELR:CSC=1, the F-divider does not affect the operation.

FDIV[1:0]	Division rate	Target main oscillation frequency
00	128 division(initial value)	4 MHz
01	256 division	8 MHz
10	384 division	12 MHz
11	512 division	16 MHz

(3) Set G-divider so that PMU clock frequency becomes 1/4 or less of the peripheral clock frequency (PCLK1).
Clock transfer between peripheral clock (PCLK1) and PMU clock (PMUCLK) needs 4 PMU clock cycles.

When the source clock of peripheral clock (PCLK1) is sub oscillation clock (CMONR.CKM="10"), set the CCPMUCR1.GDIV register so that the frequency of peripheral clock (PCLK1) is quadruple or higher the frequency of PMU clock.

Also, even when the source clock of the peripheral clock (PCLK1) is the main clock divided by 2 (CMONR.CKM="00" or CMONR.CKM="01"), and when the peripheral clock (PCLK1) is equal to or less than 128 kHz (32 kHz × 4) in the setting of DIVR0.DIVB and DIVR2.DIVP, CCPMUCR1.GDIV register should be set similarly.

GDIV[4:0]	Division ratio
00000	No 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 selected by CMONR.CKM) / (DIVR0.DIVB[2:0] division ratio) /(DIVR2.DIVP[3:0] division ratio)

Chapter 6: Clock Reset State Transitions

This chapter explains clock reset state transitions.

1. Overview
2. Device States and Transitions
3. Device State and Regulator Mode Corresponding to those States

Code : CRST-1v0-91520-2-E

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".

2. Device States and Transitions

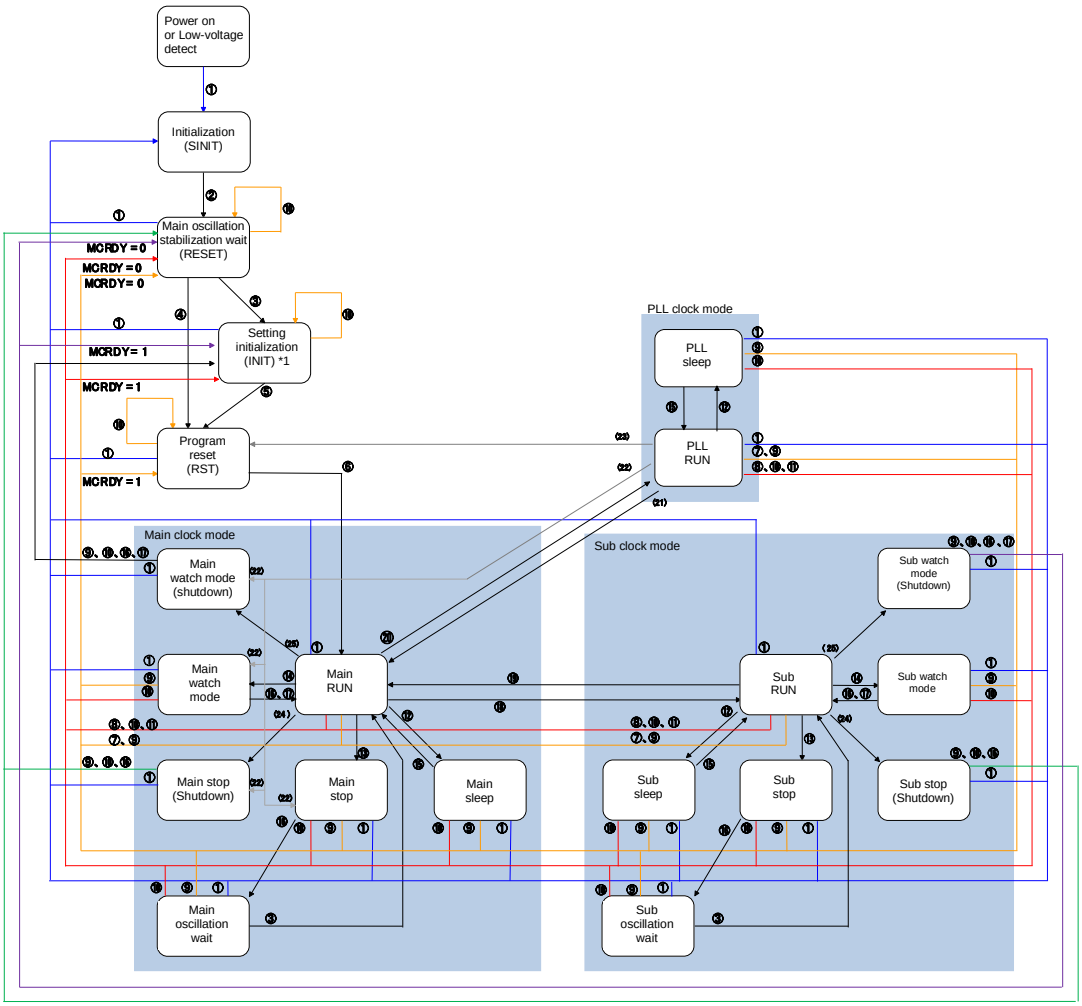
This section explains device states and transitions of clock reset state transitions.

2.1. Diagram of State Transitions

This section shows diagram of state transitions.

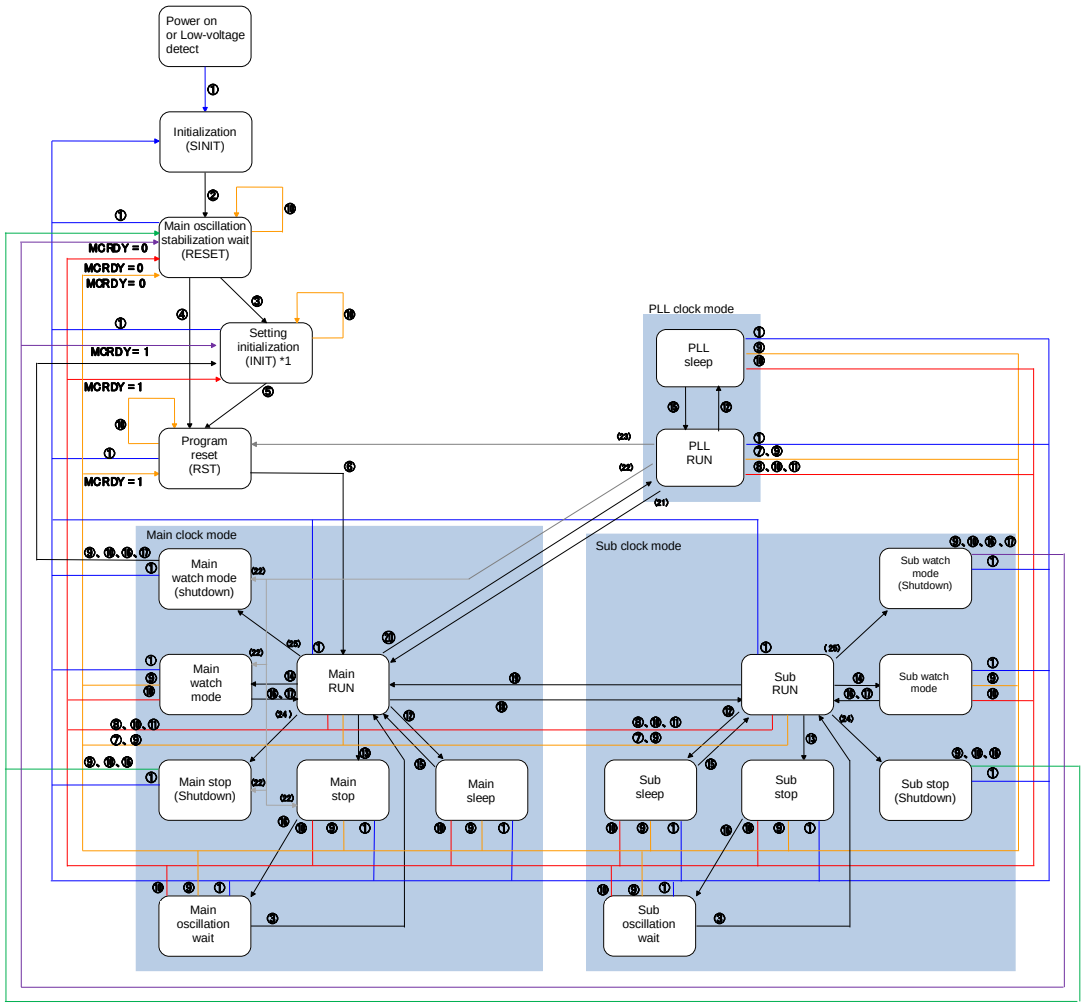
The device state transitions for this series are shown below.

Figure 2-1 Diagram of Device State Transitions [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]



① Power-on reset or internal low-voltage detection or external reset and simultaneous assertion of NMI
② Power-on reset release and internal low-voltage release and external reset and release simultaneous assertion 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)

Figure 2-2 Diagram of Device State Transitions [MB91F52xxxD]



① Power-on reset or internal low-voltage detection or external reset
② Power-on reset release and internal low-voltage release and external reset release
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 low-voltage detection
⑩ 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)

*1 : There is a register not reset when returning from the watch mode (Shutdown) and returning from the stop mode (Shutdown). See "Restrictions on Power-Shutdown and Normal Standby Control" in "CHAPTER : POWER CONSUMPTION CONTROL" for detail.

Note:

The transition may be different from above diagram when connecting to OCD tool. See "CHAPTER: ON-CHIP DEBUGGER (OCD)" for details.

2.2. Explanation of Each States

This section explains each state.

Device operation states for this series are shown below.

■ RUN State (Normal Operation)

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

■ Sleep Mode

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

■ Watch Mode State

The devices are not running. The state transits by program operations. Internal circuits other than oscillation circuits (main clock generation unit, 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 or the external 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".

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)

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 3-1 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	Main mode
Main watch mode (Shutdown)	Oscillation	Standby mode
Main stop	Stop	Main mode
Main stop (Shutdown)	Stop	Standby mode
Main Oscillation wait	Oscillation	Main mode
PLL RUN	Oscillation	Main mode
PLL sleep	Oscillation	Main mode

Table 3-2 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	Main mode
Main watch mode(Shutdown)	Oscillation	Oscillation or Stop	Standby mode
Main stop	Stop	Stop	Main 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	Main mode
Sub sleep 1	Oscillation	Oscillation	Main mode
Sub sleep 2	Stop	Oscillation	Main mode
Sub watch mode	Oscillation or Stop	Oscillation	Main mode
Sub watch mode (Shutdown)	Oscillation or Stop	Oscillation	Standby mode
Sub stop	Stop	Stop	Main mode
Sub stop (Shutdown)	Stop	Stop	Standby mode
Sub Oscillation wait 1	Oscillation	Oscillation	Main mode
Sub Oscillation wait 2	Stop	Oscillation	Main 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.

Chapter 7: Reset

This chapter explains the reset.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : RST-1v1-91520-3-E

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.

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 "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.

3. Configuration

This section explains the configuration of the reset.

Figure 3-1 Configuration Diagram of Reset [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]

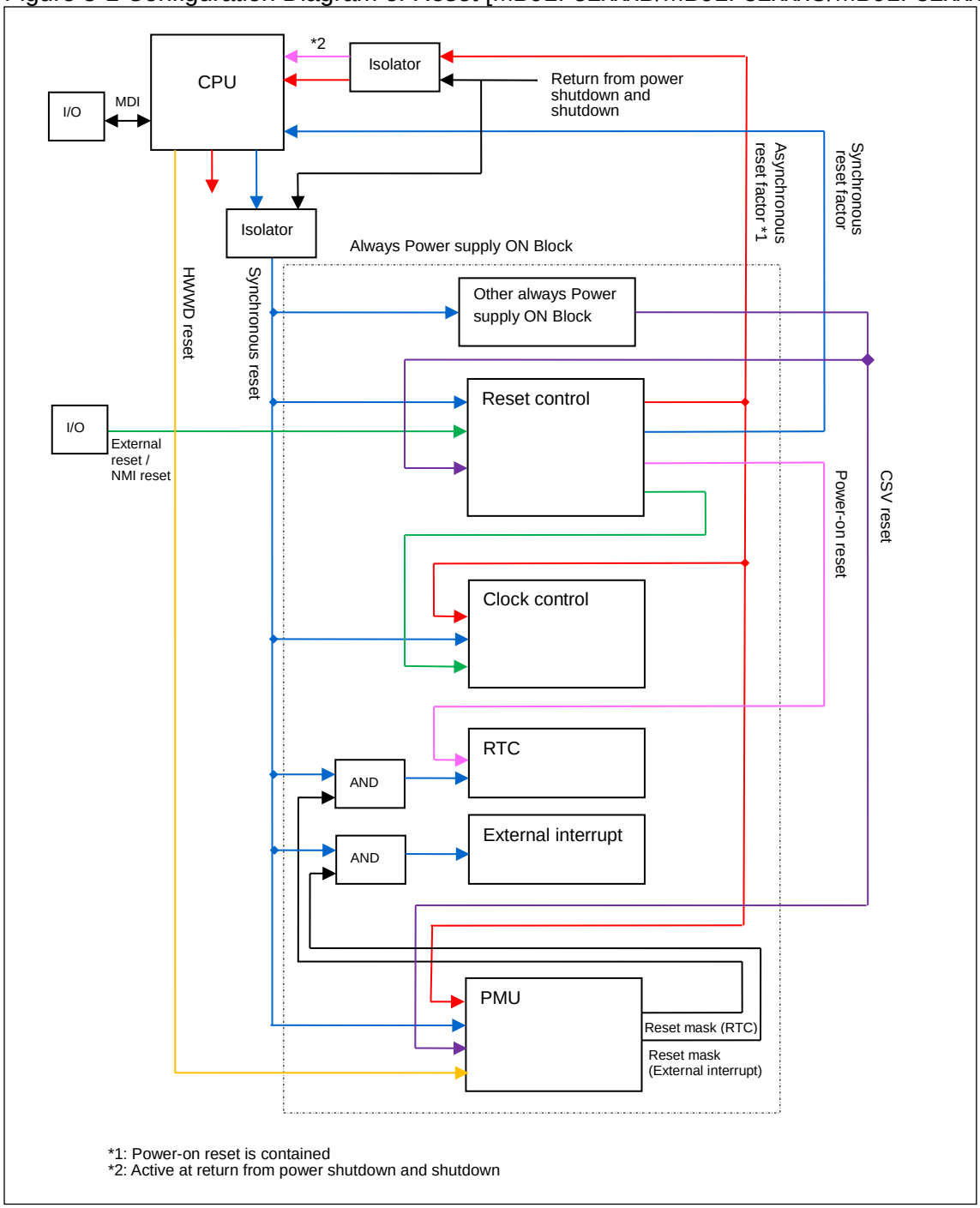


Figure 3-2 Configuration Diagram of Reset [MB91F52xxxD]

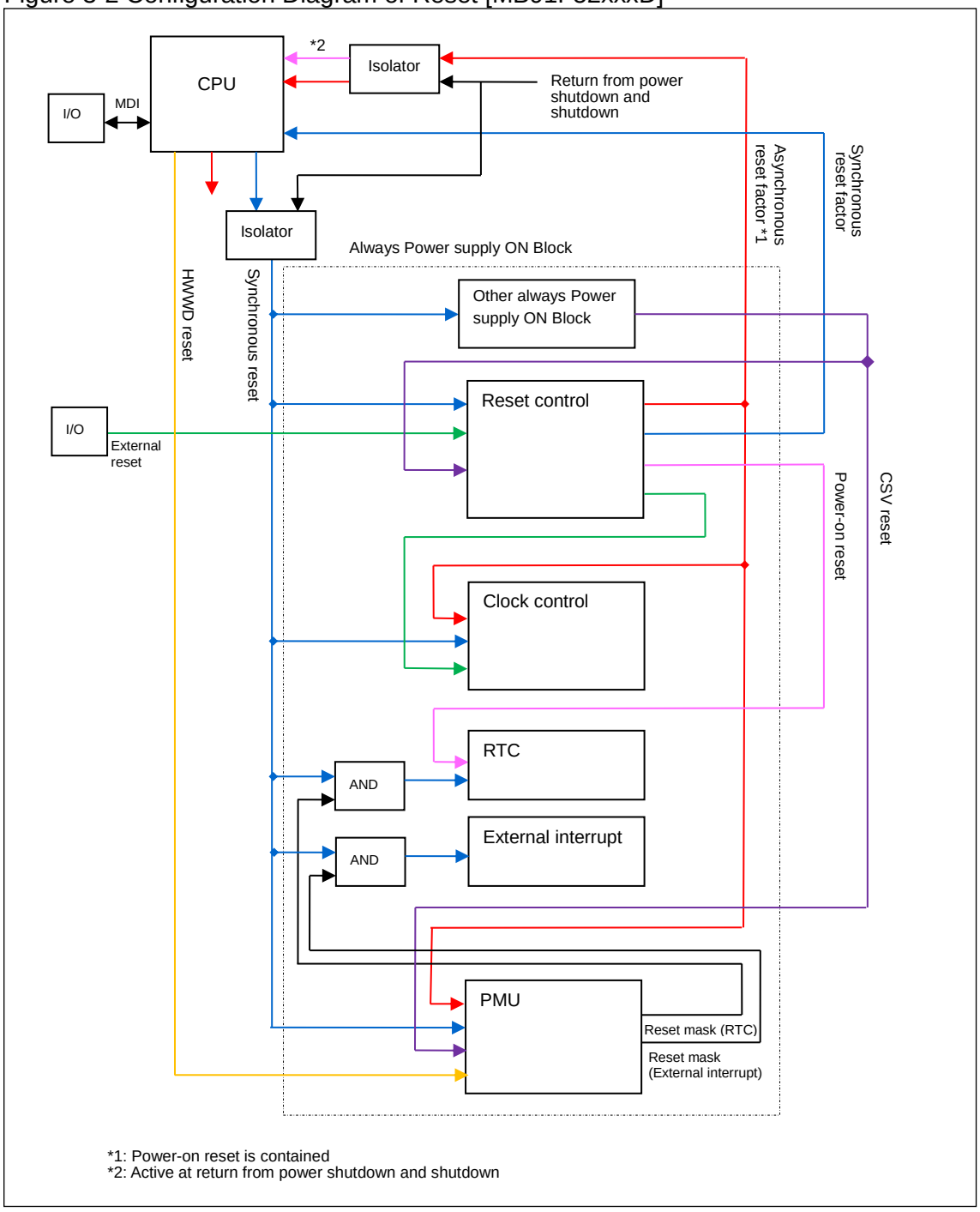


Figure 3-3 Configuration Diagram of Reset (Reset Control)
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]

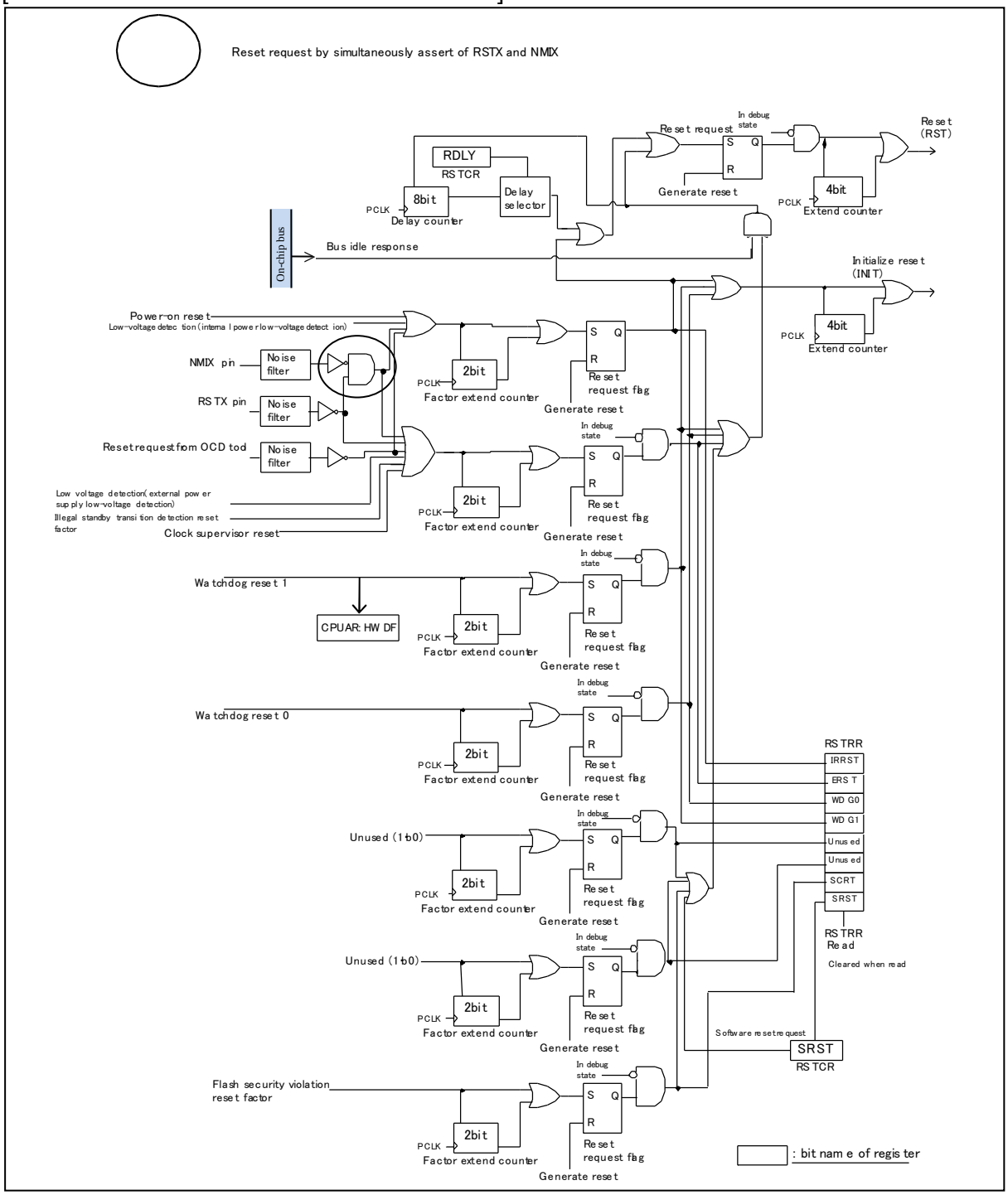


Figure 3-4 Configuration Diagram of Reset (Reset Control) [MB91F52xxxD]

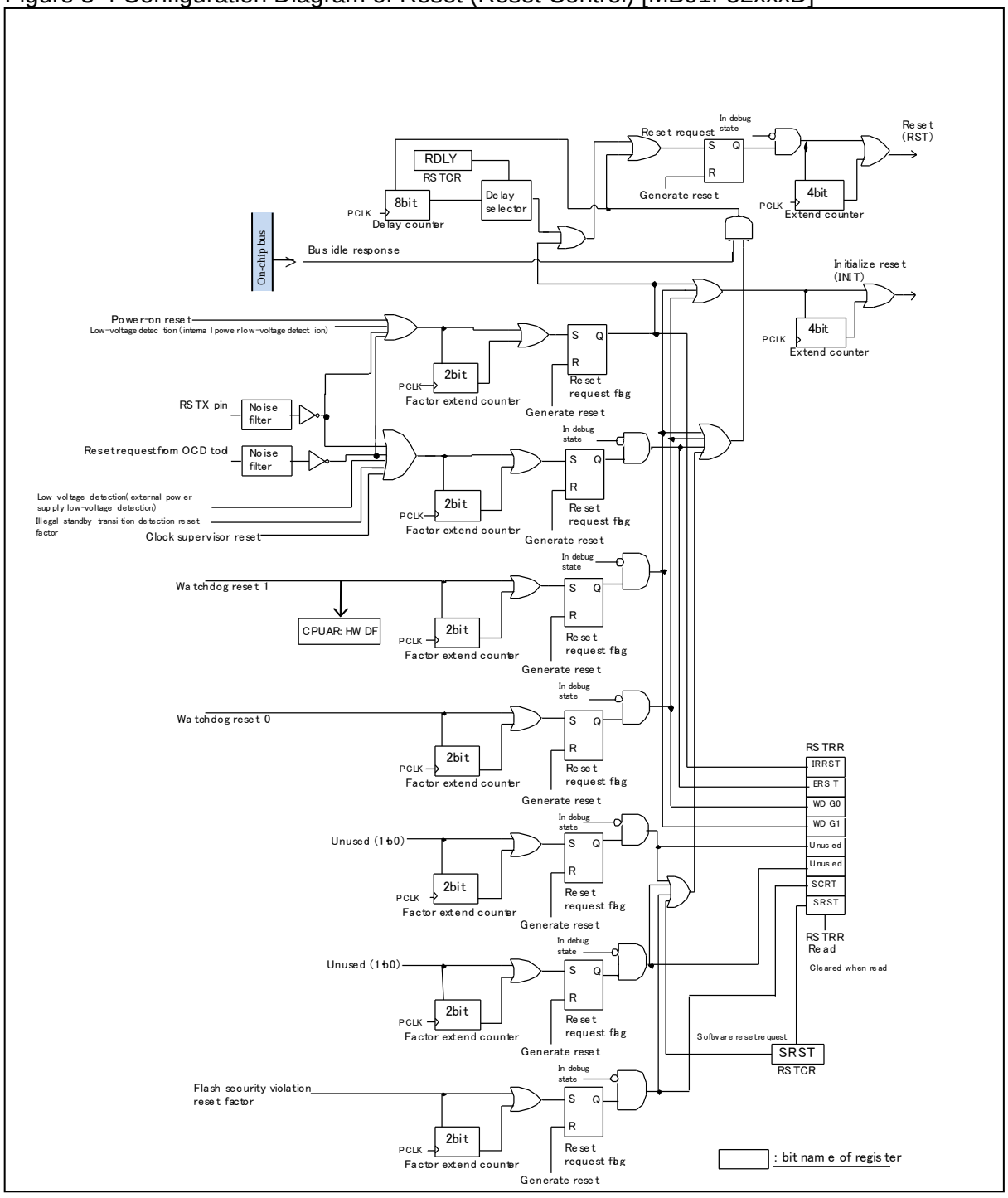
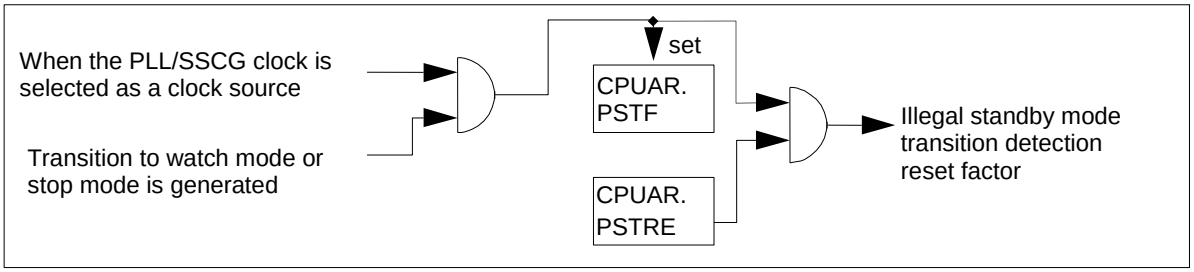


Figure 3-5 Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor



4. Registers

This section explains the registers of the reset.

Table 4-1 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 "CHANPTER : POWER CONSUMPTION CONTROL" is allocated in address 0x0482, 0x0591, and 0x0592.

4.1. Reset Source Register : RSTRR (ReSeT Result Register)

The bit configuration of the reset source register is shown.

This register displays various reset factors generated until just before.

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

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

*: Due to a reset factor.
*: These bits other than IRRST bit are undefined at power-on reset.

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.

[bit7] IRRST (IRregular ReSeT) : Irregular reset

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit indicates that any of power-on reset, internal low-voltage detection, reset timeout, or simultaneous assertion 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.

[MB91F52xxxD] This bit indicates that any of power-on reset, internal low-voltage detection, reset timeout, or assertion of RSTX 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) :

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Reset pin input, illegal standby mode transition detection, external low-voltage detection, clock supervisor reset, simultaneous assertion 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 assertion of RSTX and NMIX external pins. In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

ERST	RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset, low-voltage detection (external low-voltage detection) or simultaneous assertion of RSTX and NMIX external pins
0	Undetected
1	Detected

This bit will be cleared when it is read out.

[MB91F52xxxD] Reset pin input, illegal standby mode transition detection, external low-voltage detection, clock supervisor reset.

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.

In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

ERST	RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset, low-voltage detection (external low-voltage detection)
0	Undetected
1	Detected

This bit will be cleared when it is read out.

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

This bit indicates a reset from the watchdog timer 1.

In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

WDG1	Watchdog timer 1 reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

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

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

This bit indicates a reset from the watchdog timer 0.

In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

WDG0	Watchdog timer 0 reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

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

This bit indicates that a flash memory security violation reset has occurred.

In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

SCRT	Flash security violation reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

[bit0] SRST (Software ReSeT) : Software reset

This bit indicates a reset by writing "1" to the RSTCR:SRST bit.

In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

SRST	Software reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

4.2. Reset Control Register : RSTCR (ReSeT Control Register)

The bit configuration of the reset control register is shown.

This register controls various types of reset issuance.

■ RSTCR : Address 0481_H (Access : Byte, Half-word, Word)

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

[bit7 to bit5] RDLY[2:0] (Reset DeLaY) : Reset issue delay

These bits set the reset timeout value. A reset will be issued if all bus operations become idle or the timer has counted to the reset timeout by this bit after a reset factor has been detected. (The latter is a case of irregular reset). These bits can be written for only once after the reset.

RDLY[2:0]	Reset timeout value
000	PCLK × 2 cycles
001	PCLK × 4 cycles
010	PCLK × 8 cycles
011	PCLK × 16 cycles

RDLY[2:0]	Reset timeout value
100	PCLK × 32 cycles
101	PCLK × 64 cycles
110	PCLK × 128 cycles
111	PCLK × 256 cycles (Initial value)

[bit4 to bit1] Reserved

This has no effect on both writing and reading.

[bit0] SRST (Software ReSeT) : Software reset

You will be able to generate a software reset request by reading RSTCR after writing "1" to this bit.

After you have written "1" to this bit, any values written to RSTCR will be ignored until a reset is generated, which means that register values cannot be changed.

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

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

4.3. CPU Abnormal Operation Register : CPUAR (CPU Abnormal operation Register)

The bit configuration of the CPU abnormal operation register is shown.

This register indicates the status and settings associated with the abnormal operation of CPU.

■ CPUAR : Address $051A_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PSTRE	Reserved				PMDF	PSTF	HWDF
Initial value	0	0	0	0	*	*	*	*
Attribute	R/W	R0,WX	R0,WX	R0,WX	RX,WX	R(RM1),W	R(RM1),W	R(RM1), W

* : It will be initialized to "0" by RSTX pin asserts (including simultaneous assert with NMIX). 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

[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.

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	*
Attribute	R,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	R,W

* : 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)

Displays information on whether the immediately preceding status was 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". Writing "1" to this bit is ignored.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit is initialized only by power-on reset, internal low-voltage detection reset, and simultaneous assertion of RSTX and NMIX. So, check other reset factors before checking the recovery from shutdown using this bit.

[MB91F52xxxD] This bit is initialized only by power-on reset, internal low-voltage detection reset, and assertion of RSTX. So, check other reset factors before checking the recovery from shutdown using this bit.

[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 or internal low-voltage detection reset detection flag.

PONR_F	Power-on reset
0	No detection
1	Detection

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

5. Operation

This section explains the reset operation.

This section explains each of the reset operations for this product.

5.1. Reset Level

5.2. Reset Factor

5.3. Reset Acceptance

5.4. Reset Issue

5.5. Reset Sequence

5.6. Notes

5.1. Reset Level

The reset level is explained.

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

- Initialize reset (INIT)
- Reset (RST)

Note:

Except the registers for debug interface (OCDU), the registers initialized by the reset of both levels are the same for this product.

5.1.1. Initialize Reset (INIT)

Initialize reset (INIT) is explained.

It initializes the CPU and all registers except the ones initialized only by the power-on reset or super initialize reset (SINIT) and registers with undefined initial value. 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)

5.1.2. Reset (RST)

The reset (RST) is explained.

It initializes the CPU and all registers except the ones initialized only by the power-on reset, SINIT or INIT and registers with undefined initial value. 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.

5.2. Reset Factor

This section explains the reset factor.

This section explains each of the reset factors for this product.

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).

5.2.2. RSTX Pin Input

The RSTX pin input is shown.

It is a hardware reset input from the outside of the device.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Reset by this reset factor is detected as irregular reset only at the reset timeout or simultaneous assertion of the NMIX pin.
Other than the irregular reset detection, a reset (RST) will be issued.

[MB91F52xxxD] Reset by this reset factor is detected as irregular reset.

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.

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. Whether or not an irregular reset has been detected, an initialize reset (INIT) will be issued.

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.

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

5.2.7. Internal Low-Voltage Detection Reset

The internal low-voltage detection reset is shown.

Low-voltage detection (internal voltage) is a hardware reset input from the low-voltage detection circuit located inside of the device.

Resets due to this reset factor will be detected as an irregular reset and an initialize reset (INIT) will be issued.

See "CHAPTER : LOW-VOLTAGE DETECTION (INTERNAL LOW-VOLTAGE DETECTION)" for details on voltage detection.

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.

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
```

5.2.10. Recovery from Standby (Power Shutdown)

Recovery from standby (power shutdown) is shown.

For a majority of the block including the microcontroller, 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.

Resets due to this reset factor will issue an initialization reset (INIT).

5.3. Reset Acceptance

This section explains the reset acceptance.

This section explains the acceptance processing of each reset factor.

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

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.

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.

5.3.4. Irregular Reset

The irregular reset is shown.

If a reset is issued without confirming the completion of reset request processing, an irregular request will be generated. Once an irregular reset is generated, the following processing will be executed.

- Issue initialize reset (INIT) regardless of the type of reset factor.
- Set the bit7: IRRST bit of RSTRR register to "1".

When an irregular reset occurs, there is no guarantee that memory contents were not destroyed by the reset since a bus access may have been executed at the time of inputting the reset. The irregular reset does not necessarily mean that the memory contents were destroyed, but how the bus access was executed cannot be identified.

5.4. Reset Issue

This section explains reset issue.

A reset will be issued after a reset request has been accepted. This section explains each type of reset issue.

5.4.1. Super Initialize Reset (SINIT)

The super initialize reset (SINIT) is shown.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] The super initialize reset (SINIT) will be issued first for power-on reset, internal low-voltage detection, or simultaneous assertion of RSTX and NMIX.

[MB91F52xxxD] The super initialize reset (SINIT) will be issued first for power-on reset, internal low-voltage detection, or assert of RSTX.

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 $2^{15}\times$ main clock cycle.

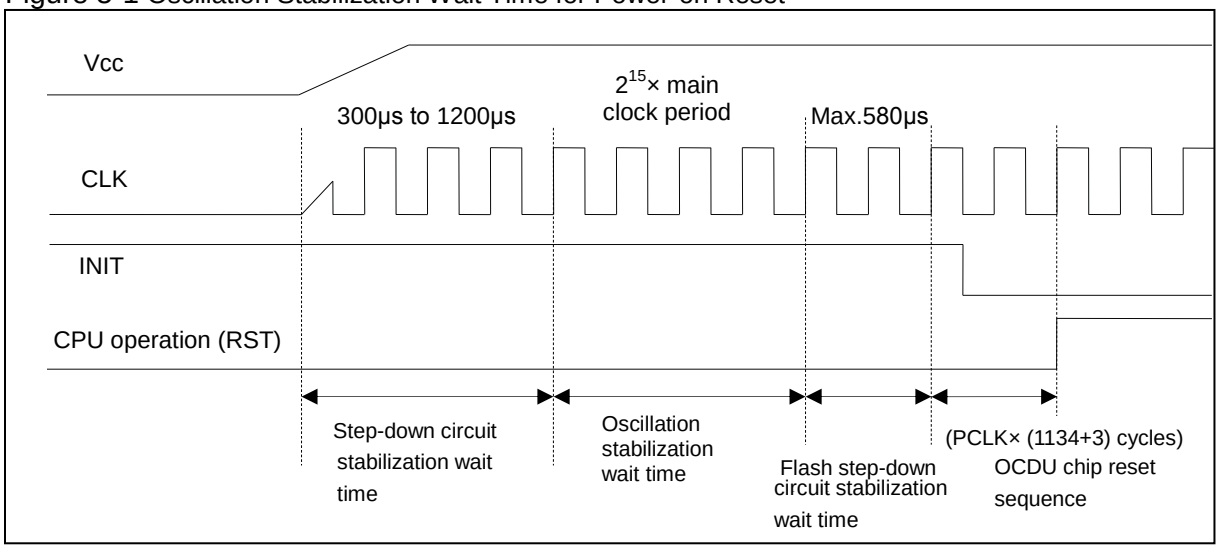
Table 5-1 Oscillation Stabilization Wait Time (SINIT)

Type	Main clock oscillation stabilization wait time
Power-on reset	$2^{15}\times$ Main clock cycle
Internal low-voltage detection	$2^{15}\times$ Main clock cycle
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Simultaneous assertion of RSTX and NMIX	$2^{15}\times$ Main clock cycle
[MB91F52xxxD] Assertion of RSTX	$2^{15}\times$ Main clock cycle

Note:

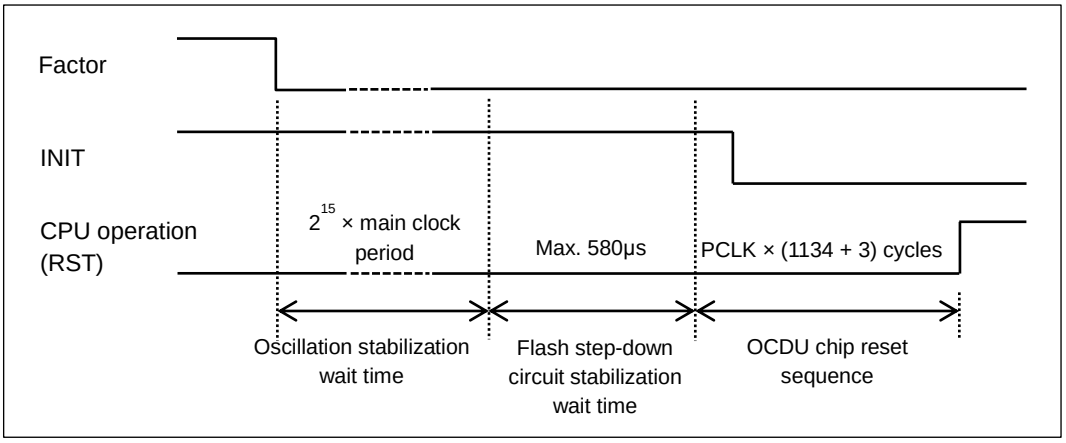
The oscillation stabilization wait time shown in the above table does not include the regulator stabilization wait time associated with the power-on and voltage restore. These stabilization wait time (300µs to 1200µs and maximum 580µs) are needed at power-on reset.

Figure 5-1 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 5-2 Super Initialize Reset (SINIT) Sequence



Because the clock settings register is initialized by reset, the period of the peripheral clock (PCLK) is 8 times the period of the main clock (MCLK).

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.

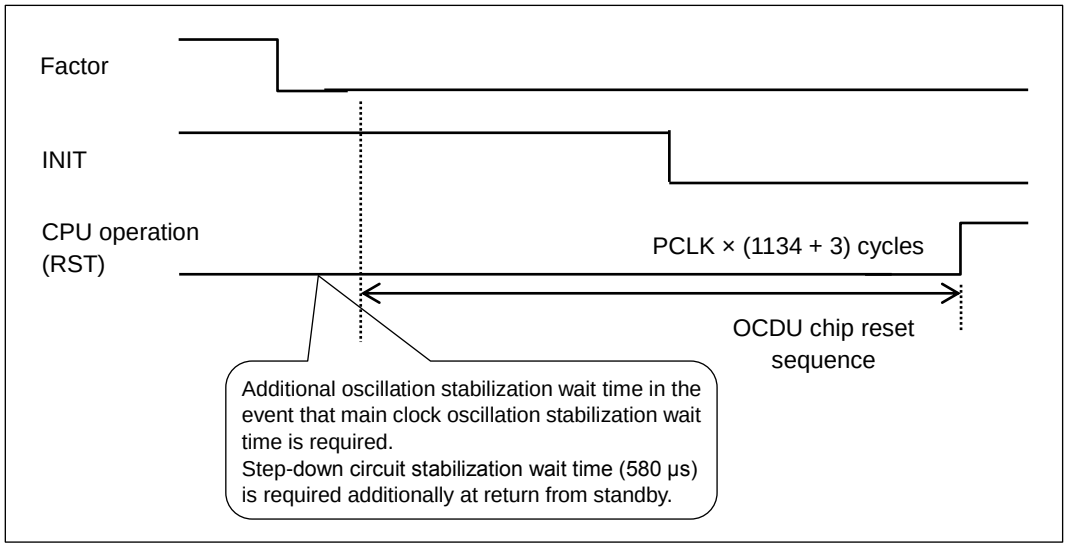
If the main clock is inactive such as in a stop mode, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15}× main clock cycle).

Table 5-2 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 5-3 Initialize Reset (INIT) Sequence



Because the clock settings register is initialized by reset, the period of the peripheral clock (PCLK) is 8 times the period of the main clock (MCLK).

5.4.3. Reset (RST)

The reset (RST) is shown.

If a reset factor that is not the SINIT or INIT level occurs, only a reset (RST) will be issued.

This reset is used for initializing the CPU and all registers except some registers (see "5.1.1. Initialize Reset (INIT)").

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

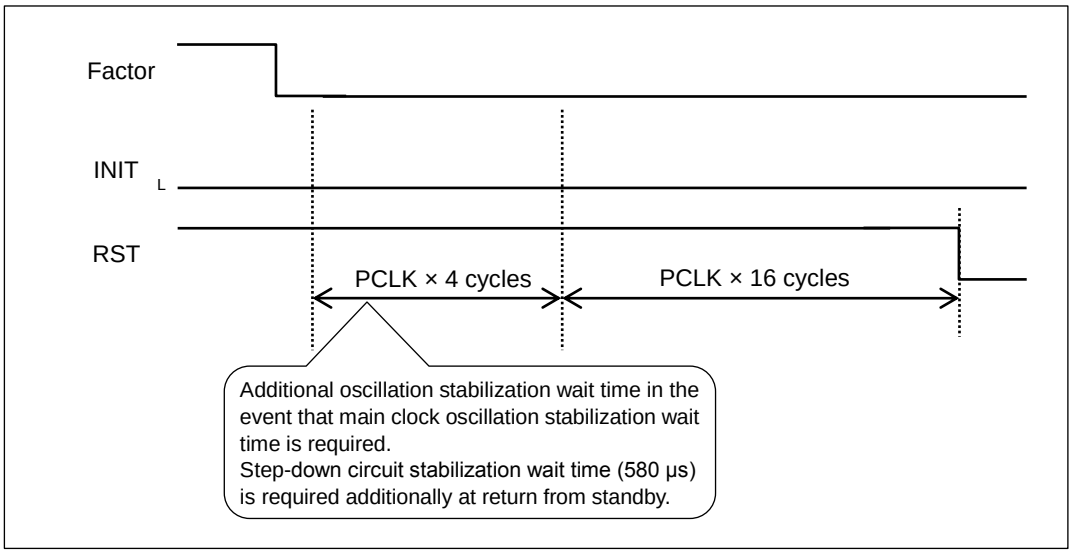
If the main clock is inactive such as in a stop mode before the reset, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15}× main clock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

Table 5-3 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 5-4 Reset (RST) Sequence



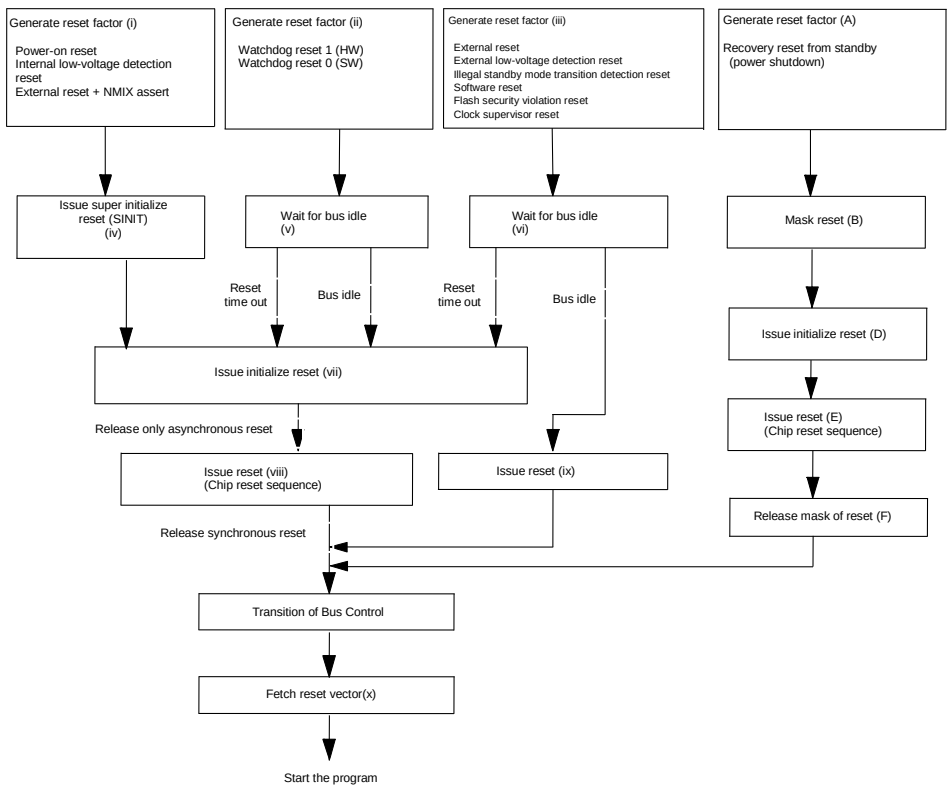
Because the clock settings register is initialized by reset, the period of the peripheral clock (PCLK) is 8 times the period of the main clock (MCLK).

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 5-5 Reset Sequence [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]



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).

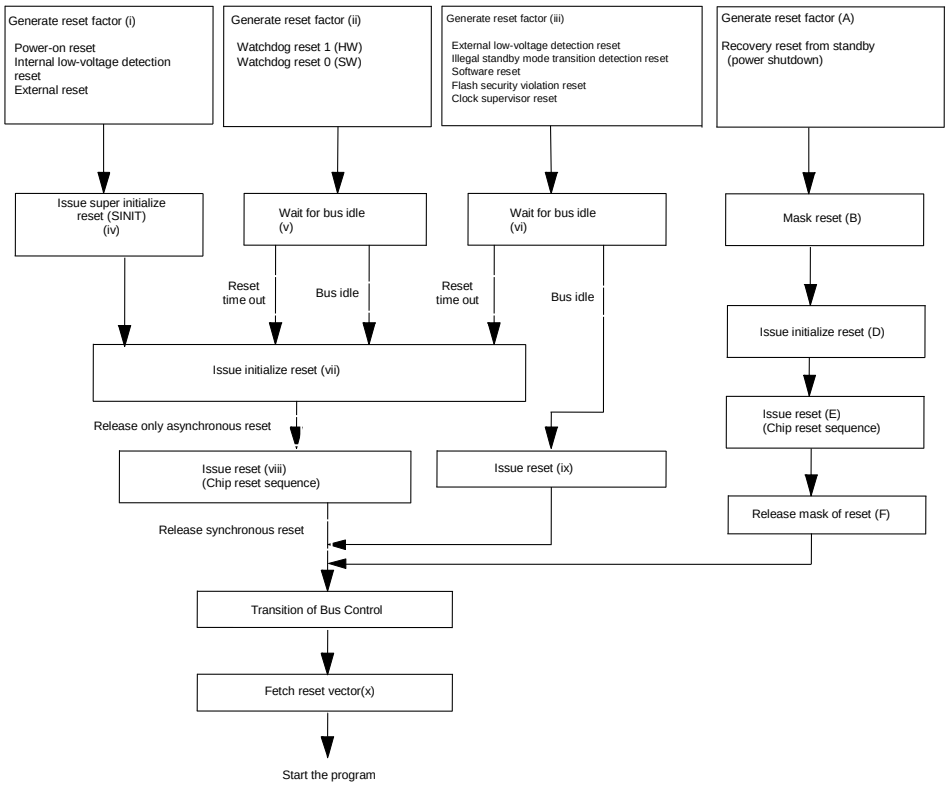
-The main clock oscillation stabilization wait time is taken during (vii), (viii), or (ix) if necessary (CMONR.MCRDY=0).

-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 factor, 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)

Figure 5-6 Reset Sequence [MB91F52xxxD]



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 factor, 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)

5.5.1. 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.

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.

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.

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.

5.5.5. Reset Vector Fetch

Reset vector fetch is shown.

After the reset release, the CPU starts fetching the reset vector.

After CPU acquires the bus control, the CPU accesses the reset vector through on-chip bus and retrieves the acquired reset vector to the PC to start running programs.

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.

5.6. Notes

Notes are shown.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] During return form standby watch mode (power-shutdown) and standby stop mode (power-shutdown), an internal reset is issued. Therefore any reset factor without power-on reset, internal low-voltage detection reset, reset by simultaneous assertion of RSTX and NMIX will not be accepted.

[MB91F52xxxD] During return form standby watch mode (power-shutdown) and standby stop mode (power-shutdown), an internal reset is issued. Therefore any reset factor without power-on reset, internal low-voltage detection reset, reset by assertion of RSTX will not be accepted.

Chapter 8: DMA Controller (DMAC)



This chapter explains the DMA controller (DMAC).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. DMA Usage Examples

Code : FR81S10_DMA-1v1-91520-2-E

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.

2. Features

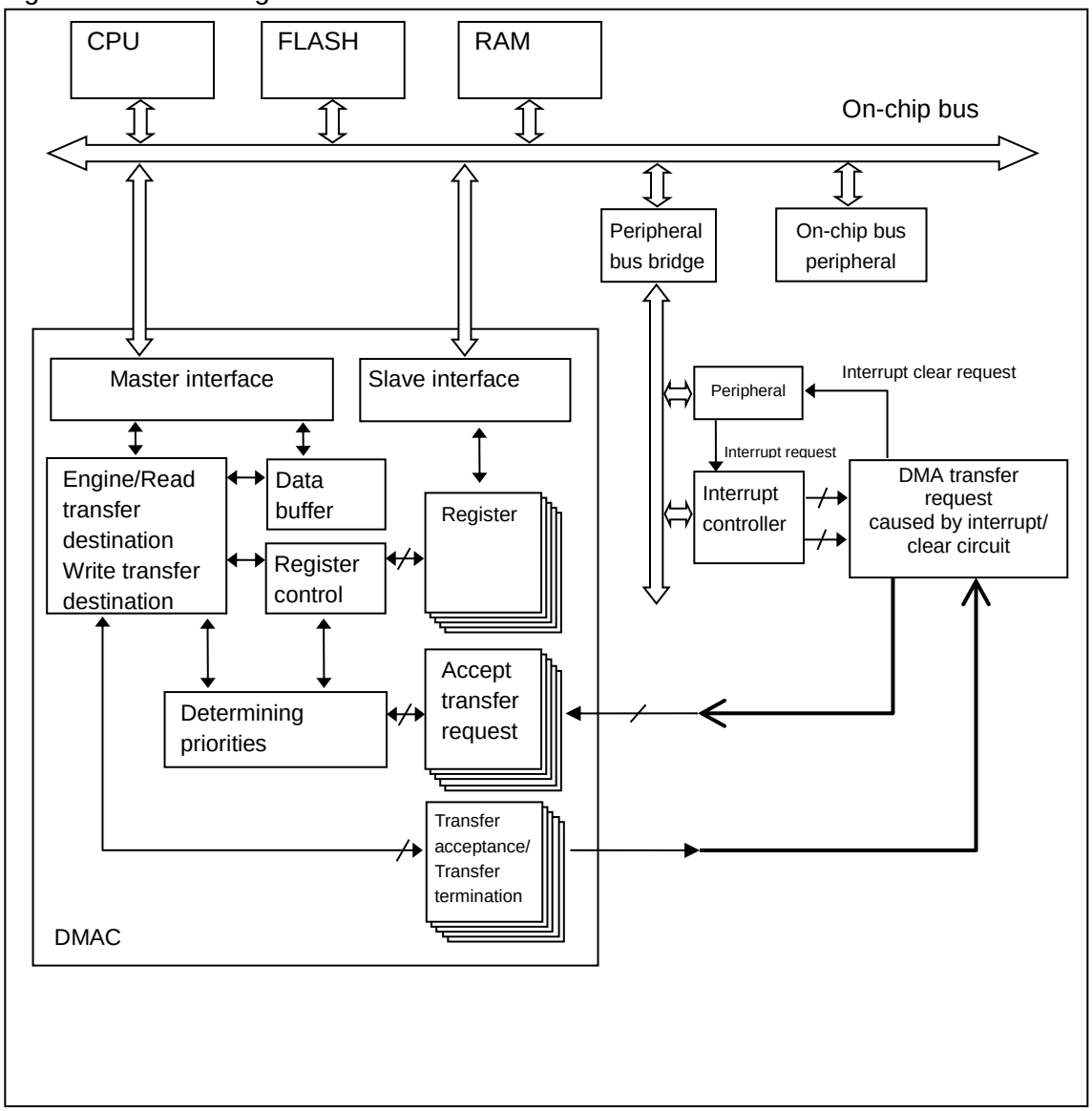
This section explains the features of the DMA controller (DMAC).

- Channels: 16 channels
- Address space: 32-bit address space (4 GB)
- Transfer mode: Block/burst transfer
- Address update: Increment/Decrement/Fixed (Address increment/decrement range : 1, 2, 4)
- Transfer size : 8-bits, 16-bits, 32-bits
- Block size: 1 to 16
- Transfer count: 1 to 65535
- Transfer request:
  - Software transfer requests
  - Transfer requests by peripheral 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)
    Or round robin
- Interrupt request : Normal completion interrupt requests, abnormal completion interrupt requests, and transfer suspend interrupt requests by transfer stop requests can be generated

3. Configuration

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

Figure 3-1 Block Diagram



4. Registers

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

Table 4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0C00	DCCR0				DMA channel control register 0
0x0C04	DCSR0		DTCR0		DMA channel status register 0 DMA transfer count register 0
0x0C08	DSAR0				DMA transfer source address register 0
0x0C0C	DDAR0				DMA transfer destination address register 0
0x0C10	DCCR1				DMA channel control register 1
0x0C14	DCSR1		DTCR1		DMA channel status register 1 DMA transfer count register 1
0x0C18	DSAR1				DMA transfer source address register 1
0x0C1C	DDAR1				DMA transfer destination address register 1
0x0C20	DCCR2				DMA channel control register 2
0x0C24	DCSR2		DTCR2		DMA channel status register 2 DMA transfer count register 2
0x0C28	DSAR2				DMA transfer source address register 2
0x0C2C	DDAR2				DMA transfer destination address register 2
0x0C30	DCCR3				DMA channel control register 3
0x0C34	DCSR3		DTCR3		DMA channel status register 3 DMA transfer count register 3
0x0C38	DSAR3				DMA transfer source address register 3
0x0C3C	DDAR3				DMA transfer destination address register 3
0x0C40	DCCR4				DMA channel control register 4
0x0C44	DCSR4		DTCR4		DMA channel status register 4 DMA transfer count register 4

Address	Registers				Register function
	+0	+1	+2	+3	
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
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

Address	Registers				Register function
	+0	+1	+2	+3	
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
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

Address	Registers				Register function
	+0	+1	+2	+3	
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

4.1. DMA Control Register: DMACR (DMA Control Register)

This section explains the DMA control register.

The DMA control register is a 32-bit register to control the entire DMAC (all channels). This register must be accessed as a 32-bit data.

■ DMACR : Address $0DF8_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	DME	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit31] DME (DMA Enable) : DMA operation enable

This bit controls the operation of the entire DMAC. When this bit is "0", a DMA transfer will not be performed even if operation of each channel is enabled. When this bit is "1", operations according to the settings for each channel are performed.

If "0" is written while a DMA transfer is in progress, the transfer is stopped in blocks specified in DCCRx.BLK.

DME	DMA operation enable
0	DMA operation disabled (Initial value)
1	DMA operation enabled

[bit30 to bit16] Reserved

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

[bit15] AT (Arbitration Type) : Priority setting

This bit configures how to determine priority for each channel. If the priority is set to "fixed" (AT = 0), ascending order, ch.0 > ch.1 > ch.2 > ch.3, is taken. If the priority is set to "round robin" (AT = 1), DMAC makes the priority of the channel which started the transfer the lowest and raises the priority of following channels one by one. The decision on priority is made on each transfer of a block unit specified in DCCRx.BLK regardless of the priority setting.

AT	Priority setting
0	Fixed (initial value)
1	Round robin

[bit14 to bit0] Reserved

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

4.2. DMA Channel Control Register 0 to 15: DCCR0 to 15 (DMA Channel Control Register 0 to 15)

This section explains the bit configuration for DMA channel control register 0 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 enable

This bit controls the operation of the channels. If the request source is set to "software", writing "1" to this bit starts a DMA transfer according to the configuration. In this case, the CE bit is automatically cleared when the transfer according to the transfer request completed.

If the request source is other than software, writing "1" to this bit makes channel operation enabled. After enabling operation, a DMA transfer starts when the corresponding transfer request is detected. In case of a request other than software, the CE bit will not be automatically cleared if transfer count reload (DCCRx.TCR) is specified. When transfer count reload is disabled, the CE bit will be cleared when all transfers are finished.

If "0" is written while the operation is going on regardless of the request source, stop transfer in blocks specified in DCCRx.BLK. When writing "1" again and detecting a new transfer request, the operation restarts.

CE	Channel operation enable
0	Disabled (initial value)
1	Enabled

[bit30 to bit27] Reserved

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

[bit26] AIE (Abnormal completion Interrupt Enable) : Abnormal completion interrupt enable

This bit controls the generation of interrupts when setting the prohibited values to the DMA channel control register (DCCR). The items not allowed to set to registers are listed below.

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

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

AIE	Abnormal completion interrupt enable
0	Disabled (initial value)
1	Enabled

[bit25] SIE (Stop Interrupt Enable) : Transfer suspend interrupt enabled by transfer stop requests

This bit controls the generation of interrupts when a DMA transfer is suspended by a transfer stop request from the transfer request source. As for the interrupt factor, refer to the status register (DCSRx).

SIE	Transfer suspend interrupt enable
0	Disabled (initial value)
1	Enabled

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

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

NIE	Normal completion interrupt enable
0	Disabled (initial value)
1	Enabled

[bit23 , bit22] Reserved

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

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

These bits select the transfer request source for the channel.

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 (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 (Source Address Count) : Transfer source address count

These bits specify the address update once for each transfer of the transfer source address. The update values when specifying "increment/decrement" will be one of the values, 1, 2, 4 depending on the transfer size (DCCRx.TS).

SAC[1:0]	Transfer Source address count
00	Address increment (initial value)
01	Address decrement
10	Reserved (setting is prohibited)
11	Address fixed

[bit11] DT (Destination Type) : Transfer destination type

The setting values are different depending on the combinations of DMA transfer request source (DCCR.RS[1:0]), transfer source address (DSAR), and transfer destination address (DDAR). 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 (Destination Address Count) : Transfer destination address count

These bits specify the address update once for each transfer of the transfer destination address. The update values when specifying "increment/decrement" will be one of the values, 1, 2, 4 depending on the transfer size (DCCRx.TS).

DAC[1:0]	Transfer destination address count
00	Address increment (initial value)
01	Address decrement
10	Reserved (setting is prohibited)
11	Address fixed

[bit7] TCR (Transfer Count Reload) : Transfer count reload

This bit specifies the transfer count register reload.

When specifying a reload, the transfer count register value is returned to the initial value at the end of the transfer. If the transfer request source is set other than "software", DCCRx.CE bit will not be cleared at the end of the transfer and the operation will go into the transfer request wait state. When disabling a reload, the transfer count register value at the end of the transfer will point to "0". In this case, DCCRx.CE bit will be cleared at the end of the transfer regardless of the transfer request source.

TCR	Transfer count reload
0	Reload disabled (initial value)
1	Reload

[bit6] Reserved

Always write "0" to this bit. The read value is "0".

[bit5, bit4] TS (Transfer Size) : Transfer size

These bits specify the transfer size. DMA transfers will be performed once with the bit width specified here.

TS[1:0]	Transfer size
00	8-bit :byte (initial value)
01	16-bit :half-word
10	32-bit :word
11	Reserved (setting is prohibited)

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

[bit3 to bit0] BLK (BlocK size) : Block size

These bits specify the block size. 1 block transfer will be repeated for the number of blocks of the transfer size specified with DCCRx.TS bit.

BLK[3:0]	Transfer count
0000	Once (initial value)
0001	Twice
0010	3 times
0011	4 times
0100	5 times
0101	6 times
0110	7 times
0111	8 times
1000	9 times
1001	10 times
1010	11 times
1011	12 times
1100	13 times
1101	14 times
1110	15 times
1111	16 times

4.3. DMA Channel Status Register 0 to 15 : DCSR0 to 15: (DMA Channel Status Register 0 to 15)

This section explains the bit configuration for DMA channel status register 0 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 channel operating state. Writing "1" to the corresponding DCCRx.CE bit for the channel makes it in the operating state. Completing transfers for as many times as set transfer count or writing "0" to DCCRx.CE makes the operation stop.

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

When having allowed the abnormal completion interrupt (DCCRx.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 suspend state by the transfer stop request

This bit indicates that a DMA transfer has been suspended by a transfer stop request from the transfer request source. When having allowed the transfer suspension interrupt (DCCRx.SIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is 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 "DCCRx.CE" bit at the time the transfer count is "0", the operation will complete normally. When having allowed the normal completion interrupt (DCCRx.NIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is 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

4.4. DMA Transfer Count Register 0 to 15 : DTCR0 to 15: (DMA Transfer Count Register 0 to 15)

This section explains the bit configuration for DMA transfer count register 0 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 transfers. DMAC decreases a transfer count at the end of each block transfer and stops the transfer when the transfer count becomes "0". If "0" is set for transfer count, transfer will not be performed. Also, the dedicated reload register is provided. If DCCRx.TCR is "1", the value is returned to the initial value after data transfer.

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.

These registers are 32-bit registers to indicate the transfer source address of each DMAC channel, which exist independently for each channel. These registers must be accessed as a 32-bit data.

■ DSAR0 to 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 DCCRx.SAC, the address is updated according to the transfer size (DCCRx.TS). Also, the dedicated reload register is provided. If DCCRx.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 DCCRx.TS.

If the DMA transfer request source has a peripheral interrupt (DCCRx.RS[1:0]=01), at least either the transfer source address (DSAR) or the transfer destination address (DDAR) must be within the address range of peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus. For details, see "■ Setting the ST Bit (Transfer source type) and DT Bit (Transfer destination type)".

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.

These registers are 32-bit registers to indicate the transfer destination address of each DMAC channel, which exist independently for each channel. These registers must be accessed as a 32-bit data.

■ DDAR0 to 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

[bit31 to bit0] DDA[31:0] (DMA Destination Address) : DMA transfer destination address

These registers indicate the transfer destination address. If an increment or a decrement is set by DCCRx.DAC, the address is updated according to the transfer size (DCCRx.TS). Also, the dedicated reload register is provided. If DCCRx.DAR is "1", the value is returned to the initial value after data transfer.

Set a value in these registers not to cause a misalignment against the transfer size to be set by DCCRx.TS.

If the DMA transfer request source has a peripheral interrupt (DCCRx.RS[1:0]=01), at least either the transfer source address (DSAR) or the transfer destination address (DDAR) must be within the address range of peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus. For details, see "■ Setting the ST Bit (Transfer source type) and DT Bit (Transfer destination type)".

4.7. DMA Transfer Suppression NMI Flag Register : DNMIR (DMA-halt by NMI Register)

This section explains the bit configuration for DMA transfer suppression flag register.

This register is 8-bit register to suppress DMA transfer by the user NMI. This register must be accessed as a 8-bit data.

■ DNMIR: Address $0DF6_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NMIH	Reserved						NMIHD
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

[bit7] NMIH (NMI Halt) : DMA suppression flag (by NMI factor)

If the NMIHD bit is "0", this flag shows an occurrence of the user NMI request. The "H" level of NMI is detected, and this bit is set to "1". To restart DMA transfer, set this bit to "0".

Writing "1" to this bit is 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.

4.8. DMA Transfer Suppression Level Register : DILVR (DMA-halt by Interrupt Level Register)

This section explains the bit configuration for DMA transfer suppression level register.

This register is 8-bit register to control the DMA transfer suppression by peripheral interrupts. This register must be accessed as a 8-bit data.

■ DILVR: Address $0DF7_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			LVL4	LVL[3:0]			
Initial value	0	0	0	1	1	1	1	1
Attribute	R0,W0	R0,W0	R0,W0	R1,WX	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved
Always write "0" to these bits. The read value is "0".

[bit4 to bit0] LVL (Level) : DMA suppression interrupt level
These bits set an interrupt level for suppression of DMA transfer. If a peripheral interrupt having an interrupt level higher than the one specified by this register occurs, the DMA transfer is suppressed. LVL4 is fixed to "1", but LVL[3:0] can be set to any level.

LVL[4:0]	DMA suppression control
11111	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.

LVL[4:0]	DMA suppression control
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.

5. Operation

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

5.1. Configuration

5.1. Configuration

This section explains the configuration for DMAC operation.

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

5.1.1. Common Items for All Channels

The common Items for all channels is shown below.

This section explains the register settings for control of the entire DMAC.

■ DMA Operation Enable

The entire DMAC operation 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.

5.1.2. Separate Items for Each Channel

The items set separately for each channel are shown.

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

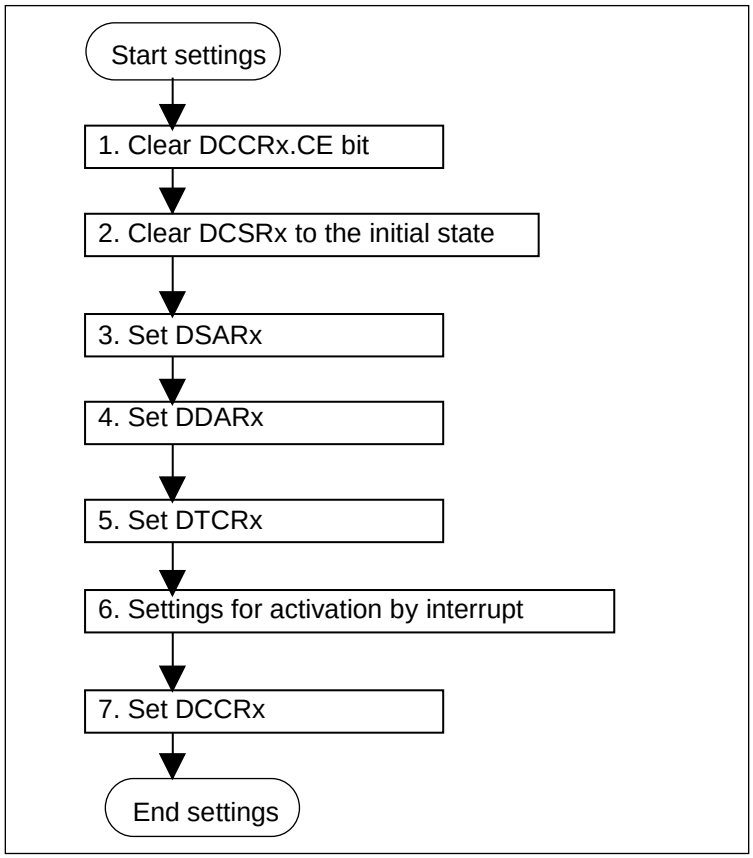
■ Register Setup Procedure

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

1. Clear the DCCRx.CE bit to disable the channel operation.
2. Clear each bit of DCSRx register to initialize the channel status flag.
3. Set the transfer source address (to be used when the transfer starts) in the DSARx register.

4. Set the transfer destination address (to be used when the transfer starts) in the DDARx register.
5. Set the transfer count in the DTCRx register. This count must be 1 or a larger value.
6. 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 DCCRx register. During this time, the channel operation is enabled when the DCCRx.CE bit is set.

Figure 5-1 Channel Register Setup Procedure



■ Transfer Source Address and the Transfer Destination Address Setting

Set the transfer source address (to be used when the transfer starts) using the DSARx.DSA.

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

Align the transfer source and destination addresses based on the transfer size (DCCRx.TS), and ignore the lower 1 bit or lower 2 bits for 16-bit or 32-bit transfer size respectively.

■ Transfer Count Setting

Set the number of times of block transfer (repeated to the end of transfer) using the DTCRx.DTC. The transfer count can be 1 to 65535 times. The DMAC transfers data (1 block data), whose length in bytes is set by the transfer size and block size (see "■ Transfer Size and Block Size Setting") for the specified number of times.

■ Channel Operation Enable

Set the channel operation control using the DCCRx.CE.

- Disable the channel operation (DCCRx.CE = 0)
- Enable the channel operation (DCCRx.CE = 1)

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

■ Interrupt Enable Setting

Enable an interrupt during abnormal completion, using the DCCRx.AIE.

- Disable an abnormal completion interrupt (DCCRx.AIE = 0)
- Enable an abnormal completion interrupt (DCCRx.AIE = 1)

Using the DCCRx.SIE, enable an interrupt to occur if data transfer is suspended by a transfer stop request.

- Disable a transfer suspend interrupt during detection of transfer stop request (DCCRx.SIE = 0)
- Enable a transfer suspend interrupt during detection of transfer stop request (DCCRx.SIE = 1)

Enable an interrupt during normal completion, using the DCCRx.NIE.

- Disable a normal completion interrupt (DCCRx.NIE = 0)
- Enable a normal completion interrupt (DCCRx.NIE = 1)

■ Transfer Request Source setting

Set the transfer request source to accept a transfer request using the DCCRx.RS.

- Request by software (DCCRx.RS = 00)
- Request by an interrupt (DCCRx.RS = 01)

■ Transfer Mode Setting

Set the DMA transfer mode using the DCCRx.TM.

- Block transfer (DCCRx.TM = 00)
- Burst transfer (DCCRx.TM = 01)

■ Setting the ST Bit (Transfer source type) and DT Bit (Transfer destination type)

Set them by following the table definition below. The DMA transfer is not supported in combination (5).

Table 5-1 ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type) Setting

	Combination of transfer request source, transfer source, and transfer destination			DMA transfer support	ST and DT bit setting
	Transfer request source (DCCR.RS[1:0])	Transfer source (DSAR)	Transfer destination (DDAR)		
(1)	Request by software (DCCR.RS[1:0] = 00)	Any combination		Supported	ST= 0, DT= 0
(2)	Peripheral interrupt (DCCR.RS[1:0] = 01)	●	□	Supported	ST= 1, DT= 0
(3)		□	●	Supported	ST= 0, DT= 1
(4)		●	●	Supported	ST= 0, DT= 1
(5)		□	□	Not supported	-

● : Address range of the peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus
□ : Other address range

If the ST and DT bits are set in a combination other than above, the interrupt may not be cleared automatically after occurrence of the DMA transfer request.

■ Transfer Address Reload Setting

Using the DCCRx.SAR, set the reload control of transfer source address at the end of transfer.

- The transfer source address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.SAR=0)
- The transfer source address is returned to the initial value at the end of transfer. (DCCRx.SAR=1)

Using the DCCRx.DAR, set the reload control of transfer destination address at the end of transfer.

- The transfer destination address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.DAR=0)
- The transfer destination address is returned to the initial value at the end of transfer. (DCCRx.DAR=1)

■ Transfer Address Update Setting

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

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

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

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

■ Transfer Count Reload Setting

Using the DCCRx.TCR, set the reload control of transfer count at the end of transfer.

- The transfer count is not reloaded after the transfer. (After the normal completion of transfer, the transfer count is set to 0.) (DCCRx.TCR=0)
- The transfer count is returned to the initial value at the end of transfer. (DCCRx.TCR=1)

■ Transfer Size and Block Size Setting

To set a transfer unit for DMA transfer (the byte count to be transferred as 1 block), set the transfer size and block size.

Using the DCCRx.TS, set the size of data to be sent by a single DMA transfer (8-bit/16-bit/32-bit).

- 8-bit (DCCRx.TS = 00)
- 16-bit (DCCRx.TS = 01)
- 32-bit (DCCRx.TS = 10)

Using the DCCRx.BLK, set the DMA transfer count for 1-block data transfer. The block size can be 1 to 16 times. In the 1-block transfer, data having the bit width being set by the transfer size (DCCRx.TS), is transferred for the number of times being set by the block size.

5.1.3. Operations

This section explains DMAC operations.

This section explains the DMAC operations as follows.

(1) Channel status check
(2) Data transfer

■ Channel Status Check

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

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

Data writing to the DCSRx.CA bit is ignored.

The DCSRx.AC, DCSRx.SP, and DCSRx.NC bits must be cleared before the DMA transfer is allowed because these bits are not cleared automatically.

■ Data Transfer

The DMAC starts DMA transfer when the transfer source address and transfer destination address are set. By receiving a transfer source read instruction, this controller reads the data, having the bit width (8-bit/16-bit/32-bit) being set by DCCRx.TS, from the transfer source address, and temporarily stores it in the data buffer inside of the DMAC. By receiving a transfer destination write instruction, the controller writes the data temporarily stored in the DMAC into the transfer destination address.

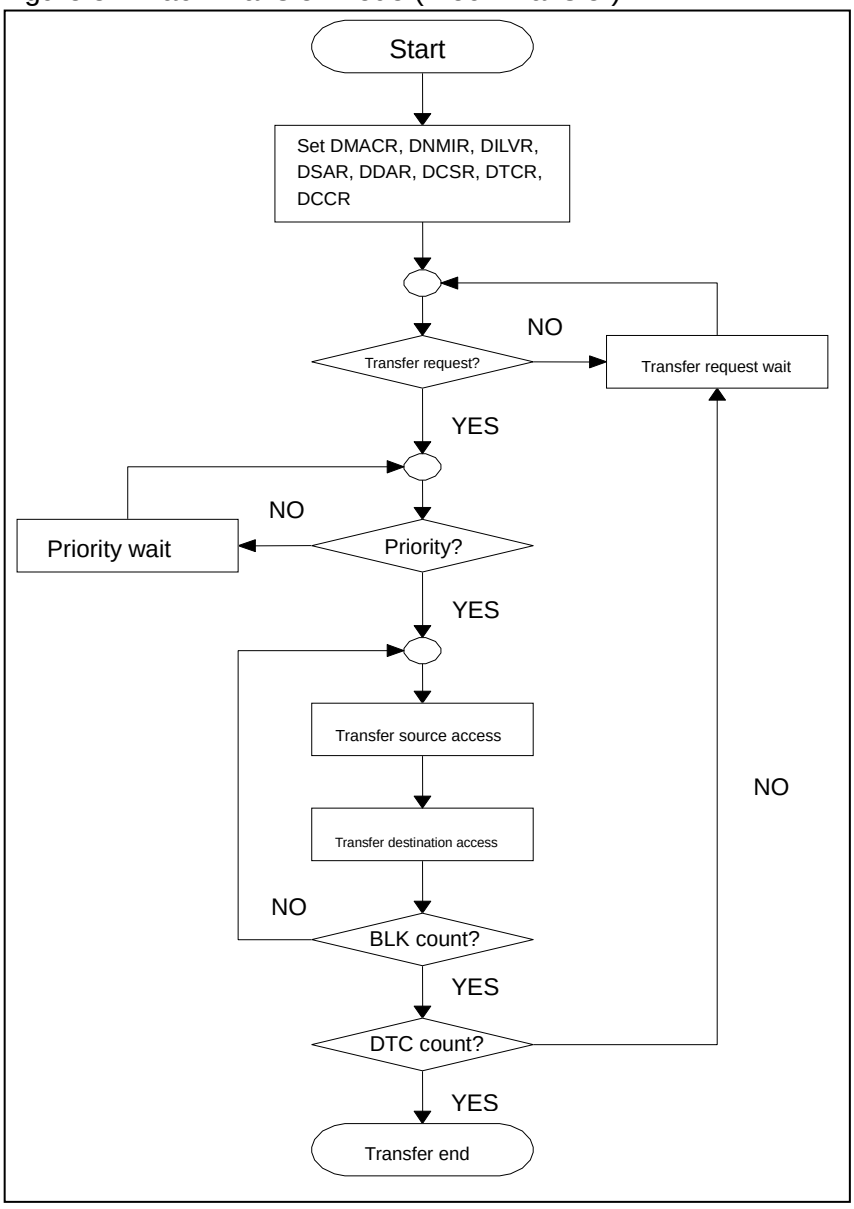
■ Transfer Mode

The transfer mode has block transfer mode or burst transfer mode.

● Block Transfer Mode

1-time transfer request causes the 1 block transfer. When a transfer request is detected after the block transfer, the next 1-block transfer occurs. These operations are repeated until the end of data transfer. During 1-block data transfer, the data having the size specified by the DCCRx.TS bit is transferred for the number of times being set by the block size.

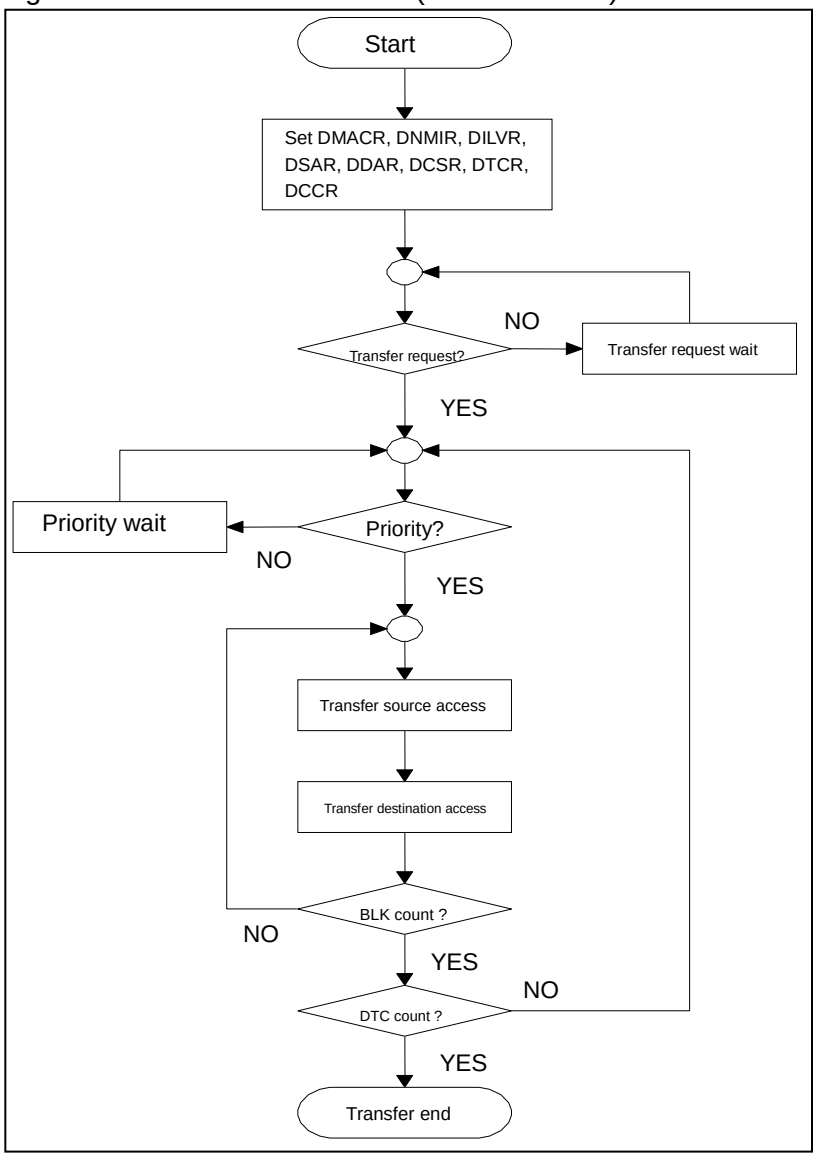
Figure 5-2 Each Transfer Mode (Block Transfer)



Burst transfer mode

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

Figure 5-3 Each Transfer Mode (Burst Transfer)



■ Transfer request

The transfer request has a request by software or a request by interrupt. The following explains the relationship between the transfer request detection conditions and the transfer mode.

● Request by software

If the DCCRx.CE bit is set to "1", a transfer request is detected. When the DMA operation is enabled (DMACR.DME=1), the priority is determined and the data transfer is started immediately. When the data transfer by the transfer request has terminated, the DCCRx.CE bit is cleared automatically.

● Request by interrupt

If the channel operation is enabled (DCCRx.CE=1), a transfer request is awaited. If a peripheral interrupt, being set by the interrupt controller, has occurred, its transfer request is detected. When the DMA operation is enabled (DMACR.DME=1), the priority is determined and the data transfer is started immediately.

When a transfer stop request is asserted from the peripheral, a transfer request is not detected.

Also, an interrupt vector to be used for transfer request must be set for each channel. See 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 5-2 Relationship between Transfer Request Detection Conditions and Transfer Mode

	Block transfer	Burst transfer
Request by software	Set the DCCRx.CE bit to "1".	Set the DCCRx.CE bit to "1".
Request by interrupt	Edge detection	Edge detection

Also, the relationship between the detected transfer request and the DMACR.DME and DCCRx.CE bits is given on Table 5-3. If the DME bit or CE bit is cleared during transfer, the block transfer is stopped.

Table 5-3 Relationship between Transfer Requests and DME/CE Bits

		DME bit	CE bit
DME/CE clear		The already detected transfer request is not cleared.	The already detected transfer request is cleared.
DME/CE setting after the transfer interrupt	Block transfer	When a new transfer request is detected, the data transfer is restarted based on the priority.	When a new transfer request is detected, the data transfer is restarted based on the priority.
	Burst transfer	When the DME bit is set, the data transfer is restarted immediately based on the priority.	

Standby recovery request by DMA transfer request

If the MCU receives a transfer request in the standby mode, the DMAC requests the MCU to recover from the standby mode. If data transfer is enabled and if a transfer request is asserted by the transfer request source, a standby recovery is requested.

Channel priority

If multiple transfer requests are issued, the DMAC starts data transfer on the channel having the highest priority. The channel priority can be fixed or can be set by round robin. The priority is determined for each block transfer or when data transfer ends.

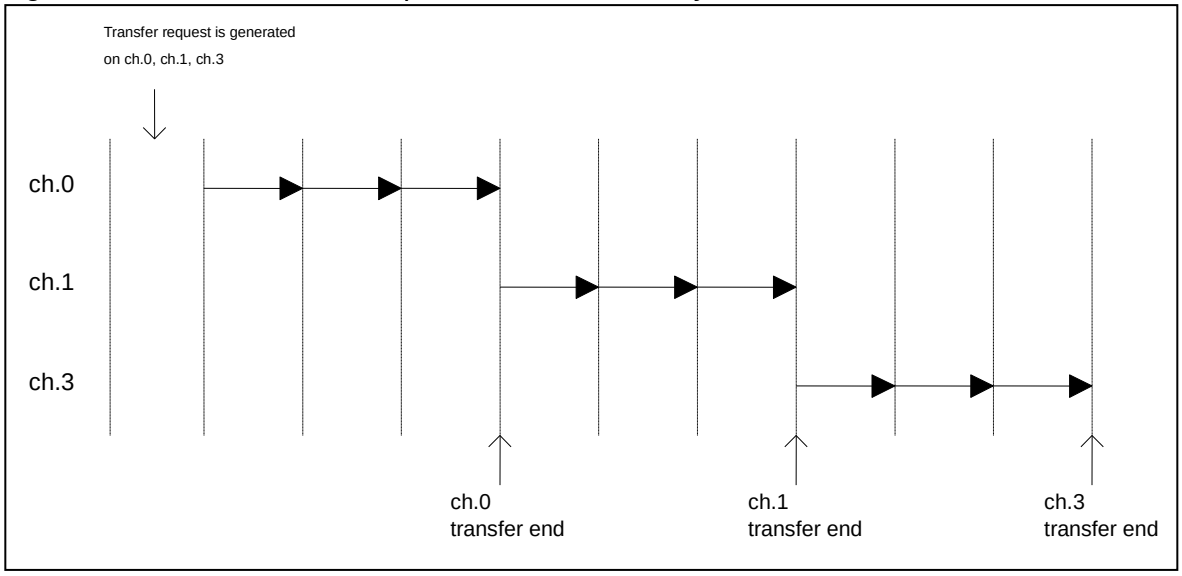
- Fixed priority (DMACR.AT = 0)
  The channel priority is fixed in the sequence of "ch.0 > ch.1 > ch.2 > ch.3". The following gives an example.

Example 1 : If transfer requests are issued on ch.0, ch.1 and ch.3 simultaneously, data transfer starts from ch. 0. When data transfer ends on ch.0, the next data transfer starts on ch.1. After data transfer on ch.1, the next data transfer starts on ch.3. The following gives transfer examples. Dotted lines in the figure show the block delimiters.

Transfer request : Requests are issued for ch.0, ch.1 and ch.3 simultaneously.

Setting : Ch.0, ch.1 and ch.3 are set to the burst transfer mode, and a data transfer count of 3.

Figure 5-4 Data Transfer Example 1 If Channel Priority Is Fixed

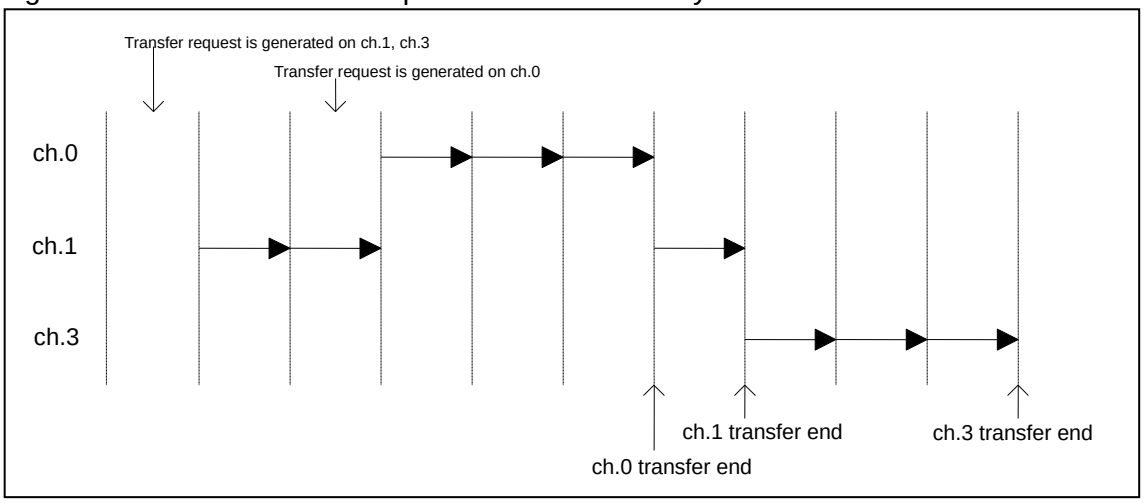


Example 2: If transfer requests are issued simultaneously for ch.1 and ch.3 and if a transfer request on ch.0 is issued during data transfer on ch.1, the data transfer on ch.1 is temporarily stopped and data transfer on ch.0 is started. During this time, the channel transition occurs in units of blocks. When the requested data transfer ends on ch.0, the data transfer is started on ch.1. Dotted lines in the figure show the block delimiters.

Transfer request : Requests are issued for ch.1 and ch.3 simultaneously. When data is transferred on ch.1, another request for transfer on ch.0 is issued.

Setting : Ch.0, ch.1 and ch.3 are set to the burst transfer mode, and data transfer count of 3.

Figure 5-5 Data Transfer Example 2 If Channel Priority Is Fixed

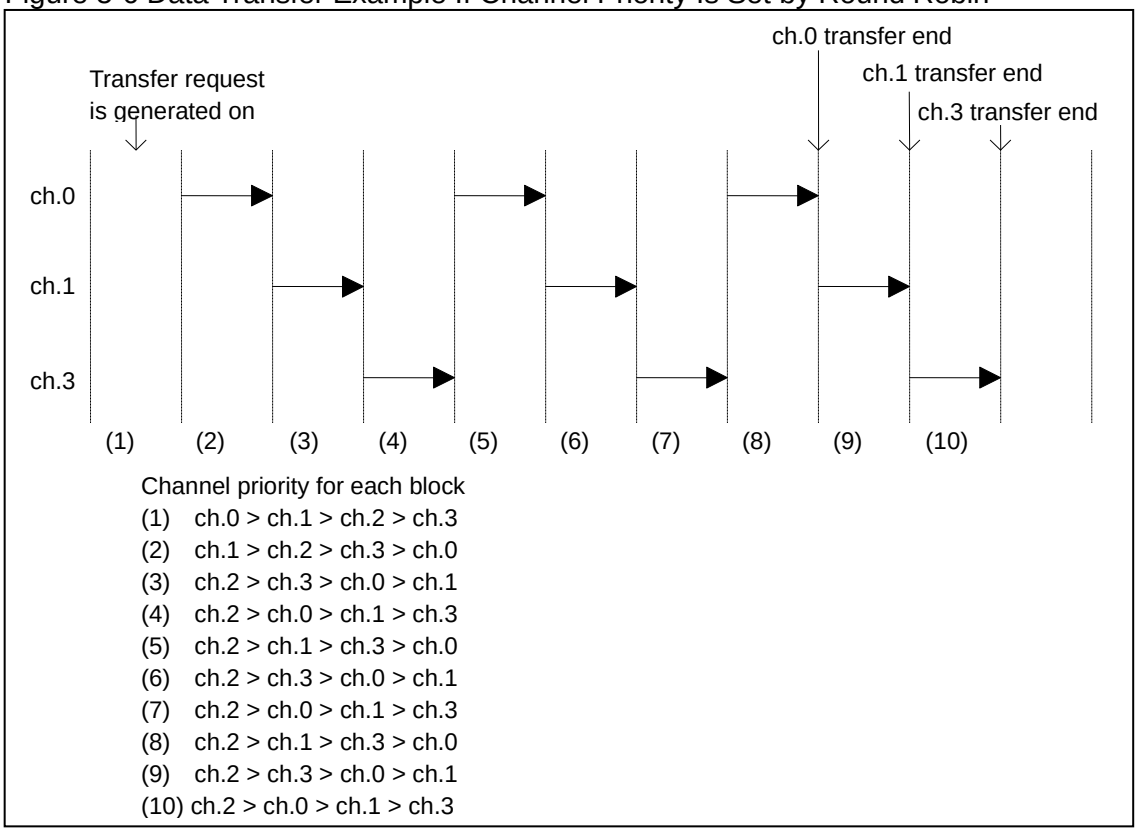


- Round robin (DMACR.AT = 1)
  When data transfer is started on a channel, its priority is set to the lowest level. A channel priority below this level is raised by one level. In the round robin, data transfer starts on a channel having the highest priority when a transfer request is issued. The priority of the channel where data transfer has started is dropped to the lowest level. The priority is determined for each of block data transfer, and data transfer is started on the channel having the highest priority. The following gives a transfer example. Dotted lines in the figure show the block delimiters.

Example : Transfer request : Requests are issued for ch.0, ch.1 and ch.3 simultaneously.

Setting : Ch.0, ch.1 and ch.3 are set to the burst transfer mode; and data transfer count of 3.

Figure 5-6 Data Transfer Example If Channel Priority Is Set by Round Robin



- **Updating of transfer address**

  The transfer source address and transfer destination address are updated each time data which size has been set by the DCCRx.TS is transferred. The address updating can be increasing, decreasing, or fixed. When increasing or decreasing, its address amount is determined by the transfer size (DCCRx.TS). If fixed, the address value does not change. Table 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 5-4 Updating of Transfer Source Address and Transfer Destination Address

Address setting		Transfer size (TS)	Address updating for each data transfer	
Transfer source (SAC)	Transfer destination (DAC)		Transfer source (DSA)	Transfer destination (DDA)
Increments ("00")	Increments ("00")	8-bit ("00")	Increments by 1	Increments by 1
		16-bit ("01")	Increments by 2	Increments by 2
		32-bit ("10")	Increments by 4	Increments by 4
	Decrements ("01")	8-bit ("00")	Increments by 1	Decrements by 1
		16-bit ("01")	Increments by 2	Decrements by 2
		32-bit ("10")	Increments by 4	Decrements by 4
	Fixed ("11")	8-bit ("00")	Increments by 1	Not updated
		16-bit ("01")	Increments by 2	
		32-bit ("10")	Increments by 4	
Decrements ("01")	Increments ("00")	8-bit ("00")	Decrements by 1	Increments by 1
		16-bit ("01")	Decrements by 2	Increments by 2
		32-bit ("10")	Decrements by 4	Increments by 4
	Decrements ("01")	8-bit ("00")	Decrements by 1	Decrements by 1
		16-bit ("01")	Decrements by 2	Decrements by 2
		32-bit ("10")	Decrements by 4	Decrements by 4
	Fixed ("11")	8-bit ("00")	Decrements by 1	Not updated
		16-bit ("01")	Decrements by 2	
		32-bit ("10")	Decrements by 4	
Fixed ("11")	Increments ("00")	8-bit ("00")	Not updated	Increments by 1
		16-bit ("01")		Increments by 2
		32-bit ("10")		Increments by 4
	Decrements ("01")	8-bit ("00")		Decrements by 1
		16-bit ("01")		Decrements by 2
		32-bit ("10")		Decrements by 4
	Fixed ("11")	8-bit ("00")		Not updated
		16-bit ("01")		
		32-bit ("10")		

● Reloading of transfer address

The DMAC can reload the transfer address after the specified number of data transfer has completed.

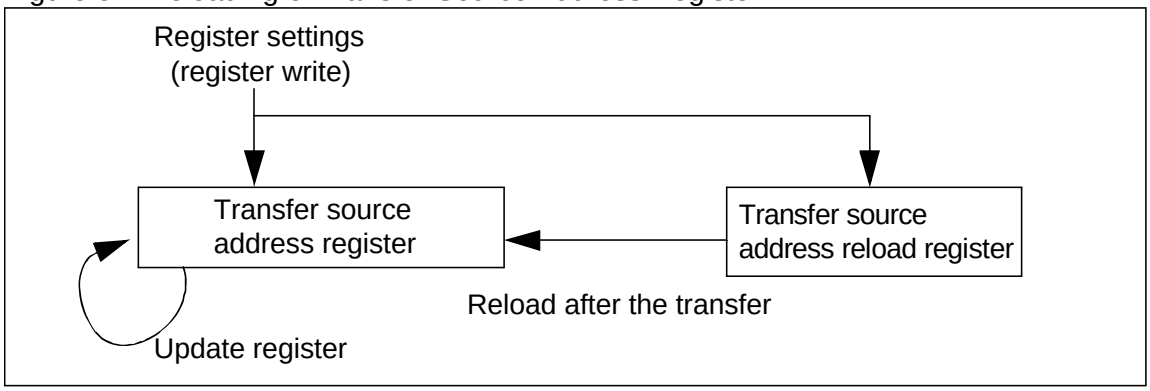
- Reloading of transfer source address

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

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

  If the specified number of transfers is suspended or abnormally terminated, the DSARx.DSA bit indicates the next access address (after the terminated address) regardless of the reload setting of the transfer source address.

Figure 5-7 Reloading of Transfer Source Address Register



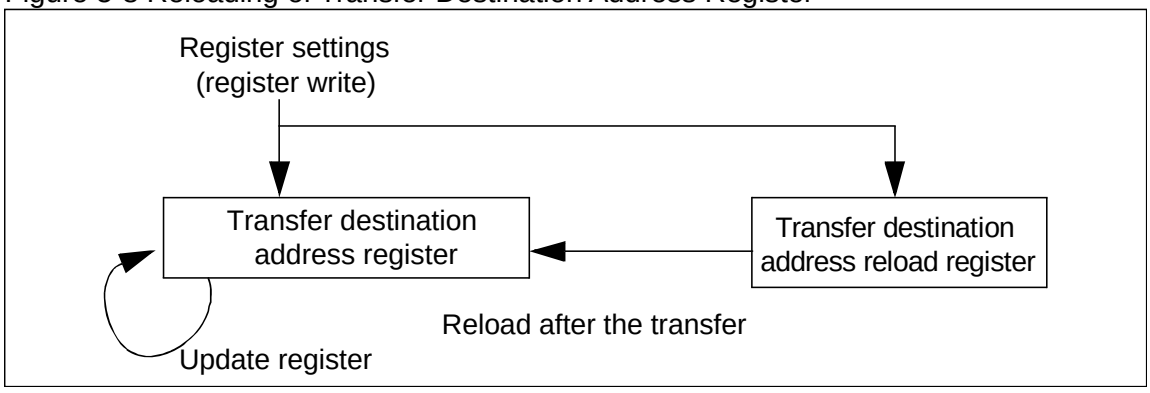
- Reloading of transfer destination address register

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

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

  If the specified number of transfers is suspended or abnormally terminated, the DDARx.DDA bit indicates the next access address (after the terminated address) regardless of the reload setting of the transfer destination address.

Figure 5-8 Reloading of Transfer Destination Address Register



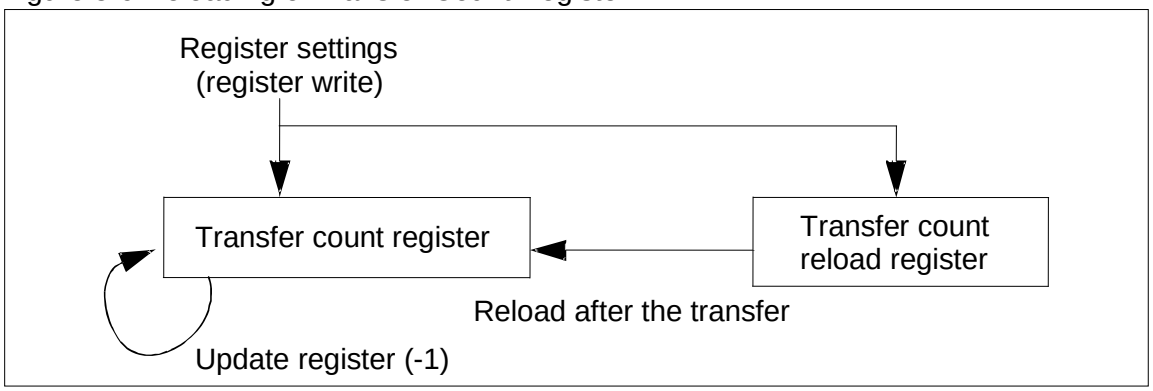
● Reloading of transfer count

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

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

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

Figure 5-9 Reloading of Transfer Count Register



The DCCRx.CE bit status varies after the data transfer, depending on the reload setting of the transfer count. The following explains the relation between the transfer count reload setting and the transfer request source.

Table 5-5 DCCRx.CE Bit at the End of Transfer

	Software request	Non-software request
If the reloading of transfer count is set	The DCCRx.CE bit is cleared	The DCCRx.CE bit is not cleared
If the reloading of transfer count is disabled	The DCCRx.CE bit is cleared	The DCCRx.CE bit is cleared

● Transfer suspension

The DMAC suspends the DMA transfer due to the following causes.

- A suspension as the DMACR.DME bit is cleared
- A suspension as the DCCRx.CE bit is cleared
- A suspension caused by the transfer stop request by the transfer request source peripheral

Data transfer is suspended in units of blocks. If data transfer is suspended, the next transfer is not started. Data transfer is stopped. The settings to restart data transfer vary depending on the suspension cause.

- A suspension as the 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 DCCRx.CE bit is cleared
  If the DCCRx.CE bit is cleared, the channel is stopped from operating. After a block of data has been transferred, the data transfer is suspended. Also, as the DCCRx.CE bit is cleared, the already detected transfer request is cleared. To restart data transfer, set the DCCRx.CE bit for the stopped channel and issue a new transfer request.

· A transfer stop request from the transfer request source peripheral

The following peripherals can issue a transfer stop request under certain conditions.

(A) Multi-function serial interface

If a PE, FRE, or ORE flag is set

(B) LIN

If a PE, FRE, or ORE flag is set

If a transfer stop request is issued, the transfer is suspended after one block of the current data has been transferred. If the data transfer is suspended, the following occur.

· The SP bit of DMA channel status registers (DCSR0 to DCSR15) is set to "1".
· The CE bit of DMA channel control registers (DCCR0 to DCCR15) 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 (DCSR0 to DCSR15) of the corresponding channel to "0".
3. Set the CE bit of DMA channel control registers (DCCR0 to DCCR15) to "1".
4. Issue a new transfer request.

Table 5-6 Settings to Restart the Suspended Data Transfer

	DME clear	CE clear	If a transfer stop request from transfer request source peripheral is detected
Setting to restart transfer	(1) Set the DME bit	(1) Set the CE bit (2) Issue a transfer request	(1) The transfer request is negated (2) The SP bit is cleared (3) The CE bit is set (4) Issue a transfer request

● Transfer termination

Data transfer can terminate normally or abnormally.

· Normal termination

The transfer terminates normally at the time when the transfers for the number of times set by the transfer count (DTCRx.DTC) end. When terminated normally, the DCSRx.NC bit of the corresponding channel is set. Also, the DCCRx.CE bit is cleared and data transfer is stopped. However, if the reloading of the transfer count has been set by non-software transfer request source, the DCCRx.CE bit of the channel is not cleared.

If writing "1" to the corresponding channel's DCCRx.CE bit at the time the transfer count (DTCRx.DTC) is "0", the DCSRx.NC bit is set in the similar way as for the normal termination. Before setting the DCCRx.CE bit to "1", be sure to set the DTCRx.DTC bit to "1" or a larger value.

- Abnormal termination
  If an inhibited value is set in the register, data transfer terminates abnormally. When terminated abnormally, the DCSRx.AC bit of the corresponding channel is set. Also, the DCCRx.CE bit is cleared and data transfer is stopped.

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

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

Interrupt request

The DMAC can issue an interrupt request at normal termination of data transfer, at abnormal termination of data transfer, or at transfer suspension by a transfer stop request. When issuing an interrupt request, set the interrupt controller as well.

Use the DMA channel status register (DCSRx) to check the interrupt request factor or to clear the interrupt request.

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

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

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

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

  Clear the interrupt request by clearing the DCSRx.AC bit of the corresponding channel.

- A transfer suspension interrupt request by a transfer stop request
  If the transfer suspension interrupt of a channel is enabled (DCCRx.AIE=1), the DMAC issues the interrupt request if data transfer is suspended by a transfer stop request. However, the DCSRx.SP bit of the corresponding channel is set regardless of the transfer suspension interrupt (DCCRx.SIE) settings.

  Clear the interrupt request by clearing the DCSRx.SP bit of the corresponding channel.

- DMA transfer suppressing

The DMA transfer is suppressed due to the following causes.

- A DMA transfer suppress request from DSU/OCD (for debugging)
- NMI
- Peripheral interrupt

The DMA transfer is suppressed in units of blocks. If data transfer is suppressed, new data transfer does not start. Data transfer is stopped. The settings to restart data transfer vary depending on the DMA transfer suppress causes.

- DMA transfer suppressing request from DSU/OCD (for debugging)
  When the DMA transfer suppressing request by DSU/OCD is asserted, a new transfer does not start and a current transfer stops with the block unit. The acknowledge is not returned to the DMA transfer suppressing from DSU/OCD.

- DMA transfer suppressing by NMI
  If the NMIHD bit is set to "0", DMAC sets NMIH flag when user NMI occurs and suppresses DMA transfer after the current block has been transferred.

  Write "0" in the NMIH flag when you restart transfer.

- DMA transfer suppressing by peripheral interrupt
  If an interrupt having the level higher than the one specified in the DILVR register occurs, the DMA transfer is suppressed after the current block has been transferred.

  When the interrupt request is cleared and the interrupt level drops to LVL[4:0] or lower level, the DMA transfer restarts.

Table 5-7 LVL[4:0] Settings to Suppress DMA Transfer

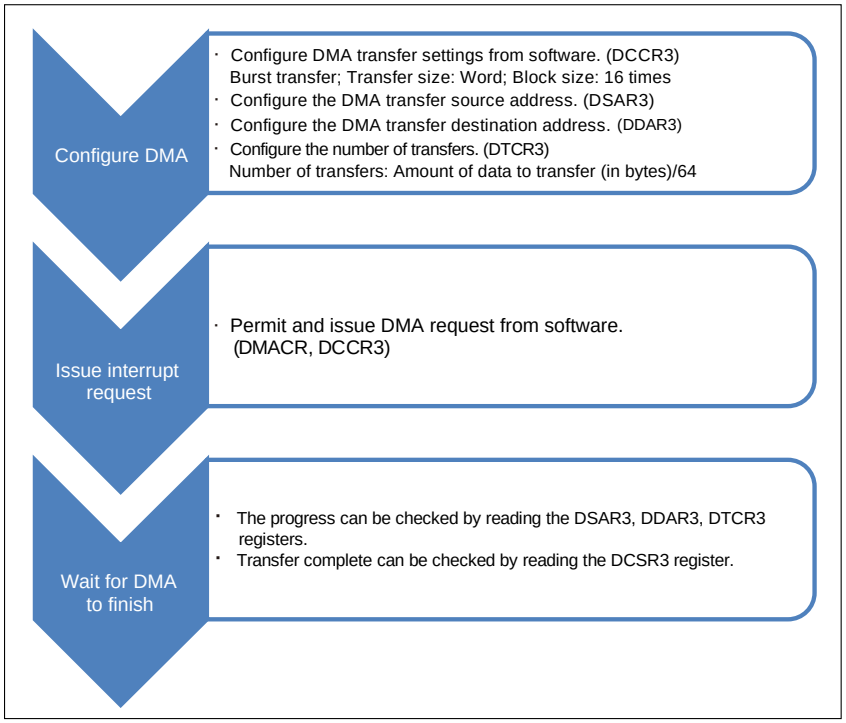
LVL[4:0]	DMA suppress control
11111	Suppresses the DMA transfer when any peripheral interrupt request is issued. (initial value)
11110	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1E_H$ is issued.
11101	Suppresses the DMA transfer when a peripheral interrupt 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.

6. DMA Usage Examples

This section explains DMA controller (DMAC) DMA usage examples.

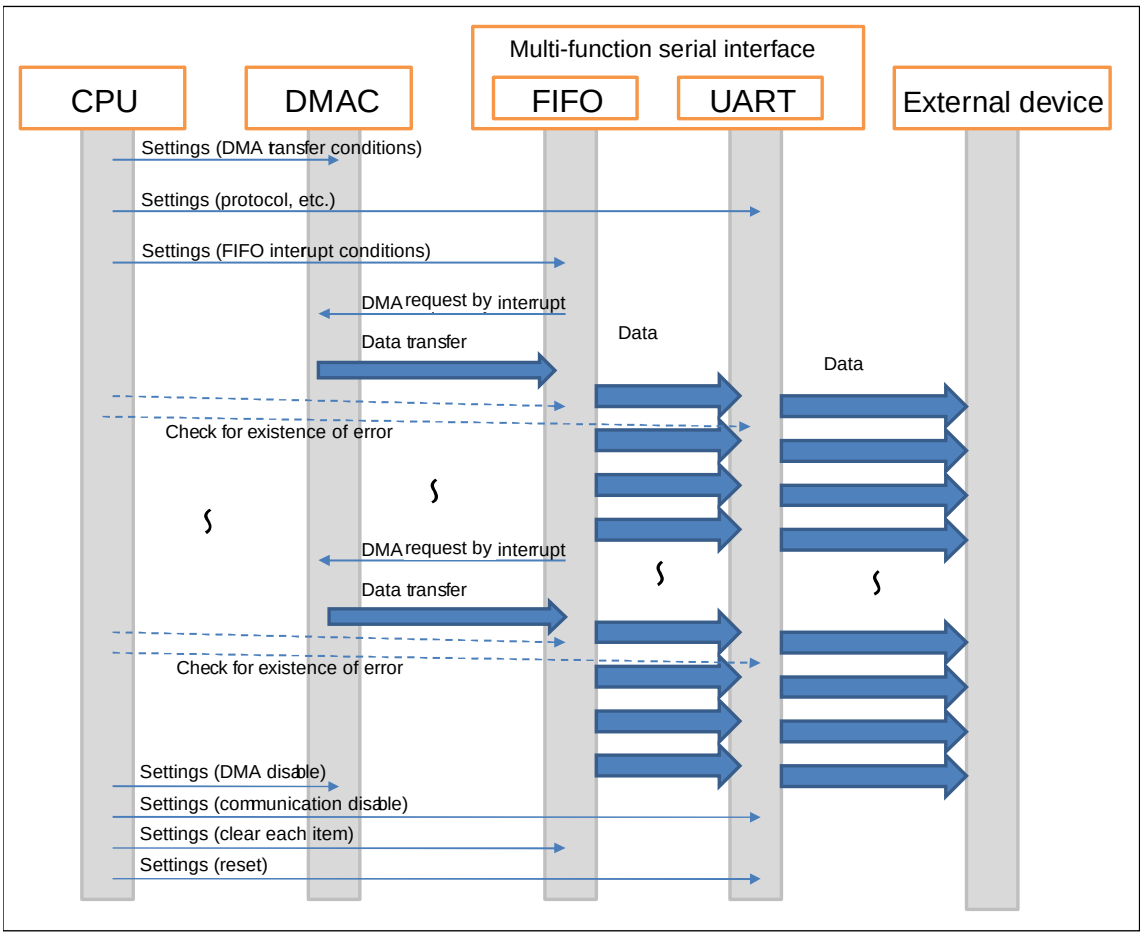
The following gives an example of memcpy instruction in every 64-byte data using the DMA. This is the simplest DMA transfer example.

Figure 6-1 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 6-2 Communication Example via the Multi-function Serial Interface That Uses DMA



Chapter 9: Generation And Clearing Of DMA Transfer Requests



This chapter explains the generation and clearing of DMA transfer requests.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : DMAREQ-1v0-91520-2-E

1. Overview

This section explains the overview of the generation and clearing of DMA transfer requests.

This product can activate DMA transfer using interrupt requests from peripheral functions. Registers used to select interrupt requests that activate DMA transfer are provided for each DMA controller (DMAC) channel. If multiple interrupt requests are assigned to one interrupt vector number, it is also necessary to specify what interrupt request flag is to be cleared by the DMA controller (DMAC).

DMA controller (DMAC) registers allow DMA transfer request generation factors (transfer request sources) to be set on interrupt requests from peripheral functions. The interrupt requests to be used can be selected by specifying the value corresponding to the interrupt vector number.

2. Features

This section explains features of the generation and clearing of DMA transfer requests.

2.1 Transfer Request Generation Setting

2.2 Interrupt Clearing Setting

2.1. Transfer Request Generation Setting

The transfer request generation setting is shown.

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.

2.2. Interrupt Clearing Setting

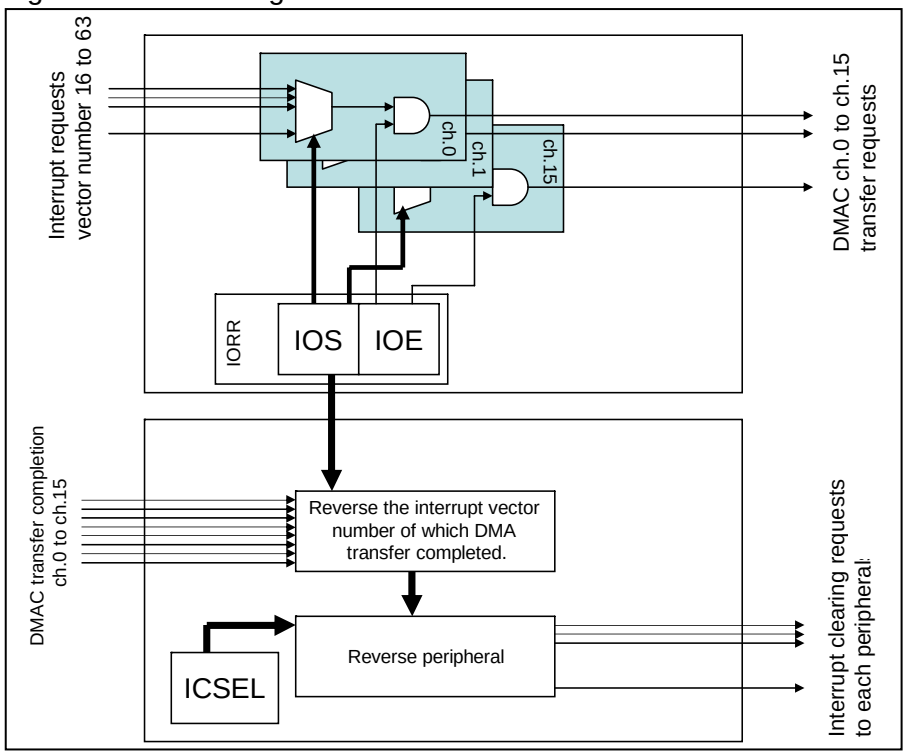
The interrupt clearing setting is shown.

After the DMA transfer ends, the interrupt source peripheral that has issued the interrupt request to be cleared is identified if the transfer request source is a vector number to which multiple interrupt source peripherals belong.

3. Configuration

This section explains the configuration of the generation and clearing of DMA transfer requests.

Figure 3-1 Block Diagram



4. Registers

This section explains registers of the generation and clearing of DMA transfer requests.

Table 4-1 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	Reserved	ICSEL5	ICSEL6	ICSEL7	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)

Address	Registers				Register function
	+0	+1	+2	+3	
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	Reserved	ICSEL13	ICSEL14	ICSEL15	DMA clear request register 13 (for vector number #52) DMA clear request register 14 (for vector number #53) DMA clear request register 15 (for vector number #54)
0x0410	ICSEL16	ICSEL17	ICSEL18	ICSEL19	DMA clear request register 16 (for vector number #55) DMA clear request register 17 (for vector number #56) DMA clear request register 18 (for vector number #57) DMA clear request register 19 (for vector number #58)
0x0414	ICSEL20	ICSEL21	ICSEL22	ICSEL23	DMA clear request register 20 (for vector number #59) DMA clear request register 21 (for vector number #60) DMA clear request register 22 (for vector number #61) DMA clear request register 23 (for vector number #45)
0x0438	ICSEL24	ICSEL25	ICSEL26	ICSEL27	DMA clear request register 24 (for vector number #49) DMA clear request register 25 (for vector number #48) DMA clear request register 26 (for vector number #50) DMA clear request register 27 (for vector number #51)
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
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

4.1. DMA Request Clear Register 0 : ICSEL0 (Interrupt Clear SELect register 0)

The bit configuration of DMA request clear register 0 is shown below.

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

■ ICSEL0 : Address 0400_H (Access : Byte, Half-word, Word)

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

[bit2 to bit0] EISEL[2:0] (External Interrupt request SELection) : Interrupt clear selection bits for external interrupts 0 to 7

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

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.

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

■ 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

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

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.

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

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							RTSEL0
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

RTSEL0	Clear target
0	Reload timer 0
1	Reload timer 1

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.

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

■ ICSEL3: Address 0403_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							RTSEL1
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit0] RTSEL1 (Reload Timer SELection) : Interrupt clear selection bit for reload timer 2/3

RTSEL1	Clear target
0	Reload timer 2
1	Reload timer 3

4.5. DMA Request Clear Register 5 : ICSEL5 (Interrupt Clear SELect register 5)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #39).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SG_RX_SEL1[2:0]		
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit2 to bit0] SG_RX_SEL1[2:0] (SG_RX SELection1) : Interrupt clear selection bits for 16-bit free-run timer 0 zero detection, compare clear, multi-function serial ch.7 transmission completion

SG_RX_SEL1[2:0]	Clear target
000	Reserved (Does not clear any interrupt)
001	Reserved (Does not clear any interrupt)
010	16-bit free-run timer 0 zero detection
011	16-bit free-run timer 0 compare clear
100	Multi-function serial ch.7 transmission completion
101 to 111	Reserved (Does not clear any interrupt)

Note:

Setting SG_RX_SEL1[2:0]= "000", "001" and "101" to "111" are prohibited. During this setting, no interrupt clear will be selected.

4.6. DMA Request Clear Register 6 : ICSEL6 (Interrupt Clear SELect register 6)

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

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

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PPGSEL0[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit3 to bit0] PPGSEL0[3:0] (PPG SELection0) : Interrupt clear selection bits for PPG0, 1, 10, 11, 20, 21, 16-bit free-run timer 1 zero detection, compare clear

PPGSEL0[3:0]	Clear target
0000	PPG0
0001	PPG1
0010	PPG10
0011	PPG11
0100	PPG20
0101	PPG21
0110	Reserved (Does not clear any interrupt)
0111	Reserved (Does not clear any interrupt)
1000	16-bit free-run timer 1 zero detection
1001	16-bit free-run timer 1 compare clear
1010 to 1111	Reserved (Does not clear any interrupt)

Note:

Setting PPGSEL0[3:0]= "0110", "0111" and "1010" to "1111" are prohibited. During this setting, no interrupt clear will be selected.

4.7. DMA Request Clear Register 7 : ICSEL7 (Interrupt Clear SELect register 7)

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

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

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PPGSEL1[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit3 to bit0] PPGSEL1[3:0] (PPG SELection1) : Interrupt clear selection bits for PPG2, 3, 12, 13, 22, 23, 16-bit free-run timer 2 zero detect, compare clear

PPGSEL1[3:0]	Clear target
0000	PPG2
0001	PPG3
0010	PPG12
0011	PPG13
0100	PPG22
0101	PPG23
0110	Reserved (Does not clear any interrupt)
0111	Reserved (Does not clear any interrupt)
1000	16-bit free-run timer 2 zero detection
1001	16-bit free-run timer 2 compare clear
1010 to 1111	Reserved (Does not clear any interrupt)

Note:

Setting PPGSEL1[3:0]= "0110", "0111" and "1010" to "1111" are prohibited. During this setting, no interrupt clear will be selected.

4.8. DMA Request Clear Register 8 : ICSEL8 (Interrupt Clear SELect register 8)

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

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

■ 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

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

4.9. DMA Request Clear Register 9 : ICSEL9 (Interrupt Clear SELect register 9)

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

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

■ 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

PPGSEL3[1:0]	Clear target
00	PPG6

PPGSEL3[1:0]	Clear target
01	PPG7
10	PPG16
11	PPG17

4.10. DMA Request Clear Register 10 : ICSEL10 (Interrupt Clear SELect register 10)

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

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

■ 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

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

4.11. DMA Request Clear Register 11 : ICSEL11 (Interrupt Clear SELect register 11)

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

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

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

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

[bit2 to bit0] PMSTSEL[2:0] (PLL, Main, Sub Timer SELection) : Interrupt clear selection for main timer / sub timer / PLL timer, multi-function serial ch.8 transmission completion, 16-bit ICU2, ICU3

PMSTSEL[2:0]	Clear target
000	Main timer
001	Sub timer
010	PLL timer
011	Multi-function serial ch.8 transmission completion
100	16-bit ICU2
101	16-bit ICU3
110	Reserved (Does not clear any interrupt)
111	Reserved (Does not clear any interrupt)

Note:

Setting PMSTSEL[2:0]= "110" and "111" are prohibited. During this setting, no interrupt clear will be selected.

4.12. DMA Request Clear Register 13 : ICSEL13 (Interrupt Clear SELect register 13)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #52).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ICUSEL0[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] ICUSEL0[1:0] : Interrupt clear selection for ICU ch.6, multi-function serial ch.10 reception completion

ICUSEL0[1:0]	Clear target
00	Reserved (Does not clear any interrupt)
01	32-bit ICU ch.6
10	Multi-function serial ch.10 reception completion
11	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL0[1:0]= "00" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.13. DMA Request Clear Register 14 : ICSEL14 (Interrupt Clear SELect register 14)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #53).

■ ICSEL14: Address $040E_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ICUSEL1[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] ICUSEL1[1:0] : Interrupt clear selection for ICU ch.7, multi-function serial ch.10 transmission completion

ICUSEL1[1:0]	Clear target
00	Reserved (Does not clear any interrupt)
01	32-bit ICU ch.7
10	Multi-function serial ch.10 transmission completion
11	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL1[1:0]= "00" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.14. DMA Request Clear Register 15 : ICSEL15 (Interrupt Clear SELect register 15)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #54).

■ ICSEL15: Address $040F_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ICUSEL2[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] ICUSEL2[1:0] : Interrupt clear selection for ICU ch.8, multi-function serial ch.11 reception completion

ICUSEL2[1:0]	Clear target
00	Reserved (Does not clear any interrupt)
01	32-bit ICU ch.8
10	Multi-function serial ch.11 reception completion
11	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL2[1:0]= "00" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.15. DMA Request Clear Register 16 : ICSEL16 (Interrupt Clear SELect register 16)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #55).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				ICUSEL3[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	R,W	R/W

[bit3 to bit0] ICUSEL3[3:0] : Interrupt clear selection for ICU ch.9, WG dead timer underflow 0, 1, 2, WG dead timer reload 0, 1, 2, WG DTTI0

ICUSEL3[3:0]	Clear target
0000	Reserved (Does not clear any interrupt)

0001	32-bit ICU ch.9
0010	WG dead timer underflow 0
0011	WG dead timer underflow 1
0100	WG dead timer underflow 2
0101	WG dead timer reload 0
0110	WG dead timer reload 1
0111	WG dead timer reload 2
1000	WG DTTI0
1001 to 1111	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL3[3:0]= "0000" is prohibited. During this setting, no interrupt clear will be selected.

4.16. DMA Request Clear Register 17 : ICSEL17 (Interrupt Clear SELect register 17)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #56).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ICUSEL4[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] ICUSEL4[1:0] : Interrupt clear selection for ICU ch.4, multi-function serial ch.11 transmission completion

ICUSEL4[1:0]	Clear target
00	32-bit ICU ch.4
01	Reserved (Does not clear any interrupt)
10	Multi-function serial ch.11 transmission completion
11	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL4[1:0]= "01" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.17. DMA Request Clear Register 18 : ICSEL18 (Interrupt Clear SELect register 18)

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

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #57).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			ICUSEL5[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

[bit4 to bit0] ICUSEL5[4:0] : Interrupt clear selection for ICU ch.5, A/D converter ch.32 to ch.47

ICUSEL5[4:0]	Clear target
00000	32-bit ICU ch.5
00001	Reserved (Does not clear any interrupt)
00010	A/D converter ch.32
00011	A/D converter ch.33
00100	A/D converter ch.34
00101	A/D converter ch.35
00110	A/D converter ch.36
00111	A/D converter ch.37
01000	A/D converter ch.38
01001	A/D converter ch.39
01010	A/D converter ch.40
01011	A/D converter ch.41
01100	A/D converter ch.42
01101	A/D converter ch.43
01110	A/D converter ch.44
01111	A/D converter ch.45
10000	A/D converter ch.46
10001	A/D converter ch.47
10010 to 11111	Reserved (Does not clear any interrupt)

Note:

Setting ICUSEL5[4:0]= "00001" and "10010" to "11111" are prohibited. During this setting, no interrupt clear will be

selected.

4.18. DMA Request Clear Register 19 : ICSEL19 (Interrupt Clear SELect register 19)

The bit configuration of DMA request clear register 19 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #58).

■ 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 OCU6, 7, 10, 11

OCUSEL0[2:0]	Clear target
000	Reserved (Does not clear any interrupt)
001	Reserved (Does not clear any interrupt)
010	32-bit OCU6
011	32-bit OCU7
100	32-bit OCU10
101	32-bit OCU11
110	Reserved (Does not clear any interrupt)
111	Reserved (Does not clear any interrupt)

Note:

Setting OCUSEL0[2:0]= "000 to 001" and "110" to "111" are prohibited. During this setting, no interrupt clear will be selected.

4.19. DMA Request Clear Register 20 : ICSEL20 (Interrupt Clear SELect register 20)

The bit configuration of DMA request clear register 20 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #59).

■ **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 OCU8, 9

OCUSEL1[2:0]	Clear target
000	Reserved (Does not clear any interrupt)
001	Reserved (Does not clear any interrupt)
010	Reserved (Does not clear any interrupt)
011	Reserved (Does not clear any interrupt)
100	32-bit OCU8
101	32-bit OCU9
110	Reserved (Does not clear any interrupt)
111	Reserved (Does not clear any interrupt)

Note:

Setting OCUSEL1[2:0]= "000" to "011" and "110" to "111" are prohibited. During this setting, no interrupt clear will be selected.

4.20. DMA Request Clear Register 21 : ICSEL21 (Interrupt Clear SELect register 21)

The bit configuration of DMA request clear register 21 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #60).

■ ICSEL21: Address 0415_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						BT_SG_SEL0[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] BT_SG_SEL0[1:0] (BT_SG Selection0) : Interrupt clear selection bits for Base Timer0 IRQ0, IRQ1

BT_SG_SEL0[1:0]	Clear target
00	Base Timer0 IRQ0
01	Base Timer0 IRQ1
10	Reserved (Does not clear any interrupt)
11	Reserved (Does not clear any interrupt)

Note:

Setting BT_SG_SEL0[1:0]= "10" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.21. DMA Request Clear Register 22 : ICSEL22 (Interrupt Clear SELect register 22)

The bit configuration of DMA request clear register 22 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #61).

■ ICSEL22: Address 0416_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						BT_SG_SEL1[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] BT_SG_SEL [1:0] (BT_SG_Selection1) : Interrupt clear selection bits for Base Timer1 IRQ0, IRQ1

BT_SG_SEL1[1:0]	Clear target
00	Base Timer1 IRQ0
01	Base Timer1 IRQ1
10	Reserved (Does not clear any interrupt)
11	Reserved (Does not clear any interrupt)

Note:

Setting BT_SG_SEL1[1:0]= "10" and "11" are prohibited. During this setting, no interrupt clear will be selected.

4.22. DMA Request Clear Register 23 : ICSEL23 (Interrupt Clear SELect register 23)

The bit configuration of DMA request clear register 23 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #45).

■ ICSEL23: Address 0417_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						MFS_SEL0[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] MFS_SEL0[1:0] (MFS_Selection) : Interrupt clear selection bits for MFS ch.8 (reception completion) / ICU0 / ICU1

MFS_SEL0[1:0]	Clear target
00	Multi-function serial ch.8 reception completion
01	16-bit ICU0
10	16-bit ICU1
11	Reserved (Does not clear any interrupt)

Note:

Setting MFS_SEL0[1:0]= "11" is prohibited. During this setting, no interrupt clear will be selected.

4.23. DMA Request Clear Register 24 : ICSEL24 (Interrupt Clear SELect register 24)

The bit configuration of DMA request clear register 24 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #49).

■ ICSEL24: Address 0438_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						MFS_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] MFS_SEL1[1:0] (MFS_Selection1) : Interrupt clear selection bits for MFS ch.9 (transmission completion) / OCU0 / OCU1

MFS_SEL1[1:0]	Clear target
00	Multi-function serial ch.9 transmission completion
01	16-bit OCU0
10	16-bit OCU1
11	Reserved (Does not clear any interrupt)

Note:

Setting MFS_SEL1[1:0]= "11" is prohibited. During this setting, no interrupt clear will be selected.

4.24. DMA Request Clear Register 25 : ICSEL25 (Interrupt Clear SELect register 25)

The bit configuration of DMA request clear register 25 is shown below.

These bits select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #48).

■ ICSEL25: Address 0439_H (Access : Byte, Half-word, Word)

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

[bit4 to bit0] AD_SEL[4:0] (AD_Selection) : Interrupt clear selection bits for ADC ch.0 to ch.31

AD_SEL[4:0]	Clear target
00000	A/D converter ch.0
00001	A/D converter ch.1
00010	A/D converter ch.2
00011	A/D converter ch.3

AD_SEL[4:0]	Clear target
00100	A/D converter ch.4
00101	A/D converter ch.5
00110	A/D converter ch.6
00111	A/D converter ch.7
01000	A/D converter ch.8
01001	A/D converter ch.9
01010	A/D converter ch.10
01011	A/D converter ch.11
01100	A/D converter ch.12
01101	A/D converter ch.13
01110	A/D converter ch.14
01111	A/D converter ch.15
10000	A/D converter ch.16
10001	A/D converter ch.17
10010	A/D converter ch.18
10011	A/D converter ch.19
10100	A/D converter ch.20
10101	A/D converter ch.21
10110	A/D converter ch.22
10111	A/D converter ch.23
11000	A/D converter ch.24
11001	A/D converter ch.25
11010	A/D converter ch.26
11011	A/D converter ch.27
11100	A/D converter ch.28
11101	A/D converter ch.29
11110	A/D converter ch.30
11111	A/D converter ch.31

4.25. DMA Request Clear Register 26 : ICSEL26 (Interrupt Clear SELect register 26)

The bit configuration of DMA request clear register 26 is shown below.

This bit selects the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #50).

■ ICSEL26: Address 043A$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							OCU_SEL2
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] OCU_SEL2 (OCU_Selection2) : Interrupt clear selection bit for OCU2 / OCU3

OCU_SEL2	Clear target
0	16-bit OCU2
1	16-bit OCU3

4.26. DMA Request Clear Register 27 : ICSEL27 (Interrupt Clear SELect register 27)

The bit configuration of DMA request clear register 27 is shown below.

This bit selects the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #51).

■ ICSEL27: Address 043B$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							OCU_SEL3
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] OCU_SEL3 (OCU_Selection3) : Interrupt clear selection bit for OCU4 / OCU5

OCU_SEL3	Clear target
0	16-bit OCU4
1	16-bit OCU5

4.27. 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 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

[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).

5. Operation

This section explains the operation of the generation and clearing of DMA transfer requests.

5.1. Configuration

5.2. Notes

5.1. Configuration

The configuration of the operation is shown.

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.

5.2. Notes

The notes are shown.

- 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)

Chapter 10: FixedVector Function



This chapter explains the FixedVector function.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

Code : FR81SFVEC-1v1-91520-2-E

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.

2. Features

This section explains the features of the FixedVector function.

- Interrupt vector on reset returned by the FixedVector function
  - MB91F522 0x0007_0024
  - MB91F523 0x0007_0024
  - MB91F524 0x0007_0024
  - MB91F525 0x0007_0024
  - MB91F526 0x0007_0024

3. Configuration

This section explains the configuration of the FixedVector function.

See "Figure 3-3" in "CHAPTER: FLASH MEMORY" for the configuration diagram.

4. Registers

This section explains the registers of the FixedVector function.

None.

5. Operation

This section explains the operation of the FixedVector function.

5.1. Operation After Reset Released

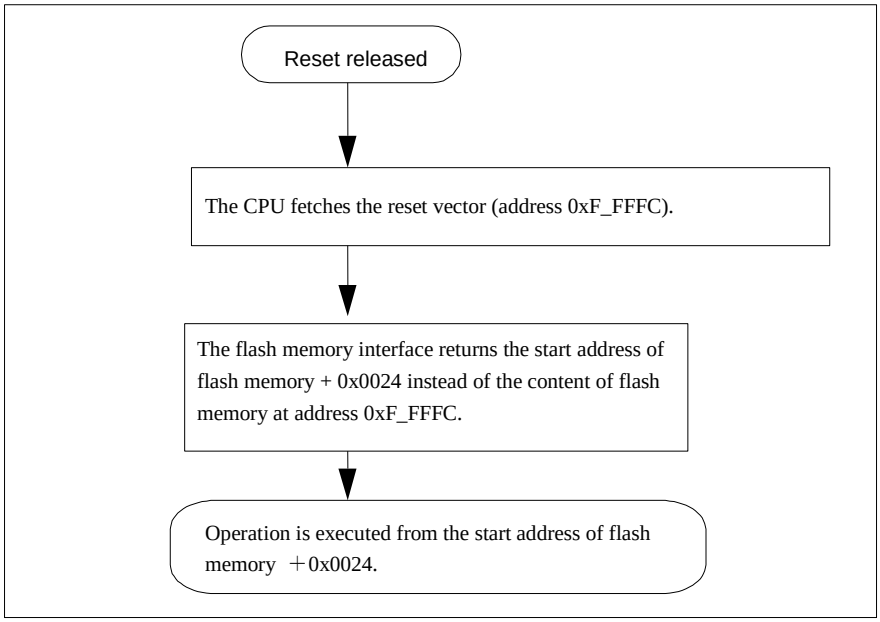
5.2. Usage

5.1. Operation After Reset Released

The operation after reset released is shown below.

In the following flow, the start address of flash memory + 0x0024 is returned instead of the content of 0xF_FFFC in flash memory when the reset is released.

Figure 5-1 Operation Flow after Reset



5.2. Usage

The usage is shown below.

After the reset is released, this series executes from the start address of flash memory + 0x0024 instead of the value written at address 0x000F_FFFC.

6. Notes

This section explains the notes of the FixedVector function.

During reads from addresses 0x000F_FFFC to 0x000F_FFFF other than reset vector fetch (Example: the call destination when INT #00H is executed while TBR is its initial value (=0x000F_FC00)), the content of flash memory at the addresses 0x000F_FFFC to 0x000F_FFFF is returned.

Chapter 11: I/O Ports

This chapter explains the I/O ports.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : IO-1v0-91520-2-E

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.

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 2-1 Diagram of I/O Multiplexing, I/O Relocation



- **Key code function**
  This function is for error writing protection. If writing is not executed to the key code register (KEYCDR) according to the specified method, writing to the target register will become invalid. Also, word access for the target register cannot be executed. The following are the key code target registers.

  · Data direction register
  · Port function register
  · Extended port function register
  · Port pull-up/down enable register
  · Analog input control register
  · DA control register
  · Port input enable register

3. Configuration

This section explains the configuration of the I/O ports.

No configuration diagram is provided.

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 19
0x0004	PDR04	PDR05	PDR06	PDR07	
0x0008	PDR08	PDR09	PDR10	PDR11	
0x000C	PDR12	PDR13	PDR14	PDR15	
0x0010	Reserved	Reserved	Reserved	Reserved	
0x0014	Reserved	Reserved	Reserved	Reserved	
0x0018	PDR16	PDR17	PDR18	PDR19	
0x0E00	DDR00	DDR01	DDR02	DDR03	Data direction register 00 to 19 (Key code target registers)
0x0E04	DDR04	DDR05	DDR06	DDR07	
0x0E08	DDR08	DDR09	DDR10	DDR11	
0x0E0C	DDR12	DDR13	DDR14	DDR15	
0x0E10	Reserved	Reserved	Reserved	Reserved	
0x0E14	Reserved	Reserved	Reserved	Reserved	
0x0E18	DDR16	DDR17	DDR18	DDR19	
0x0E20	PFR00	PFR01	PFR02	PFR03	Port function register 00 to 19 (Key code target registers)
0x0E24	PFR04	PFR05	PFR06	PFR07	
0x0E28	PFR08	PFR09	PFR10	PFR11	
0x0E2C	PFR12	PFR13	PFR14	PFR15	
0x0E30	Reserved	Reserved	Reserved	Reserved	
0x0E34	Reserved	Reserved	Reserved	Reserved	
0x0E38	PFR16	PFR17	PFR18	PFR19	

Address	Registers				Register function
	+0	+1	+2	+3	
0x0E40	PDDR00	PDDR01	PDDR02	PDDR03	Input data direct read register 00 to 19
0x0E44	PDDR04	PDDR05	PDDR06	PDDR07	
0x0E48	PDDR08	PDDR09	PDDR10	PDDR11	
0x0E4C	PDDR12	PDDR13	PDDR14	PDDR15	
0x0E50	Reserved	Reserved	Reserved	Reserved	
0x0E54	Reserved	Reserved	Reserved	Reserved	
0x0E58	PDDR16	PDDR17	PDDR18	PDDR19	
0x0E60	EPFR00	EPFR01	EPFR02	EPFR03	Extended port function register 00 to 63 (Key code target registers)
0x0E64	EPFR04	EPFR05	EPFR06	EPFR07	
0x0E68	EPFR08	EPFR09	EPFR10	EPFR11	
0x0E6C	EPFR12	EPFR13	EPFR14	EPFR15	
0x0E70	Reserved	Reserved	Reserved	Reserved	
0x0E74	Reserved	Reserved	Reserved	Reserved	
0x0E78	Reserved	Reserved	EPFR26	EPFR27	Extended port function register 00 to 63 (Key code target registers)
0x0E7C	EPFR28	EPFR29	Reserved	Reserved	
0x0E80	Reserved	EPFR33	EPFR34	EPFR35	
0x0E84	EPFR36	Reserved	Reserved	Reserved	
0x0E88	Reserved	Reserved	EPFR42	EPFR43	
0x0E8C	EPFR44	EPFR45	Reserved	Reserved	
0x0E90	Reserved	Reserved	Reserved	Reserved	
0x0E94	Reserved	Reserved	Reserved	Reserved	
0x0E98	EPFR56	EPFR57	EPFR58	EPFR59	
0x0E9C	EPFR60	EPFR61	EPFR62	EPFR63	
0x0EA0	Reserved	Reserved	Reserved	Reserved	Reserved
0x0EA4	Reserved	Reserved	Reserved	Reserved	
0x0EA8	Reserved	Reserved	Reserved	Reserved	
0x0EAC	Reserved	Reserved	Reserved	Reserved	
0x0EB0	Reserved	Reserved	Reserved	Reserved	
0x0EB4	Reserved	Reserved	Reserved	Reserved	

Address	Registers				Register function
	+0	+1	+2	+3	
0x0EC0	PPER00	PPER01	PPER02	PPER03	Port pull-up/down enable register 00 to 19 (Key code target registers)
0x0EC4	PPER04	PPER05	PPER06	PPER07	
0x0EC8	PPER08	PPER09	PPER10	PPER11	
0x0ECC	PPER12	PPER13	PPER14	PPER15	
0x0ED0	Reserved	Reserved	Reserved	Reserved	
0x0ED4	Reserved	Reserved	Reserved	Reserved	
0x0ED8	PPER16	PPER17	PPER18	PPER19	
0x0EE0	Reserved	Reserved	Reserved	Reserved	Reserved
0x0EE4	Reserved	Reserved	Reserved	Reserved	
0x0EE8	Reserved	Reserved	Reserved	Reserved	
0x0EEC	Reserved	Reserved	Reserved	Reserved	
0x0EF0	Reserved	Reserved	Reserved	Reserved	
0x0EF4	Reserved	Reserved	Reserved	Reserved	
0x0F00	Reserved	Reserved	Reserved	Reserved	
0x0F04	Reserved	Reserved	Reserved	Reserved	
0x0F08	Reserved	Reserved	Reserved	Reserved	
0x0F0C	Reserved	Reserved	Reserved	Reserved	
0x0F10	Reserved	Reserved	Reserved	Reserved	
0x0F14	Reserved	Reserved	Reserved	Reserved	
0x0F18	Reserved	Reserved	Reserved	Reserved	
0x0F1C	Reserved	Reserved	Reserved	Reserved	
0x0F20	Reserved	Reserved	Reserved	Reserved	
0x0F24	Reserved	Reserved	Reserved	Reserved	
0x0F28	Reserved	Reserved	Reserved	Reserved	Reserved
0x0F2C	Reserved	Reserved	Reserved	Reserved	
0x0F34	Reserved	Reserved	Reserved	Reserved	
0x0F38	Reserved	Reserved	Reserved	Reserved	
0x0F40	PORTEN	Reserved	Reserved	Reserved	Port input enable register (Key code target registers)
0x0F44	KEYCDR				Key code register

Address	Registers				Register function
	+0	+1	+2	+3	
0x01B8	EPFR64	EPFR65	EPFR66	EPFR67	Extended port function register 64 to 88(Key code target registers)
0x01BC	EPFR68	EPFR69	EPFR70	EPFR71	
0x01C0	EPFR72	EPFR73	EPFR74	EPFR75	
0x01C4	EPFR76	EPFR77	EPFR78	EPFR79	
0x01C8	EPFR80	EPFR81	EPFR82	EPFR83	
0x01CC	EPFR84	EPFR85	EPFR86	EPFR87	
0x01D0	EPFR88	Reserved	Reserved	Reserved	

4.1. Port Data Register 00 to 19 : PDR00 to 19 (Port Data Register 00 to 19)

The bit configuration of port data register 00 to 19 is shown below.

These registers hold the output levels of the pins corresponding to individual ports that are in output mode.

■ PDR00 to PDR19 : 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 (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.P[7], PDR14.P[7:5, 1:0], PDR15.P[7:6] are reserved bits. Both writing to and reading from these bits have no effect.
PDR13.P[6:5] are reserved bits in the dual clock products. Both writing to and reading from these bits have no effect.
Some devices of the MB91520 series have ports missing. For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". As for those bits allocated in the missing ports, both writing and reading have no effect.

4.2. Data Direction Register 00 to 19 : DDR00 to 19 (Data Direction Register 00 to 19)

The bit configuration of data direction register 00 to 19 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 to DDR19 are key code target registers.

■ DDR00 to DDR19 : 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 (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

DDR13.P[7], DDR14.P[7:5, 1:0], DDR15.P[7:6] are reserved bits. Both writing to and reading from these bits have no effect.
DDR13.P[6:5] are reserved bits in the dual clock products. Both writing to and reading from these bits have no effect.
Some devices of the MB91520 series have ports missing. For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". As for those bits allocated in the missing ports, both writing and reading have no effect.

4.3. Port Function Register 00 to 19 : PFR00 to 19 (Port Function Register 00 to 19)

The bit configuration of port function register 00 to 19 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 must be set for the port function.
PFR00 to PFR19 are key code target registers.

■ PFR00 to PFR19 : Address $0E20_H$, $0E21_H$, ·· (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	*	*	*	*	*	*	*	*
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*:Initial value of the each bits can be referred " I/O Map" in "APPENDIX".

[bit7 to bit0] P (Port) : Port function selection bits
These bits 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)

PFR13.P[7], PFR14.P[7:5, 1:0], PFR15.P[7:6] are reserved bits. Both writing to and reading from these bits have no effect.
PFR13.P[6:5] are reserved bits in the dual clock products. Both writing to and reading from these bits have no effect.
Some devices of the MB91520 series have ports missing. For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". As for those bits allocated in the missing ports, both writing and reading have no effect.

4.4. Input Data Direct Register 00 to 19 : PDDR00 to 19 (Port Data Direct Register 00 to 19)

The bit configuration of input data direct register 00 to 19 is shown below.

These registers can always show the voltage levels of individual external pins. These registers can always be read unconditionally.

■ PDDR00 to PDDR19 : 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 (Port) : Read bits
The value of the external pins can be read.
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	"L" level
1	"H" level

PDDR13.P[7], PDDR14.P[7:5, 1:0], PDDR15.P[7:6] are reserved bits. Both writing to and reading from these bits have no effect.
PDDR13.P[6:5] are reserved bits in the dual clock products. Both writing to and reading from these bits have no effect.
Some devices of the MB91520 series have ports missing. For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". As for those bits allocated in the missing ports, both writing and reading have no effect.

4.5. Port Pull-up/down Enable Register 00 to 19 : PPER00 to 19 (Port Pull-up/down Enable Register 00 to 19)

The bit configuration of port pull-up/down enable register 00 to 19 is shown below.

These registers enable pull-up or pull-down of each port. These registers are functioned for input condition pins only.
PPER00 to PPER19 are key code target registers.

■ PPER00 to PPER19 : 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 (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

This series does not have pull-down function.
See "List of Pin Functions" and "I/O Circuit Types" of "CHAPTER: OVERVIEW" for the existence of pull-up function.

PPER13.P[7], PPER14.P[7:5, 1:0], PPER15.P[7:6] are reserved bits. Both writing to and reading from these bits have no effect.
PPER13.P[6:5] are reserved bits in the dual clock products. Both writing to and reading from these bits have no effect.
Some devices of the MB91520 series have ports missing. For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". As for those bits allocated in the missing ports, both writing and reading have no effect.

4.6. Extended Port Function Register 00 to 88 : EPFR00 to 88 (Extended Port Function Register 00 to 88)

The bit configuration of extended port function register 00 to 88 is show below.

These registers control switching between the peripheral and the external bus, I/O relocation and I/O multiplexing. 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).

EPFR00 to EPFR88 are key code target registers.

4.6.1. Extended Port Function Register 00, 01, 56 : EPFR00, EPFR01, EPFR56

The bit configuration of extended port function register 00, 01, 56 is shown.

These registers select input pins for input capture. (I/O relocation)

■ **EPFR00 : Address $0E60_H$ (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

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.

■ EPFR01 : Address $0E61_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ICU7E	Reserved	ICU6E	Reserved	ICU5E	ICU4E[1:0]	
Initial value	1	0	1	0	1	0	0	0
Attribute	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W	R/W	R/W

ICU4E[1:0]: Input capture ch.4 input pin selection.
ICU5E: Input capture ch.5 input pin selection.
ICU6E: Input capture ch.6 input pin selection.
ICU7E: Input capture ch.7 input pin selection.

■ EPFR56 : Address $0E98_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					ICU9E	Reserved	ICU8E
Initial value	1	1	1	1	1	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R1,WX	R/W

ICU8E: Input capture ch.8 input pin selection.
ICU9E: Input capture ch.9 input pin selection.

ICUnE[1:0] (n=0 to 3)	Operation
00	Input from the ICUn_0 pin
01	Input from the ICUn_1pin
10	Input from the ICUn_2 pin
11	Input from the ICUn_3 pin

ICUnE[1:0] (n=4)	Operation
00	Input from the ICUn_0 pin
01	Input from the ICUn_1pin
10	Input from the ICUn_2 pin
11	Reserved (Input from the ICUn_2 pin)

ICUnE (n=5 to 9)	Operation
0	Input from the ICUn_0 pin
1	Input from the ICUn_1pin

EPFR01 [bit7, bit5, bit3] Reserved
EPFR56 [bit7 to bit3, bit1] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.2. Extended Port Function Register 02 to 05, 57 to 60 : EPFR02 to 05, 57 to 60

The bit configuration of extended port function register 02 to 05, 57 to 60 is shown.

These registers enable reload timer output and select output/input pins. (I/O relocation and I/O multiplexing)

■ EPFR02 : Address $0E62_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT0E[1:0]		TIN0E[1:0]	
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

TOT0E[1:0] Reload timer ch.0 TOT output pin selection
TIN0E[1:0] Reload timer ch.0 TIN input pin selection

■ EPFR03 : Address $0E63_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT1E[2:0]			Reserved	TIN1E
Initial value	1	1	1	0	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R1,WX	R/W

TOT1E[2:0] Reload timer ch.1 TOT output pin selection
TIN1E Reload timer ch.1 TIN input pin selection

■ EPFR04 : Address $0E64_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT2E[1:0]		Reserved	TIN2E
Initial value	1	1	1	1	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX	R/W

TOT2E[1:0] Reload timer ch.2 TOT output pin selection
TIN2E Reload timer ch.2 TIN input pin selection

■ EPFR05 : Address $0E65_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT3E[1:0]		TIN3E[1:0]	
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

TOT3E[1:0] Reload timer ch.3 TOT output pin selection
TIN3E[1:0] Reload timer ch.3 TIN input pin selection

■ EPFR57 : Address $0E99_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT4E[1:0]		Reserved	TIN4E
Initial value	1	1	1	1	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX	R/W

TOT4E[1:0] Reload timer ch.4 TOT output pin selection
TIN4E Reload timer ch.4 TIN input pin selection

■ EPFR58 : Address $0E9A_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT5E[1:0]		Reserved	TIN5E
Initial value	1	1	1	1	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX	R/W

TOT5E[1:0] Reload timer ch.5 TOT output pin selection
TIN5E Reload timer ch.5 TIN input pin selection

■ EPFR59 : Address $0E9B_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT6E[1:0]		Reserved	TIN6E
Initial value	1	1	1	1	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX	R/W

TOT6E[1:0] Reload timer ch.6 TOT output pin selection
TIN6E Reload timer ch.6 TIN input pin selection

■ EPFR60 : Address $0E9C_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			Reserved	TOT7E[1:0]		Reserved	TIN7E
Initial value	1	1	1	1	0	0	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX	R/W

TOT7E[1:0] Reload timer ch.7 TOT output pin selection
TIN7E Reload timer ch.7 TIN input pin selection

TOTnE[2:0] (n=1)	Operation
000	No output
xx1	Output from the TOTn_0 pin
x1x	Output from the TOTn_1 pin
1xx	Output from the TOTn_2 pin

TOTnE[1:0] (n=0, 2 to 7)	Operation
00	No output
01	Output from the TOTn_0 pin
10	Output from the TOTn_1 pin
11	Reserved (Output from the TOTn_1 pin)

TINnE[1:0] (n=0, 3)	Operation
00	Input from the TINn_0 pin
01	Input from the TINn_1 pin
10	Input from the TINn_2 pin
11	Reserved (Input from the TINn_2 pin)

TINnE (n=1, 2, 4 to 7)	Operation
0	Input from the TINn_0 pin
1	Input from the TINn_1 pin

EPFR02 [bit7 to bit4] Reserved
EPFR03 [bit7 to bit5, bit1] Reserved
EPFR04 [bit7 to bit4, bit1] Reserved
EPFR05 [bit7 to bit4] Reserved
EPFR57 [bit7 to bit4, bit1] Reserved
EPFR58 [bit7 to bit4, bit1] Reserved
EPFR59 [bit7 to bit4, bit1] Reserved
EPFR60 [bit7 to bit4, bit1] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.3. Extended Port Function Register 06 to 09, 33 to 36, 61 to 64 : EPFR06 to 09, 33 to 36, 61 to 64

The bit configuration of extended port function register 06 to 09, 33 to 36, 61 to 64 is shown.

These registers enable multi-function serial interface output. (I/O relocation and I/O multiplexing)

The relocation of I^2C of ch.3 to ch.8, ch.10, and ch.11 doesn't correspond. Please set the register so that the relocation of _0 is selected when you select I^2C with ch.3 to ch.8, ch.10, and ch.11.

■ EPFR35 : Address $0E83_H$ (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

SOT0E[1:0] Multi-function serial interface ch.0 SOT output 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

■ EPFR36 : Address $0E84_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SOT1E	SCK1E[1:0]		Reserved
Initial value	1	1	1	1	0	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R1,WX

SOT1E Multi-function serial interface ch.1 SOT output pin selection
SCK1E[1:0] Multi-function serial interface ch.1 SCK output/input pin selection

■ EPFR06 : Address $0E66_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SOT2E[1:0]		SCK2E	Reserved
Initial value	1	1	1	1	0	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R1,WX

SOT2E[1:0] Multi-function serial interface ch.2 SOT output pin selection
SCK2E Multi-function serial interface ch.2 SCK output/input pin selection

■ EPFR07 : Address 0E67$_H$ (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

SOT3E[1:0] Multi-function serial interface ch.3 SOT output pin selection
SCK3E[1:0] Multi-function serial interface ch.3 SCK output/input pin selection
SIN3E Multi-function serial interface ch.3 SIN input pin selection

■ EPFR08 : Address 0E68$_H$ (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

SOT4E[1:0] Multi-function serial interface ch.4 SOT output pin selection
SCK4E[1:0] Multi-function serial interface ch.4 SCK output/input pin selection
SIN4E Multi-function serial interface ch.4 SIN input pin selection

■ EPFR09 : Address 0E69$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT5E	SCK5E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT5E Multi-function serial interface ch.5 SOT output pin selection
SCK5E Multi-function serial interface ch.5 SCK output/input pin selection

■ EPFR33 : Address 0E81$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT6E	SCK6E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT6E Multi-function serial interface ch.6 SOT output pin selection
SCK6E Multi-function serial interface ch.6 SCK output/input pin selection

■ EPFR34 : Address $0E82_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT7E	SCK7E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT7E Multi-function serial interface ch.7 SOT output pin selection
SCK7E Multi-function serial interface ch.7 SCK output/input pin selection

■ EPFR61 : Address $0E9D_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT8E	SCK8E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT8E Multi-function serial interface ch.8 SOT output pin selection
SCK8E Multi-function serial interface ch.8 SCK output/input pin selection

■ EPFR62 : Address $0E9E_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT9E	SCK9E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT9E Multi-function serial interface ch.9 SOT output pin selection
SCK9E Multi-function serial interface ch.9 SCK output/input pin selection

■ EPFR63 : Address $0E9F_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT10E[1:0]		SCK10E[1:0]		Reserved
Initial value	1	1	1	0	0	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R1,WX

SOT10E[1:0] Multi-function serial interface ch.10 SOT output pin selection
SCK10E[1:0] Multi-function serial interface ch.10 SCK output/input pin selection

■ EPFR64 : Address $01B8_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SOT11E	SCK11E	Reserved
Initial value	1	1	1	1	1	0	0	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R1,WX

SOT11E Multi-function serial interface ch.11 SOT output pin selection
SCK11E Multi-function serial interface ch.11 SCK output/input pin selection

SOT0, 2, 10 pin selection

SOTnE[1:0] (n=0, 2, 10)	Operation
00	No output
01	Output from the SOTn_0 pin
1x	Output from the SOTn_1 pin

SOT1, 5 to 9, 11 pin selection

SOTnE (n=1, 5 to 9, 11)	Operation
0	No output
1	Output from the SOTn_0 pin

SOT3, 4 pin selection

SOTnE[1:0] (n=3, 4)	Operation
00	No output
01	Output from the SOTn_0 pin
10	Output from the SOTn_1 pin
11	Output from the SOTn_2 pin

SCK0, 1, 10 pin selection

SCKnE[1:0] (n=0, 1, 10)	Operation
00	Non input/output from the SCKn_0
01	Input from the SCKn_0 / output from the SCKn
10	Input from the SCKn_1 / output from the SCKn_1
11	Reserved (Input from the SCKn_1 / output from the SCKn_1)

SCK2, 5 to 9, 11 pin selection

SCKnE (n=2, 5 to 9, 11)	Operation
0	Non input/output from the SCKn_0
1	Input from the SCKn_0 / output from the SCKn

SCK3, 4 pin selection

SCKnE[1:0] (n=3, 4)	Operation
00	Non input/output from the SCKn_0
01	Input from the SCKn_0 / output from the SCKn
10	Input from the SCKn_1 / output from the SCKn_1
11	Input from the SCKn_2 / output from the SCKn_2

SIN0, 3, 4 pin selection

SINnE (n=0, 3, 4)	Operation
0	Input from the SINn_0 pin
1	Input from the SINn_1 pin

EPFR35 [bit7 to bit5] Reserved
EPFR36 [bit7 to bit4, bit0] Reserved
EPFR06 [bit7 to bit4, bit0] Reserved
EPFR07 [bit7 to bit5] Reserved
EPFR08 [bit7 to bit5] Reserved
EPFR09 [bit7 to bit3, bit0] Reserved
EPFR33 [bit7 to bit3, bit0] Reserved
EPFR34 [bit7 to bit3, bit0] Reserved
EPFR61 [bit7 to bit3, bit0] Reserved
EPFR62 [bit7 to bit3, bit0] Reserved
EPFR63 [bit7 to bit5, bit0] Reserved
EPFR64 [bit7 to bit3, bit0] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.4. Extended Port Function Register 10 to 15, 45, 71 to 78 : EPFR10 to 15, 45, 71 to 78

The bit configuration of extended port function register 10 to 15, 45, 71 to 78 is shown.

These registers enable PPG output and select output pins. (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[1:0]		PPG0E[1:0]	
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

PPG0E[1:0] PPG ch.0 output pin selection
PPG1E[1:0] PPG ch.1 output pin selection

■ EPFR11 : Address 0E6B$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PPG3E[1:0]		PPG2E[1:0]	
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

PPG2E[1:0] PPG ch.2 output pin selection
PPG3E[1:0] PPG ch.3 output pin selection

■ EPFR12 : Address 0E6C$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PPG5E[1:0]		PPG4E[1:0]	
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

PPG4E[1:0] PPG ch.4 output pin selection
PPG5E[1:0] PPG ch.5 output pin selection

■ EPFR13 : Address 0E6D$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPG7E	PPG6E
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

PPG6E PPG ch.6 output pin selection
PPG7E PPG ch.7 output pin selection

■ EPFR14 : Address 0E6E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPG9E	PPG8E
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

PPG8E PPG ch.8 output pin selection
PPG9E PPG ch.9 output pin selection

■ EPFR15 : Address 0E6F$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PPG12E	PPG11E	PPG10E
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

PPG10E PPG ch.10 output pin selection
PPG11E PPG ch.11 output pin selection
PPG12E PPG ch.12 output pin selection

■ EPFR45 : Address 0E8D$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PPG17E[1:0]		PPG16E[1:0]		PPG15E	PPG14E	PPG13E
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

PPG13E PPG ch.13 output pin selection
PPG14E PPG ch.14 output pin selection
PPG15E PPG ch.15 output pin selection
PPG16E[1:0] PPG ch.16 output pin selection
PPG17E[1:0] PPG ch.17 output pin selection

■ EPFR71 : Address 01BF$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PPG21E	Reserved	PPG20E	Reserved	PPG19E	Reserved	PPG18E
Initial value	1	0	1	0	1	0	1	0
Attribute	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W

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

■ EPFR72 : Address 01C0$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG25E[1:0]		PPG24E[1:0]		PPG23E[1:0]		Reserved	PPG22E
Initial value	0	0	0	0	0	0	1	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R1,WX	R/W

PPG22E PPG ch.22 output pin selection
PPG23E[1:0] PPG ch.23 output pin selection
PPG24E[1:0] PPG ch.24 output pin selection
PPG25E[1:0] PPG ch.25 output pin selection

■ EPFR73 : Address 01C1$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG29E[1:0]		PPG28E[1:0]		PPG27E[1:0]		PPG26E[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

PPG26E[1:0] PPG ch.26 output pin selection
PPG27E[1:0] PPG ch.27 output pin selection
PPG28E[1:0] PPG ch.28 output pin selection
PPG29E[1:0] PPG ch.29 output pin selection

■ EPFR74 : Address 01C2$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG33E[1:0]		PPG32E[1:0]		PPG31E[1:0]		PPG30E[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

PPG30E[1:0] PPG ch.30 output pin selection
PPG31E[1:0] PPG ch.31 output pin selection
PPG32E[1:0] PPG ch.32 output pin selection
PPG33E[1:0] PPG ch.33 output pin selection

■ EPFR75 : Address 01C3$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG37E[1:0]		PPG36E[1:0]		PPG35E[1:0]		PPG34E[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

PPG34E[1:0] PPG ch.34 output pin selection
PPG35E[1:0] PPG ch.35 output pin selection
PPG36E[1:0] PPG ch.36 output pin selection
PPG37E[1:0] PPG ch.37 output pin selection

■ EPFR76 : Address 01C4$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG41E[1:0]		PPG40E[1:0]		PPG39E[1:0]		PPG38E[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

PPG38E[1:0] PPG ch.38 output pin selection
PPG39E[1:0] PPG ch.39 output pin selection
PPG40E[1:0] PPG ch.40 output pin selection
PPG41E[1:0] PPG ch.41 output pin selection

■ EPFR77 : Address $01C5_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG45E	PPG44E[1:0]		PPG43E[1:0]		PPG42E
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

PPG42E PPG ch.42 output pin selection
PPG43E[1:0] PPG ch.43 output pin selection
PPG44E[1:0] PPG ch.44 output pin selection
PPG45E PPG ch.45 output pin selection

■ EPFR78 : Address $01C6_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPG47E	PPG46E
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

PPG46E PPG ch.46 output pin selection
PPG47E PPG ch.47 output pin selection

PPGnE[1:0] (n=0 to 5, 16, 17, 23 to 41, 43, 44)	Operation
00	No output
01	Output from the PPGn_0 pin
1x	Output from the PPGn_1 pin

PPGnE (n=6 to 15, 18 to 22, 42, 45 to 47)	Operation
0	No output
1	Output from the PPGn_0 pin

EPFR10 [bit7 to bit4] Reserved
EPFR11 [bit7 to bit4] Reserved
EPFR12 [bit7 to bit4] Reserved
EPFR13 [bit7 to bit2] Reserved
EPFR14 [bit7 to bit2] Reserved
EPFR15 [bit7 to bit3] Reserved
EPFR45 [bit7] Reserved
EPFR71 [bit7, bit5, bit3, bit1] Reserved
EPFR72 [bit1] Reserved
EPFR77 [bit7, bit6] Reserved
EPFR78 [bit7 to bit2] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.5. Extended Port Function Register 79, 80 : EPFR79, 80

The bit configuration of extended port function register 79, 80 is shown.

These registers select PPG trigger input pins.

■ EPFR79 : Address $01C7_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TRG5E[1:0]		TRG4E	TRG3E	TRG2E	TRG1E	TRG0E[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

TRG0E[1:0] : PPG trigger ch.0 input pin selection
TRG1E: PPG trigger ch.1 input pin selection
TRG2E: PPG trigger ch.2 input pin selection
TRG3E: PPG trigger ch.3 input pin selection
TRG4E: PPG trigger ch.4 input pin selection
TRG5E[1:0]: PPG trigger ch.5 input pin selection

■ EPFR80 : Address $01C8_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TRG9E	TRG8E	TRG7E	TRG6E[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

TRG6E[1:0]: PPG trigger ch.6 input pin select
TRG7E: PPG trigger ch.7 input pin select
TRG8E: PPG trigger ch.8 input pin select
TRG9E: PPG trigger ch.9 input pin select

TRGnE[1:0] (n=0, 5, 6)	Operation
00	Input from the TRGn_0 pin (Initial value)
01	Input from the TRGn_1 pin
1x	Input from the TRGn_2 pin

TRGnE (n=1 to 4, 7 to 9)	Operation
0	Input from the TRGn_0 pin (Initial value)
1	Input from the TRGn_1 pin

EPFR80 [bit7 to bit5] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.6. Extended Port Function Register 86 : EPFR86

The bit configuration of extended port function register 86 is shown.

This register enables CAN output and selects input pins. (I/O relocation and I/O multiplexing)

■ EPFR86 : Address $01CE_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			RX1E	TX2E	TX1E	TX0E[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

TXnE (n=0 to 2) : CAN channel n transmission data output enabled

TXnE (n=1, 2)	Operation
0	CAN channel n output disabled (Initial value)
1	CAN channel n output enabled

TX0E[1:0]	Operation
00	CAN channel 0 output disabled (Initial value)
01	CAN channel TX0(128) output enabled
1x	CAN channel TX0_1 output enabled

RX1E CAN channel 1 reception data input pin select

RX1E	Operation
0	CAN channel RX1(64) input enabled (Initial value)
1	CAN channel RX1_1 input enabled

EPFR86 [bit7 to bit5] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.7. Extended Port Function Register 26 : EPFR26

The bit configuration of extended port function register 26 is shown.

This register enables Base Timer output and selects output and input pins. (I/O relocation and I/O multiplexing)

■ EPFR26 : Address 0E7A$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TIB1E[1:0]		TIB0E[1:0]		TIA1E[1:0]		TIA0E[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

TIBnE[1:0] (n=0, 1) : Base Timer TIOBn input pin select

TIBnE[1:0] (n=0, 1)	Operation
x0	Input from the base timer TIOBn_0 (Initial value)
x1	Input from the base timer TIOBn_1

TIA0E[1:0]: Base Timer TIOA0 output pin select

TIA0E[1:0]	Operation
00	Base timers TIOA0_0, TIOA0_1 output disabled (Initial value)
01	Base timer TIOA0_0 output enabled
1x	Base timer TIOA0_1 output enabled

TIA1E[1:0]: Base Timer TIOA1 output/input pin select

TIA1E[1:0]	Operation
00	Base timers TIOA1_0, TIOA1_1 output disabled, Input from the base timer TIOA1_0 (Initial value)
01	Base timer TIOA1_0 output enabled, Input from the base timer TIOA1_0
1x	Base timer TIOA1_1 output enabled, Input from the base timer TIOA1_1

4.6.8. Extended Port Function Register 27 : EPFR27

The bit configuration of extended port function register 27 is shown.

This register enables real time clock output.

■ EPFR27 : Address $0E7B_H$ (Access : Byte, Half-word, Word)

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

[bit4] WOTE Real time clock over flow output enable

WOTE	Operation
0	Real time clock over flow output disabled (Initial value)
1	Real time clock over flow output enabled

EPFR27 [bit7 to bit5, bit3 to bit0] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.9. Extended Port Function Register 28 : EPFR28

The bit configuration of extended port function register 28 is shown.

This register enables 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		FRCK5E	FRCK4E	FRCK3E	Reserved	FRCK1E	Reserved
Initial value	1	1	0	0	0	1	0	1
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R1,WX	R/W	R1,WX

[bit1] FRCK1E : Free-run timer ch.1 clock input selection
[bit3] FRCK3E : Free-run timer ch.3 clock input selection
[bit4] FRCK4E : Free-run timer ch.4 clock input selection
[bit5] FRCK5E : Free-run timer ch.5 clock input selection

FRCKnE (n=1,3 to 5)	Operation
0	Input from the FRCKn_0 (Initial value)
1	Input from the FRCKn_1

EPFR28 [bit7, bit6, bit2, bit0] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.10. Extended Port Function Register 29, 81, 82 : EPFR29, 81, 82

The bit configuration of extended port function register 29, 81, 82 is shown.

These registers 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
	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

■ **EPFR81 : Address $01C9_H$ (Access : Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OCU7E[1:0]		OCU6E[1:0]		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

■ **EPFR82 : Address $01CA_H$ (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=6 to 11) Output compare channel n output enable

OCUnE[1:0] (n=6 to 11)	Operation
00	No output
01	Output from the OCUn_0
10	Output from the OCUn_1
11	Setting is prohibited

EPFR29 [bit7 to bit0] Reserved
EPFR81 [bit3 to bit0] Reserved

4.6.11. Extended Port Function Register 83 : EPFR83

The bit configuration of extended port function register 83 is shown.

This register selects up/down counter pins. (I/O relocation)

■ EPFR83 : Address 01CB$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ZIN1E	BIN1E	AIN1E	ZIN0E		BIN0E	AIN0E
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

[bit6] ZIN1E :
[bit5] BIN1E :
[bit4] AIN1E :
[bit3, bit2] ZIN0E [1:0] :
[bit1] BIN0E :
[bit0] AIN0E :

AINnE (n=0,1)	Operation
0	Input from the AINn_0 pin (Initial value)
1	Input from the AINn_1 pin

Same BIN0E, BIN1E, ZIN1E

ZINnE[1:0] (n=0)	Operation
00	Input from the ZINn_0 pin (Initial value)
01	Input from the ZINn_1 pin
1x	Input from the ZINn_2 pin

EPFR83 [bit7] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.12. Extended Port Function Register 42 : EPFR42

The bit configuration of extended port function register 42 is shown.

This register selects D/A converter output signal. (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

EPFR42 [bit7 to bit2] Reserved

These bits always read "1". Writing has no effect on operation.

4.6.13. Extended Port Function Register 43, 44 : EPFR43, 44

The bit configuration of extended port function register 43, 44 is shown.

These registers select external interrupt pins. (I/O relocation)

■ **EPFR43 : Address 0E8B$_H$ (Access : Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	INT7E	Reserved		INT4E	INT3E	INT2E	INT1E	Reserved
Initial value	0	1	1	0	0	0	0	1
Attribute	R/W	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R1,WX

■ **EPFR44 : Address 0E8C$_H$ (Access : Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	INT14E	INT13E	Reserved			INT9E	Reserved
Initial value	1	0	0	1	1	1	0	1
Attribute	R1,WX	R/W	R/W	R1,WX	R1,WX	R1,WX	R/W	R1,WX

INTnE (n=0 to 15)	Operation
0	Input from the INTn_0 pin (Initial value)
1	Input from the INTn_1 pin

EPFR43 [bit6, bit5, bit0] Reserved
EPFR44 [bit7, bit4 to bit2, bit0] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.14. Extended Port Function Register 48 to 51 : EPFR48 to 51

The bit configuration of extended port function register 48 to 51 is shown.

These registers are reserved. These must not be used.

■ **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

EPFR48 [bit7 to bit0] Reserved
EPFR49 [bit7 to bit0] Reserved
EPFR50 [bit7 to bit0] Reserved
EPFR51 [bit7 to bit0] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.15. Extended Port Function Register 65 to 70 : EPFR65 to 70

The bit configuration of extended port function register 65 to 70 is shown.

These registers enable serial chip select output.

■ EPFR65 : Address 01B9H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SCSO40E		SCSO3E		Reserved	SCSO2E	SCSO1E	
Initial value	0	0	0	0	1	0	0	0
Attribute	R/W	R/W	R/W	R/W	R1,WX	R/W	R/W	R/W

SCSO1E[1:0] : Serial chip select 1 input/output pin select
SCSO2E : Serial chip select 2 output pin select
SCSO3E[1:0] : Serial chip select 3 input/output pin select
SCSO40E[1:0] : Serial chip select 40 input/output pin select

■ EPFR66 : Address 01BAH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		SCSO43E		SCSO42E		SCSO41E	
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

SCSO41E[1:0] : Serial chip select 41 output pin select
SCSO42E[1:0] : Serial chip select 42 output pin select
SCSO43E[1:0] : Serial chip select 43 output pin select

■ EPFR67 : Address 01BBH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SCSO53E	SCSO52E	SCSO51E	SCSO50E
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

SCSO50E : Serial chip select 50 output pin select
SCSO51E : Serial chip select 51 output pin select
SCSO52E : Serial chip select 52 output pin select
SCSO53E : Serial chip select 53 output pin select

■ EPFR68 : Address 01BCH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SCSO63E	SCSO62E	SCSO61E	SCSO60E
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

SCSO60E : Serial chip select 60 output pin select
SCSO61E : Serial chip select 61 output pin select
SCSO62E : Serial chip select 62 output pin select
SCSO63E : Serial chip select 63 output pin select

■ EPFR69 : Address 01BDH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SCSO73E	SCSO72E	SCSO71E	SCSO70E
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

SCSO70E : Serial chip select 70 output pin select
SCSO71E : Serial chip select 71 output pin select
SCSO72E : Serial chip select 72 output pin select
SCSO73E : Serial chip select 73 output pin select

■ EPFR70 : Address 01BEH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SCSO11E	SCSO10E[1:0]		SCSO9E	SCSO8E
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

SCSO8E : Serial chip select 8 output pin select
SCSO9E : Serial chip select 9 output pin select
SCSO10E[1:0]: Serial chip select 10 input/output pin select
SCSO11E : Serial chip select 11 output pin select

SCSOnE (n=1,3,40 to 43, 10)	Operation
00	Input from the SCSn_0 pin, output disabled (Initial value) *
01	Input from the SCSn_0 pin / Output from the SCSn_0 pin *
1x	Input from the SCSn_1 pin / Output from the SCSn_1 pin *

*:SCSOnE(n=41,42,43) is output only.

SCSOnE (n=2,50 to 53, 60 to 63, 70 to 73, 8, 9, 11)	Operation
0	Input from the SCSn_0 pin, output disabled (Initial value) *
1	Input from the SCSn_0 pin / Output from the SCSn_0 pin *

*:SCSOnE(n=51,52,53,61,62,63,71,72,73) is output only.

EPFR65 [bit3] Reserved
EPFR66 [bit7, bit6] Reserved
EPFR67 [bit7 to bit4] Reserved
EPFR68 [bit7 to bit4] Reserved
EPFR69 [bit7 to bit4] Reserved
EPFR70 [bit7 to bit5] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.16. Extended Port Function Register 84, 85 : EPFR84, 85

The bit configuration of extended port function register 84, 85 is shown.

These registers enable wave generator output and select I/O pins. (I/O multiplexing and I/O relocation)

■ EPFR84 : Address 01CCH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RTO3E		RTO2E		RTO1E		RTO0E	
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

RTO0E[1:0] : Wave generator real time output ch.0 output pin select
RTO1E[1:0] : Wave generator real time output ch.1 output pin select
RTO2E[1:0] : Wave generator real time output ch.2 output pin select
RTO3E[1:0] : Wave generator real time output ch.3 output pin select

■ EPFR85 : Address 01CDH (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		DTTI0E		RTO5E		RTO4E	
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

RTO4E[1:0] : Wave generator real time output ch.4 output pin select
RTO5E[1:0] : Wave generator real time output ch.5 output pin select
DTTI0E[1:0] : Wave generator ch.0 to ch.5 input pin select

RTOnE (n=0 to 5)	Operation
00	RTOn output disabled (Initial value)
01	RTOn_0 output enabled
1x	RTOn_1 output enabled

DTTI0E	Operation
00	Input from the DTTI_0 pin (Initial value)
01	Input from the DTTI_1 pin
1x	Input from the DTTI_2 pin

EPFR85 [bit7, bit6] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.17. Extended Port Function Register 87 : EPFR87

The bit configuration of extended port function register 87 is shown.

These registers select A/D converter external trigger input.

■ EPFR87 : Address 01CF$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ADTG1E	ADTG0E
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

ADTG0E : A/D converter external trigger input ch.0 input pin select
ADTG1E : A/D converter external trigger input ch.1 input pin select

ADTGnE (n=0, 1)	Operation
0	Input from the ADTGn_0 pin (Initial value)
1	Input from the ADTGn_1 pin

EPFR87 [bit7 to bit2] Reserved
These bits always read "1". Writing has no effect on operation.

4.6.18. Extended Port Function Register 88 : EPFR88

The bit configuration of extended port function register 88 is shown.

This register enables clock monitor output. (I/O multiplexing)

■ EPFR88 : Address $01D0_H$ (Access: Byte, Half-word, Word)

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

[bit0] MONKCLKE : Clock monitor output pin select

MONKCLKE	Operation
0	MONCLK output disabled (Initial value)
1	MONCLK output enabled

EPFR88 [bit7 to bit1] Reserved
These bits always read "1". Writing has no effect on operation.

4.7. Port Input Enable Register: PORTEN (PORT ENable register)

The bit configuration of the port input enable register is shown below.

This register releases the port input block. At a power-on reset, inputs to most pins are blocked in order to avoid pass-through current fluctuations before the ports are configured by software. For information on pins whose inputs are blocked, see "C. 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. The PORTEN is the target key code register.

■ PORTEN : Address 0F40$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							GPORTEN
Initial value	1	1	1	1	1	1	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 pins are set to block input. See the "Pin Status" table in the "APPENDIX" for the pins that are input-blocked.
1	Input block by this bit is released.

4.8. KEY CoDe Register : KEYCDR

The bit configuration of key code register is shown.

This register sets register writing that includes the error writing protection function. If writing to this register is not executed according to the specified method, writing to the target register will become invalid. This register is only enabled for half-word access.

■ KEYCDR : Address 0F44$_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	KEY1	KEY0	SIZE	RADR12	RADR11	RADR10	RADR9	RADR8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RADR7	RADR6	RADR5	RADR4	RADR3	RADR2	RADR1	RADR0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit15, bit14] KEY1, KEY0: Key code

Key code setting bits. It is necessary to write continuously to this bit according to the order "00", "01", "10", and "11".

Note:

When the writing order becomes different, the key code setting will become invalid and it will be necessary to reset them from the beginning.

[bit13] SIZE: Access size

This bit sets the access size for writing to the key code target register. Write the same data to the bit when writing the key code according to the order "00", "01", "10", and "11".

SIZE	Description
0	Set byte access
1	Set half-word access

Notes:

- When different data is written while writing the key code "00", "01", "10", and "11", the key code setting will become invalid and it will be necessary to reset it from the beginning.
- Word access for the key code target register is prohibited.

[bit12 to bit0] RADR[12:0]: Port address

These bits set the lower 13 bits of the address for the key code target register. Write the same data to the bit when writing the key code according to the order "00", "01", "10", and "11".

Notes:

- When different data is written while writing the key code "00", "01", "10", and "11", the key code setting will become invalid and it will be necessary to reset them from the beginning.
- Key code setting might be canceled because of the DMA transfer. Read the value of the target register, and check if the value is updated.

5. Operation

This section explains operations of I/O ports.

5.1. Pin I/O Assignment

5.2. EPFR setting priority

5.3. Notes on Input I/O Relocation Setting

5.4. Noise Filter

5.5. Input blocked by GPORTEN

5.6. Notes on Pins with the A/D Converter Function

5.7. Setting when Using the Base Timer TIOA1 Pin

5.8. Key Code Register Function Settings

5.9. Operation at Wake Up from Power Shutdown

5.10. Notes on switching the I/O port function

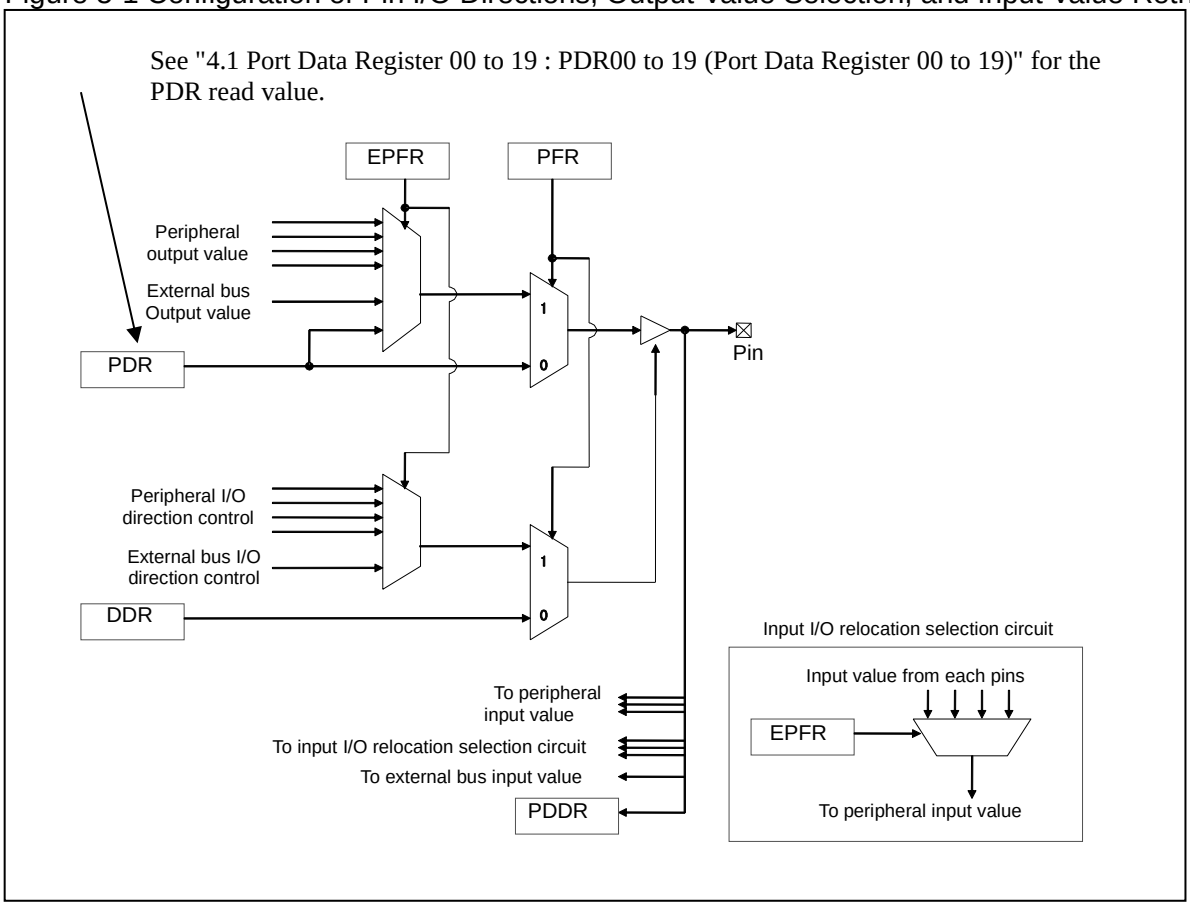
5.11. Input blocked when specific peripheral functions are used

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 5-1 Configuration of Pin I/O Directions, Output Value Selection, and Input Value Retrieval



As explained in the pertinent section concerning pin assignment, first change the PFR setting to enable the port function. Since the pin functions as a port, also set the DDR and PDR values in advance if necessary. Note that the I/O direction of the pin is once set as specified by the DDR. For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode". For information on the setting method, see "CHAPTER: 12-BIT A/D CONVERTER".

5.1.1. Peripheral I/O (Bidirectional) Pin Assignment

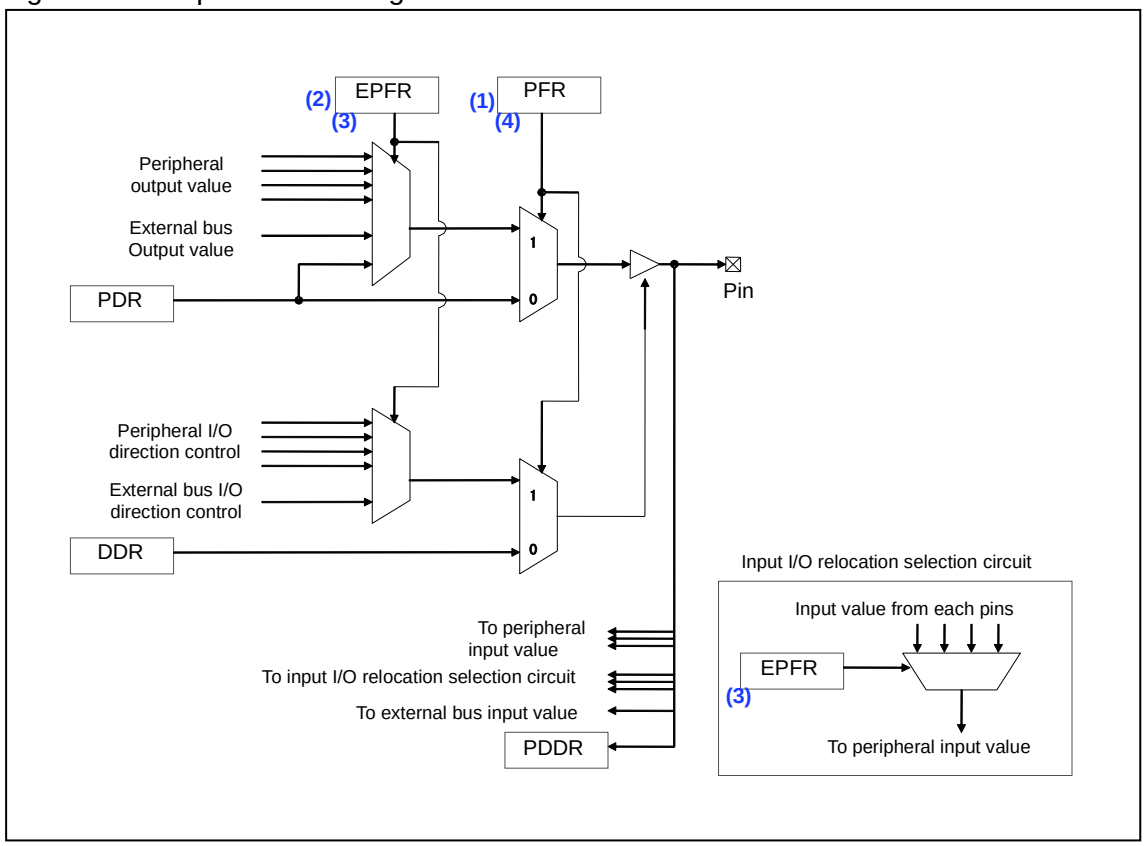
The peripheral I/O (bidirectional) pin assignment is shown below.

■ Preparation

- Since the pin once functions as a port as the result of step (1), set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode". For information on the setting method, see "CHAPTER: 12-bit 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 terminal to be used with the EPFR of the relevant peripheral.
(4)Set the PFR for the peripheral.

Figure 5-2 Peripheral I/O Assignment Procedure



5.1.2. Peripheral Input Assignment

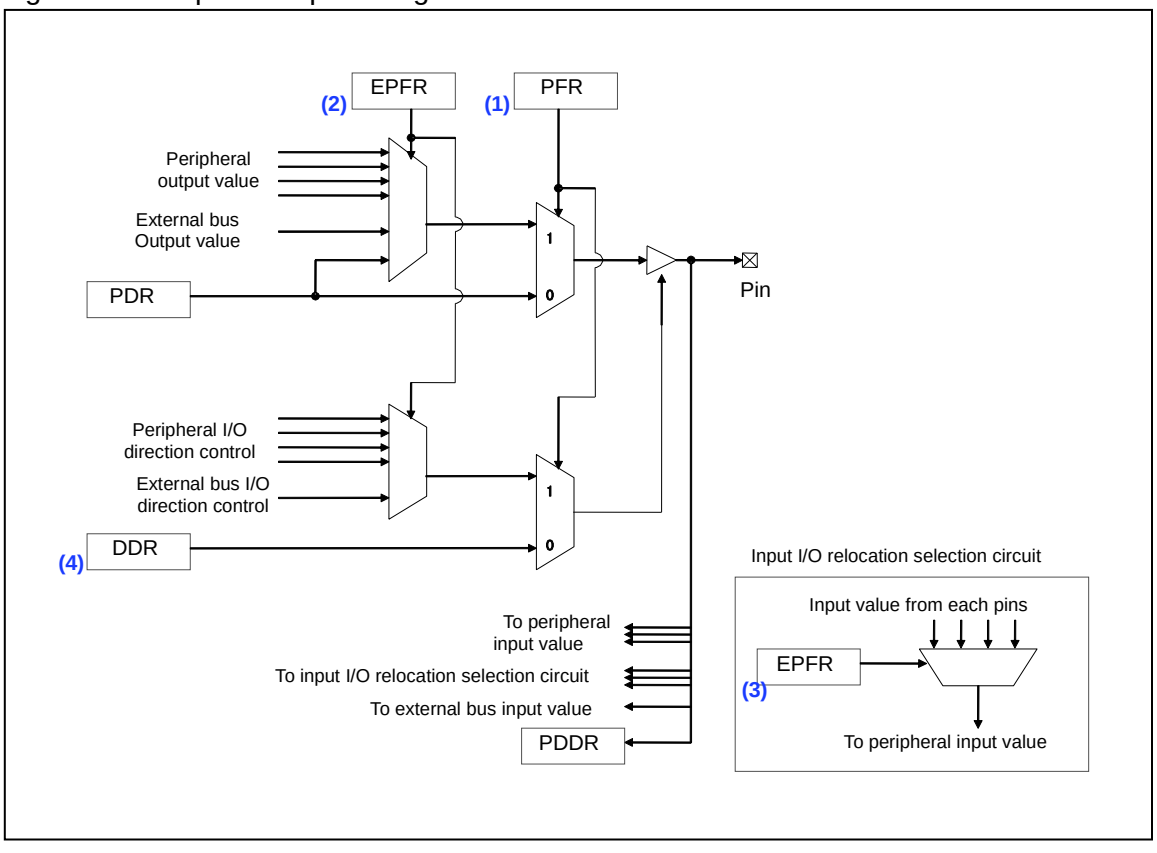
The peripheral input assignment is shown below.

■ 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: 12-bit A/D CONVERTER".

(1)Set the PFR of the applicable pin to enable the port function.
(2)Disable the EPFRs for all other peripherals that use the relevant pin.
(3)If the relevant peripheral has the I/O relocation function, set the EPFR of the relevant peripheral.
(4)Set the DDR for input.

Figure 5-3 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 PPG10_0 are assigned to the same pin (pin number 106, P102), external interrupt 10(0) can be generated at the PPG10(0) output by setting the pin to peripheral output of PPG10_0.

5.1.3. Peripheral Output Assignment

The peripheral output assignment is shown below.

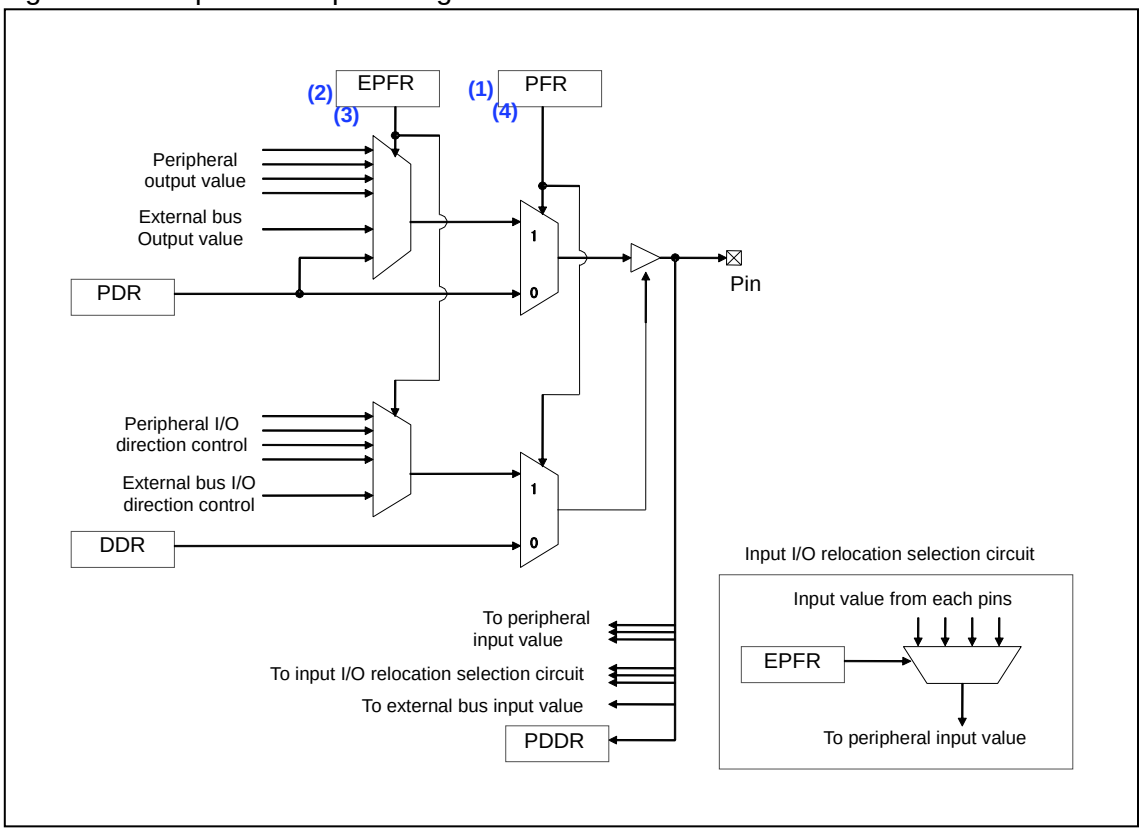
The setting method is the same as that described in "5.1.1 Peripheral I/O (Bidirectional) Pin Assignment".

■ 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: 12-bit 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 use 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 pin to be used with the EPFR of the relevant peripheral.
(4)Set the PFR for the peripheral.

Figure 5-4 Peripheral Output Assignment Procedure



5.1.4. External Bus Assignment

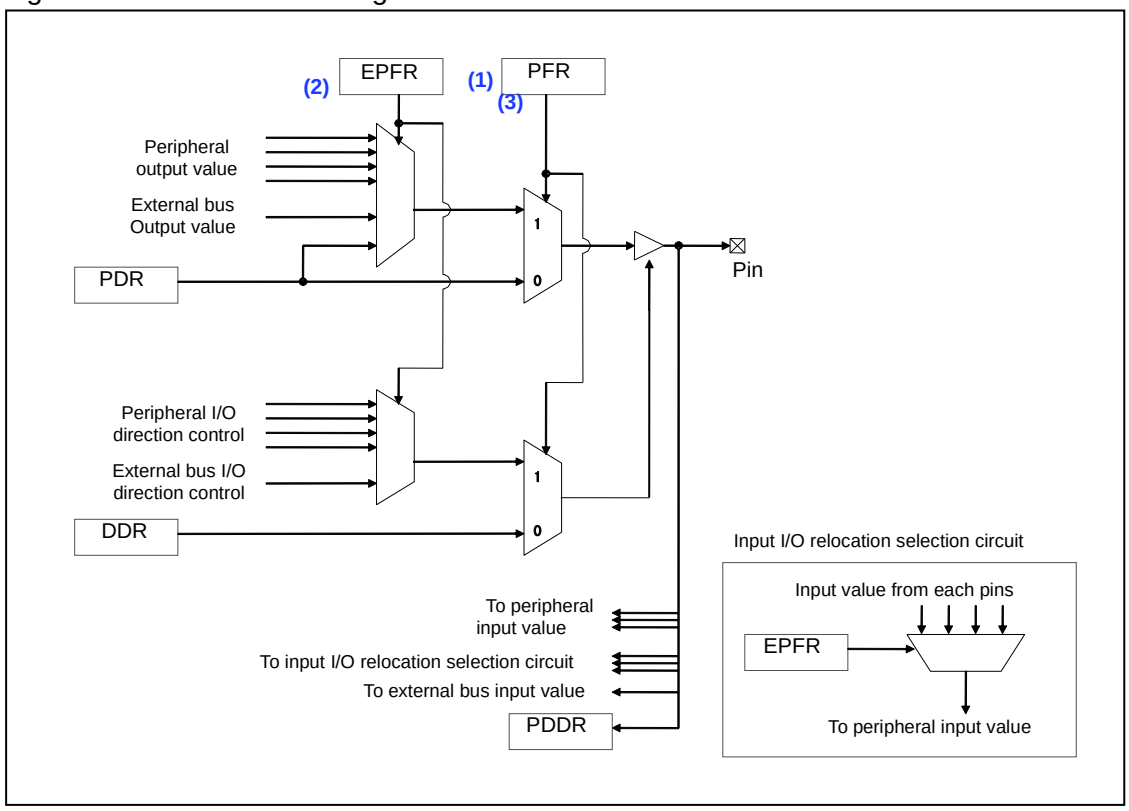
The external bus assignment is shown below.

■ 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: 12-bit 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 5-5 External Bus Assignment Procedure



5.1.5. Port Function (Input) Assignment

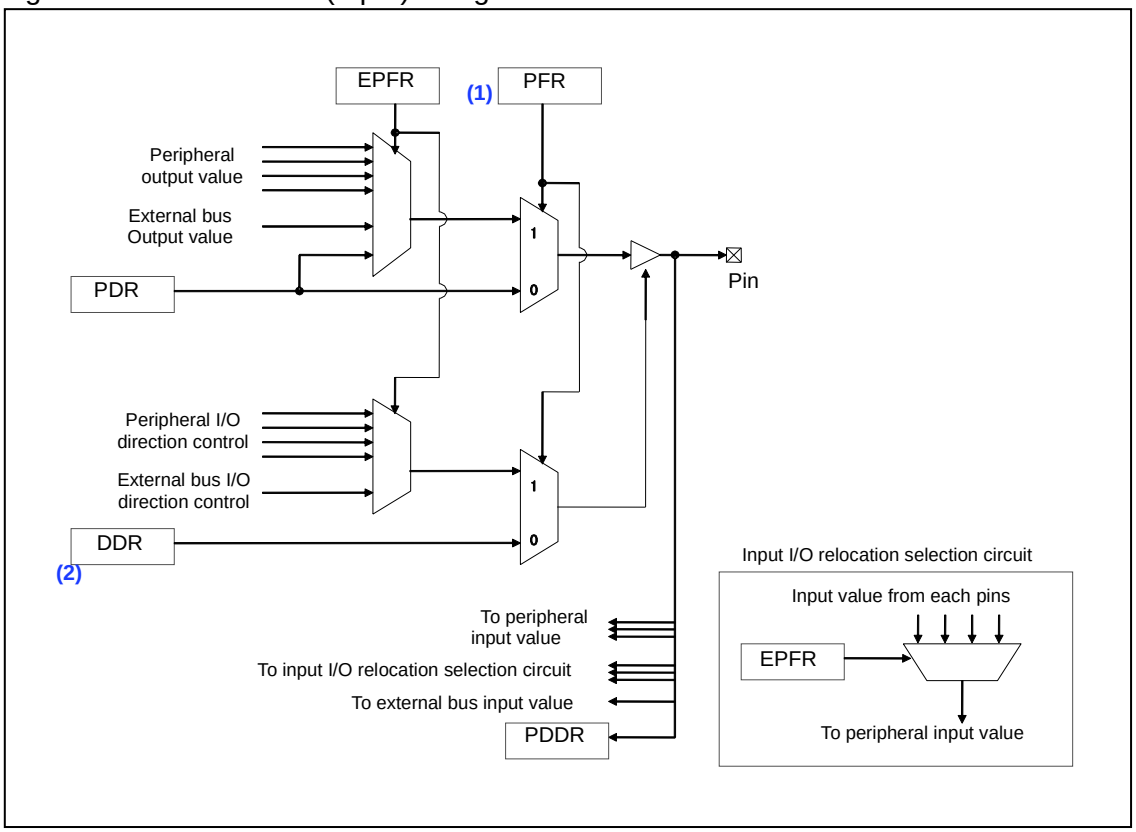
The port function (input) assignment is shown below.

■ 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: 12-bit A/D CONVERTER".

(1)Set the PFR to enable the port function.
(2)Set the DDR for input.

Figure 5-6 Port Function (Input) Assignment Procedure



5.1.6. Port Function (Output) Assignment

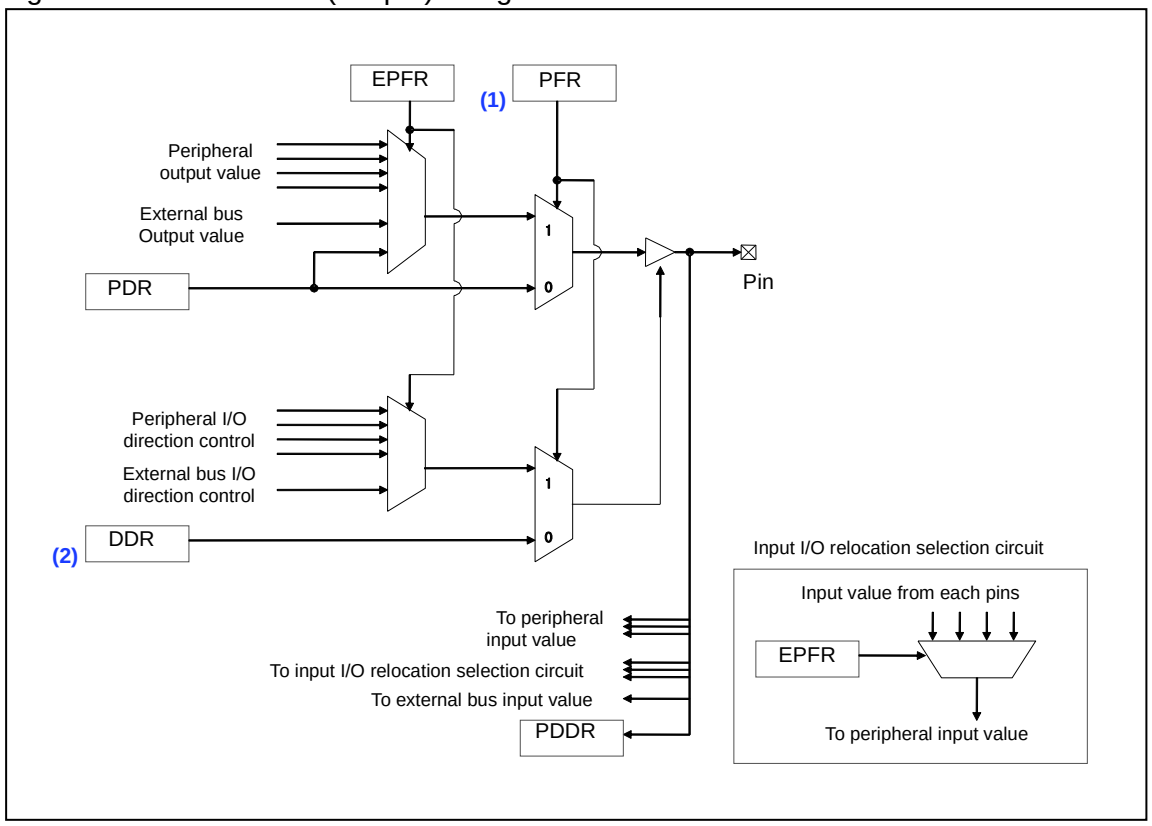
The port function (Output) assignment is shown below.

■ 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: 12-bit A/D CONVERTER".

(1)Set the PFR to enable the port function.
(2)Set the DDR for output.

Figure 5-7 Port Function (Output) Assignment Procedure



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: 12-bit A/D CONVERTER".

Since the A/D converter assignment is given the highest priority, no other configuration is required.

5.1.8. D/A converter output assignment

The D/A converter output assignment is shown below.

Same as "5.1.3. Peripheral Output Assignment"

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. D/A converter output
2. CAN
3. Multi-function serial interface
4. PPG
5. Real time clock
6. Base timer
7. Reload timer
8. Output compare
9. Clock monitor
10. Waveform generator

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.

5.4. Noise Filter

The noise filter is shown.

If an external pin is used to receive input for the following functions, the value that is entered through the noise filter is treated as the input level:

- Port function
- External interrupt request
- Free-run Timer
- Reload timer
- PPG
- Input capture
- A/D converter trigger input
- Base timer
- UP/DOWN COUNTER
- Dead timer interrupt request

Note:

Please refer to "Pins of Each Function" in the chapter of "OVERVIEW" for details.

5.5. 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 "C. Pin Status in CPU Status" in " APPENDIX " for the pin that becomes input blocked. See "4.7. Port Input Enable Register: PORTEN (PORT ENable register)"for the method of releasing the input blocked state.

When the state of a pin to be the input blocked state is read during the input blocked by GPORTEN, "0" is always read out.

5.6. 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 (digital port, peripheral function), set the relevant bit of the A/D converter analog input enable register (ADER) to "Analog input disable" in advance. In this case an A/D conversion should not be done on this analog input, because the digital inputs of this port pin are fixed at "0" during A/D conversion. For information on the setting method, see "CHAPTER: 12-bit 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 PFR19) and extended port function register (EPFR00 to EPFR88) settings.

5.7. 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 "5.1.2 Peripheral Input Assignment") and set for peripheral output for all cases other than base timer I/O mode 1 (see "5.1.3 Peripheral Output Assignment").

5.8. Key Code Register Function Settings

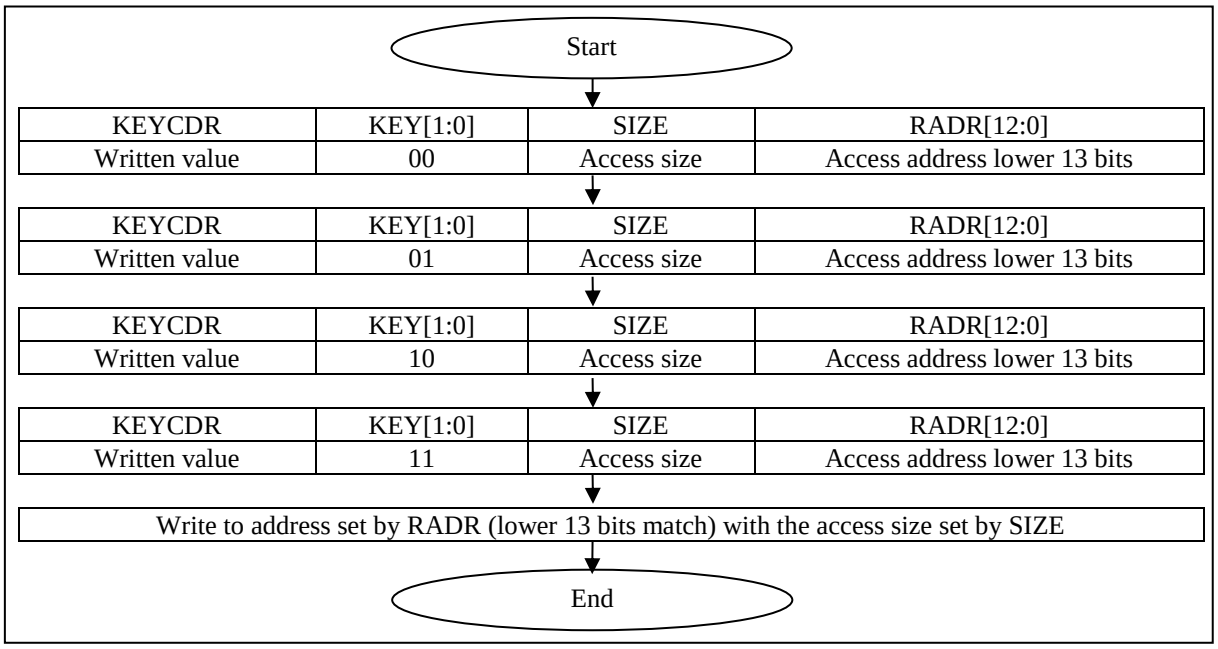
Setting when using the Key Code Register is shown.

The following settings are necessary for the key code register (KEYCDR) in order to write to the key code target register.

- Set KEY1+KEY0+Access Size (SIZE)+Access address (RADR[12:0]) to the key code register using half-word.
- Write (KEY1, KEY0) continuously according to the order (0, 0), (0, 1), (1, 0), and (1, 1). Set the address and access size to the same value four times when (KEY1, KEY0) is written four times.

The following is a flow chart.

Figure 5-8 Key Code Flow Chart



If the following conditions apply, the key code will not be released and writing will not be executed to the target register. In this case, it is necessary to set the key code register again from the beginning.

- When writing order for (KEY1,KEY0) is different
- When the data written to the SIZE bit is changed in the middle
- When the data written to the RADR bit is changed in the middle
- When the access size written to the SIZE bit is different from the size when accessing the actual target register
- When the address (lower 13 bits) written to the RADR bit is different from the address (lower 13 bits) when accessing the actual target register
- When the key code register and register related to the port are read while writing to the key code register

Notes:

- The key code setting might be canceled by DMA transfer. Read the value written in the object register, and confirm whether the value has been changed.

- While debugging by the on-chip debugger (OCD), the key code setting is canceled when the break function is executed during the key code setting.
- The DDR, PFR, EPFR, PPER, PORTEN, ADER, and DACR are the target key code registers. It is necessary to set the key code in order to execute writing.

5.9. 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, maintenance of the I/O state is kept during the wake-up from power shutdown. After completion of wake-up, this state shall be canceled and the register setting of the I/O port shall be effective.

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.

5.10. Notes on switching the I/O port function

Notes on switching the I/O port function are shown below.

When I/O port is switched from port function to resource or from resource to port function, the value of PDR may be output momentarily.

It happens if port function is changed from "input to output" or "output to input" at the time of switching.

If this output may cause a problem for the system, please write a value to PDR in advance at a level that will not cause a problem.

5.11. Input blocked when specific peripheral functions are used

A note regarding blocked input when specific peripheral functions are used is shown below.

When a pin is used as the A/D function and the state of the pin is read, "0" is always read.

Chapter 12: Interrupt Control (Interrupt Controller)

This chapter explains the interrupt control (interrupt controller).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : INTCNT-1v0-91520-2-E

1. Overview

This section explains overview the of the interrupt control (interrupt controller).

The interrupt controller performs arbitration of interrupt requests.

2. Features

This section explains features of the interrupt control (interrupt controller).

This module is composed of the following parts.

- ICR register
- Interrupt priority determination circuit
- Interrupt level and interrupt vector generation circuit

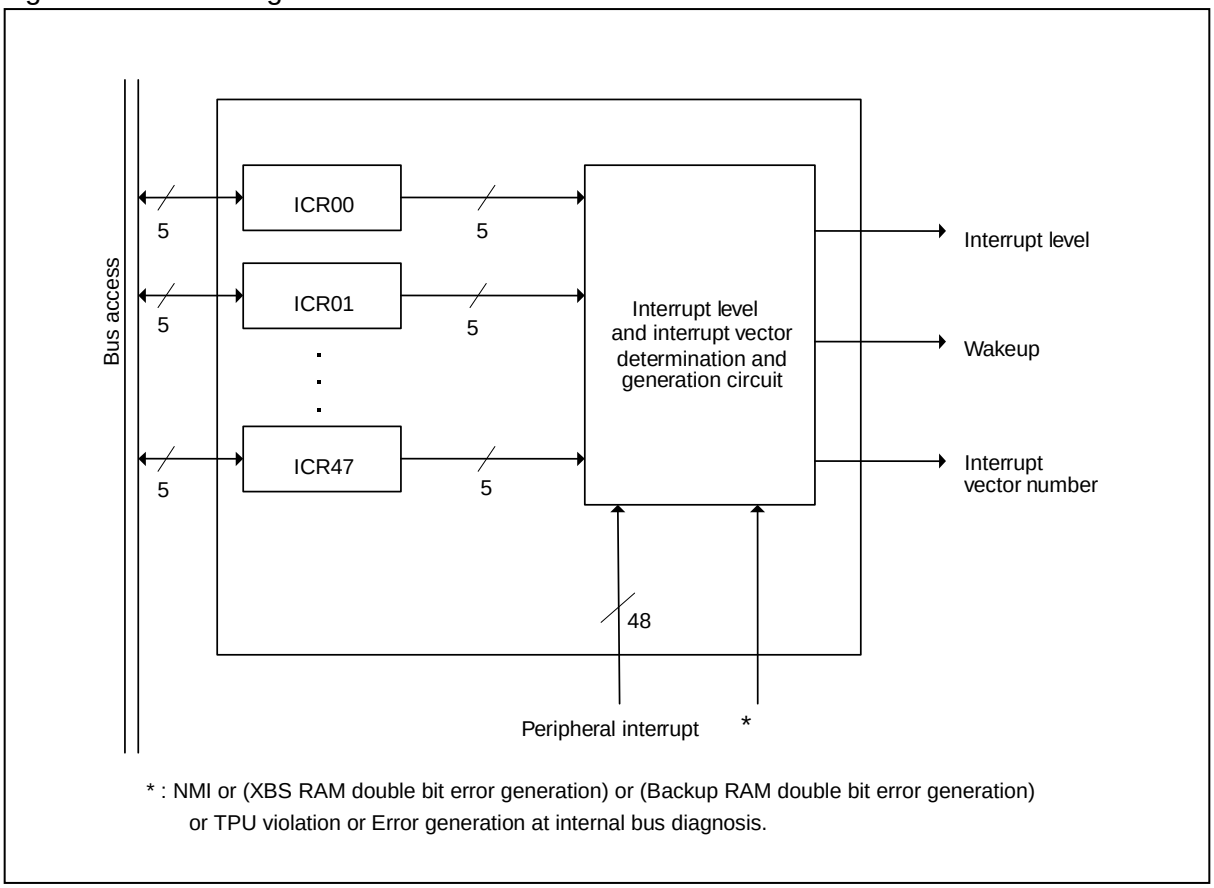
This module has the following functions.

- Detecting NMI requests and peripheral interrupt requests
- Priority determination (by level and interrupt vector)
- Transmitting the interrupt level of the factor with the highest priority to the CPU
- Transmitting the interrupt vector number of the factor with the highest priority to the CPU
- Generating wakeup requests by NMI / interrupts that occur with a level other than "11111"

3. Configuration

This section explains the configuration of the interrupt control (interrupt controller).

Figure 3-1 Block Diagram



* : NMI or (XBS RAM double bit error generation) or (Backup RAM double bit error generation) or TPU violation or Error generation at internal bus diagnosis.

4. Registers

This section explains the registers of the interrupt control (interrupt controller).

Table 4-1 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	

4.1. Interrupt Control Registers 00 to 47 : ICR00 to ICR47 (Interrupt Control Register 00 to 47)

The bit configuration of the interrupt control registers 00 to 47 is shown below.

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.

5. Operation

This section explains the operation of the interrupt control (interrupt controller).

5.1. Setting

5.2. Starting

5.3. Determining Priorities

5.4. Recovering From Stop Mode

5.5. Recovering From Standby Mode (Power shutdown)

5.1. Setting

This section explains the setting of the interrupt control (interrupt controller).

1. Configure the ICR register of the interrupt vector number corresponding to the peripheral for which you want to generate the interrupt.
2. Configure the peripheral where you want to generate the interrupt. (Configure interrupt output as enabled on the peripheral.)

5.2. Starting

This section explains the starting of the interrupt control (interrupt controller).

Start the configured peripheral.

5.3. Determining Priorities

The determining priorities are shown below.

This module selects the highest priority interrupt among interrupt factors that occur simultaneously and outputs the interrupt level and interrupt vector number for the interrupt factors to the CPU.

The criteria for determining the priority of interrupt factors are as follows.

1. NMI
2. Factors that meet the following conditions
   - If the value of the interrupt level is not 31 (5’b11111). (31 indicates interrupts disabled)
   - The factors where the value of the interrupt level is the smallest.
   - When the interrupt level is the same (except for 31), the factors that has the smallest interrupt vector number from amongst these.

If no interrupt factors is selected by the above criteria, 31 (5’b11111) is output as the interrupt level. The interrupt vector number at this time is undefined.

5.4. Recovering From Stop Mode

The recovering from stop mode is shown below.

The function for using an interrupt request to recover from stop mode is performed by this module. If an interrupt request (the interrupt level is anything other than "5’b11111") is generated from a peripheral (including NMI), a request is generated to the clock control unit to recover from stop mode.

As the interrupt priority judgment unit restarts operation once the clock supply starts after recovery from stop mode, the CPU is able to execute instructions until the interrupt priority judgment unit produces a result.

For interrupts that are not used as sources for recovering from stop mode, set the interrupt level of the corresponding interrupt control registers (ICR00 to ICR47) to "5’b11111" (interrupts disabled).

5.5. Recovering From Standby Mode (Power shutdown)

The recovering from standby mode (Power shutdown) is show below.

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 state cannot change to the power shutdown state. The instruction execution is continued as it is.

When the interrupt level is at ICR=0x1F (interrupt disable) and there is a state in which an interrupt factor is generated, this does not result in a standby return factor. Thus, even though the state once changes to the power shutdown state, it returns immediately after completion of the power shutdown return sequence because the state has the power shutdown return factor. (It is executed from the reset vector.)

Chapter 13: External Interrupt Input

This chapter explains the external interrupt input.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Setting
7. Q&A
8. Notes

Code : BG04-1v0-91520-2-E

1. Overview

This section explains the overview of the external interrupt input.

Interrupt request input from external interrupt input pins (INT0 to INT15).

2. Features

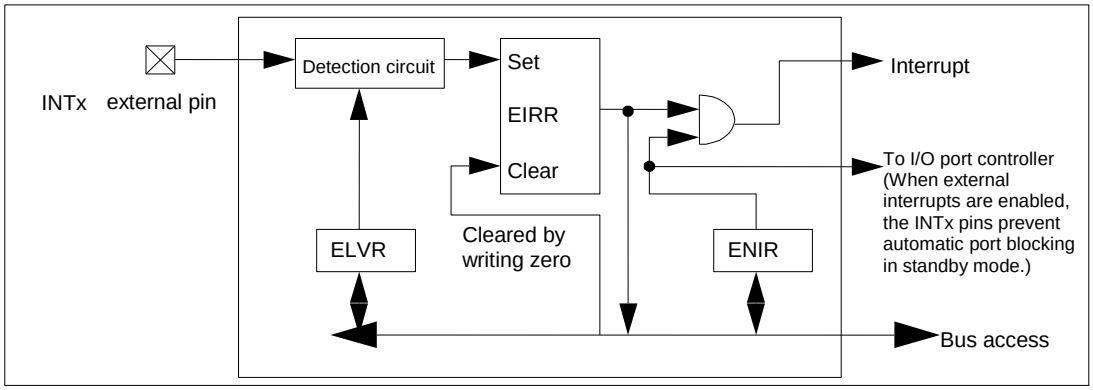
This section explains features of the external interrupt input.

- Sixteen types of external interrupt input pins (INT0 to INT15)
- Interrupt detection factors:4 types: ("L" level, "H" level, rising edge, and falling edge)

3. Configuration

This section explains the configuration of the external interrupt input.

Figure 3-1 Block Diagram



4. Registers

This section explains registers of the external interrupt input.

■ List of External Pins

Channel	Base_addr	External pins (INT)				
		MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK, MB91F52xL
0	0x0550	INT0_0	INT0_0	INT0_0	INT0_0	INT0_0
1	0x0550	INT1_1	INT1_1	INT1_0/ INT1_1	INT1_0/ INT1_1	INT1_0/ INT1_1
2	0x0550	INT2_1	INT2_1	INT2_0/ INT2_1	INT2_0/ INT2_1	INT2_0/ INT2_1
3	0x0550	INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1
4	0x0550	INT4_1	INT4_1	INT4_1	INT4_0/ INT4_1	INT4_0/ INT4_1
5	0x0550	INT5_0	INT5_0	INT5_0	INT5_0	INT5_0
6	0x0550	INT6_0	INT6_0	INT6_0	INT6_0	INT6_0
7	0x0550	INT7_1	INT7_1	INT7_0/ INT7_1	INT7_0/ INT7_1	INT7_0/ INT7_1
8	0x0554	INT8_0	INT8_0	INT8_0	INT8_0	INT8_0
9	0x0554	INT9_1	INT9_1	INT9_1	INT9_0/ INT9_1	INT9_0/ INT9_1
10	0x0554	INT10_0	INT10_0	INT10_0	INT10_0	INT10_0
11	0x0554	INT11_0	INT11_0	INT11_0	INT11_0	INT11_0
12	0x0554	INT12_0	INT12_0	INT12_0	INT12_0	INT12_0
13	0x0554	INT13_1	INT13_1	INT13_0/ INT13_1	INT13_0/ INT13_1	INT13_0/ INT13_1
14	0x0554	INT14_1	INT14_1	INT14_1	INT14_1	INT14_0/ INT14_1
15	0x0554	INT15_0	INT15_0	INT15_0	INT15_0	INT15_0

Table 4-1 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

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

Notes:

- 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 is invalid.

- 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.

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.

Notes:

- 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.

4.3. External Interrupt Request Level Register 0/1 : ELVR0/ELVR1 (External interrupt LeVel Register 0/1)

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 0552_H (Access: Byte, Half-word, Word)**

■ **ELVR1 : Address 0556_H (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".

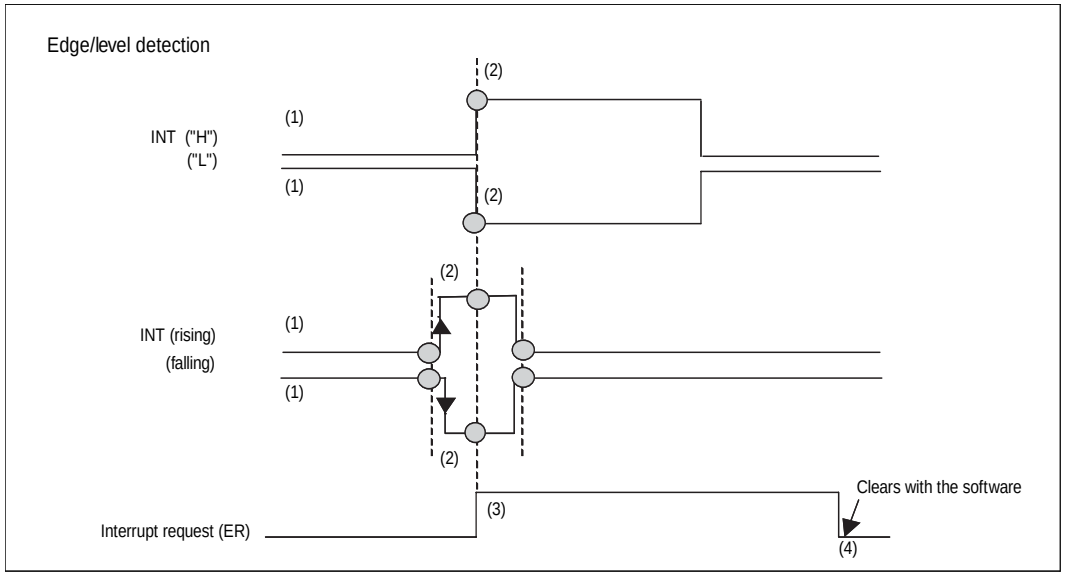
Notes:

- 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.

5. Operation

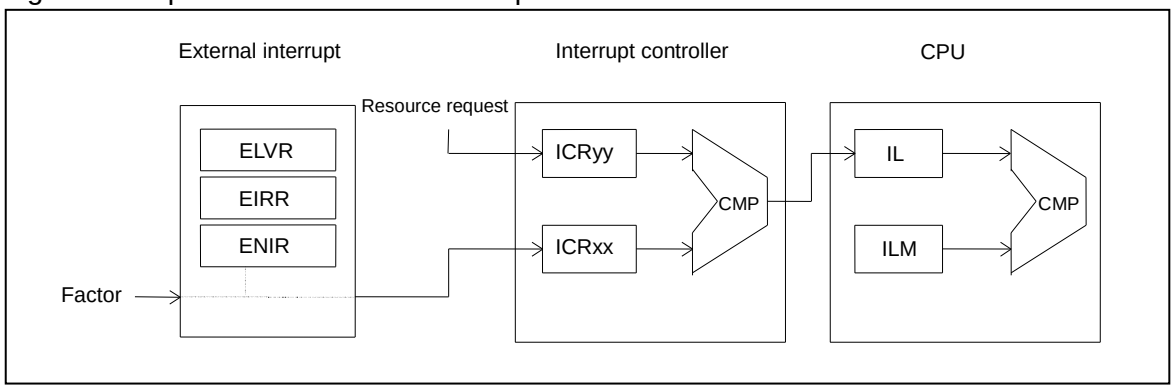
This section explains the operation of the external interrupt input.

Figure 5-1 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 5-2 Operation of External Interrupt



1. Operation of external interrupt
   This module generates the interrupt request signal to the interrupt controller when a request set in the ELVR register is input in the corresponding pin after setting a request level and the enable register. The corresponding interrupt will be generated when the interrupt from this resource was found to have the highest priority in the result for examining the priority in interrupts concurrently occurred in the interrupt controller.

2. Transition to standby mode
   Channels not to be used should be moved to disable state before letting them go into the standby mode. External pins enter an input blocked state at standby mode, but external pins of external interrupt enabled channels enter an input enabled state.

3. Setting procedure of external interrupts
   When setting registers which reside in the external interrupt unit, follow the steps shown below:

   (1) Disable the corresponding bit for the enable register.
   (2) Set the corresponding bit for the request level setting register.
   (3) Read the request level register.
   (4) Clear the corresponding bit for the factor register.
   (5) Enable the corresponding bit for the enable register.
   (Note that concurrent writes of 16-bit data are allowed in step (4) and (5).)

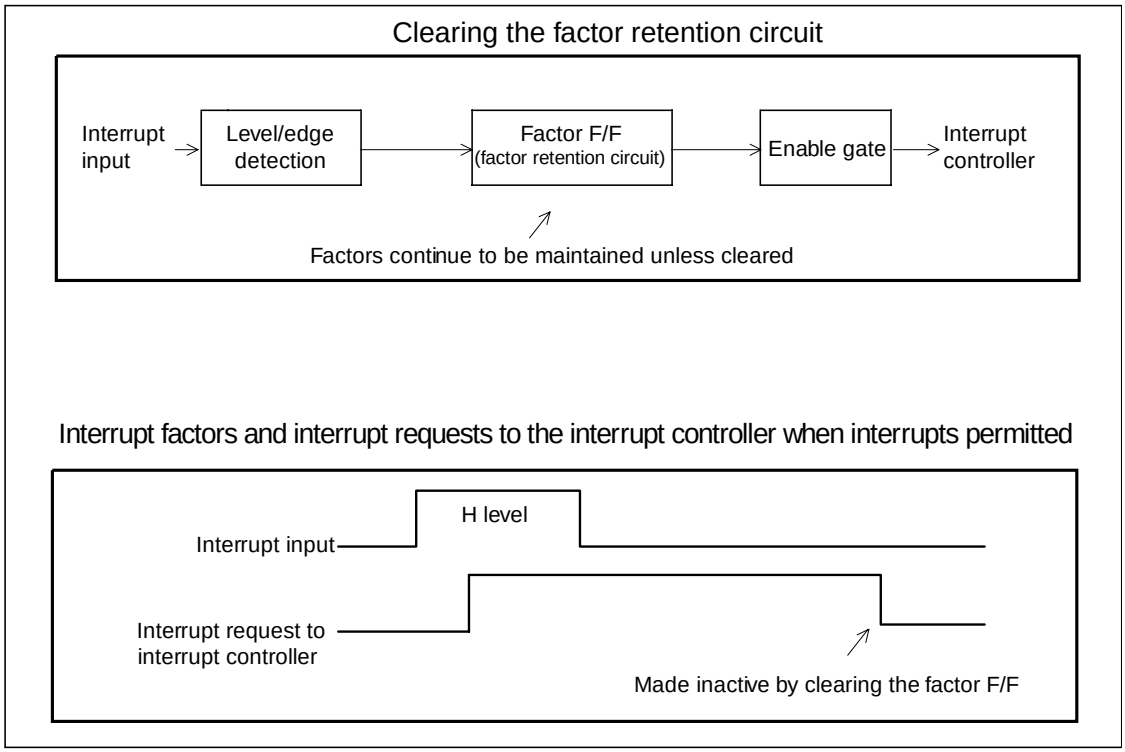
   The enable register must be disabled before you can set the registers in this module. The factor register must be cleared before you can set the enable register to enable state.

   This has to be done to avoid generating erroneous interrupt factors at the time of setting register or in interrupt enable state.

4. External interrupt factor
   Requests to the interrupt controller will continue to be active although a request input from outside is canceled, because there is an internal factor retention circuit.

   To cancel requests going toward the interrupt controller, the factor register should be cleared.

Figure 5-3 Clearing the Factor Retention Circuit and Interrupt Factor and Interrupt Request to Interrupt Controller in Interrupt Enable State



6. Setting

This section explains settings of the external interrupt input.

Table 6-1 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 "7. Q&A".
Specifying external pins to be used for input.	See "CHAPTER: I/O PORTS".	See "CHAPTER: I/O PORTS".
External interrupt	An input from the external pin → Input signal to pins INT0 to INT15	—

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".

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.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] 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 causes (power-on reset, internal low-voltage detection, and simultaneous assertion of RSTX and NMIX) are recognized. At this time, the register of the external interrupt input is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the external interrupt input register before using it.

[MB91F52xxxD] 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 causes (power-on reset, internal low-voltage detection, and assertion of RSTX) are recognized. At this time, the register of the external interrupt input is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the external interrupt input register before using it.

■ Note for using the external interrupt as source for recovering from the watch mode with power-shutdown

Set the interrupt levels that are used as sources for recovering from the watch mode with power-shutdown to ‘31’, before CPU state changes to the watch mode with power-shutdown. And don’t use NMIX pin as source for recovering from the watch mode with power-shutdown.

Chapter 14: NMI Input

This chapter explains the NMI input.

1. Overview
2. Features
3. Configuration
4. Register
5. Operation
6. Usage Example

Code : FR81S10_NMI-1v1-91520-2-E

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.

2. Features

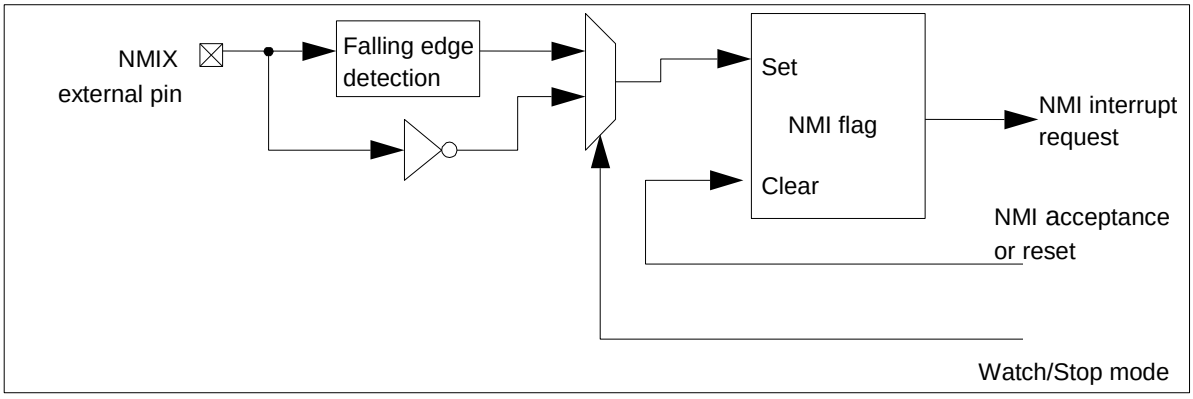
This section explains features of the NMI input

Can be used in stop mode (Power-shutdown is included) and watch mode. Don’t use NMI input as source for recovering from the watch mode (Power-shutdown).

3. Configuration

This section explains the configuration of the NMI input.

Figure 3-1 Block Diagram



4. Register

This section explains the register of the NMI input.

This function has no register.

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

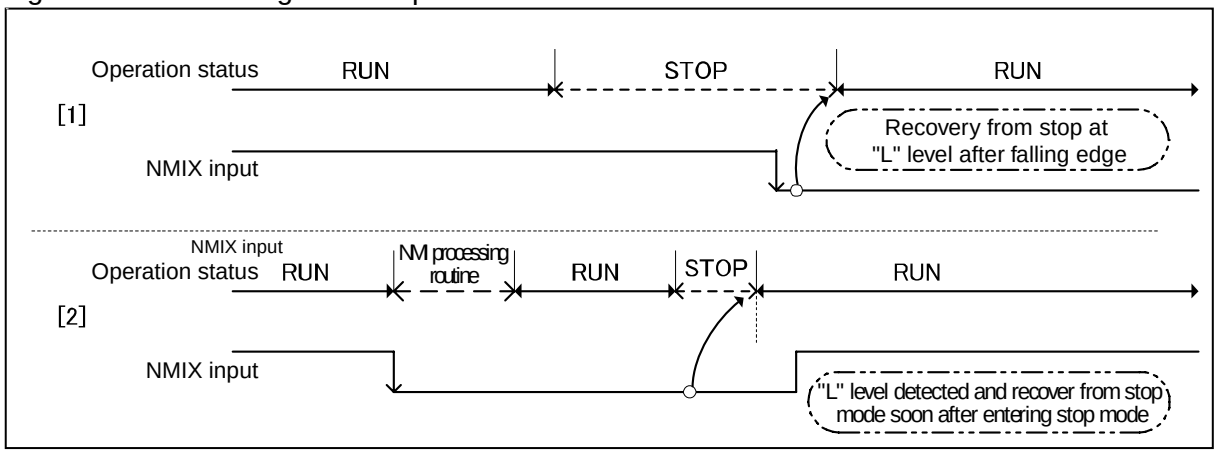
The NMI request detector has an NMI flag that is set for an NMI request and is cleared only if an interrupt for the NMI itself is accepted or reset occurs. The NMI flag cannot be read or written.

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 5-1). 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.

Figure 5-1 Recovering from Stop Mode



Note:

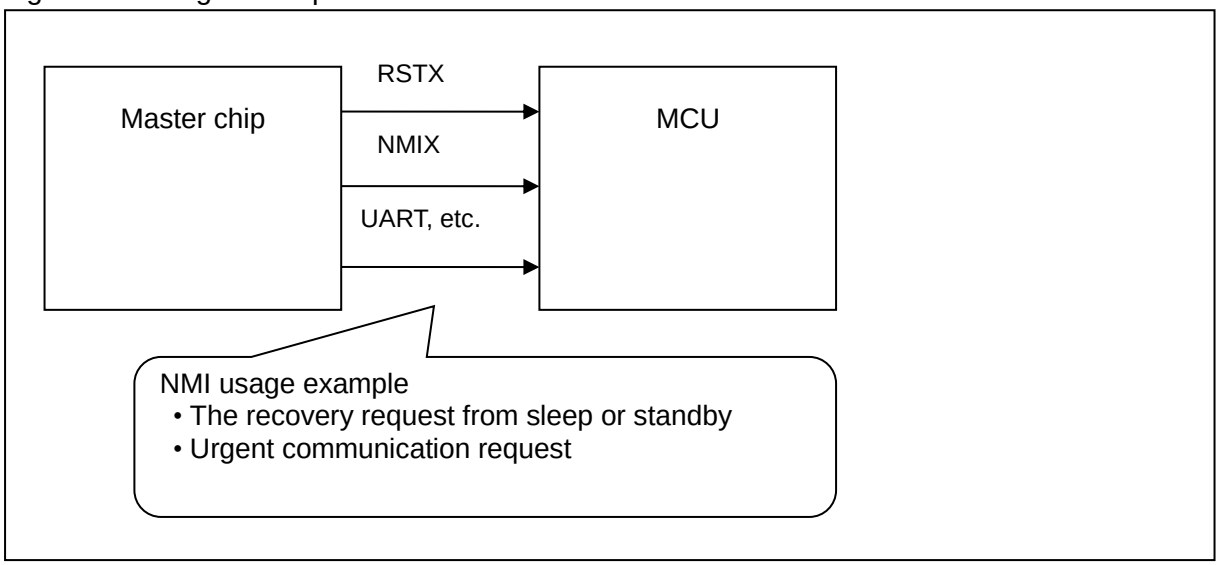
- The watch mode and the watch mode (power-shutdown) are similarly controlled.
- An internal reset is issued at the return from standby mode (power-shutdown), and no NMI request is accepted.

6. Usage Example

This section explains a usage example of the NMI input.

This section gives an example of using the NMI function.

Figure 6-1 Usage Example



Chapter 15: Delay Interrupt

This chapter explains the delay interrupt.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Restrictions

Code : FR81S10_DINT-1v1-91520-2-E

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.

2. Features

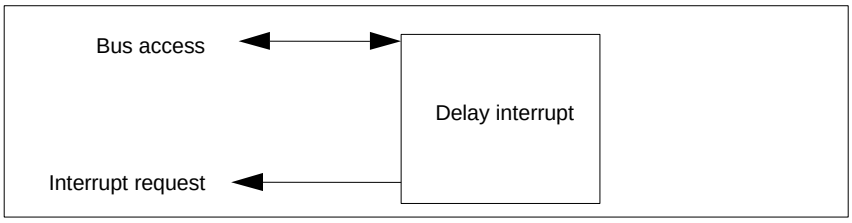
This section explains features of the delay interrupt.

The delay interrupt can be generated by writing to a register.

3. Configuration

This section explains the configuration of the delay interrupt.

Figure 3-1 Block Diagram



4. Registers

This section explains registers of the delay interrupt.

Address	Registers				Register function
	+0	+1	+2	+3	
0x0044	DICR	Reserved	Reserved	Reserved	Delay Interrupt Control Register

■ Delay Interrupt Control Register : DICR (Delay Interrupt Control Register)

This register controls the delay interrupts.

- **DICR : Address 0044_H (Access: Byte)**

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

[bit0] DLYI (DeLaY Interrupt enable) : Delay Interrupt Enable Bit

This bit generates and clears the delay interrupt factor.

DLYI	Description
"0" write	Clears the delay interrupt factor
"1" write	Generates the delay interrupt factor

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 factor. Writing "0" to this bit cancels the delay interrupt source.

This bit functions like a standard interrupt factor flag and should be cleared in the interrupt routine at the same time as when switching a task.

6. Restrictions

This section explains restrictions of the delay interrupt.

Do not use delay interrupts for DMA transfer requests.

Chapter 16: Interrupt Request Batch Read

This chapter explains the overview, features, and configuration of the interrupt request batch read.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : BIRPR-1v1-91520-2-E

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 FR81-family CPU.

2. Features

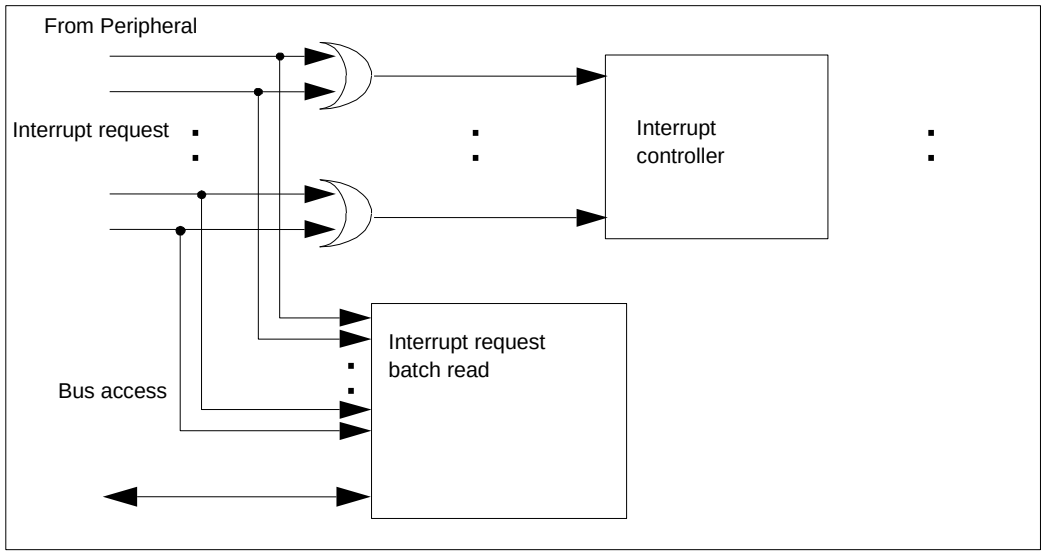
This section shows features of the interrupt request batch read.

Using this module, you can easily check whether interrupts have been generated.

3. Configuration

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

Figure 3-1 Block Diagram



4. Registers

This section explains the registers of the interrupt request batch read.

Table 4-1 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	Reserved	Reserved	IRPR3H	IRPR3L	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)

Address	Registers				Register function
	+0	+1	+2	+3	
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	IRPR15L	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) Interrupt request batch read register 15 lower-order (#35)

#nn : Interrupt vector number (decimal)

4.1. Interrupt Request Batch Read Register 0 upper-order : IRPR0H (Interrupt Request Peripheral Read register 0H)

The bit configuration of the interrupt request batch read register 0 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #18)

■ IRPR0H : Address 0418_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RTIR0	RTIR1	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 interrupt request
[bit6] RTIR1 (Reload Timer Interrupt Request 1) : Reload timer 1 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.2. Interrupt Request Batch Read Register 0 lower-order : IRPR0L (Interrupt Request Peripheral Read register 0L)

The bit configuration of the interrupt request batch read register 0 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #19)

■ IRPR0L : Address 0419_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RTIR2	RTIR3	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RTIR2 (Reload Timer Interrupt Request 2) : Reload timer 2 interrupt request
[bit6] RTIR3 (Reload Timer Interrupt Request 3) : Reload Timer 3 Interrupt Request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.3. Interrupt Request Batch Read Register 1 upper-order : IRPR1H (Interrupt Request Peripheral Read register 1H)

The bit configuration of the interrupt request batch read register 1 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #20)

■ IRPR1H : Address 041A$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RXIR0	ISIR0	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RXIR0 (multifunction serial interface RX Interrupt Request 0) : Multi-function serial interface ch.0 reception completion interrupt request

[bit6] ISIR0 (multifunction 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.

4.4. Interrupt Request Batch Read Register 1 lower-order : IRPR1L (Interrupt Request Peripheral Read register 1L)

The bit configuration of the interrupt request batch read register 1 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #22)

■ IRPR1L : Address 041B$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RXIR1	ISIR1	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RXIR1 (multifunction serial interface RX Interrupt Request 1) : Multi-function serial interface ch.1 reception completion interrupt request

[bit6] ISIR1 (multifunction 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.

4.5. Interrupt Request Batch Read Register 3 upper-order : IRPR3H (Interrupt Request Peripheral Read register 3H)

The bit configuration of the interrupt request batch read register 3 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #40)

■ IRPR3H : Address $041E_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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 Request10) : 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.

4.6. Interrupt Request Batch Read Register 3 lower-order : IRPR3L (Interrupt Request Peripheral Read register 3L)

The bit configuration of the interrupt request batch read register 3 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #41)

■ IRPR3L : Address 041F$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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.

4.7. Interrupt Request Batch Read Register 4 upper-order : IRPR4H (Interrupt Request Peripheral Read register 4H)

The bit configuration of the interrupt request batch read register 4 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #42)

■ IRPR4H : Address 0420$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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.

4.8. Interrupt Request Batch Read Register 4 lower-order : IRPR4L (Interrupt Request Peripheral Read register 4L)

The bit configuration of the interrupt request batch read register 4 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #43)

■ IRPR4L : Address 0421_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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.

4.9. Interrupt Request Batch Read Register 5 upper-order : IRPR5H (Interrupt Request Peripheral Read register 5H)

The bit configuration of the interrupt request batch read register 5 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #44)

■ IRPR5H : Address 0422_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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.

4.10. Interrupt Request Batch Read Register 5 lower-order : IRPR5L (Interrupt Request Peripheral Read register 5L)

The bit configuration of the interrupt request batch read register 5 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #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.

4.11. Interrupt Request Batch Read Register 6 upper-order : IRPR6H (Interrupt Request Peripheral Read register 6H)

The bit configuration of the interrupt request batch read register 6 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #45)

■ IRPR6H : Address 0424_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		RXIR8	ISIR8	Reserved			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit5] RXIR8 (multifunction serial interface 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

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.12. Interrupt Request Batch Read Register 6 lower-order : IRPR6L (Interrupt Request Peripheral Read register 6L)

The bit configuration of the interrupt request batch read register 6 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #46)

■ IRPR6L : Address 0425_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MTIR	STIR	PTIR	TXIR8	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] 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) : Multi-function serial interface ch.8 transmission completion interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.13. Interrupt Request Batch Read Register 7 upper-order : IRPR7H (Interrupt Request Peripheral Read register 7H)

The bit configuration of the interrupt request batch read register 7 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #47)

■ IRPR7H : Address 0426_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	SUBIR	Reserved	RXIR9	ISIR9	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX

[bit6] SUBIR (SUB Interrupt Request) : Clock calibration (sub) 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.

4.14. Interrupt Request Batch Read Register 7 lower-order : IRPR7L (Interrupt Request Peripheral Read register 7L)

The bit configuration of the interrupt request batch read register 7 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #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.

4.15. Interrupt Request Batch Read Register 8 upper-order : IRPR8H (Interrupt Request Peripheral Read register 8H)

The bit configuration of the interrupt request batch read register 8 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #50)

■ IRPR8H : Address 0428_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		FRTIR4	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[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.

4.16. Interrupt Request Batch Read Register 8 lower-order : IRPR8L (Interrupt Request Peripheral Read register 8L)

The bit configuration of the interrupt request batch read register 8 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #51)

■ IRPR8L : Address 0429_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	FRTIR3	FRTIR5	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[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.

4.17. Interrupt Request Batch Read Register 9 upper-order : IRPR9H (Interrupt Request Peripheral Read register 9H)

The bit configuration of the interrupt request batch read register 9 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #52)

■ IRPR9H : Address $042A_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ICUIR6	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] 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.

4.18. 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 shown.

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
	Reserved	ICUIR7	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] 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.

4.19. Interrupt Request Batch Read Register 10 upper-order : IRPR10H (Interrupt Request Peripheral Read register 10H)

The bit configuration of the interrupt request batch read register 10 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #54)

■ IRPR10H : Address $042C_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ICUIR8	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] 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.

4.20. Interrupt Request Batch Read Register 10 lower-order : IRPR10L (Interrupt Request Peripheral Read register 10L)

The bit configuration of the interrupt request batch read register 10 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #55)

■ **IRPR10L : Address 042D$_H$ (Access : Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ICUIR9	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] 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.

4.21. Interrupt Request Batch Read Register 11 upper-order : IRPR11H (Interrupt Request Peripheral Read register 11H)

The bit configuration of the interrupt request batch read register 11 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #56)

■ **IRPR11H : Address 042E$_H$ (Access : Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR4	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,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

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.22. Interrupt Request Batch Read Register 11 lower-order : IRPR11L (Interrupt Request Peripheral Read register 11L)

The bit configuration of the interrupt request batch read register 11 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #57)

■ IRPR11L : Address 042F$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR5	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,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

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.23. Interrupt Request Batch Read Register 12 upper-order : IRPR12H (Interrupt Request Peripheral Read register 12H)

The bit configuration of the interrupt request batch read register 12 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #58)

■ IRPR12H : Address 0430$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		OCUIR6	OCUIR7	OCUIR10	OCUIR11	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX

[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.

4.24. Interrupt Request Batch Read Register 12 lower-order : IRPR12L (Interrupt Request Peripheral Read register 12L)

The bit configuration of the interrupt request batch read register 12 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #59)

■ IRPR12L : Address 0431_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	OCUIR8	OCUIR9	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R0,WX	R0,WX

[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.

4.25. Interrupt Request Batch Read Register 13 upper-order : IRPR13H (Interrupt Request Peripheral Read register 13H)

The bit configuration of the interrupt request batch read register 13 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #60)

■ IRPR13H : Address 0432_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT0IR0	BT0IR1	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT0IR0 (BT0 Interrupt Request 0) : Base timer ch.0 interrupt request 0
[bit6] BT0IR1 (BT0 Interrupt Request 1) : Base timer ch.0 interrupt request 1

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.26. Interrupt Request Batch Read Register 13 lower-order : IRPR13L (Interrupt Request Peripheral Read register 13L)

The bit configuration of the interrupt request batch read register 13 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #61)

■ IRPR13L : Address 0433_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT1IR0	BT1IR1	Reserved	Reserved	Reserved	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT1IR0 (BT1 Interrupt Request 0) : Base timer ch.1 interrupt request 0
[bit6] BT1IR1 (BT1 Interrupt Request 1) : Base timer ch.1 interrupt request 1

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.27. Interrupt Request Batch Read Register 14 upper-order : IRPR14H (Interrupt Request Peripheral Read register 14H)

The bit configuration of the interrupt request batch read register 14 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #62)

■ IRPR14H : Address 0434_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMAC0IR	DMAC1IR	DMAC2IR	DMAC3IR	DMAC4IR	DMAC5IR	DMAC6IR	DMAC7IR
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] DMAC0IR (DMAC 0 Interrupt Request) : DMAC ch.0 interrupt request
[bit6] DMAC1IR (DMAC 1 Interrupt Request) : DMAC ch.1 interrupt request
[bit5] DMAC2IR (DMAC 2 Interrupt Request) : DMAC ch.2 interrupt request
[bit4] DMAC3IR (DMAC 3 Interrupt Request) : DMAC ch.3 interrupt request
[bit3] DMAC4IR (DMAC 4 Interrupt Request) : DMAC ch.4 interrupt request
[bit2] DMAC5IR (DMAC 5 Interrupt Request) : DMAC ch.5 interrupt request

[bit1] DMAC6IR (DMAC 6 Interrupt Request) : DMAC ch.6 interrupt request
[bit0] DMAC7IR (DMAC 7 Interrupt Request) : DMAC ch.7 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

4.28. 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 shown.

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.

4.29. Interrupt Request Batch Read Register 15 upper-order : IRPR15H (Interrupt Request Peripheral Read register 15H)

The bit configuration of the interrupt request batch read register 15 upper-order is shown.

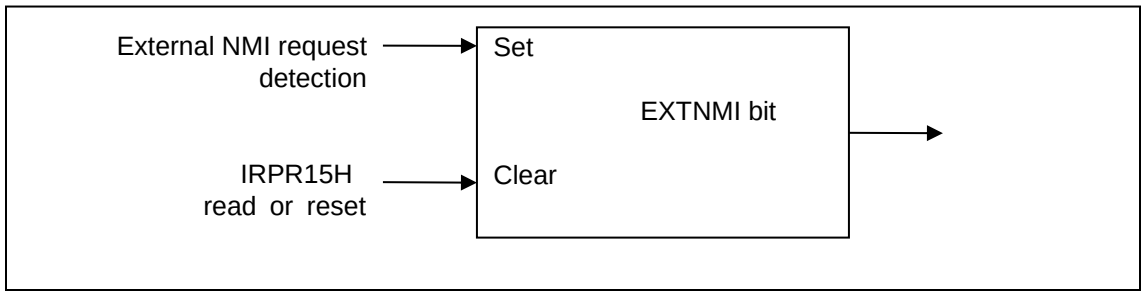
This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #15)

■ IRPR15H : Address 0436_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EXTNMI	XB_ECC_DE	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.



[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.

An internal reset is issued at the return from standby mode (power-shutdown), no NMI request can be maintained.

4.30. Interrupt Request Batch Read Register 15 lower-order : IRPR15L (Interrupt Request Peripheral Read register 15L)

The bit configuration of the interrupt request batch read register 15 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #35)

■ IRPR15L : Address 0437_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CANIR1	XBTC	XBIC	XBTE	BRTC	BRIC	BRTE	Reserved
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	R0,WX

[bit7] CANIR1(CAN Interrupt Request 1) : CAN ch.1 interrupt request

[bit6] XBTC(XBs ram Test Completed interrupt request) : XBS RAM test completed interrupt request

[bit5] XBIC(XBs ram Initialization Completed interrupt request) : XBS RAM initialization completed request interrupt request

[bit4] XBTE(XBs ram Test Error interrupt request) : XBS RAM test error interrupt request

[bit3] BRTC(Backup RAM Test Completed interrupt request) : Backup RAM test completed interrupt request

[bit2] BRIC(Backup RAM Initialization Completed interrupt request) : Backup RAM initialization completed interrupt request

[bit1] BRTE(Backup RAM Test Error interrupt request) : Backup RAM test error interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

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.

Chapter 17: PPG

This chapter explains the PPG.

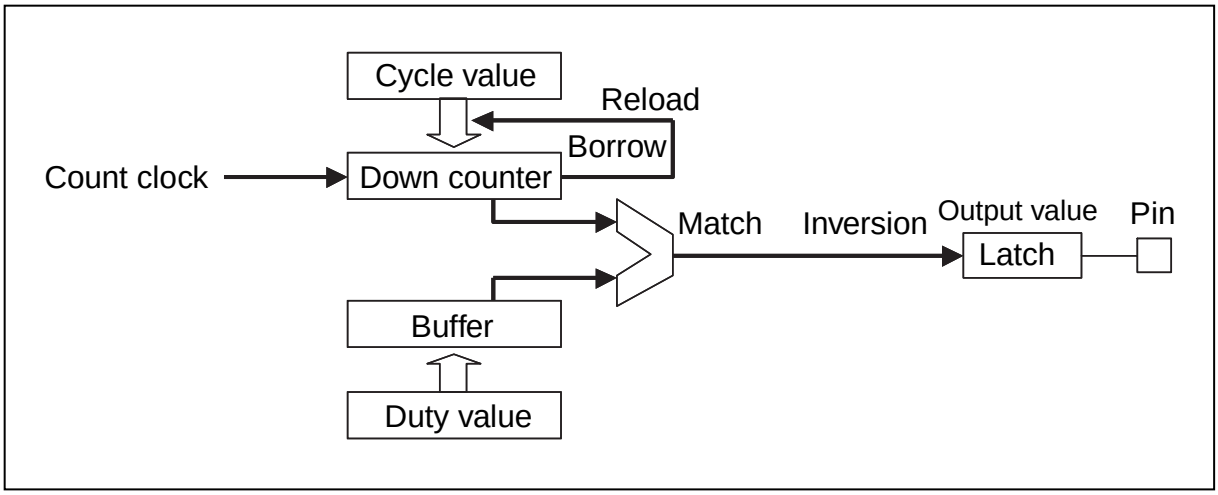
1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

Code : FS30-4v3-91520-2-E

1. Overview

This section explains the overview of the PPG.

The programmable pulse generator (PPG) is used to obtain one-shot (rectangular wave) or pulse width modulation (PWM) outputs. The PPG can easily adapt itself to a wide range of applications because the cycle and duty of its output can be programmed by software.



The numbers of available external output pins are shown below:

MB91F52xB (64pin) : 21
MB91F52xD (80pin) : 27
MB91F52xF (100pin) : 34
MB91F52xJ (120pin) : 38
MB91F52xK (144pin) : 44
MB91F52xL (176pin) : 48

2. Features

This section explains features of the PPG.

● Clamp output

- Normal polarity: Output clamped to "L"
- Inverted polarity: Output clamped to "H"

● Count clock

- One of the following 4 count clocks is selected:
  Outputs obtained by dividing the frequency of the peripheral clock by 1, 4, 16, and 64.

Cycle

- Setting range = Duty value to 65535 (specified by a 16-bit register)
- Cycle = Count clock × (PCSR register value + 1)
  (Example) Count clock = 32 MHz (31.25 ns), PCSR value = 63999
  Cycle = 31.25ns × (63999+1) = 2ms

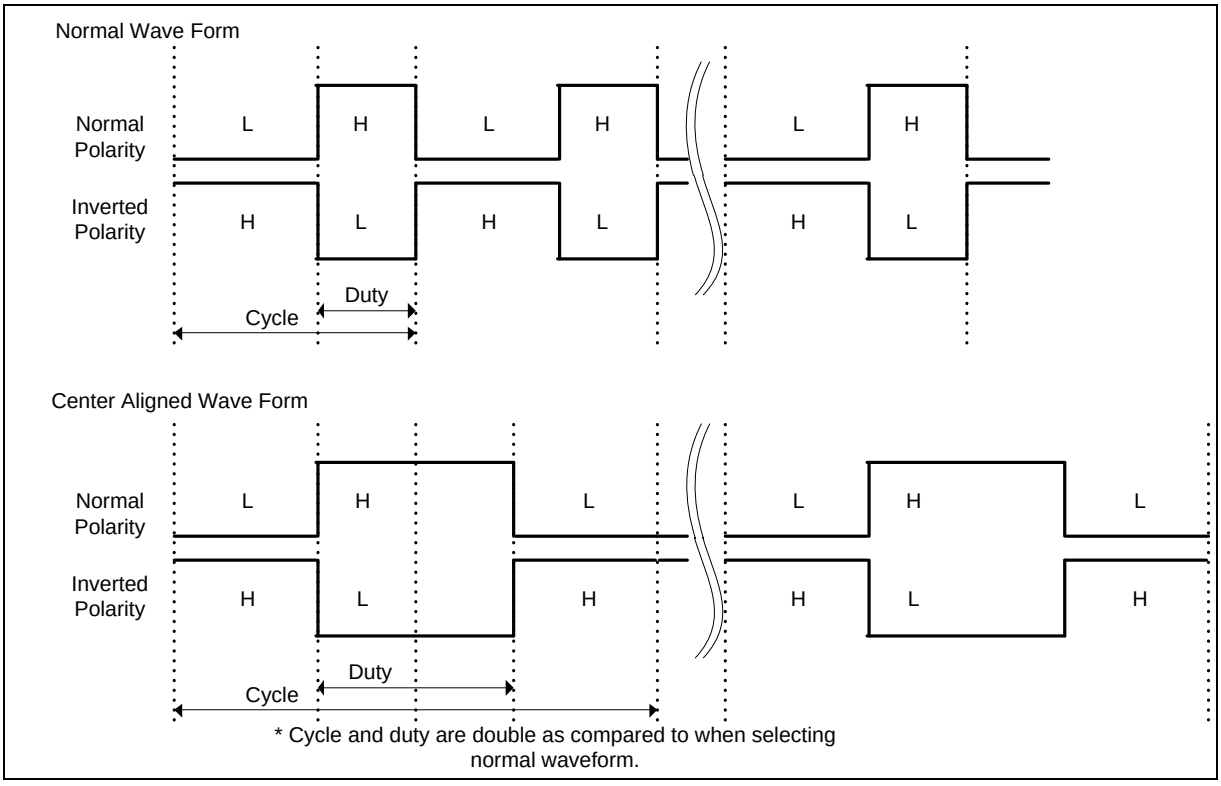
Note: Cycle Setting (PHCSR/PLCSR) for the High/Low format at the PPG communication mode is also similar.

Duty

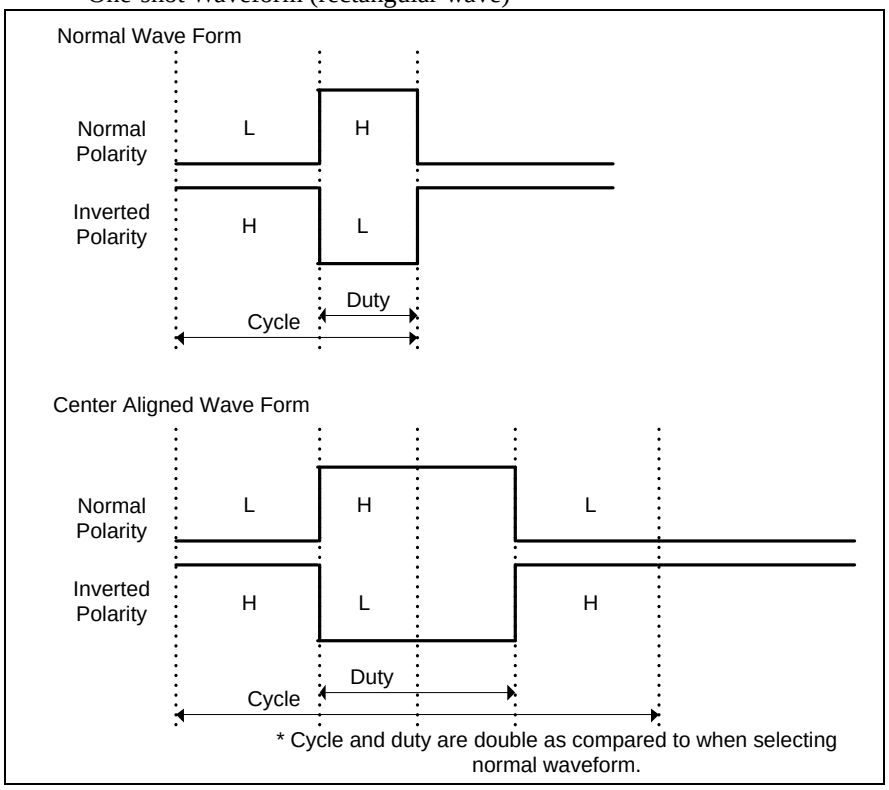
- Setting range = 0 to cycle value (specified by a 16-bit register)
- Duty = Count clock (PDUT register value + 1)
  Note: Duty setting (PHDUT/PLDUT) for the High/Low format at the PPG communication mode is also similar.

Output Waveforms

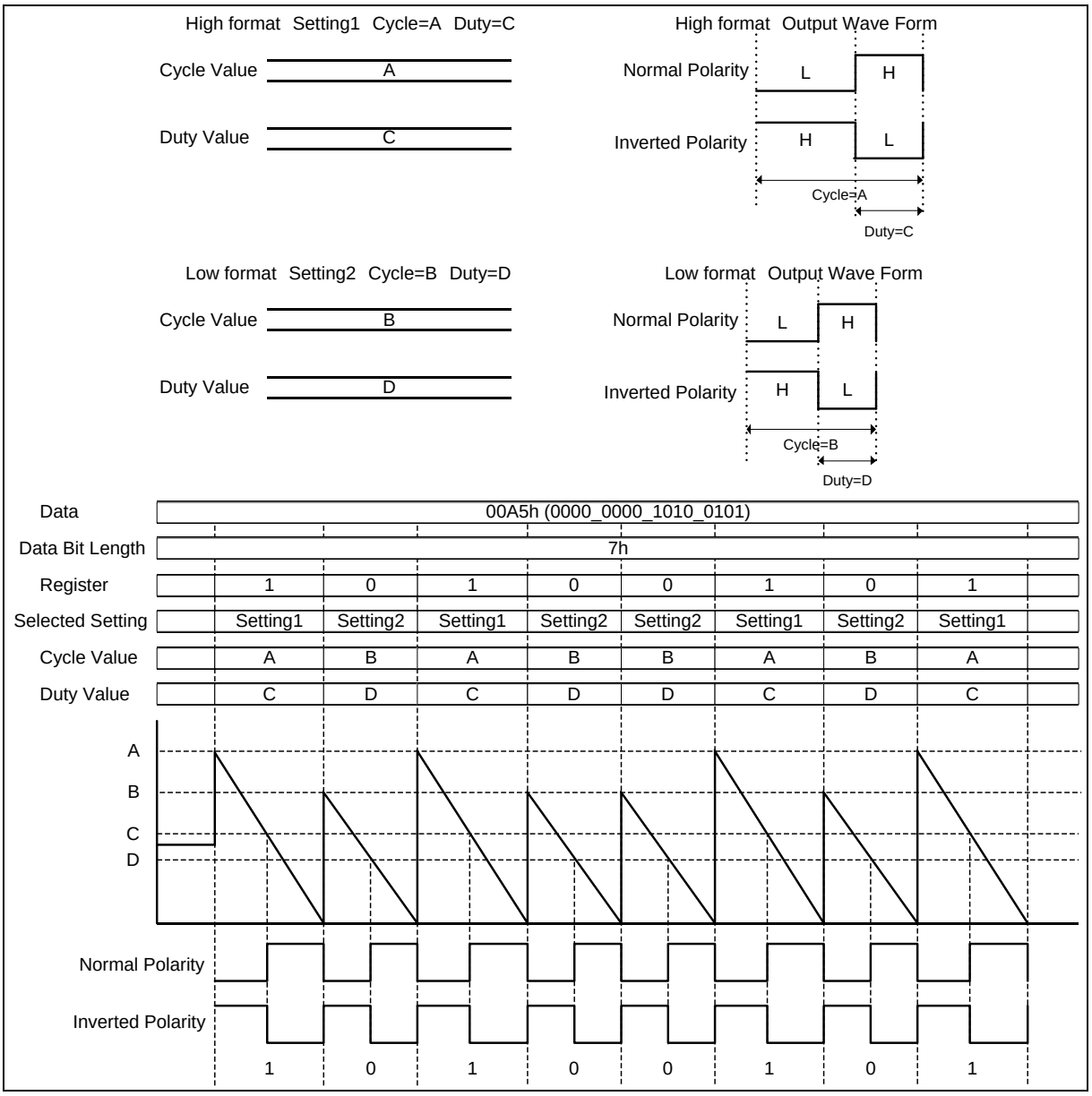
- PWM Waveform



- One-shot Waveform (rectangular wave)



- High/Low format Waveform
  The cycle of the High format and the Low format and duty are set respectively, and outputting waveform is changed according to the data setting.



Interrupt factors

- One of the following six interrupts is selected:
  - Software trigger or external trigger (TRG pin)
  - 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
  - Timing Point Capture match
  - Empty flag of PPG communication data register

Activation triggers

- Software trigger
- External trigger (TRG pin)

  The activation trigger is input from an external.

  The activation trigger is selected from one of the following triggers:
  - Internal trigger (EN0 to EN47)
  - External trigger (TRG pins 0 to 11)
  - Reload timer 0/1

GATE function

PPG is activated/stopped by GATE signals from the waveform generator.

Start Delay Mode

- Support for PWM, One-shot operation, Normal Wave Form, and Center Aligned Wave Form.
- Setting range = 0 to 65535 (specified by a 16-bit register)
- Delay range = Count Clock × (PSDR Resister value + 1)

  (Example) Normal Wave Form: Count Clock = 32MHz (31.25ns), PSDR=63999

  Cycle = 31.25ns × (63999 + 1) = 2ms

  (Example) Center Aligned Wave Form: Count Clock = 32MHz (31.25ns), PSDR=63999

  Cycle = 31.25ns × {(63999 + 1) × 2} = 4ms

Timing Point Capture Mode

The AD activation trigger is generated according to the timing of the Timing Point Capture setting value.

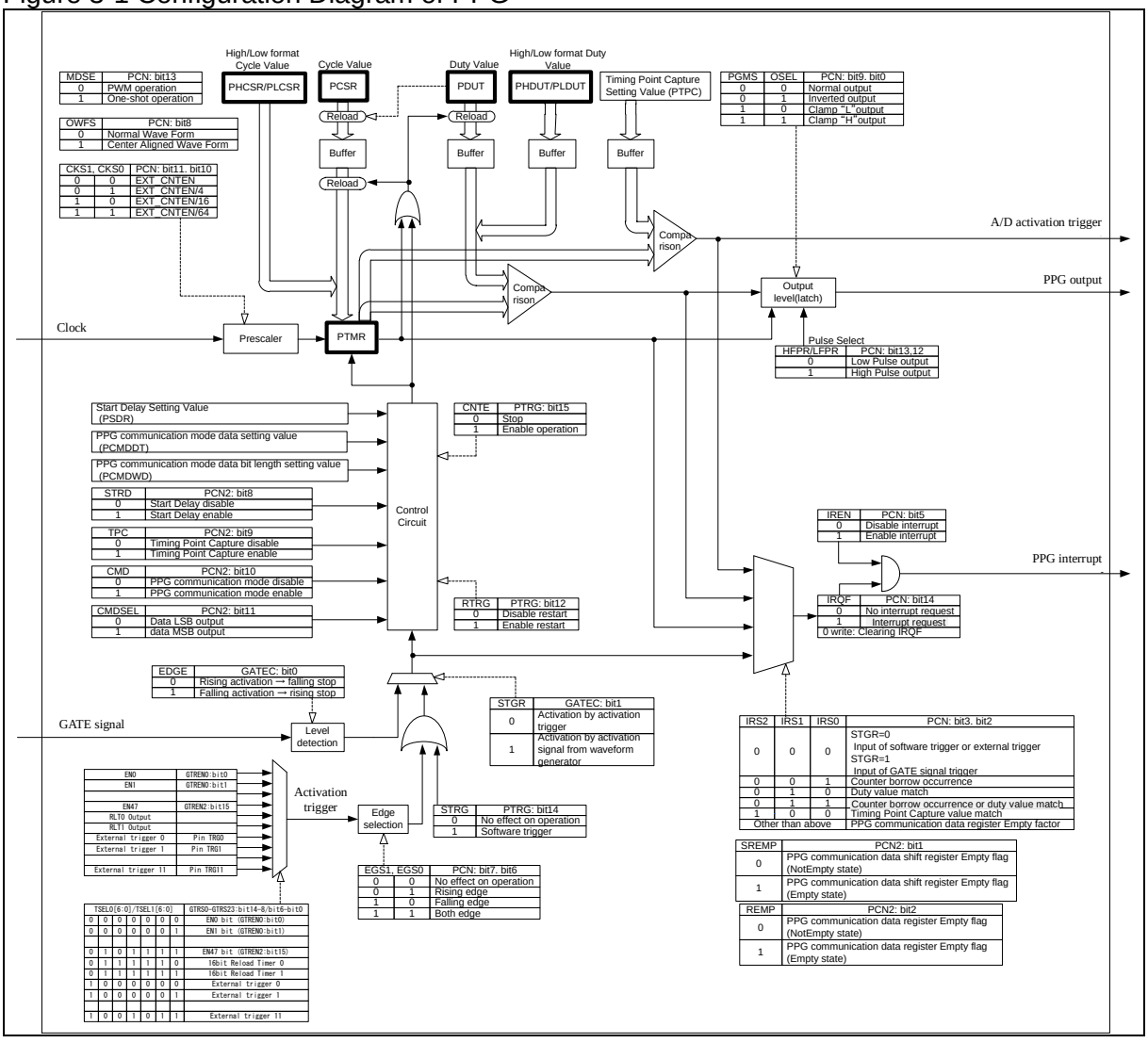
PPG communication Mode

Cycle of High format and Low format, and setting of duty.

3. Configuration

This section explains the configuration of the PPG.

Figure 3-1 Configuration Diagram of PPG



MDSE	PCN: bit13
0	PWM operation
1	One-shot operation

PGMS	OSEL	PCN: bit9, bit0
0	0	Normal output
0	1	Inverted output
1	0	Clamp "L" output
1	1	Clamp "H" output

OWFS	PCN: bit8
0	Normal Wave Form
1	Center Aligned Wave Form

CKS1, CKS0	PCN: bit11, bit10
0 0	EXT_CNTEN
0 1	EXT_CNTEN/4
1 0	EXT_CNTEN/16
1 1	EXT_CNTEN/64

HFPR/LFPR	PCN: bit13,12
0	Low Pulse output
1	High Pulse output

CNTE	PTRG: bit15
0	Stop
1	Enable operation

STRD	PCN2: bit8
0	Start Delay disable
1	Start Delay enable

TPC	PCN2: bit9
0	Timing Point Capture disable
1	Timing Point Capture enable

CMD	PCN2: bit10
0	PPG communication mode disable
1	PPG communication mode enable

CMDSEL	PCN2: bit11
0	Data LSB output
1	data MSB output

RTRG	PTRG: bit12
0	Disable restart
1	Enable restart

IREN	PCN: bit5
0	Disable interrupt
1	Enable interrupt

IRQF	PCN: bit14
0	No interrupt request
1	Interrupt request
0 write: Clearing IRQF	

EDGE	GATEC: bit0
0	Rising activation → falling stop
1	Falling activation → rising stop

STGR	GATEC: bit1
0	Activation by activation trigger
1	Activation by activation signal from waveform generator

IRS2	IRS1	IRS0	PCN: bit3, bit2
0	0	0	STGR=0: Input of software trigger or external trigger STGR=1: Input of GATE signal trigger
0	0	1	Counter borrow occurrence
0	1	0	Duty value match
0	1	1	Counter borrow occurrence or duty value match
1	0	0	Timing Point Capture value match
Other than above			PPG communication data register Empty factor

STRG	PTRG: bit14
0	No effect on operation
1	Software trigger

EGS1, EGS0	PCN: bit7, bit6	
0	0	No effect on operation
0	1	Rising edge
1	0	Falling edge
1	1	Both edge

SREMP	PCN2: bit1
0	PPG communication data shift register Empty flag (NotEmpty state)
1	PPG communication data shift register Empty flag (Empty state)

REMP	PCN2: bit2
0	PPG communication data register Empty flag (NotEmpty state)
1	PPG communication data register Empty flag (Empty state)

TSEL0[6:0]/TSEL1[6:0]	GTRS0-GTRS23: bit14-8/bit6-bit0
0 0 0 0 0 0 0	EN0 bit (GTREN0: bit0)
0 0 0 0 0 0 1	EN1 bit (GTREN0: bit1)
0 1 0 1 1 1 1	EN47 bit (GTREN2: bit15)
0 1 1 1 1 1 0	16bit Reload Timer 0
0 1 1 1 1 1 1	16bit Reload Timer 1
1 0 0 0 0 0 0	External trigger 0
1 0 0 0 0 0 1	External trigger 1
1 0 0 1 0 1 1	External trigger 11

4. Registers

This section explains registers of the PPG.

■ List of External Pins

Channel	External Pins (PPG Output)					
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK	MB91F52xL
0	No	No	No	PPG0_1	PPG0_0/ PPG0_1	PPG0_0/ PPG0_1
1	PPG1_0/ PPG1_1	PPG1_0/ PPG1_1	PPG1_0/ PPG1_1	PPG1_0/ PPG1_1	PPG1_0/ PPG1_1	PPG1_0/ PPG1_1
2	PPG2_0	PPG2_0	PPG2_0/ PPG2_1	PPG2_0/ PPG2_1	PPG2_0/ PPG2_1	PPG2_0/ PPG2_1
3	PPG3_1	PPG3_1	PPG3_1	PPG3_1	PPG3_0/ PPG3_1	PPG3_0/ PPG3_1
4	PPG4_1	PPG4_1	PPG4_1	PPG4_1	PPG4_0/ PPG4_1	PPG4_0/ PPG4_1
5	No	PPG5_1	PPG5_1	PPG5_1	PPG5_0/ PPG5_1	PPG5_0/ PPG5_1
6	No	No	PPG6_0	PPG6_0	PPG6_0	PPG6_0
7	PPG7_0	PPG7_0	PPG7_0	PPG7_0	PPG7_0	PPG7_0
8	No	PPG8_0	PPG8_0	PPG8_0	PPG8_0	PPG8_0
9	No	No	PPG9_0	PPG9_0	PPG9_0	PPG9_0
10	PPG10_0	PPG10_0	PPG10_0	PPG10_0	PPG10_0	PPG10_0
11	PPG11_0	PPG11_0	PPG11_0	PPG11_0	PPG11_0	PPG11_0
12	PPG12_0	PPG12_0	PPG12_0	PPG12_0	PPG12_0	PPG12_0
13	PPG13_0	PPG13_0	PPG13_0	PPG13_0	PPG13_0	PPG13_0
14	No	No	PPG14_0	PPG14_0	PPG14_0	PPG14_0
15	No	PPG15_0	PPG15_0	PPG15_0	PPG15_0	PPG15_0
16	PPG16_1	PPG16_1	PPG16_1	PPG16_0/ PPG16_1	PPG16_0/ PPG16_1	PPG16_0/ PPG16_1
17	PPG17_1	PPG17_1	PPG17_1	PPG17_0/ PPG17_1	PPG17_0/ PPG17_1	PPG17_0/ PPG17_1
18	No	PPG18_0	PPG18_0	PPG18_0	PPG18_0	PPG18_0
19	PPG19_0	PPG19_0	PPG19_0	PPG19_0	PPG19_0	PPG19_0
20	PPG20_0	PPG20_0	PPG20_0	PPG20_0	PPG20_0	PPG20_0
21	No	No	PPG21_0	PPG21_0	PPG21_0	PPG21_0
22	No	No	No	PPG22_0	PPG22_0	PPG22_0
23	PPG23_1	PPG23_1	PPG23_1	PPG23_1	PPG23_0/ PPG23_1	PPG23_0/ PPG23_1
24	PPG24_0	PPG24_0	PPG24_0	PPG24_0	PPG24_0	PPG24_0/ PPG24_1
25	No	No	PPG25_0	PPG25_0	PPG25_0	PPG25_0/ PPG25_1
26	No	PPG26_0	PPG26_0	PPG26_0	PPG26_0	PPG26_0/ PPG26_1

Channel	External Pins (PPG Output)					
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK	MB91F52xL
27	PPG27_0	PPG27_0	PPG27_0	PPG27_0	PPG27_0	PPG27_0/ PPG27_1
28	No	No	PPG28_0	PPG28_0	PPG28_0	PPG28_0/ PPG28_1
29	No	PPG29_0	PPG29_0	PPG29_0	PPG29_0	PPG29_0/ PPG29_1
30	PPG30_0	PPG30_0	PPG30_0	PPG30_0	PPG30_0	PPG30_0/ PPG30_1
31	PPG31_0	PPG31_0	PPG31_0	PPG31_0	PPG31_0	PPG31_0/ PPG31_1
32	No	No	No	PPG32_0	PPG32_0	PPG32_0/ PPG32_1
33	No	No	No	PPG33_0	PPG33_0	PPG33_0/ PPG33_1
34	No	No	No	No	PPG34_0	PPG34_0/ PPG34_1
35	PPG35_0	PPG35_0	PPG35_0	PPG35_0	PPG35_0	PPG35_0/ PPG35_1
36	No	No	No	No	PPG36_0	PPG36_0/ PPG36_1
37	PPG37_0	PPG37_0	PPG37_0	PPG37_0	PPG37_0	PPG37_0/ PPG37_1
38	No	No	No	No	PPG38_0	PPG38_0/ PPG38_1
39	No	No	No	No	PPG39_0	PPG39_0/ PPG39_1
40	No	No	No	No	PPG40_1	PPG40_0/ PPG40_1
41	No	No	No	No	PPG41_1	PPG41_0/ PPG41_1
42	No	No	No	No	No	PPG42_0
43	PPG43_1	PPG43_1	PPG43_1	PPG43_1	PPG43_1	PPG43_0/ PPG43_1
44	No	No	PPG44_1	PPG44_1	PPG44_1	PPG44_0/ PPG44_1
45	No	No	No	No	No	PPG45_0
46	No	No	No	No	No	PPG46_0
47	No	No	No	No	No	PPG47_0

Channel	External Pins (PPG Trigger Input)					
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK	MB91F52xL
0	TRG0_1/ TRG0_2	TRG0_1/ TRG0_2	TRG0_1/ TRG0_2	TRG0_1/ TRG0_2	TRG0_0/ TRG0_1/ TRG0_2	TRG0_0/ TRG0_1/ TRG0_2
1	No	No	TRG1_1	TRG1_1	TRG1_0/ TRG1_1	TRG1_0/ TRG1_1
2	No	TRG2_1	TRG2_1	TRG2_1	TRG2_0/ TRG2_1	TRG2_0/ TRG2_1
3	TRG3_0/ TRG3_1	TRG3_0/ TRG3_1	TRG3_0/ TRG3_1	TRG3_0/ TRG3_1	TRG3_0/ TRG3_1	TRG3_0/ TRG3_1
4	No	No	No	TRG4_0/ TRG4_1	TRG4_0/ TRG4_1	TRG4_0/ TRG4_1
5	No	No	No	TRG5_0/ TRG5_1	TRG5_0/ TRG5_1	TRG5_0/ TRG5_1/ TRG5_2
6	No	No	TRG6_0	TRG6_0/ TRG6_1	TRG6_0/ TRG6_1	TRG6_0/ TRG6_1/ TRG6_2
7	TRG7_0	TRG7_0	TRG7_0	TRG7_0/ TRG7_1	TRG7_0/ TRG7_1	TRG7_0/ TRG7_1
8	TRG8_0	TRG8_0	TRG8_0	TRG8_0	TRG8_0	TRG8_0/ TRG8_1
9	No	No	No	No	TRG9_0	TRG9_0/ TRG9_1
10	No	No	No	No	No	TRG10_0
11	No	No	No	No	No	TRG11_0

The registers of PPG are listed below.

■ List of PPG Registers

Address	+0	+1	+2	+3
0x1A40	PPG (ch.0) control status register (PCN0)		PPG (ch.0) cycle setting register (PCSR0)	
0x1A44	PPG (ch.0) duty setting register (PDUT0)		PPG (ch.0) timer register (PTMR0)	
0x1A48	PPG (ch.0) control status register2 (PCN200)		PPG (ch.0) Start Delay value setting register (PSDR0)	
0x1A4C	PPG (ch.0) Timing Point Capture value setting register (PTPC0)		PPG (ch.0) communication mode data bit length setting register (PCMDWD0)	
0x1A50	PPG (ch.0) communication mode High format cycle setting register (PHCSR0)		PPG (ch.0) communication mode Low format cycle setting register (PLCSR0)	
0x1A54	PPG (ch.0) communication mode High format duty setting register (PHDUT0)		PPG (ch.0) communication mode Low format duty setting register (PLDUT0)	
0x1A58	PPG (ch.0) communication mode data setting register (PCMDDT0)		Reserved	
0x1A5C	PPG (ch.1) control status register (PCN1)		PPG (ch.1) cycle setting register (PCSR1)	
0x1A60	PPG (ch.1) duty setting register (PDUT1)		PPG (ch.1) timer register (PTMR1)	
0x1A64	PPG (ch.1) control status register2 (PCN201)		PPG (ch.1) Start Delay value setting register (PSDR1)	
0x1A68	PPG (ch.1) Timing Point Capture value setting register (PTPC1)		PPG (ch.1) communication mode data bit length setting register (PCMDWD1)	
0x1A6C	PPG (ch.1) communication mode High format cycle setting register (PHCSR1)		PPG (ch.1) communication mode Low format cycle setting register (PLCSR1)	
0x1A70	PPG (ch.1) communication mode High format duty setting register (PHDUT1)		PPG (ch.1) communication mode Low format duty setting register (PLDUT1)	
0x1A74	PPG (ch.1) communication mode data setting register (PCMDDT1)		Reserved	
0x1A78	PPG (ch.2) control status register (PCN2)		PPG (ch.2) cycle setting register (PCSR2)	
0x1A7C	PPG (ch.2) duty setting register (PDUT2)		PPG (ch.2) timer register (PTMR2)	
0x1A80	PPG (ch.2) control status register2 (PCN202)		PPG (ch.2) Start Delay value setting register (PSDR2)	
0x1A84	PPG (ch.2) Timing Point Capture value setting register (PTPC2)		PPG (ch.2) communication mode data bit length setting register (PCMDWD2)	
0x1A88	PPG (ch.2) communication mode High format cycle setting register (PHCSR2)		PPG (ch.2) communication mode Low format cycle setting register (PLCSR2)	
0x1A8C	PPG (ch.2) communication mode High format duty setting register (PHDUT2)		PPG (ch.2) communication mode Low format duty setting register (PLDUT2)	
0x1A90	PPG (ch.2) communication mode data setting register (PCMDDT2)		Reserved	
0x1A94	PPG (ch.3) control status register (PCN3)		PPG (ch.3) cycle setting register (PCSR3)	
0x1A98	PPG (ch.3) duty setting register (PDUT3)		PPG (ch.3) timer register (PTMR3)	
0x1A9C	PPG (ch.3) control status register2 (PCN203)		PPG (ch.3) Start Delay value setting register (PSDR3)	
0x1AA0	PPG (ch.3) Timing Point Capture value setting register (PTPC3)		PPG (ch.3) communication mode data bit length setting register (PCMDWD3)	
0x1AA4	PPG (ch.3) communication mode High format cycle setting register (PHCSR3)		PPG (ch.3) communication mode Low format cycle setting register (PLCSR3)	
0x1AA8	PPG (ch.3) communication mode High format duty setting register (PHDUT3)		PPG (ch.3) communication mode Low format duty setting register (PLDUT3)	

Address	+0	+1	+2	+3
0x1AAC	PPG (ch.3) communication mode data setting register (PCMDDT3)		Reserved	
0x1AB0	PPG (ch.4) control status register (PCN4)		PPG (ch.4) cycle setting register (PCSR4)	
0x1AB4	PPG (ch.4) duty setting register (PDUT4)		PPG (ch.4) timer register (PTMR4)	
0x1AB8	PPG (ch.4) control status register2 (PCN204)		PPG (ch.4) Start Delay value setting register (PSDR4)	
0x1ABC	PPG (ch.4) Timing Point Capture value setting register (PTPC4)		Reserved	
0x1AC0	PPG (ch.5) control status register (PCN5)		PPG (ch.5) cycle setting register (PCSR5)	
0x1AC4	PPG (ch.5) duty setting register (PDUT5)		PPG (ch.5) timer register (PTMR5)	
0x1AC8	PPG (ch.5) control status register2 (PCN205)		PPG (ch.5) Start Delay value setting register (PSDR5)	
0x1ACC	PPG (ch.5) Timing Point Capture value setting register (PTPC5)		Reserved	
0x1AD0	PPG (ch.6) control status register (PCN6)		PPG (ch.6) cycle setting register (PCSR6)	
0x1AD4	PPG (ch.6) duty setting register (PDUT6)		PPG (ch.6) timer register (PTMR6)	
0x1AD8	PPG (ch.6) control status register2 (PCN206)		PPG (ch.6) Start Delay value setting register (PSDR6)	
0x1ADC	PPG (ch.6) Timing Point Capture value setting register (PTPC6)		Reserved	
0x1AE0	PPG (ch.7) control status register (PCN7)		PPG (ch.7) cycle setting register (PCSR7)	
0x1AE4	PPG (ch.7) duty setting register (PDUT7)		PPG (ch.7) timer register (PTMR7)	
0x1AE8	PPG (ch.7) control status register2 (PCN207)		PPG (ch.7) Start Delay value setting register (PSDR7)	
0x1AEC	PPG (ch.7) Timing Point Capture value setting register (PTPC7)		Reserved	
0x1AF0	PPG (ch.8) control status register (PCN8)		PPG (ch.8) cycle setting register (PCSR8)	
0x1AF4	PPG (ch.8) duty setting register (PDUT8)		PPG (ch.8) timer register (PTMR8)	
0x1AF8	PPG (ch.8) control status register2 (PCN208)		PPG (ch.8) Start Delay value setting register (PSDR8)	
0x1AFC	PPG (ch.8) Timing Point Capture value setting register (PTPC8)		Reserved	
0x1B00	PPG (ch.9) control status register (PCN9)		PPG (ch.9) cycle setting register (PCSR9)	
0x1B04	PPG (ch.9) duty setting register (PDUT9)		PPG (ch.9) timer register (PTMR9)	
0x1B08	PPG (ch.9) control status register2 (PCN209)		PPG (ch.9) Start Delay value setting register (PSDR9)	
0x1B0C	PPG (ch.9) Timing Point Capture value setting register (PTPC9)		Reserved	
0x1B10	PPG (ch.10) control status register (PCN10)		PPG (ch.10) cycle setting register (PCSR10)	
0x1B14	PPG (ch.10) duty setting register (PDUT10)		PPG (ch.10) timer register (PTMR10)	
0x1B18	PPG (ch.10) control status register2 (PCN210)		PPG (ch.10) Start Delay value setting register (PSDR10)	
0x1B1C	PPG (ch.10) Timing Point Capture value setting register (PTPC10)		Reserved	
0x1B20	PPG (ch.11) control status register (PCN11)		PPG (ch.11) cycle setting register (PCSR11)	
0x1B24	PPG (ch.11) duty setting register (PDUT11)		PPG (ch.11) timer register (PTMR11)	
0x1B28	PPG (ch.11) control status register2 (PCN211)		PPG (ch.11) Start Delay value setting register (PSDR11)	

Address	+0	+1	+2	+3
0x1B2C	PPG (ch.11) Timing Point Capture value setting register (PTPC11)		Reserved	
0x1B30	PPG (ch.12) control status register (PCN12)		PPG (ch.12) cycle setting register (PCSR12)	
0x1B34	PPG (ch.12) duty setting register (PDUT12)		PPG (ch.12) timer register (PTMR12)	
0x1B38	PPG (ch.12) control status register2 (PCN212)		PPG (ch.12) Start Delay value setting register (PSDR12)	
0x1B3C	PPG (ch.12) Timing Point Capture value setting register (PTPC12)		Reserved	
0x1B40	PPG (ch.13) control status register (PCN13)		PPG (ch.13) cycle setting register (PCSR13)	
0x1B44	PPG (ch.13) duty setting register (PDUT13)		PPG (ch.13) timer register (PTMR13)	
0x1B48	PPG (ch.13) control status register2 (PCN213)		PPG (ch.13) Start Delay value setting register (PSDR13)	
0x1B4C	PPG (ch.13) Timing Point Capture value setting register (PTPC13)		Reserved	
0x1B50	PPG (ch.14) control status register (PCN14)		PPG (ch.14) cycle setting register (PCSR14)	
0x1B54	PPG (ch.14) duty setting register (PDUT14)		PPG (ch.14) timer register (PTMR14)	
0x1B58	PPG (ch.14) control status register2 (PCN214)		PPG (ch.14) Start Delay value setting register (PSDR14)	
0x1B5C	PPG (ch.14) Timing Point Capture value setting register (PTPC14)		Reserved	
0x1B60	PPG (ch.15) control status register (PCN15)		PPG (ch.15) cycle setting register (PCSR15)	
0x1B64	PPG (ch.15) duty setting register (PDUT15)		PPG (ch.15) timer register (PTMR15)	
0x1B68	PPG (ch.15) control status register2 (PCN215)		PPG (ch.15) Start Delay value setting register (PSDR15)	
0x1B6C	PPG (ch.15) Timing Point Capture value setting register (PTPC15)		Reserved	
0x1B70	PPG (ch.16) control status register (PCN16)		PPG (ch.16) cycle setting register (PCSR16)	
0x1B74	PPG (ch.16) duty setting register (PDUT16)		PPG (ch.16) timer register (PTMR16)	
0x1B78	PPG (ch.16) control status register2 (PCN216)		PPG (ch.16) Start Delay value setting register (PSDR16)	
0x1B7C	PPG (ch.16) Timing Point Capture value setting register (PTPC16)		Reserved	
0x1B80	PPG (ch.17) control status register (PCN17)		PPG (ch.17) cycle setting register (PCSR17)	
0x1B84	PPG (ch.17) duty setting register (PDUT17)		PPG (ch.17) timer register (PTMR17)	
0x1B88	PPG (ch.17) control status register2 (PCN217)		PPG (ch.17) Start Delay value setting register (PSDR17)	
0x1B8C	PPG (ch.17) Timing Point Capture value setting register (PTPC17)		Reserved	
0x1B90	PPG (ch.18) control status register (PCN18)		PPG (ch.18) cycle setting register (PCSR18)	
0x1B94	PPG (ch.18) duty setting register (PDUT18)		PPG (ch.18) timer register (PTMR18)	
0x1B98	PPG (ch.18) control status register2 (PCN218)		PPG (ch.18) Start Delay value setting register (PSDR18)	
0x1B9C	PPG (ch.18) Timing Point Capture value setting register (PTPC18)		Reserved	
0x1BA0	PPG (ch.19) control status register (PCN19)		PPG (ch.19) cycle setting register (PCSR19)	
0x1BA4	PPG (ch.19) duty setting register (PDUT19)		PPG (ch.19) timer register (PTMR19)	
0x1BA8	PPG (ch.19) control status register2 (PCN219)		PPG (ch.19) Start Delay value setting register (PSDR19)	

Address	+0	+1	+2	+3
0x1BAC	PPG (ch.19) Timing Point Capture value setting register (PTPC19)		Reserved	
0x1BB0	PPG (ch.20) control status register (PCN20)		PPG (ch.20) cycle setting register (PCSR20)	
0x1BB4	PPG (ch.20) duty setting register (PDUT20)		PPG (ch.20) timer register (PTMR20)	
0x1BB8	PPG (ch.20) control status register2 (PCN220)		PPG (ch.20) Start Delay value setting register (PSDR20)	
0x1BBC	PPG (ch.20) Timing Point Capture value setting register (PTPC20)		Reserved	
0x1BC0	PPG (ch.21) control status register (PCN21)		PPG (ch.21) cycle setting register (PCSR21)	
0x1BC4	PPG (ch.21) duty setting register (PDUT21)		PPG (ch.21) timer register (PTMR21)	
0x1BC8	PPG (ch.21) control status register2 (PCN221)		PPG (ch.21) Start Delay value setting register (PSDR21)	
0x1BCC	PPG (ch.21) Timing Point Capture value setting register (PTPC21)		Reserved	
0x1BD0	PPG (ch.22) control status register (PCN22)		PPG (ch.22) cycle setting register (PCSR22)	
0x1BD4	PPG (ch.22) duty setting register (PDUT22)		PPG (ch.22) timer register (PTMR22)	
0x1BD8	PPG (ch.22) control status register2 (PCN222)		PPG (ch.22) Start Delay value setting register (PSDR22)	
0x1BDC	PPG (ch.22) Timing Point Capture value setting register (PTPC22)		Reserved	
0x1BE0	PPG (ch.23) control status register (PCN23)		PPG (ch.23) cycle setting register (PCSR23)	
0x1BE4	PPG (ch.23) duty setting register (PDUT23)		PPG (ch.23) timer register (PTMR23)	
0x1BE8	PPG (ch.23) control status register2 (PCN223)		PPG (ch.23) Start Delay value setting register (PSDR23)	
0x1BEC	PPG (ch.23) Timing Point Capture value setting register (PTPC23)		Reserved	
0x1BF0	PPG (ch.24) control status register (PCN24)		PPG (ch.24) cycle setting register (PCSR24)	
0x1BF4	PPG (ch.24) duty setting register (PDUT24)		PPG (ch.24) timer register (PTMR24)	
0x1BF8	PPG (ch.24) control status register2 (PCN224)		PPG (ch.24) Start Delay value setting register (PSDR24)	
0x1BFC	PPG (ch.24) Timing Point Capture value setting register (PTPC24)		Reserved	
0x1C00	PPG (ch.25) control status register (PCN25)		PPG (ch.25) cycle setting register (PCSR25)	
0x1C04	PPG (ch.25) duty setting register (PDUT25)		PPG (ch.25) timer register (PTMR25)	
0x1C08	PPG (ch.25) control status register2 (PCN225)		PPG (ch.25) Start Delay value setting register (PSDR25)	
0x1C0C	PPG (ch.25) Timing Point Capture value setting register (PTPC25)		Reserved	
0x1C10	PPG (ch.26) control status register (PCN26)		PPG (ch.26) cycle setting register (PCSR26)	
0x1C14	PPG (ch.26) duty setting register (PDUT26)		PPG (ch.26) timer register (PTMR26)	
0x1C18	PPG (ch.26) control status register2 (PCN226)		PPG (ch.26) Start Delay value setting register (PSDR26)	
0x1C1C	PPG (ch.26) Timing Point Capture value setting register (PTPC26)		Reserved	
0x1C20	PPG (ch.27) control status register (PCN27)		PPG (ch.27) cycle setting register (PCSR27)	
0x1C24	PPG (ch.27) duty setting register (PDUT27)		PPG (ch.27) timer register (PTMR27)	
0x1C28	PPG (ch.27) control status register2 (PCN227)		PPG (ch.27) Start Delay value setting register (PSDR27)	

Address	+0	+1	+2	+3
0x1C2C	PPG (ch.27) Timing Point Capture value setting register (PTPC27)		Reserved	
0x1C30	PPG (ch.28) control status register (PCN28)		PPG (ch.28) cycle setting register (PCSR28)	
0x1C34	PPG (ch.28) duty setting register (PDUT28)		PPG (ch.28) timer register (PTMR28)	
0x1C38	PPG (ch.28) control status register2 (PCN228)		PPG (ch.28) Start Delay value setting register (PSDR28)	
0x1C3C	PPG (ch.28) Timing Point Capture value setting register (PTPC28)		Reserved	
0x1C40	PPG (ch.29) control status register (PCN29)		PPG (ch.29) cycle setting register (PCSR29)	
0x1C44	PPG (ch.29) duty setting register (PDUT29)		PPG (ch.29) timer register (PTMR29)	
0x1C48	PPG (ch.29) control status register2 (PCN229)		PPG (ch.29) Start Delay value setting register (PSDR29)	
0x1C4C	PPG (ch.29) Timing Point Capture value setting register (PTPC29)		Reserved	
0x1C50	PPG (ch.30) control status register (PCN30)		PPG (ch.30) cycle setting register (PCSR30)	
0x1C54	PPG (ch.30) duty setting register (PDUT30)		PPG (ch.30) timer register (PTMR30)	
0x1C58	PPG (ch.30) control status register2 (PCN230)		PPG (ch.30) Start Delay value setting register (PSDR30)	
0x1C5C	PPG (ch.30) Timing Point Capture value setting register (PTPC30)		Reserved	
0x1C60	PPG (ch.31) control status register (PCN31)		PPG (ch.31) cycle setting register (PCSR31)	
0x1C64	PPG (ch.31) duty setting register (PDUT31)		PPG (ch.31) timer register (PTMR31)	
0x1C68	PPG (ch.31) control status register2 (PCN231)		PPG (ch.31) Start Delay value setting register (PSDR31)	
0x1C6C	PPG (ch.31) Timing Point Capture value setting register (PTPC31)		Reserved	
0x1C70	PPG (ch.32) control status register (PCN32)		PPG (ch.32) cycle setting register (PCSR32)	
0x1C74	PPG (ch.32) duty setting register (PDUT32)		PPG (ch.32) timer register (PTMR32)	
0x1C78	PPG (ch.32) control status register2 (PCN232)		PPG (ch.32) Start Delay value setting register (PSDR32)	
0x1C7C	PPG (ch.32) Timing Point Capture value setting register (PTPC32)		Reserved	
0x1C80	PPG (ch.33) control status register (PCN33)		PPG (ch.33) cycle setting register (PCSR33)	
0x1C84	PPG (ch.33) duty setting register (PDUT33)		PPG (ch.33) timer register (PTMR33)	
0x1C88	PPG (ch.33) control status register2 (PCN233)		PPG (ch.33) Start Delay value setting register (PSDR33)	
0x1C8C	PPG (ch.33) Timing Point Capture value setting register (PTPC33)		Reserved	
0x1C90	PPG (ch.34) control status register (PCN34)		PPG (ch.34) cycle setting register (PCSR34)	
0x1C94	PPG (ch.34) duty setting register (PDUT34)		PPG (ch.34) timer register (PTMR34)	
0x1C98	PPG (ch.34) control status register2 (PCN234)		PPG (ch.34) Start Delay value setting register (PSDR34)	
0x1C9C	PPG (ch.34) Timing Point Capture value setting register (PTPC34)		Reserved	
0x1CA0	PPG (ch.35) control status register (PCN35)		PPG (ch.35) cycle setting register (PCSR35)	
0x1CA4	PPG (ch.35) duty setting register (PDUT35)		PPG (ch.35) timer register (PTMR35)	
0x1CA8	PPG (ch.35) control status register2 (PCN235)		PPG (ch.35) Start Delay value setting register (PSDR35)	

Address	+0	+1	+2	+3
0x1CAC	PPG (ch.35) Timing Point Capture value setting register (PTPC35)		Reserved	
0x1CB0	PPG (ch.36) control status register (PCN36)		PPG (ch.36) cycle setting register (PCSR36)	
0x1CB4	PPG (ch.36) duty setting register (PDUT36)		PPG (ch.36) timer register (PTMR36)	
0x1CB8	PPG (ch.36) control status register2 (PCN236)		PPG (ch.36) Start Delay value setting register (PSDR36)	
0x1CBC	PPG (ch.36) Timing Point Capture value setting register (PTPC36)		Reserved	
0x1CC0	PPG (ch.37) control status register (PCN37)		PPG (ch.37) cycle setting register (PCSR37)	
0x1CC4	PPG (ch.37) duty setting register (PDUT37)		PPG (ch.37) timer register (PTMR37)	
0x1CC8	PPG (ch.37) control status register2 (PCN237)		PPG (ch.37) Start Delay value setting register (PSDR37)	
0x1CCC	PPG (ch.37) Timing Point Capture value setting register (PTPC37)		Reserved	
0x1CD0	PPG (ch.38) control status register (PCN38)		PPG (ch.38) cycle setting register (PCSR38)	
0x1CD4	PPG (ch.38) duty setting register (PDUT38)		PPG (ch.38) timer register (PTMR38)	
0x1CD8	PPG (ch.38) control status register2 (PCN238)		PPG (ch.38) Start Delay value setting register (PSDR38)	
0x1CDC	PPG (ch.38) Timing Point Capture value setting register (PTPC38)		Reserved	
0x1CE0	PPG (ch.39) control status register (PCN39)		PPG (ch.39) cycle setting register (PCSR39)	
0x1CE4	PPG (ch.39) duty setting register (PDUT39)		PPG (ch.39) timer register (PTMR39)	
0x1CE8	PPG (ch.39) control status register2 (PCN239)		PPG (ch.39) Start Delay value setting register (PSDR39)	
0x1CEC	PPG (ch.39) Timing Point Capture value setting register (PTPC39)		Reserved	
0x1CF0	PPG (ch.40) control status register (PCN40)		PPG (ch.40) cycle setting register (PCSR40)	
0x1CF4	PPG (ch.40) duty setting register (PDUT40)		PPG (ch.40) timer register (PTMR40)	
0x1CF8	PPG (ch.40) control status register2 (PCN240)		PPG (ch.40) Start Delay value setting register (PSDR40)	
0x1CFC	PPG (ch.40) Timing Point Capture value setting register (PTPC40)		Reserved	
0x1D00	PPG (ch.41) control status register (PCN41)		PPG (ch.41) cycle setting register (PCSR41)	
0x1D04	PPG (ch.41) duty setting register (PDUT41)		PPG (ch.41) timer register (PTMR41)	
0x1D08	PPG (ch.41) control status register2 (PCN241)		PPG (ch.41) Start Delay value setting register (PSDR41)	
0x1D0C	PPG (ch.41) Timing Point Capture value setting register (PTPC41)		Reserved	
0x1D10	PPG (ch.42) control status register (PCN42)		PPG (ch.42) cycle setting register (PCSR42)	
0x1D14	PPG (ch.42) duty setting register (PDUT42)		PPG (ch.42) timer register (PTMR42)	
0x1D18	PPG (ch.42) control status register2 (PCN242)		PPG (ch.42) Start Delay value setting register (PSDR42)	
0x1D1C	PPG (ch.42) Timing Point Capture value setting register (PTPC42)		Reserved	
0x1D20	PPG (ch.43) control status register (PCN43)		PPG (ch.43) cycle setting register (PCSR43)	
0x1D24	PPG (ch.43) duty setting register (PDUT43)		PPG (ch.43) timer register (PTMR43)	
0x1D28	PPG (ch.43) control status register2 (PCN243)		PPG (ch.43) Start Delay value setting register (PSDR43)	

Address	+0	+1	+2	+3
0x1D2C	PPG (ch.43) Timing Point Capture value setting register (PTPC43)		Reserved	
0x1D30	PPG (ch.44) control status register (PCN44)		PPG (ch.44) cycle setting register (PCSR44)	
0x1D34	PPG (ch.44) duty setting register (PDUT44)		PPG (ch.44) timer register (PTMR44)	
0x1D38	PPG (ch.44) control status register2 (PCN244)		PPG (ch.44) Start Delay value setting register (PSDR44)	
0x1D3C	PPG (ch.44) Timing Point Capture value setting register (PTPC44)		Reserved	
0x1D40	PPG (ch.45) control status register (PCN45)		PPG (ch.45) cycle setting register (PCSR45)	
0x1D44	PPG (ch.45) duty setting register (PDUT45)		PPG (ch.45) timer register (PTMR45)	
0x1D48	PPG (ch.45) control status register2 (PCN245)		PPG (ch.45) Start Delay value setting register (PSDR45)	
0x1D4C	PPG (ch.45) Timing Point Capture value setting register (PTPC45)		Reserved	
0x1D50	PPG (ch.46) control status register (PCN46)		PPG (ch.46) cycle setting register (PCSR46)	
0x1D54	PPG (ch.46) duty setting register (PDUT46)		PPG (ch.46) timer register (PTMR46)	
0x1D58	PPG (ch.46) control status register2 (PCN246)		PPG (ch.46) Start Delay value setting register (PSDR46)	
0x1D5C	PPG (ch.46) Timing Point Capture value setting register (PTPC46)		Reserved	
0x1D60	PPG (ch.47) control status register (PCN47)		PPG (ch.47) cycle setting register (PCSR47)	
0x1D64	PPG (ch.47) duty setting register (PDUT47)		PPG (ch.47) timer register (PTMR47)	
0x1D68	PPG (ch.47) control status register2 (PCN247)		PPG (ch.47) Start Delay value setting register (PSDR47)	
0x1D6C	PPG (ch.47) Timing Point Capture value setting register (PTPC47)		Reserved	

■ List of GATE Function Control Registers Map

Address	+0	+1	+2	+3
0x19DC	Reserved	GATE function control register 0 (GATEC0)	Reserved	GATE function control register 2 (GATEG2)
0x19E0	Reserved	GATE function control register 4 (GATEC4)	Reserved	Reserved

■ List of PPG Control Registers Map

Address	+0	+1	+2	+3
0x19E8	General-purpose trigger selection register 0 (GTRS0)		General-purpose trigger selection register 1 (GTRS1)	
0x19EC	General-purpose trigger selection register 2 (GTRS2)		General-purpose trigger selection register 3 (GTRS3)	

Address	+0	+1	+2	+3
0x19F0	General-purpose trigger selection register 4 (GTRS4)		General-purpose trigger selection register 5 (GTRS5)	
0x19F4	General-purpose trigger selection register 6 (GTRS6)		General-purpose trigger selection register 7 (GTRS7)	
0x19F8	General-purpose trigger selection register 8 (GTRS8)		General-purpose trigger selection register 9 (GTRS9)	
0x19FC	General-purpose trigger selection register 10 (GTRS10)		General-purpose trigger selection register 11 (GTRS11)	
0x1A00	General-purpose trigger selection register 12 (GTRS12)		General-purpose trigger selection register 13 (GTRS13)	
0x1A04	General-purpose trigger selection register 14 (GTRS14)		General-purpose trigger selection register 15 (GTRS15)	
0x1A08	General-purpose trigger selection register 16 (GTRS16)		General-purpose trigger selection register 17 (GTRS17)	
0x1A0C	General-purpose trigger selection register 18 (GTRS18)		General-purpose trigger selection register 19 (GTRS19)	
0x1A10	General-purpose trigger selection register 20 (GTRS20)		General-purpose trigger selection register 21 (GTRS21)	
0x1A14	General-purpose trigger selection register 22 (GTRS22)		General-purpose trigger selection register 23 (GTRS23)	
0x1A38	General-purpose trigger setting register 0 (GTREN0)		General-purpose trigger setting register 1 (GTREN1)	
0x1A3C	General-purpose trigger setting register 2 (GTREN2)		Reserved	Reserved

4.1. PPG Control Status Register : PCN0 to PCN 47

The bit configuration of PPG control status register is shown.

The PPG control status register (PCN) controls the operation and status of the PPG.

■ PPG CoNtrol status register (PCN): Address Base_addr + 00_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CNTE	STRG	MDSE	RTRG	CKS1	CKS0	PGMS	OWFS
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0/W	R/W	R/W	R/W	R/W	R/W	R/W

	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

Note: The each bits of the PPG control status register (PCN), except for Bit13 MDSE: mode selection bit and Bit8 OWFS: PPG output waveform selection bit, will become effective immediately by writing in the register.

[bit15] CNTE : Timer operation enable bit

CNTE	Explanation
0	The timer operation is stopped.
1	The timer operation is enabled.

This bit is the timer operation enable bit.
If this bit is set to "0", the PPG operation is stopped.
If this bit is set to "1", the PPG operation is enabled.

[bit14] STRG : Software trigger bit

STRG	Explanation
0	The operation is not influenced by the value written to this bit (The read value is always "0").
1	The PPG is activated by a software trigger that is generated independent of the external trigger (at the TRG pin). This trigger is not influenced by the trigger input edge selection bits (EGS1, EGS0).

This bit is the software trigger bit.
If this bit is set to "0", the operation is not influenced by the value written to this bit.
If this bit is set to "1", The PPG is activated by a software trigger that is generated independent of the external trigger (at the TRG pin). This trigger is not influenced by the trigger input edge selection bits (EGS1, EGS0).

[bit13] MDSE : Mode selection bit

MDSE	Explanation
0	PWM operation
1	One-shot operation

This bit selects type of output waveform.
If this bit is set to "0", the PWM operation is enabled and the consecutive pulse is generated.
If this bit is set to "1", the pulse is output only once.

Note:
This bit is effective for each cycle (trigger generation or counter borrow generation).

[bit12] RTRG : Restart enable bit

RTRG	Explanation
0	Restart disabled
1	Restart enabled

This bit enables/disables to restart PPG operation.
If this bit is set to "0", the PPG is disabled from restarting.
If this bit is set to "1", the PPG is enabled to restart. When the restart enable bit is set to "1", the PPG is enabled to restart as triggered (by software/internal/external).

[bit11, bit10] CKS1, CKS0 : Count clock selection bits

CKS1	CKS0	Explanation
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

This bit is the count clock selection bits of down counter.

[bit9] PGMS : PPG output mask selection bit

PGMS	Explanation
0	No output mask
1	Output mask

This bit specifies whether to mask the PPG output.
If this bit is set to "0", the PPG output is not masked.
If this bit is set to "1", the PPG output is masked.

Note:
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 PPG output polarity selection bit (PCN:OSEL).
(If OSEL = 0, the output is maintained at the "L" level.)
When this bit is set from "1" to "0" to cancel the PPG output mask, perform the setting within the period between the beginning of cycle and the duty match.

[bit8] OWFS : PPG output waveform selection bit

OWFS	Explanation
0	Normal Wave Form is output.
1	Center Aligned Wave Form is output.

This bit selects the PPG output waveform.
If this bit is set to "0", the PPG outputs the Normal Wave Form.
If this bit is set to "1", the PPG outputs the Center Aligned Wave Form.

Note:

This bit is effective for each cycle (trigger generation or counter borrow generation).

[bit7, bit6] EGS1, EGS0 : Trigger input edge selection bits

EGS1, EGS0		Explanation
0	0	No edge selection (only software triggers are possible.)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising or falling)

These bits select the trigger input edge.

Note:

If EGS1=0 and EGS0=0, only the trigger with software trigger (PCN:STRG) is possible. The input from an external trigger (TRG pin) to PPG is disabled.
Other settings of EGS1 and EGS0 influence only the input of an external trigger (TRG pin).
The trigger of PPG to start by writing "1" to the software trigger (PCN:STRG) is not influenced by the setting of PCN:EGS1 and EGS0.

[bit5] IREN : Interrupt request enable bit

IREN	Explanation
0	Interrupt request disabled
1	Interrupt request enabled

This bit enables/disables interrupt requests.
If this bit is set to "0", interrupt requests are disabled.
If this bit is set to "1", interrupt requests are enabled.

[bit4] IRQF : Interrupt request flag bit

IRQF		Explanation
0	Read	No interrupt request
	Write	Clears the interrupt request flag.
1	Read	No interrupt request
	Write	Writing of "1" does not influence operation.

Note:

If this bit is set to "0" when the interrupt request flag (IRQF) = "1", the interrupt request flag (IRQF = 1) that is set by hardware takes precedence.

[bit3, bit2] IRS1, IRS0 : Interrupt factor selection bits

IRS1, IRS0		Explanation
0	0	STGR=0: Software trigger or external trigger (TRG pin) input STGR=1: GATE signal trigger input
0	1	Counter borrow occurrence
1	0	Counter and duty value match
1	1	Counter borrow occurrence or counter and duty value match

Note:

See the following figures for the relationship between output waveforms and interrupt generation locations:

- In the case of the PPG output waveform selection bit (OWFS="0"):
  - Figure 5-1 Example of PWM Operation (Normal Wave Form Selected)
  - Figure 5-3 Example of One-shot Operation (Normal Wave Form Selected)
- In the case of the PPG output waveform selection bit (OWFS="1"):
  - Figure 5-2 Example of PWM Operation (Center Aligned Wave Form Selected)
  - Figure 5-4 Example of One-shot Operation (Center Aligned Wave Form Selected)

[bit1] Reserved

This bit must be set to "0".

[bit0] OSEL : PPG output polarity selection bit

OSEL	Explanation
0	Normal polarity
1	Inverted polarity

This bit selects the PPG output polarity.
If this bit is set to "0", the normal polarity is selected.
If this bit is set to "1", the inverted polarity is selected.

Note:

If the PPG output mask selection bit (PCN:PGMS) is set to "1", setting the PPG output polarity selection bit (OSEL) to "0" or "1" causes the output to be clamped to "L" or "H", respectively.

4.2. PPG Cycle Setting Register : PCSR0 to PCSR47

The bit configuration of the PPG cycle setting register is shown.

The PPG cycle setting register (PCSR) specifies the cycle of the PPG output waveform.

■ PPG cycle setting register (PCSR): Address Base_addr + 02_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

[bit15 to bit0] D15 to D0 : PPG cycle setting bits

D15 to D0	Function
Cycle of the PPG output waveform	

Note: These bits are write-only.

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.

Notes:

- If the PPG output waveform selection bit (PCN.OWFS)="0" (Normal Wave Form) is selected, the waveform is output at the cycle of a set value of PPG cycle setting register.
- If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, the waveform is output at twice the cycle of a set value of PPG cycle setting register.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.3. PPG Duty Setting Register : PDUT0 to PDUT47

The bit configuration of the PPG duty setting register is shown.

The PPG duty setting register (PDUT) specifies the duty of the PPG output waveform.

■ PPG duty setting register (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

[bit15 to bit0] D15 to D0 : PPG duty setting bits

D15 to D0	Function
Duty of the PPG output waveform	

Note: These bits are write-only.

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.
Be sure to set a value that is smaller than the value set to the PPG cycle setting register (PCSR) to the PPG duty setting register.

Notes:

- If the PPG output waveform selection bit (PCN.OWFS)="0" (Normal Wave Form) is selected, the waveform is output at the duty of a set value of PPG duty setting register.
- If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, the waveform is output at twice the duty of a set value of PPG duty setting register.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed the value is not written at an upper and lower bit position.

4.4. PPG Timer Register : PTMR0 to PTMR47

The bit configuration of the PPG timer register is shown.

The PPG timer register (PTMR) allows the PPG timer countdown value to be read.

■ PPG timer register (PTMR): Address Base_addr + 06_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D9	D8
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
	D7	D6	D5	D4	D3	D2	D1	D0
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

[bit15 to bit0] D15 to D0 : PPG timer value bits

D15 to D0	Function
Timer down count value	

Note: These bits are read-only.

The count value of the 17-bit down counter can be read from these bits.

- If the Normal Wave Form (OWFS="0") is selected, the lower 16 bits are read.
- If the Center Aligned Wave Form (OWFS="1") is selected, the upper 16 bits are read.

4.5. PPG Control Status Register2 : PCN200 to PCN247

The bit configuration of the PPG control status register2 is shown.

The PPG control status register2 (PCN2) controls the operation and status of the PPG.

■ PPG control status register2 (PCN2): Address Base_addr + 08_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	LFPR	HFPR	CMDSEL	CMD	TPC	STRD
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
	Reserved	Reserved	Reserved	Reserved	Reserved	REMP	SREMP	IRS2
Initial value	0	0	0	0	0	1	1	0
Attribute	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0	R,W0	R,W0	R/W

[bit15, bit14] Reserved bits

- The read value of these bits is always "0".
- These bits must always be written to "0".

[bit13] LFPR : Low format pulse polarity selection bit

LFPR	Explanation
0	Output from Low pulse (When PCN.OSEL=1: the High pulse output)
1	Output from High pulse (When PCN.OSEL=1: the Low pulse output)

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "0". This bit must always be written to "0".

[bit12] HFPR : High format pulse polarity selection bit

HFPR	Explanation
0	Output from Low pulse (When PCN.OSEL=1: the High pulse output)
1	Output from High pulse (When PCN.OSEL=1: the Low pulse output)

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "0". This bit must always be written to "0".

[bit11] CMDSEL : PPG communication mode data read selection bit

CMDSEL	Explanation
0	Output from LSB bit position of PCMDDT set in PCMDWD
1	Output from MSB bit position of PCMDDT set in PCMDWD

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "0". This bit must always be written to "0".

[bit10] CMD : PPG communication mode enable bit

CMD	Explanation
0	PPG communication mode disable
1	PPG communication mode enable

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "0". This bit must always be written to "0".

[bit9] TPC : Timing Point Capture enable bit

TPC	Explanation
0	Timing Point Capture mode disable
1	Timing Point Capture mode enable

[bit8] STRD : Start Delay mode enable bit

STRD	Explanation
0	Start Delay mode disable
1	Start Delay mode enable

[bit7 to bit3] Reserved bits

- The read value of these bits is always "0".
- These bits must always be written to "0".

[bit2] REMP : PPG communication data register Empty flag bit

REMP	Explanation
0	No interrupt (state of Not Empty)
1	Interrupt (state of Empty)

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "1".

[bit1] SREMP : PPG communication data shift register Empty flag bit

SREMP	Explanation
0	No interrupt (state of Not Empty)
1	Interrupt (state of Empty)

Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "1".

[bit0] IRS2 : Interrupt factor selection2 bit

IRS2	IRS1	IRS0	Explanation
0	0	0	STGR=0 : Software trigger or external trigger (TRG pin) input STGR=1 : GATE signal trigger input
0	0	1	Borrow occurrence on the counter
0	1	0	Counter matched with the specified duty value
0	1	1	Borrow occurrence on the counter or counter matched with the specified duty value
1	0	0	Timing Point Capture value match
Other value			PPG communication data register Empty factor

4.6. Start Delay Value Setting Register : PSDR0 to PSDR47

The bit configuration of the Start Delay value setting register is shown.

The Start Delay value setting register (PSDR) sets the delay value to shift the phase of PPG output waveform.

■ Start Delay value setting register (PSDR): Address Base_addr + $0A_H$ (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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D0 : Start Delay value setting bits

The phase from the activation trigger generation to PPG waveform output is adjusted according to the following calculations.

(Start Delay value setting register + 1) × Count clock

Notes:

- If the PPG output waveform selection bit (PCN.OWFS)="0" (Normal Wave Form) is selected, the delay value is the set value of the Start Delay value setting register.
- If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, the delay value is doubling the set value of the Start Delay value setting register.

- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.7. Timing Point Capture Value Setting Register : PTPC0 to PTPC47

The bit configuration of the Timing Point Capture value setting register is shown.

Timing Point Capture sets the timing that generates an interrupt and the A/D activation trigger.

■ Timing Point Capture value setting register (PTPC): Address Base_addr + $0C_H$ (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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D0 : Timing Point Capture value setting bits

These bits set the timing that generates an interrupt and the A/D activation trigger.

Interrupt and A/D activation trigger is generated according to the timing after (Timing Point Capture setting value + 1 (*)) from the activation trigger. (*: When OWFS=0 is set)

Notes:

- Be sure to set the register to become "Timing Point Capture setting value < PPG cycle setting value".
- The value when the PPG output waveform selection bit (PCN.OWFS)="0" (Normal Wave Form) is selected is set to the Timing Point Capture value.
- If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, a set value of the Timing Point Capture value setting register is doubling (PCN.OWFS)="0".
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.8. PPG Communication Mode High Format Cycle Setting Register : PHCSR0 to PHCSR3

The bit configuration of the PPG communication mode High format cycle setting register is shown.

The PPG communication mode High format cycle setting register (PHCSR) sets the cycle for the High format.

■ PPG communication mode High format cycle setting register (PHCSR): Address Base_addr + 10_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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W

[bit15 to bit0] D15 to D0 : PPG communication mode High format cycle setting bits

When borrow of the counter is generated, the value is automatically transferred from the PPG communication mode High format cycle setting register to the counter.

Notes:

- In the PPG communication mode, the setting of PPG output waveform selection bit (PCN.OWFS) and mode selection bit (PCN.MDSE) does not influence operation.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.9. PPG Communication Mode Low Format Cycle Setting Register : PLCSR0 to PLCSR3

The bit configuration of the PPG communication mode Low format cycle setting register is shown.

The PPG communication mode Low format cycle setting register (PLCSR) sets the cycle for the Low format.

■ PPG communication mode Low format cycle setting register (PLCSR): Address Base_addr + 12_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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W

[bit15 to bit0] D15 to D0 : PPG communication mode Low format cycle setting bits

When borrow of the counter is generated, the value is automatically transferred from the PPG communication mode Low format cycle setting register to the counter.

Notes:

- In the PPG communication mode, the setting of PPG output waveform selection bit (PCN.OWFS) and mode selection bit (PCN.MDSE) does not influence operation.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.10. PPG Communication Mode High Format Duty Setting Register : PHDUT0 to PHDUT3

The bit configuration of the PPG communication mode High format duty setting register is shown.

The PPG communication mode High format duty setting register (PHDUT) sets the duty for the High format.

■ PPG communication mode High format duty setting register (PHDUT): Address Base_addr + 14_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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W

[bit15 to bit0] D15 to D0 : PPG communication mode High format duty setting bits

When borrow of the counter is generated, the value is automatically transferred from the PPG communication mode High format duty setting register to the counter.

Notes:

- Be sure to set a value that is smaller than the value set to the PPG communication mode High format cycle setting register (PHCSR) to the PPG communication mode High format duty setting register.
- In the PPG communication mode, the setting of PPG output waveform selection bit (PCN.OWFS) and mode selection bit (PCN.MDSE) does not influence operation.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.11. PPG Communication Mode Low Format Duty Setting Register : PLDUT0 to PLDUT3

The bit configuration of the PPG communication mode Low format duty setting register is shown.

The PPG communication mode Low format duty setting register (PLDUT) sets the duty for the Low format.

■ PPG communication mode Low format duty setting register (PLDUT): Address Base_addr + 16_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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,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	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W	R1,W

[bit15 to bit0] D15 to D0 : PPG communication mode Low format duty setting bits

When borrow of the counter is generated, the value is automatically transferred from the PPG communication mode Low format duty setting register to the counter.

Notes:

- Be sure to set a value that is smaller than the value set to the PPG communication mode Low format cycle setting register (PLCSR) to the PPG communication mode Low format duty setting register.
- In the PPG communication mode, the setting of PPG output waveform selection bit (PCN.OWFS) and mode selection bit (PCN.MDSE) does not influence operation.
- Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.12. PPG Communication Mode Data Setting Register : PCMDDT0 to PCMDDT3

The bit configuration of the PPG communication mode data setting register is shown.

The PPG communication mode data setting register (PCMDDT) sets the control of the High/Low format waveform output.

■ PPG communication mode data setting register (PCMDDT): Address Base_addr + 18_H (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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D0 : PPG communication mode data setting bits

These bits control the PPG High/Low format waveform output.

When the register setting value is "1", the High format waveform is output. When the register setting value is "0", the Low format waveform is output.

Note:

Be sure to access this register by the word (16-bit) format. If this register is byte accessed, the value is not written at an upper and lower bit position.

4.13. PPG Communication Mode Data Bit Length Setting Register : PCMDWD0 to PCMDWD3

The bit configuration of the PPG communication mode data bit length setting register is shown.

The PPG communication mode data bit length setting register (PCMDWD) sets the bit length of the High/Low format waveform output.

■ PPG communication mode data bit length setting register (PCMDWD): Address Base_addr + $0E_H$ (Byte, Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial value	1	1	1	1	1	1	1	1
Attribute	R1/W1	R1/W1	R1/W1	R1/W1	R1/W1	R1/W1	R1/W1	R1/W1

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	D3	D2	D1	D0
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 bit8] Reserved bits

- The read value of these bits is always "1".
- These bits must always be written to "1".

[bit7 to bit4] Reserved bits

- The read value of these bits is always "0".
- These bits must always be written to "0".

[bit3 to bit0] D3 to D0 : PPG communication mode data bit length setting bits

These bits control the bit length of the PPG High/Low format waveform output.

"0000b" ; PPG communication 1 bit (PCMDDT bit0)
"0001b" : PPG communication 2 bits (PCMDDT bit1 to bit0)
"0010b" : PPG communication 3 bits (PCMDDT bit2 to bit0)
"0011b" : PPG communication 4 bits (PCMDDT bit3 to bit0)
"0100b" : PPG communication 5 bits (PCMDDT bit4 to bit0)
"0101b" : PPG communication 6 bits (PCMDDT bit5 to bit0)
"0110b" : PPG communication 7 bits (PCMDDT bit6 to bit0)
"0111b" : PPG communication 8 bits (PCMDDT bit7 to bit0)
"1000b" : PPG communication 9 bits (PCMDDT bit8 to bit0)
"1001b" : PPG communication 10 bits (PCMDDT bit9 to bit0)
"1010b" : PPG communication 11 bits (PCMDDT bit10 to bit0)
"1011b" : PPG communication 12 bits (PCMDDT bit11 to bit0)
"1100b" : PPG communication 13 bits (PCMDDT bit12 to bit0)
"1101b" : PPG communication 14 bits (PCMDDT bit13 to bit0)
"1110b" : PPG communication 15 bits (PCMDDT bit14 to bit0)
"1111b" : PPG communication 16 bits (PCMDDT bit15 to bit0)

4.14. GATE Function Control Register : GATEC0, GATEC2, GATEC4

The bit configuration of the GATE function control register is shown.

The GATE function control register (GATEC) controls the operation of the GATE function.

■ GATE function control register (GATEC): Address $19DD_H$, $19DF_H$, $19E1_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	STGR	EDGE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0	R/W	R/W

[bit7 to bit2] Reserved bits

- The read value of these bits is always "0".
- These bits must always be written to "0".

[bit1] STGR : GATE function selection bit

STGR	Explanation
0	PPG is activated by the activation trigger.
1	PPG is activated and stopped according to the GATE signal from a waveform generator.

[bit0] EDGE : GATE function activation effective edge selection bit

EDGE	Explanation
0	PPG is activated by the rising of the GATE signals, and stopped by the falling. PPG activates during "H".
1	PPG is activated by the falling of the GATE signals, and stopped by the rising. PPG activates during "L".

Note:

Be sure to set GATE function control register (GATEC) before activating PPG. Please change neither the GATE selection bit (STGR) nor polarity selection bit (EDGE) of the GATE function control register (GATEC) during the PPG operation.

4.15. General-purpose Trigger Selection Register : GTRS0 to GTRS23

The bit configuration of the general-purpose trigger selection register is shown.

The General-purpose trigger selection register (GTRS) is used to select the trigger input to PPG.

■ General-purpose trigger selection register (GTRS): Address $19E8_H$ to $1A16_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	TSELii_6	TSELii_5	TSELii_4	TSELii_3	TSELii_2	TSELii_1	TSELii_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	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TSELii_6	TSELii_5	TSELii_4	TSELii_3	TSELii_2	TSELii_1	TSELii_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

Note: "ii" of "TSELii_" is an index that shows the number of PPG0 to PPG47.

[bit15] / [bit7] Reserved bits

- The read value of these bits is always "0".
- Writing has no effect on operation.

[bit14 to bit8] / [bit6 to bit0] TSELii [6:0]: Activation trigger selection bits

- bit14 to bit8 : selection bit of the activation trigger of PPG2n+1(n=0 to 23).
- bit6 to bit0 : selection bit of the activation trigger of PPG2n(n=0 to 23).

TSELii [6:0]							Activation trigger selection
0	0	0	0	0	0	0	Internal trigger (EN0) selected
0	0	0	0	0	0	1	Internal trigger (EN1) selected
0	0	0	0	0	1	0	Internal trigger (EN2) selected
0	0	0	0	0	1	1	Internal trigger (EN3) selected
0	0	0	0	1	0	0	Internal trigger (EN4) selected
0	0	0	0	1	0	1	Internal trigger (EN5) selected
0	0	0	0	1	1	0	Internal trigger (EN6) selected
0	0	0	0	1	1	1	Internal trigger (EN7) selected
0	0	0	1	0	0	0	Internal trigger (EN8) selected
0	0	0	1	0	0	1	Internal trigger (EN9) selected
0	0	0	1	0	1	0	Internal trigger (EN10) selected

TSELii [6:0]							Activation trigger selection
0	0	0	1	0	1	1	Internal trigger (EN11) selected
0	0	0	1	1	0	0	Internal trigger (EN12) selected
0	0	0	1	1	0	1	Internal trigger (EN13) selected
0	0	0	1	1	1	0	Internal trigger (EN14) selected
0	0	0	1	1	1	1	Internal trigger (EN15) selected
0	0	1	0	0	0	0	Internal trigger (EN16) selected
0	0	1	0	0	0	1	Internal trigger (EN17) selected
0	0	1	0	0	1	0	Internal trigger (EN18) selected
0	0	1	0	0	1	1	Internal trigger (EN19) selected
0	0	1	0	1	0	0	Internal trigger (EN20) selected
0	0	1	0	1	0	1	Internal trigger (EN21) selected
0	0	1	0	1	1	0	Internal trigger (EN22) selected
0	0	1	0	1	1	1	Internal trigger (EN23) selected
0	0	1	1	0	0	0	Internal trigger (EN24) selected
0	0	1	1	0	0	1	Internal trigger (EN25) selected
0	0	1	1	0	1	0	Internal trigger (EN26) selected
0	0	1	1	0	1	1	Internal trigger (EN27) selected
0	0	1	1	1	0	0	Internal trigger (EN28) selected
0	0	1	1	1	0	1	Internal trigger (EN29) selected
0	0	1	1	1	1	0	Internal trigger (EN30) selected
0	0	1	1	1	1	1	Internal trigger (EN31) selected
0	1	0	0	0	0	0	Internal trigger (EN32) selected
0	1	0	0	0	0	1	Internal trigger (EN33) selected
0	1	0	0	0	1	0	Internal trigger (EN34) selected
0	1	0	0	0	1	1	Internal trigger (EN35) selected
0	1	0	0	1	0	0	Internal trigger (EN36) selected
0	1	0	0	1	0	1	Internal trigger (EN37) selected
0	1	0	0	1	1	0	Internal trigger (EN38) selected
0	1	0	0	1	1	1	Internal trigger (EN39) selected
0	1	0	1	0	0	0	Internal trigger (EN40) selected
0	1	0	1	0	0	1	Internal trigger (EN41) selected
0	1	0	1	0	1	0	Internal trigger (EN42) selected
0	1	0	1	0	1	1	Internal trigger (EN43) selected
0	1	0	1	1	0	0	Internal trigger (EN44) selected
0	1	0	1	1	0	1	Internal trigger (EN45) selected
0	1	0	1	1	1	0	Internal trigger (EN46) selected

TSELii [6:0]							Activation trigger selection
0	1	0	1	1	1	1	Internal trigger (EN47) selected
0	1	1	1	1	1	0	16-bit reload timer 0 selected
0	1	1	1	1	1	1	16-bit reload timer 1 selected
1	0	0	0	0	0	0	External trigger 0 selected
1	0	0	0	0	0	1	External trigger 1 selected
1	0	0	0	0	1	0	External trigger 2 selected
1	0	0	0	0	1	1	External trigger 3 selected
1	0	0	0	1	0	0	External trigger 4 selected
1	0	0	0	1	0	1	External trigger 5 selected
1	0	0	0	1	1	0	External trigger 6 selected
1	0	0	0	1	1	1	External trigger 7 selected
1	0	0	1	0	0	0	External trigger 8 selected
1	0	0	1	0	0	1	External trigger 9 selected
1	0	0	1	0	1	0	External trigger 10 selected
1	0	0	1	0	1	1	External trigger 11 selected

Note:

A trigger input to PPG is selected.
On selected PPGn, the PPG will be activated when the edge selected by the trigger input edge selection bits (PCN:EGS1, RGS0) is detected upon the selected activation trigger.

4.16. General-purpose Trigger Setting Register : GTREN0 to GTREN2

The bit configuration of the general-purpose trigger setting register is shown.

The general-purpose trigger setting register (GTREN) controls the generation of internal trigger to the PPG.

■ General-purpose trigger setting register 0 (GTREN0): Address $1A38_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
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
	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

■ General-purpose trigger setting register 1 (GTREN1): Address $1A3A_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24
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
	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
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

■ General-purpose trigger setting register 2 (GTREN2): Address $1A3C_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EN47	EN46	EN45	EN44	EN43	EN42	EN41	EN40
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
	EN39	EN38	EN37	EN36	EN35	EN34	EN33	EN32
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

GTREN0, GTREN1, GTREN2 EN47 to EN0 Internal trigger input bits

EN47 to EN0	Explanation
0	Sets the level to "L".
1	Sets the level to "H".

These bits are used to generate a trigger at a specified internal trigger level.

If these bits are set to "0", a level "L" trigger is generated.
If these bits are set to "1", a level "H" trigger is generated.

Notes:

- If an internal trigger (one of EN0 to EN47) is selected by the PPG activation trigger selection bits (TSELii_[6:0]), the selected EN serves as the PPG trigger input bit.
- When the state selected by the trigger input edge selection bits (PCN:EGS1, EGS0) is generated by software with the use of the trigger input bit (selected one of EN0 to EN47), the trigger input bit serves as the PPG activation trigger.

5. Operation

This section explains the operation of the PPG.

5.1. PWM Operation (Normal Wave Form)

5.2. PWM Operation (Center Aligned Wave Form Selected)

5.3. One-shot Operation (Normal Wave Form Selected)

5.4. One-shot Operation (Center Aligned Wave Form Selected)

5.5. Restart Operation

5.6. GATE Operation

5.7. Start Delay Mode Operation (PWM Normal Wave Form Selected)

5.8. Timing Point Capture Mode Operation (PWM Normal Wave Form Selected)

5.9. PPG Communication Mode Operation

5.10. PPG Communication Activation

5.11. PPG Communication Operation

5.12. PPG Communication Forced Stop and Restart operation

5.13. PPG Output Pulse Polarity Selection

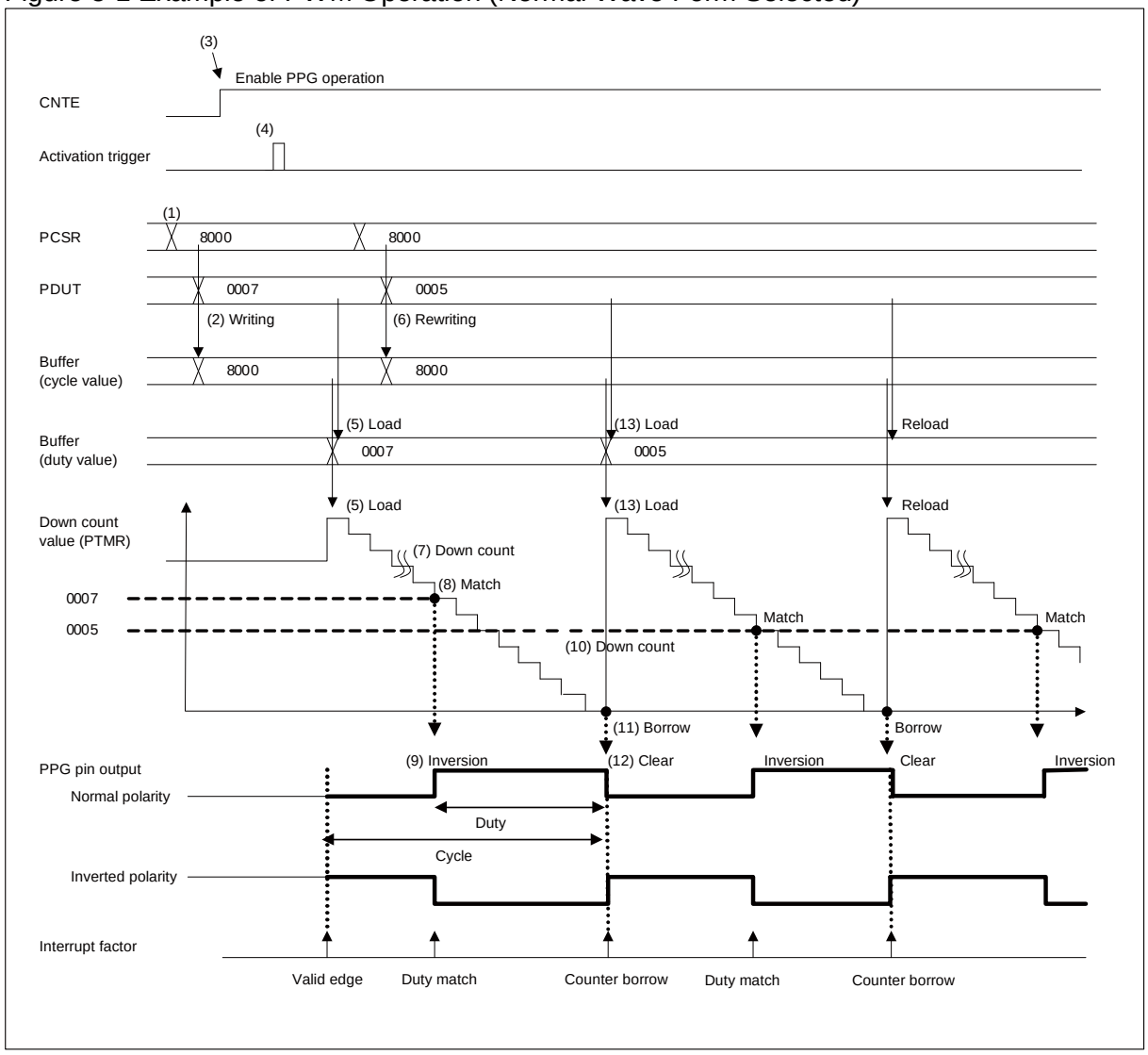
5.14. Interrupt

5.1. PWM Operation (Normal Wave Form)

The PWM operation (Normal Wave Form) is explained.

During the PWM operation, variable-duty pulses are output at the PPG pin.

Figure 5-1 Example of PWM Operation (Normal Wave Form Selected)



Setting and operation procedure:

(1) Writing of PCSR (cycle value)
(2) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(6) Rewriting of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(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 of PPG pin output level (restoration to normal state)
(13) Reloading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(14) Repetition of steps (7) to (13)

Calculation formulas:

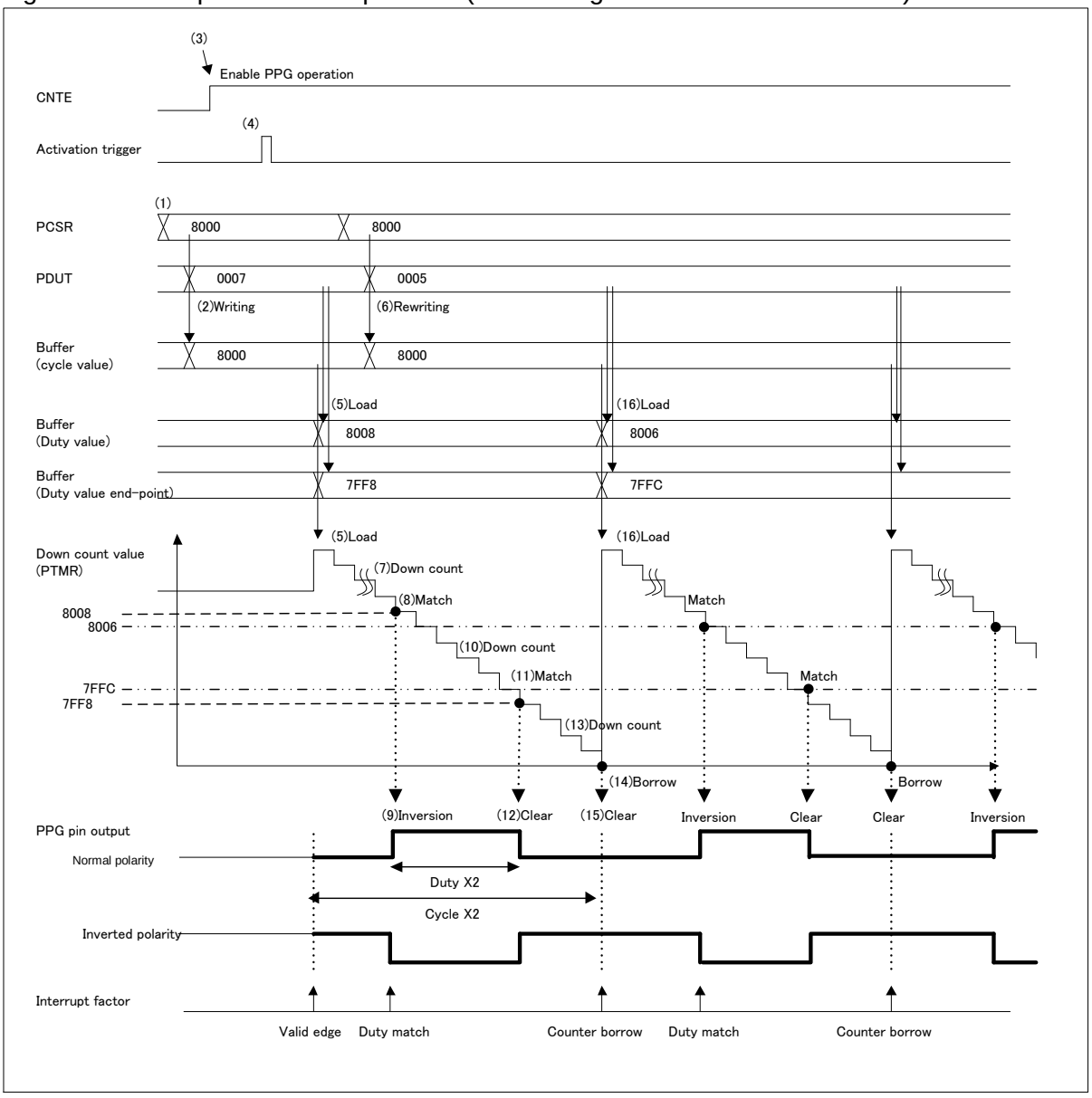
- Cycle = {Cycle value (PCSR) + 1} × Count clock
- Duty = {Duty value (PDUT) + 1} × Count clock
- Time to pulse output = {Cycle value (PCSR) - Duty value (PDUT)} × Count clock

5.2. PWM Operation (Center Aligned Wave Form Selected)

The PWM operation (Center Aligned Wave Form selected) is explained.

During the PWM operation, variable-duty pulses are output at the PPG pin. If the PPG output waveform selection bit (PCN:OWFS)="1" is selected (Center Aligned Wave Form), the output waveform is generated by doubling the value of the PPG cycle setting register (PCSR) and the value of the PPG duty setting register (PDUT)

Figure 5-2 Example of PWM Operation (Center Aligned Wave Form Selected)



Setting and operation procedure:

(1) Writing of PCSR (cycle value)
(2) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading of the [(Cycle value) × 2 + 1] to the down count value (PTMR) and the duty value to the buffer (duty value) and the buffer (duty value end point)
Duty value (Output level inversion timing) = (Duty value + Cycle value + 1)
Duty value end point (Output level clear timing) = (Cycle value - Duty value - 1)
(6) Rewriting of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(7) Counter decrement
(8) The down counter matches the duty value (output level inversion timing)
(9) Output level inversion at the PPG pin
(10) Counter decrement
(11) The down counter matches the duty value end point (output level clear timing)
(12) Clearing of PPG pin output level (restoration to normal state)
(13) Counter decrement
(14) Counter borrow occurrence
(15) Clearing of PPG pin output level (restoration to normal state)
(16) Reloading of the [(Cycle value) × 2 + 1] to the down count value (PTMR) and the duty value to the buffer (duty value) and the buffer (duty value end point)
(17) Repetition of steps (7) to (16)

Calculation formulas:

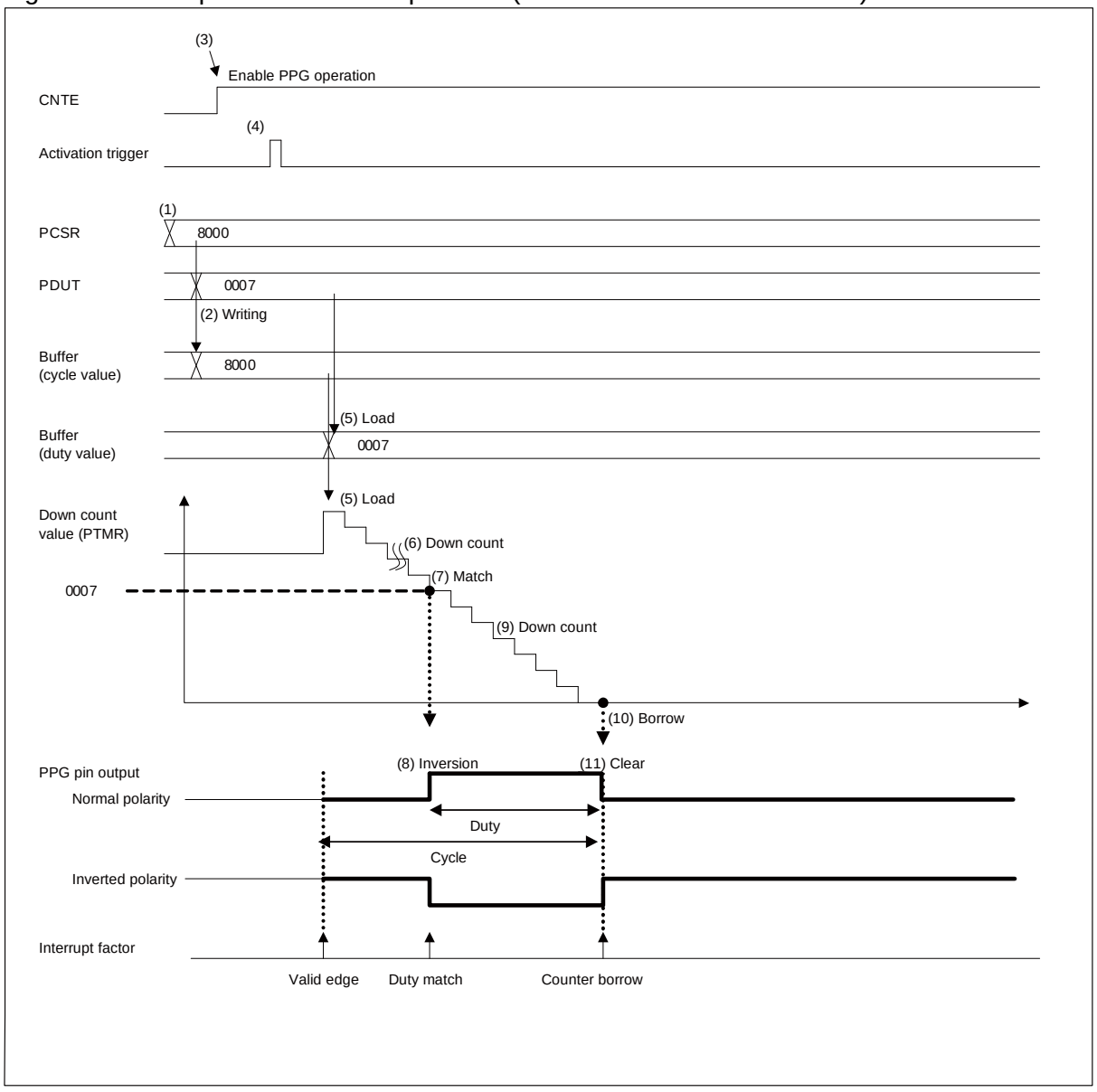
- Cycle = {(Cycle value (PCSR) + 1) × 2} × Count clock
- Duty = {(Duty value (PDUT) + 1) × 2} × Count clock
- Time to pulse output = {Cycle value (PCSR) - Duty value (PDUT)} × Count clock

5.3. One-shot Operation (Normal Wave Form Selected)

The One-shot operation (Normal Wave Form selected) is explained.

During the one-shot operation, one-shot pulses are output at the PPG pin.

Figure 5-3 Example of One-shot Operation (Normal Wave Form Selected)



Setting and operation procedure:

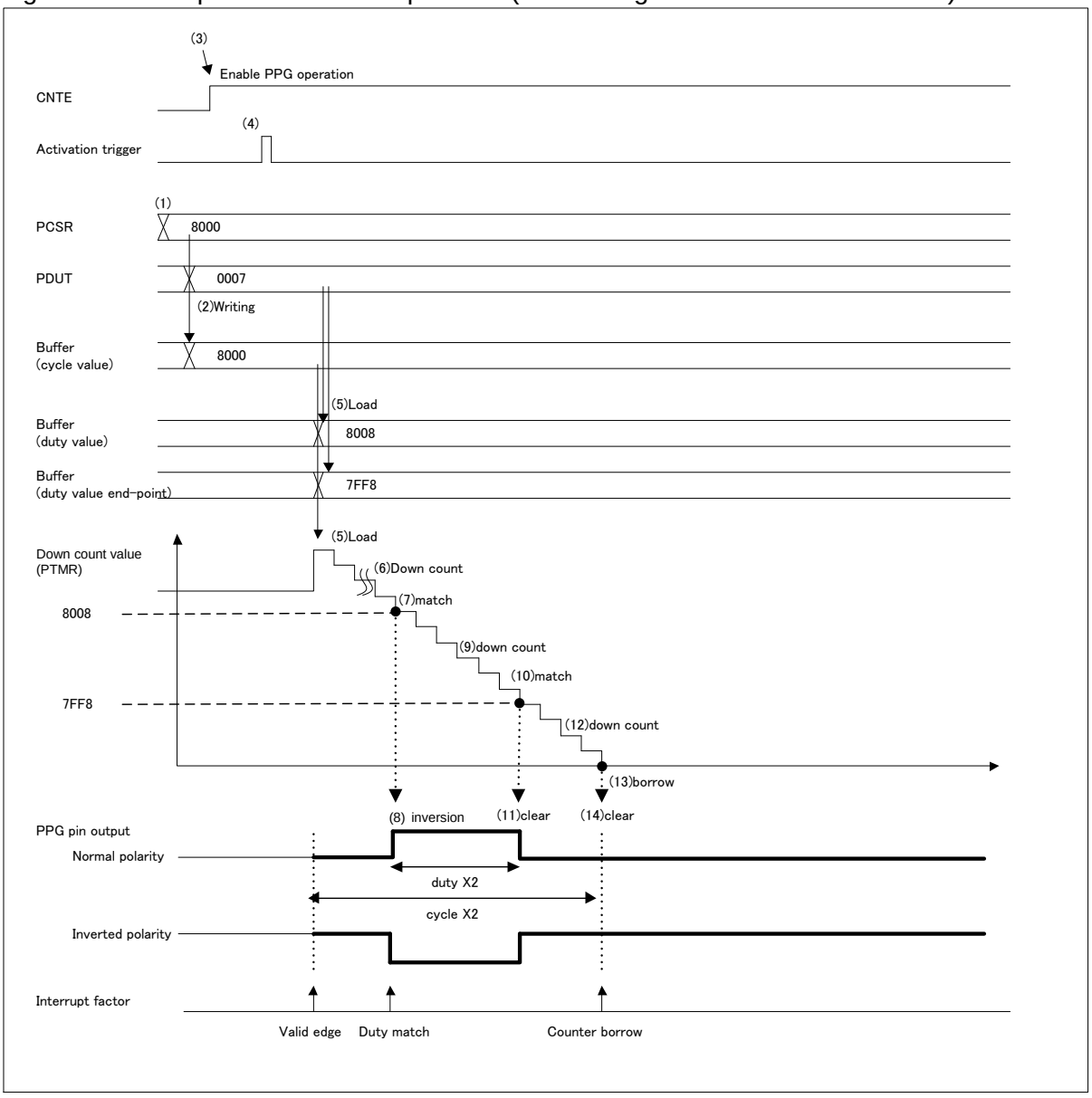
(1) Writing of PCSR (cycle value)
(2) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(6) Counter 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 of PPG pin output level (restoration to normal state)
(12) End of operation sequence

5.4. One-shot Operation (Center Aligned Wave Form Selected)

The One-shot operation (Center Aligned Wave Form selected) is explained.

During the one-shot operation, one-shot pulses are output at the PPG pin. If the PPG output waveform selection bit (PCN.OWFS)="1" is selected (Center Aligned Wave Form), the output waveform is generated by doubling the value of the PPG cycle setting register (PCSR) and the value of the PPG duty setting register (PDUT).

Figure 5-4 Example of One-shot Operation (Center Aligned Wave Form Selected)



Setting and operation procedure:

(1) Writing of PCSR (cycle value)
(2) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading of the [(Cycle value) × 2 + 1] to the down count value (PTMR) and the duty value to the buffer (duty value) and the buffer (duty value end point)
    Duty value (Output level inversion timing) = (Duty value + Cycle value + 1)
    Duty value end point (Output level clear timing) = (Cycle value - Duty value - 1)
(6) Counter decrement
(7) The down counter matches the duty value (output level inversion timing)
(8) Output level inversion at the PPG pin
(9) Counter decrement
(10) The down counter matches the duty value end point (output level clear timing)
(11) Clearing of PPG pin output level (restoration to normal state)
(12) Counter decrement
(13) Counter borrow occurrence
(14) Clearing of PPG pin output level (restoration to normal state)
(15) End of operation sequence

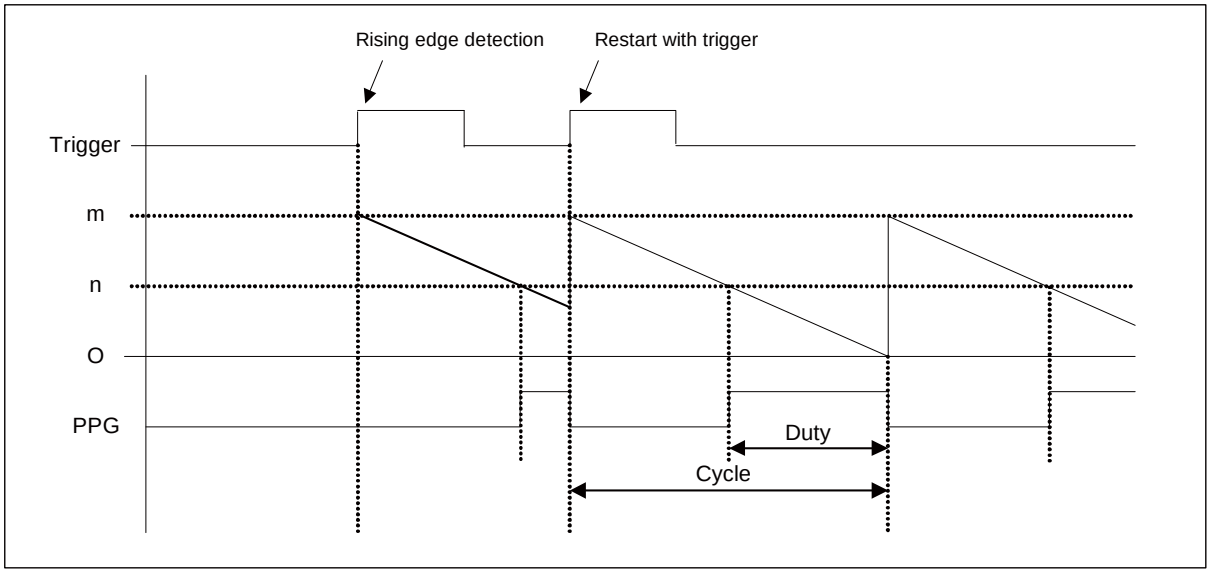
5.5. Restart Operation

The restart operation is explained.

Restart operation is as follows:

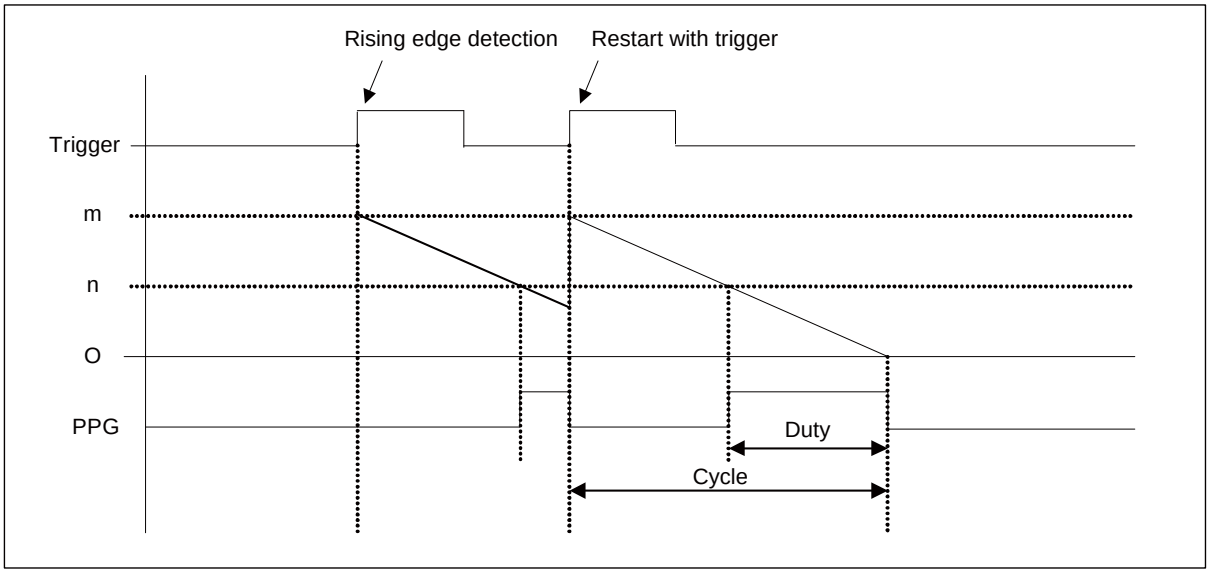
■ Restart during PWM Operation

Figure 5-5 Restart during PWM Operation (Normal Waveform Selected)



■ Restart during One-shot Operation

Figure 5-6 Restart during One-shot Operation (Normal Waveform Selected)



Regardless of whether PWM operation or one-shot operation is being performed, the second or subsequent trigger does not influence operation (does not cause restart) if the restart enable bit (PCN:RTRG) is "0" (restart disabled).

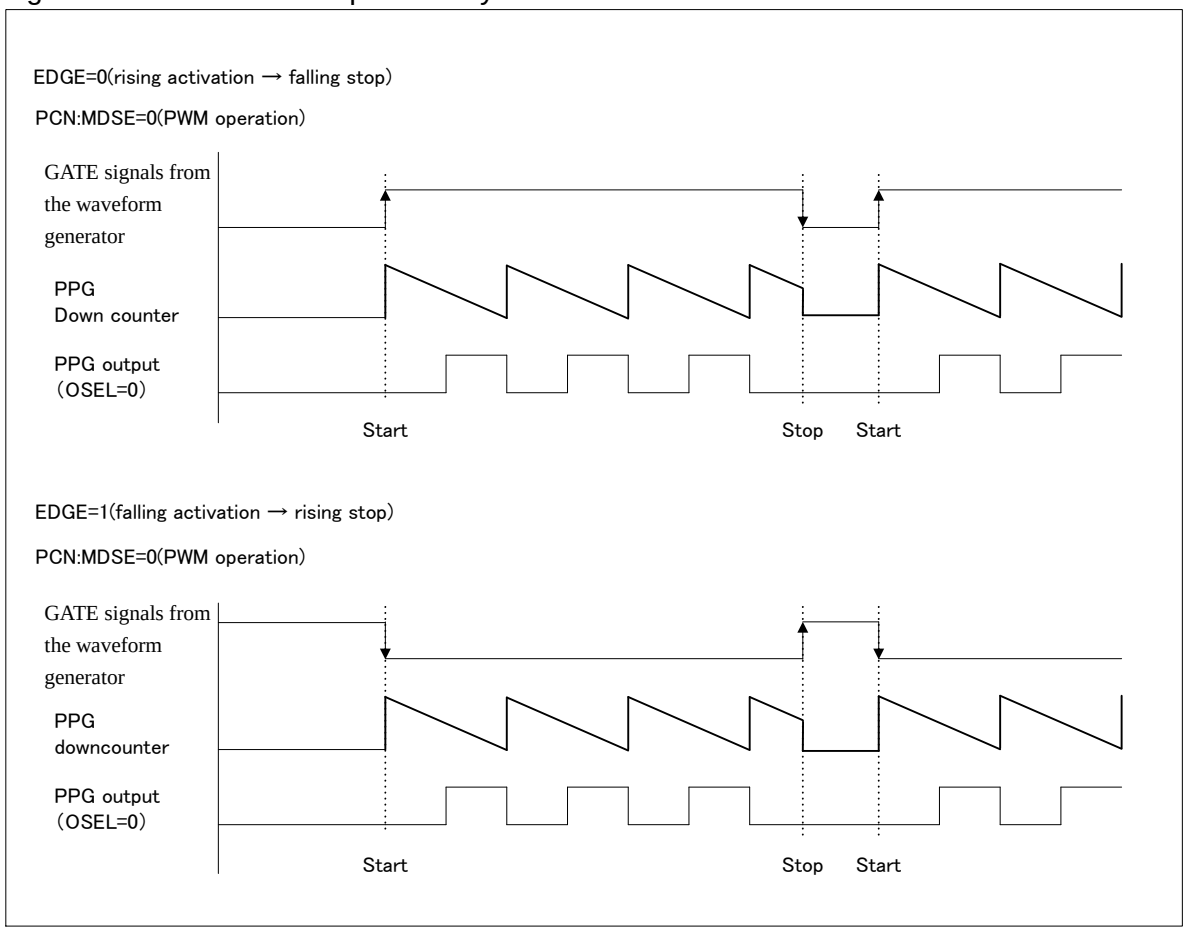
However, if, in the case of one-shot operation, the second or subsequent trigger occurs after one-shot operation, it works as a restart trigger.

5.6. GATE Operation

The GATE operation is explained.

PPG can be activated/stopped by GATE signals from the waveform generator. PPG activation valid time can be controlled by the EDGE bit of the GATE control register (GATEC:EDGE) and GATE signals from the waveform generator.

Figure 5-7 PPG Counter Operation by GATE Function



If the one-shot pulse operation (PCN:MDSE= "1") is set when PPG operation is selected (GATEC:STGR) by GATE function, continuous pulses are output from the PPG output pin in the same way as the PWM operation.

Note:

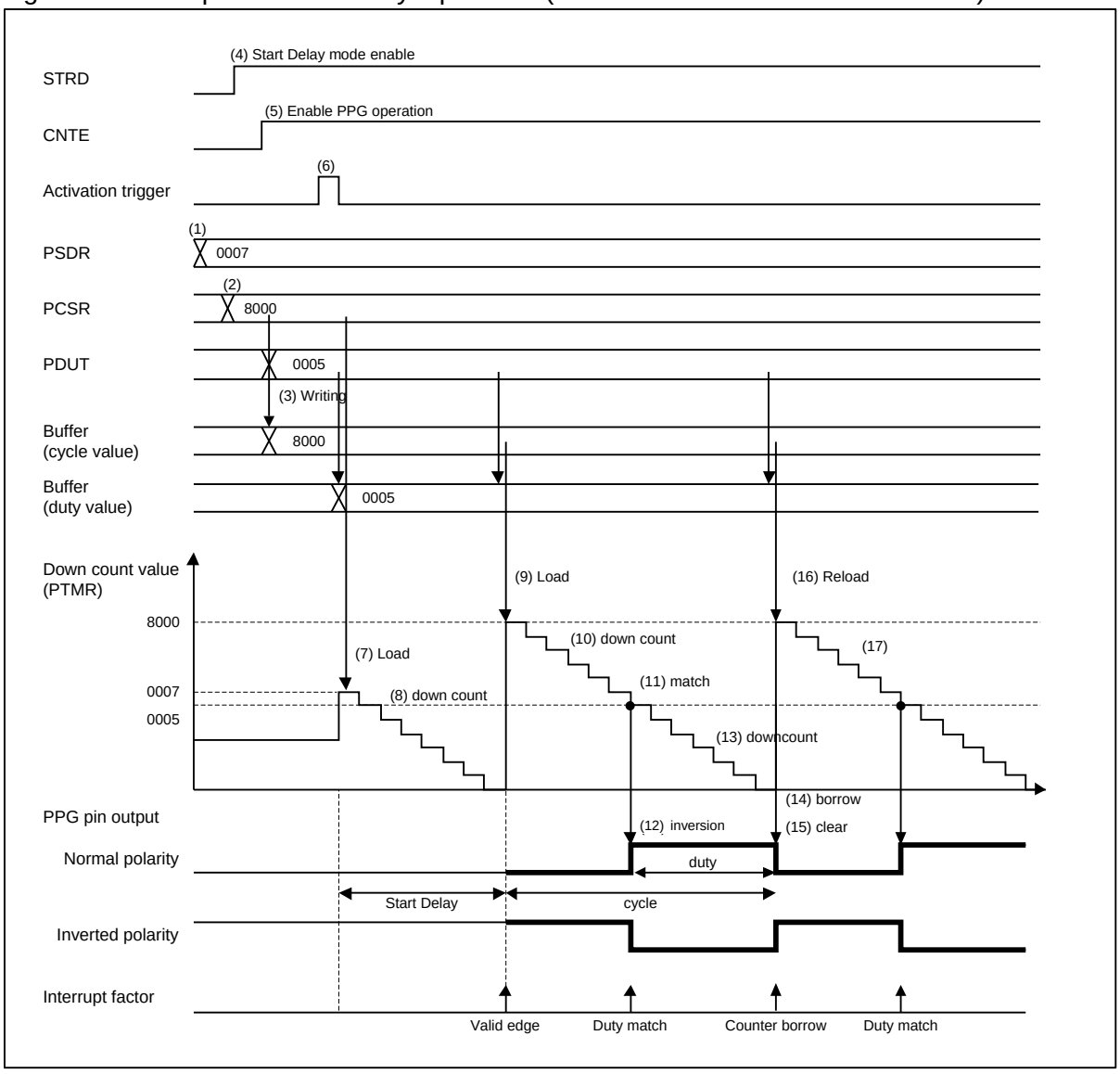
- If the GATE signal has changed from "1" to "0" (when EDGE="0") during the PPG operation, the PPG down counter value (PTMG) will be maintained and the PPG output will be changed to "L" and will be stopped.
- When the GATE function is enabled (STGR = "1") and the restart enable register is valid (RTRG = "1"), restart operation does not start even when another activation trigger is input.

5.7. Start Delay Mode Operation (PWM Normal Wave Form Selected)

The Start Delay mode operation (PWM Normal Wave Form selected) is explained.

In the Start Delay operation, a valid edge of PPG waveform output is delayed for the Start Delay value setting period. If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, the delay value is doubling the set value of the Start Delay value setting register. Moreover, the PWM operation, the one shot operation, Normal Wave Form, and Center Aligned Wave Form are supported.

Figure 5-8 Example of Start Delay Operation (PWM Normal Wave Form Selected)



Setting and operation procedure:

(1) Writing of PSDR (Delay value)
(2) Writing of PCSR (cycle value)
(3) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(4) Enabling Start Delay mode
(5) Enabling of PPG operation
(6) Activation trigger generation
(7) Loading of the Delay values
(8) Counter decrement (Delay value set by (1))
(9) Loading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(10) Counter decrement
(11) The down counter matches the duty value
(12) Output level inversion at the PPG pin

(13) Counter decrement
(14) Counter borrow occurrence
(15) Clearing of PPG pin output level (restoration to normal state)
(16) Reloading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(17) Repetition of steps (10) to (16)

Calculation formulas:

- Start Delay value = {Start Delay value (PSDR) + 1} × Count clock
- 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

Note:

The calculating formula when Center Aligned Wave Form is selected is as follows.

Calculation formulas:

- Start Delay value = {(Register setting value (PSDR) + 1) × 2} × Count clock
- Cycle = {(Cycle value (PCSR) + 1) × 2} × Count clock
- Duty = {(Duty value (PDUT) + 1) × 2} × Count clock
- Time to pulse output = {Cycle value (PCSR) - Duty value (PDUT)} × Count clock

Notes:

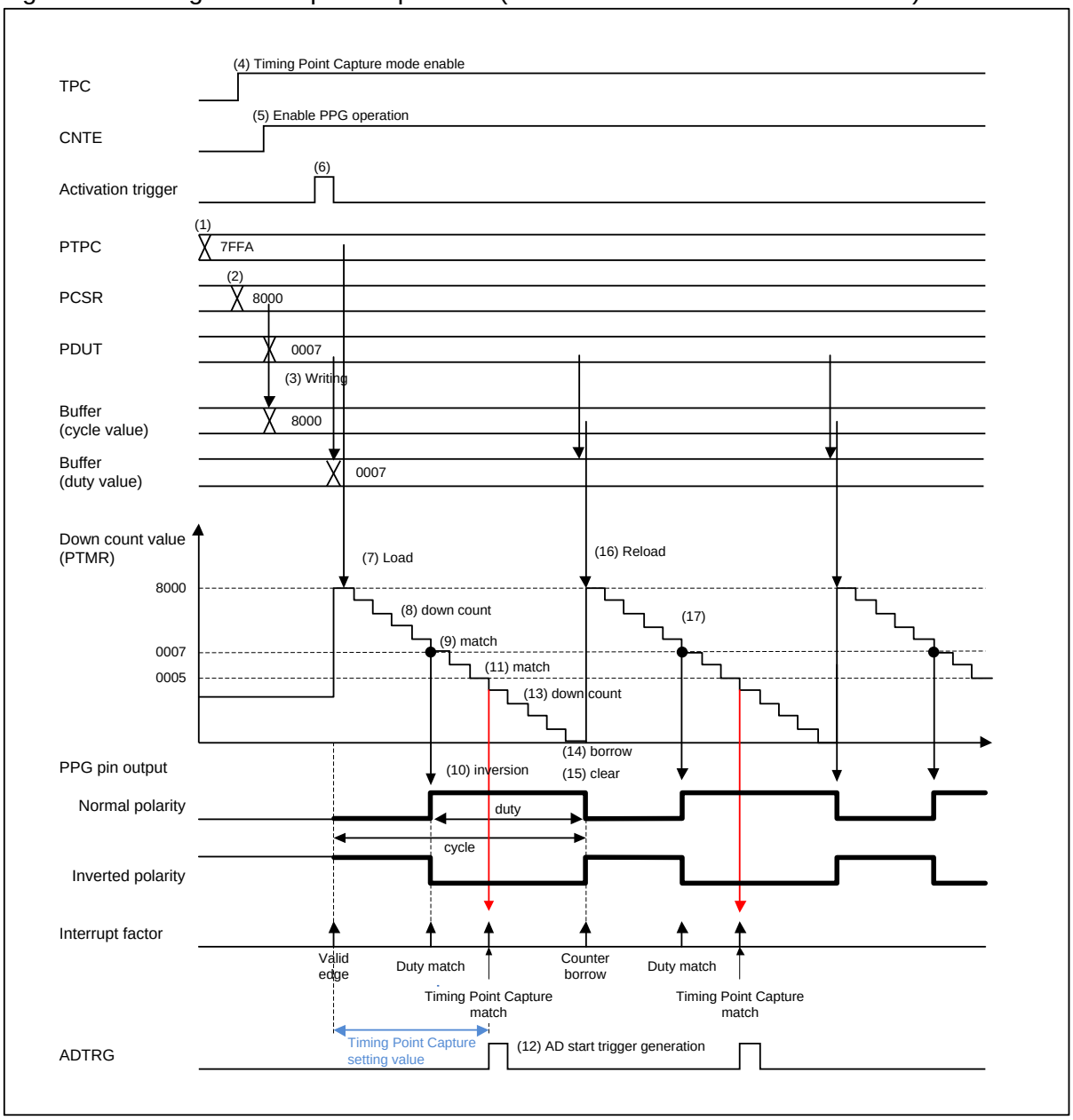
- When the Start Delay value (PSDR) is rewritten during the PPG operation in the Start Delay mode, the Start Delay value becomes effective after prohibiting operating once and generating the activation trigger. (The Start Delay value becomes effective with the activation trigger.)
- The Start Delay setting period (PSDR) is waited again when the restart request is generated during the Start Delay operation (waiting time period). Moreover, the PPG waveform output is stopped and the Start Delay setting period is waited again when the restart is requested while outputting PPG waveform by the Start Delay mode enable (STRD)="1".
- Be sure to prohibit operating once when "0" is written in the Start Delay mode enable (STRD) during the Start Delay period. Moreover, if the activation trigger is not generated, the Start Delay mode disable (STRD=0) does not become effective.

5.8. Timing Point Capture Mode Operation (PWM Normal Wave Form Selected)

The Timing Point Capture mode operation (PWM Normal Wave Form selected) is explained.

In the Timing Point Capture mode operation, the interrupt and the AD activation trigger are generated according to the timing of the Timing Point Capture setting value. If the PPG output waveform selection bit (PCN.OWFS)="1" (Center Aligned Wave Form) is selected, the doubling setting value of the Timing Point Capture value setting register and the down counter value are compared. Moreover, the PWM operation, the one shot operation, Normal Wave Form, and Center Aligned Wave Form are supported.

Figure 5-9 Timing Point Capture Operation (PWM Normal Wave Form Selected)



Setting and operation procedure:

(1) Writing of PTPC (Timing Point Capture value)
(2) Writing of PCSR (cycle value)
(3) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(4) Enabling Timing Point Capture mode
(5) Enabling of PPG operation
(6) Activation trigger generation
(7) Loading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(8) Counter decrement
(9) The down counter matches the duty value
(10) Output level inversion at the PPG pin
(11) Passage of Timing Point Capture setting period from activation trigger
(12) Interrupt and A/D activation trigger generation
(13) Counter decrement
(14) Counter borrow occurrence
(15) Clearing of PPG pin output level (restoration to normal state)
(16) Reloading of the cycle value to the down count value (PTMR) and the duty value to the buffer (duty value)
(17) Repetition of steps (8) to (16)

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
- Timing Point Capture interrupt and A/D activation trigger generation : Passage of {PTPC register setting value + 1} period from the activation trigger

---

Note:

The calculating formula when Center Aligned Wave Form is selected is as follows.

Calculation formulas:

- Cycle = {(Cycle value (PCSR) + 1) × 2} × Count clock
- Duty = {(Duty value (PDUT) + 1) × 2} × Count clock
- Time to pulse output = {Cycle value (PCSR) - Duty value (PDUT)} × Count clock
- Timing Point Capture interrupt and A/D activation trigger generation : Passage of {(PTPC register setting value + 1) × 2} period from the activation trigger

---

Notes:

- The Timing Point Capture value setting (PTPC) has to set smaller than the cycle value (PCSR). When the value that is larger than cycle value (PCSR) is set, the A/D activation trigger or the Timing Point Capture match interrupt is not generated.
- The value becomes effective at the next cycle after rewriting when the Timing Point Capture value (PTPC) is rewritten during the PPG operation.
- When "0" is written in the Timing Point Capture mode enable (TPC) during the PPG operation, neither the interrupt by the Timing Point Capture value match nor the A/D activation trigger is generated. Be sure to set the setting according to the procedure when Timing Point Capture mode enable (TPC) is set again.

5.9. PPG Communication Mode Operation

The PPG communication mode operation is explained.

In the PPG communication mode operation, the waveform according to the setting is output by setting the cycle/duty of the High/Low format, the data, and the bit length. The operation of the PPG communication function is different from usual PWM operation and the one-shot operation. (There is no Center Aligned Wave Form, and only Normal Wave Form corresponds.)

In the PPG communication mode, the following registers are valid or invalid.

There is no influence on the PPG communication operation though writing in an invalid register is possible.

Valid registers:

Software trigger (STRG), Count clock selection (CKS1, CKS0),
PPG output mask selection (PGMS), Trigger input selection (EGS1, EGS0),
Interrupt request enable (IREN), Interrupt factor selection (IRS1, IRS0),
Interrupt request flag (IRQF) (*1),
PPG output polarity selection (OSEL),
PPG timer (PTMR), GATE function control (GATEC),
Low format pulse polarity selection (LFPR), high format pulse polarity selection (HFPR),
PPG communication mode data reading selection (CMDSEL),
PPG communication mode enable (CMD),
PPG communication data register Empty flag (REMP) (*2),
PPG communication data shift register Empty flag (SREMP) (*2),
Interrupt factor selection2 (IRS2),
High format cycle setting (PHCSR), Low format cycle setting (PLCSR),
High format duty setting (PHDUT),
Low format duty setting (PLDUT),
Communication mode data setting (PCMDDT),
Communication mode data bit length setting (PCMDWD)

Invalid registers:

Timer operation enable (CNTE), Mode selection (MDSE), Restart enable (RTRG),
PPG output waveform selection (OWFS),
PPG cycle setting (PCSR), PPG duty setting (PDUT),
Timing Point Capture enable (TPC), Start Delay enable (STRD),
Start Delay value setting (PSDR), Timing Point Capture value setting (PTPC)

(*1) IRS[2:0]=000b to 100b of the interrupt selection cannot be set during PPG communication; however, the registers are enabled.
(*2) Cannot be set because this is a read only register.

5.10. PPG Communication Activation

The PPG communication activation is explained.

The PPG communication mode starts by setting the PPG communication enable (CMD), the cycle setting (PHCSR, PLCSR), the duty setting (PHDUT/PLDUT), the data (PCMDDT), and the data bit length (PCMDWD), and then set the activation trigger at the end. There is no restriction in the order of setting registers other than the activation trigger.

Figure 5-10 Example of PPG Communication Mode Operation (Activation Operation Case 1)

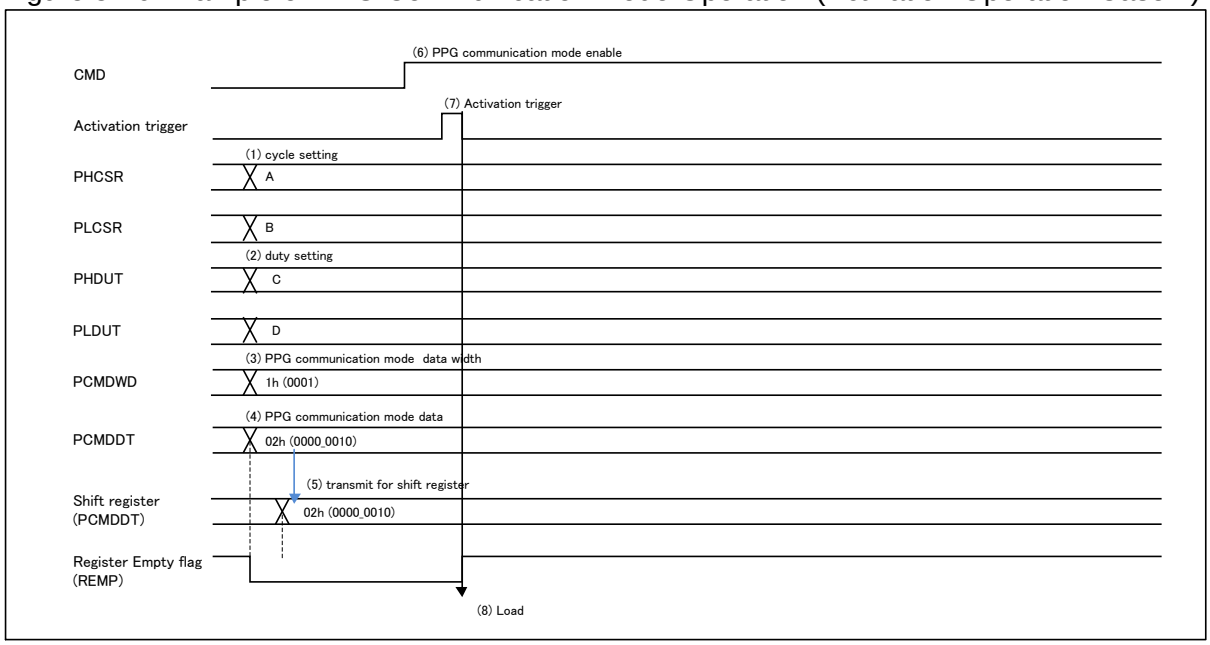
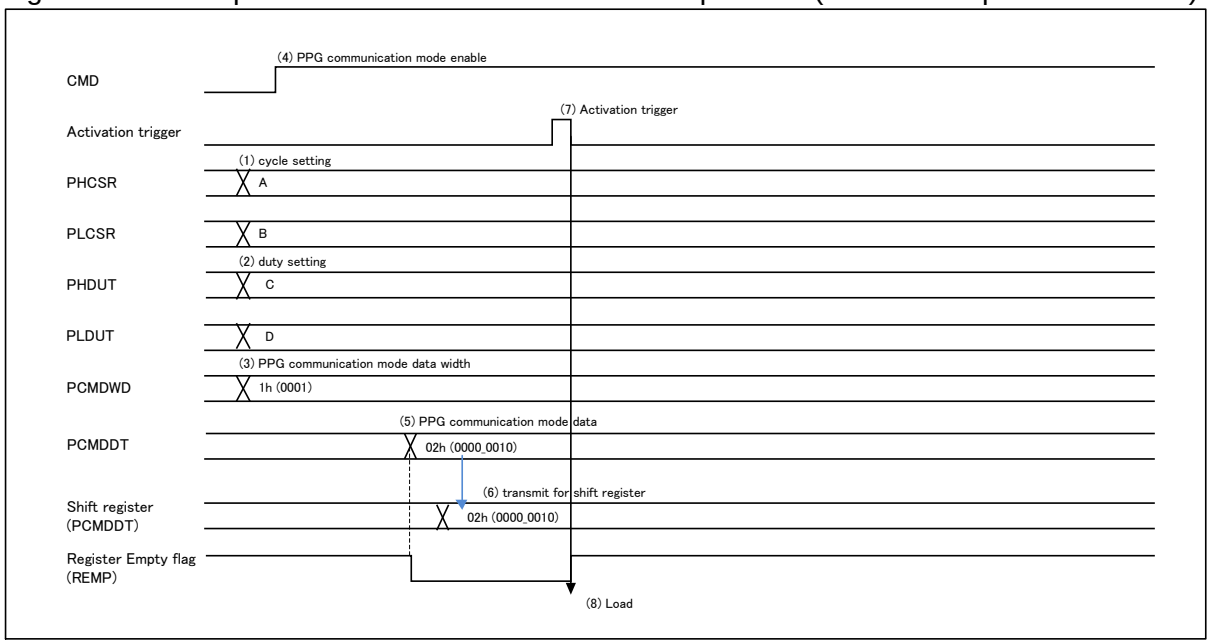


Figure 5-11 Example of PPG Communication Mode Operation (Activation Operation Case 2)



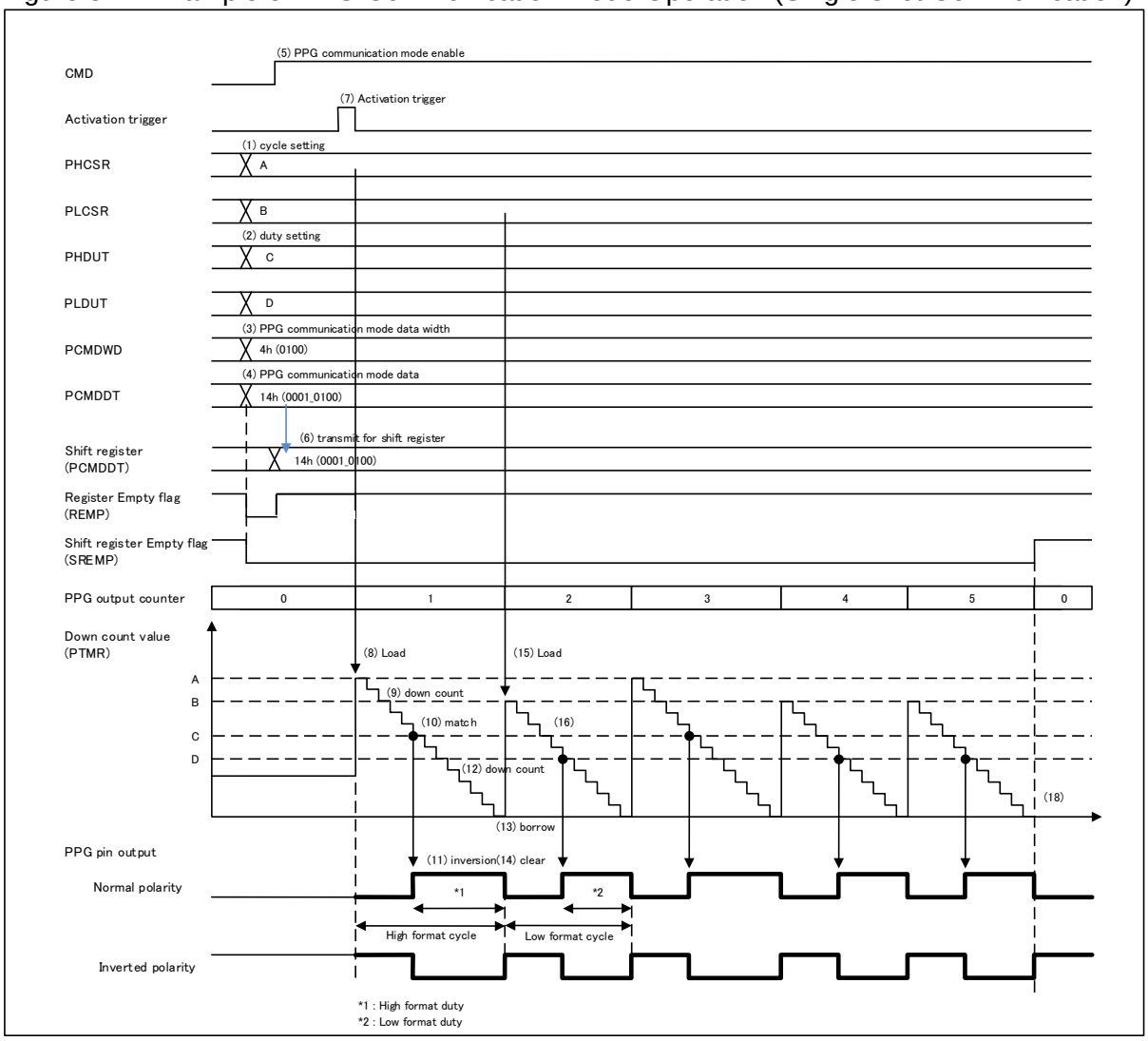
5.11. PPG Communication Operation

The PPG communication operation is explained.

After it is initialized, the waveform according to the PPG communication data bit length (PCMDWD) setting is output. If the PPG communication data setting (PCMDDT) is not rewritten before completion of the waveform output (Figure 5-12), PPG communication ends. If the PPG communication data setting (PCMDDT) is rewritten, the waveform output is completed according to the bit length setting. Then, PPG communication starts again with the rewritten data setting (PCMDDT) (Figure 5-13).

■ PPG Single-shot Communication Operation

Figure 5-12 Example of PPG Communication Mode Operation (Single-shot Communication)



Setting and operation procedure:

(1) Writing of PHCSR/PLCSR (High/Low format cycle values)
(2) Writing of PHDUT/PLDUT (High/Low format duty value)
(3) Writing of PCMDWD (PPG communication mode data width)
(4) Writing of PCMDDT (PPG communication mode data)
(5) Enabling of PPG communication mode
(6) Transmitting PCMDDT (PPG communication mode data) to the shift register
(7) Activation trigger generation
(8) Loading of the cycle value to the down count value (PTMR) and the duty value (Which cycle and duty of the High or Low format is loaded is determined according to the PCMDDT, PCMDWD, and CMDSEL setting)
(9) Counter decrement
(10) The down counter matches the duty value
(11) Output level inversion at the PPG pin
(12) Counter decrement
(13) Counter borrow occurrence
(14) Clearing of PPG pin output level (restoration to normal state)
(15) Reloading of the cycle value to the down count value (PTMR) and the duty value
(16) Repetition of steps (9) to (15) according to the setting of PCMDDT and PCMDWD
(17) Operation sequence completion

Notes:

- There is no restriction by the order of setting above-mentioned (1) to (5). However, the PPG communication operation does not start if the setting is not written all of (1) to (5).
- The PPG communication does not start when the activation trigger is generated without completing the setting of above-mentioned (1) to (5). Moreover, in this case, it is necessary to set (1) to (5) again to clear the setting once.

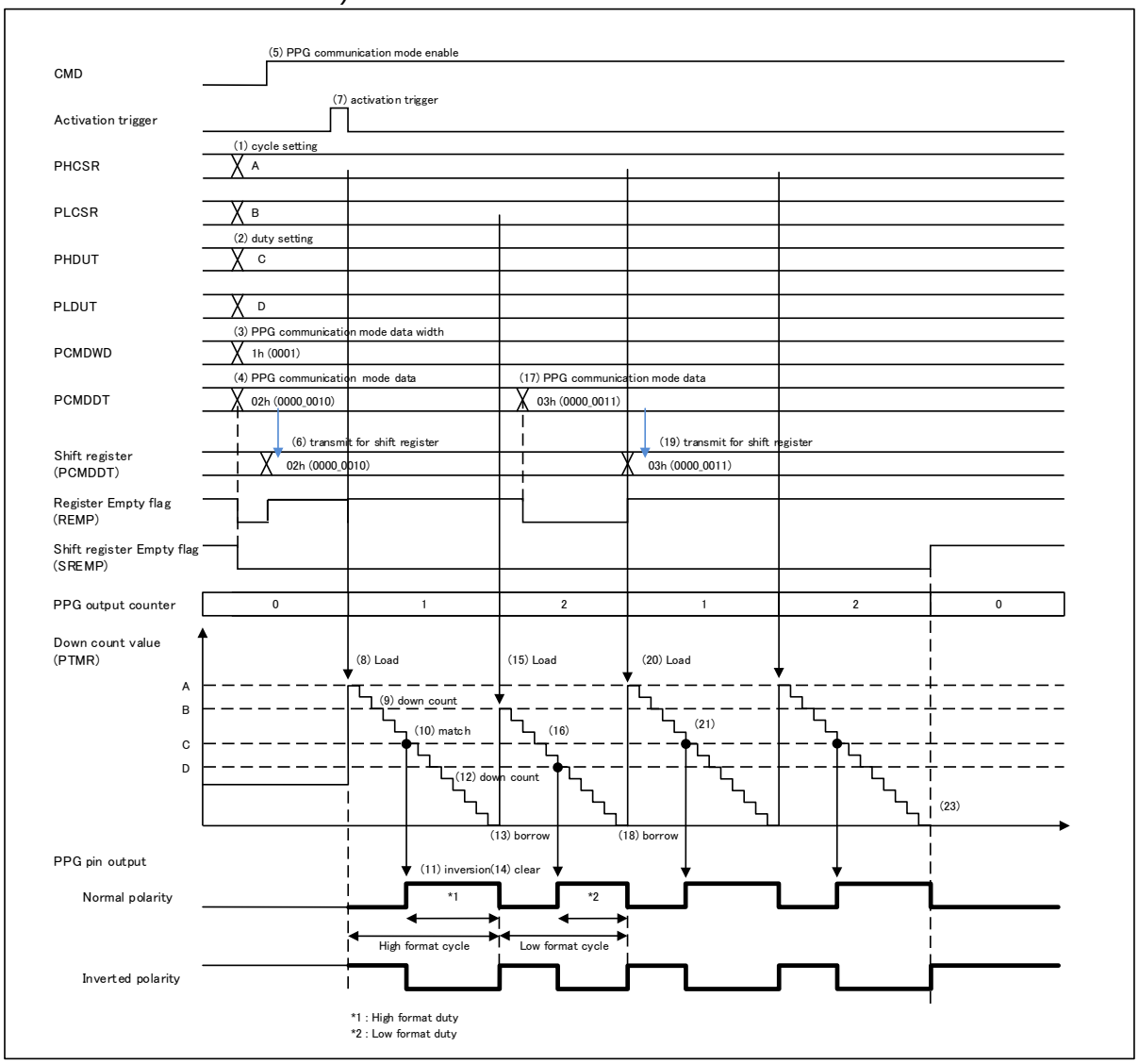
Note:

Calculation formulas: The calculating formula is common in all modes.

- 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

■ PPG Continuousness Communication Operation

Figure 5-13 Example of PPG Communication Mode Operation (Continuousness Communication)



Setting and operation procedure:

(1) Writing of PHCSR/PLCSR (High/Low format cycle values)
(2) Writing of PHDUT/PLDUT (High/Low format duty value)
(3) Writing of PCMDWD (PPG communication mode data width)
(4) Writing of PCMDDT (PPG communication mode data)
(5) Enabling of PPG communication mode
(6) Transmitting PCMDDT (PPG communication mode data) to the shift register
(7) Activation trigger generation
(8) Loading of the cycle value to the down count value (PTMR) and the duty value (Which cycle and duty of the High or Low format is loaded is determined according to the PCMDDT, PCMDWD, and CMDSEL setting)
(9) Counter decrement
(10) The down counter matches the duty value
(11) Output level inversion at the PPG pin
(12) Counter decrement
(13) Counter borrow occurrence
(14) Clearing of PPG pin output level (restoration to normal state)
(15) Reloading of the cycle value to the down count value (PTMR) and the duty value
(16) Repetition of steps (9) to (15) according to the setting of PCMDDT and PCMDWD
(17) Writing of PCMDDT (PPG communication mode data)
(18) Counter borrow occurrence
(19) Transmitting PCMDDT (PPG communication mode data) to the shifter
(20) Loading of the cycle value to the down count value (PTMR) and the duty value (Which cycle and duty of the High or Low format is loaded is determined according to the PCMDDT, PCMDWD, and CMDSEL setting)
(21) Repetition of steps (9) to (15) according to the setting of PCMDDT and PCMDWD
(22) Operation sequence completion

Notes:

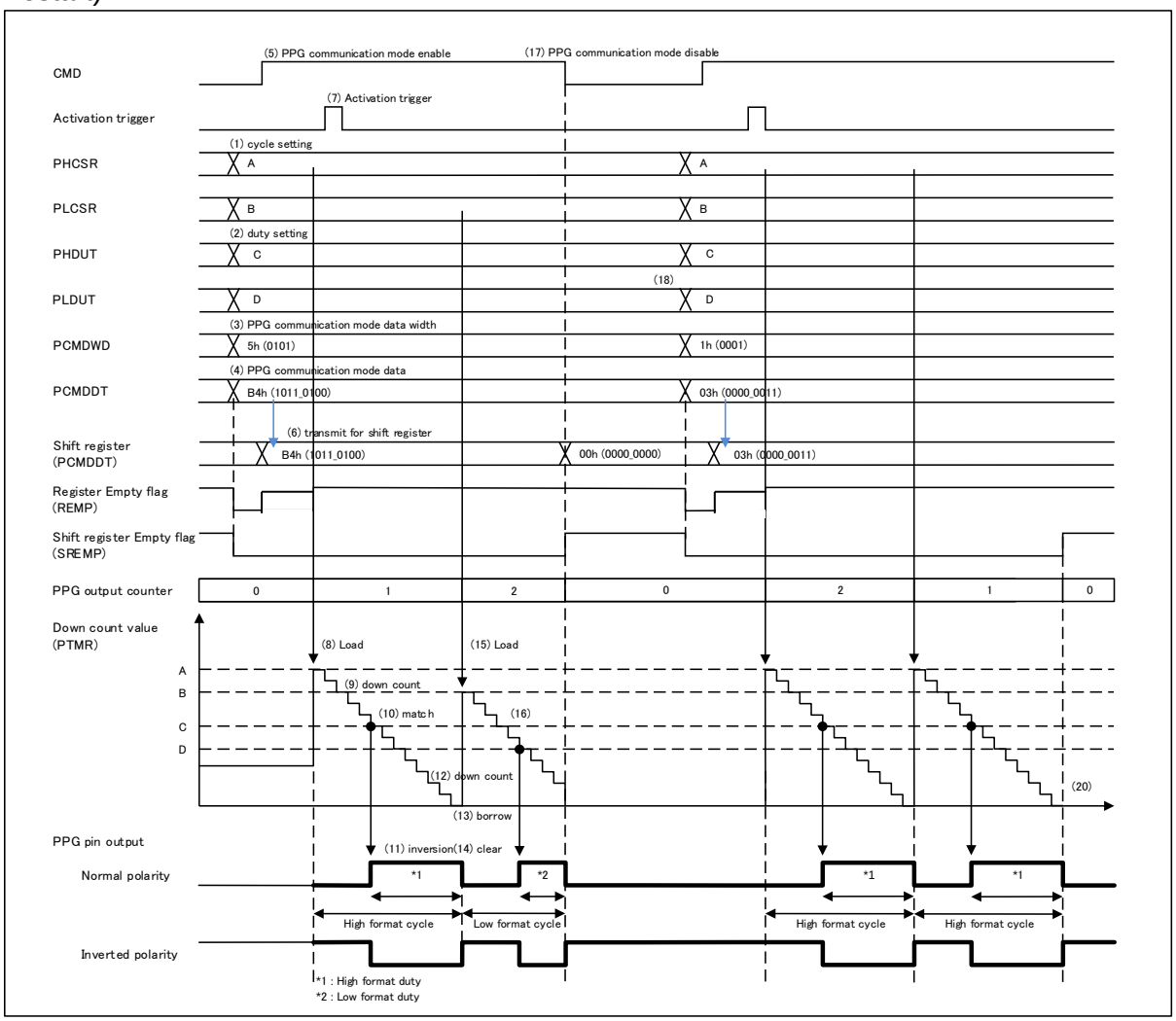
- There is no restriction by the order of setting above-mentioned (1) to (5). However, the PPG communication operation does not start if the setting is not written all of (1) to (5).
- The PPG communication does not start when the activation trigger is generated without completing the setting of above-mentioned (1) to (5). Moreover, in this case, it is necessary to set (1) to (5) again to clear the setting once.

5.12. PPG Communication Forced Stop and Restart operation

The PPG communication forced stop and restart operation is explained.

The PPG communication stops and the internal circuit other than the setting register are initialized when "0" is written in the PPG communication mode setting register (CMD) during the PPG communication mode operation. Moreover, to restart the PPG communication operation, it is necessary to write the setting in all related registers.

Figure 5-14 Example of PPG Communication Mode Operation (Communication Forced Stop and Restart)



Setting and operation procedure:

(1) Writing of PHCSR/PLCSR (High/Low format cycle values)
(2) Writing of PHDUT/PLDUT (High/Low format duty value)
(3) Writing of PCMDWD (PPG communication mode data width)
(4) Writing of PCMDDT (PPG communication mode data)
(5) Enabling of PPG communication mode
(6) Transmitting PCMDDT (PPG communication mode data) to the shift register
(7) Activation trigger generation
(8) Loading of the cycle value to the down count value (PTMR) and the duty value (Which cycle and duty of the High or Low format is loaded is determined according to the PCMDDT, PCMDWD, and CMDSEL setting)
(9) Counter decrement
(10) The down counter matches the duty value
(11) Output level inversion at the PPG pin
(12) Counter decrement
(13) Counter borrow occurrence
(14) Clearing of PPG pin output level (restoration to normal state)
(15) Reloading of the cycle value to the down count value (PTMR) and the duty value
(16) Repetition of steps (9) to (15) according to the setting of PCMDDT and PCMDWD
(17) PPG communication mode disable (internal circuit and flag register clear)
(18) Resetting all of (1) to (7), repetition of steps (8) to (15) according to the setting of PCMDDT and PCMDWD
(19) Operation sequence completion

Notes:

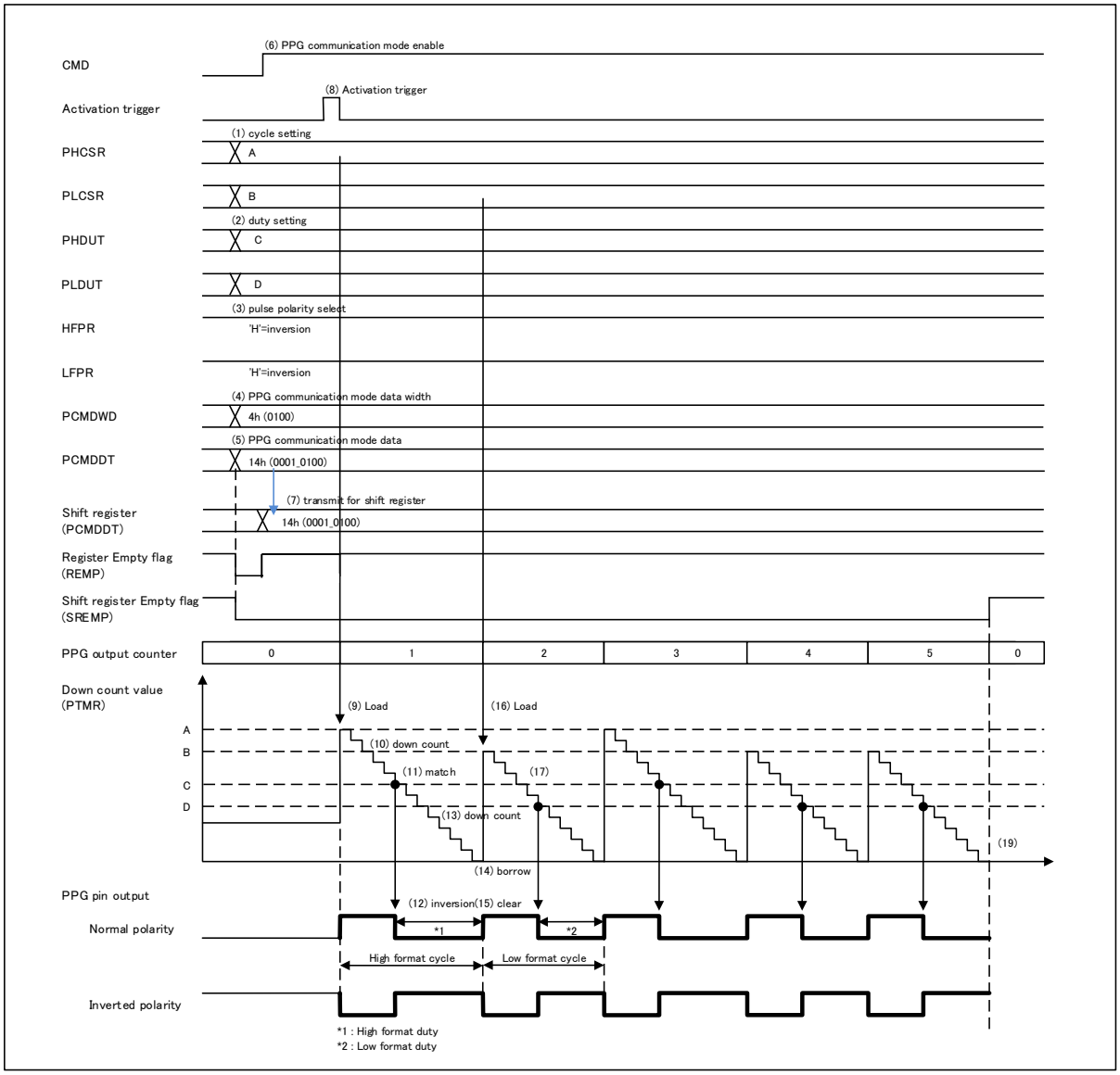
- There is no restriction by the order of setting above-mentioned (1) to (5). However, the PPG communication operation does not start if the setting is not written all of (1) to (5).
- The PPG communication does not start when the activation trigger is generated without completing the setting of above-mentioned (1) to (5).
  Moreover, in this case, it is necessary to set (1) to (5) again to clear the setting once.

5.13. PPG Output Pulse Polarity Selection

The PPG output pulse polarity selection is explained.

The PPG waveform can be output from the High pulse by writing "1" in the High/Low format pulse polarity selection register (HFPR/LFPR). (Output from the Low pulse in "0" setting.)

Figure 5-15 Example of PPG Communication Mode Operation (Output Pulse Selection Mode "1" Setting)



Setting and operation procedure:

(1) Writing of PHCSR/PLCSR (High/Low format cycle values)
(2) Writing of PHDUT/PLDUT (High/Low format duty value)
(3) Writing of HFPR/LFPR (High/Low format pulse polarity selection)
(4) Writing of PCMDWD (PPG communication mode data width)
(5) Writing of PCMDDT (PPG communication mode data)
(6) Enabling of PPG communication mode
(7) Transmitting PCMDDT (PPG communication mode data) to the shift register
(8) Activation trigger generation
(9) Loading of the cycle value to the down count value (PTMR) and the duty value (Which cycle and duty of the High or Low format is loaded is determined according to the PCMDDT, PCMDWD, and CMDSEL setting)
(10) Counter decrement
(11) The down counter matches the duty value
(12) Output level inversion at the PPG pin
(13) Counter decrement
(14) Counter borrow occurrence
(15) Clearing of PPG pin output level (restoration to normal state)
(16) Reloading of the cycle value to the down count value (PTMR) and the duty value
(17) Repetition of steps (10) to (16) according to the setting of PCMDDT and PCMDWD
(18) Operation sequence completion

Notes:

- There is no restriction by the order of setting above-mentioned (1) to (6). However, the PPG communication operation does not start if the setting is not written all of (1), (2), and (4) to (6).
- The PPG communication does not start when the activation trigger is generated without completing the setting of above-mentioned (1) to (5).
  Moreover, in this case, it is necessary to set (1) to (5) again to clear the setting once.

5.14. Interrupt

The interrupt is explained.

The interrupt request is generated for the following either.

- Software trigger, External trigger, and GATE signal trigger
- Counter borrow occurrence (The set cycle is matched)
- Match of duty
- Match of Timing Point Capture value
- PPG communication data register Empty flag

The generated interrupt factor is different depending on the operation mode of PPG.

Table 5-1 Correspondence of Operation Mode and Interrupt Request

Interrupt request	PWM operation and one-shot operation	Timing Point Capture mode	PPG communication mode
Software trigger, External trigger, and GATE signal trigger	○	○	×
Counter borrow occurrence	○	○	×
Match of counter and duty value	○	○	×
Counter borrow occurrence, or match of counter and duty value	○	○	×
Match of Timing Point Capture value	×	○	×
PPG communication data register Empty factor	×	×	○

○: Supported
×: Not supported

Table 5-2 shows the register related to the interrupt of each operation mode.

Table 5-2 Interrupt Setting of Each Operation Mode

Interrupt setting	Setting of interrupt factor	Interrupt flag	Interrupt request enable	Clear of interrupt request
PWM operation, One shot operation	PCNn.IRS[1:0] =00 to 11	PCNn.IRQF=1	PCNn.IREN=1	"0" is written in PCN.IRQF.
Timing point capture mode	PCN2n.IRS[2:0] =000 to 100	PCNn.IRQF=1	PCNn.IREN=1	"0" is written in PCN.IRQF.
PPG communication mode	PCN2n.IRS[2:0] =101 to 111	PCN2n.REMP=1 or PSC2n.SREMP=1	PCNn.IREN=1	"0" is written in PCN.IRQF.

(n=0 to 47)
PCN: PPG control register
PCN2: PPG control register2

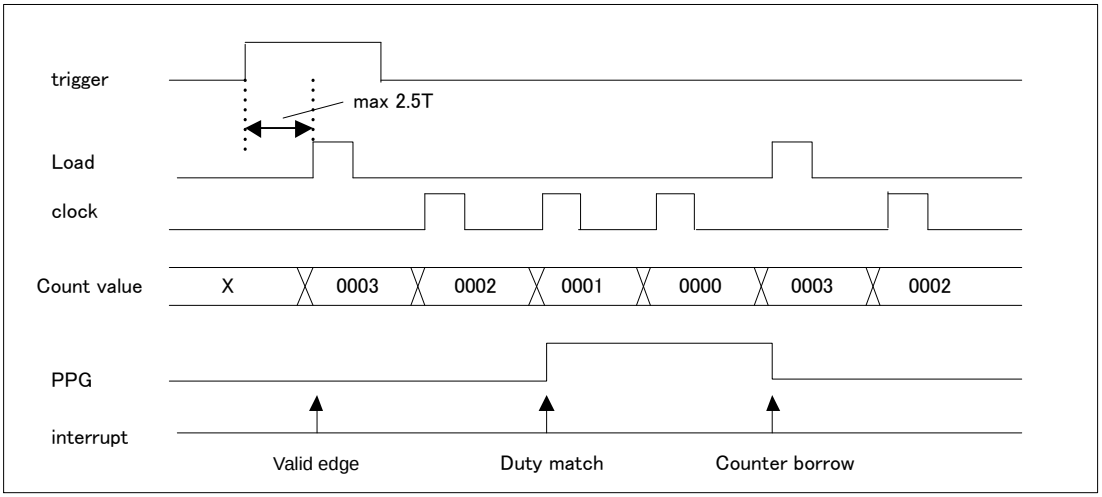
6. Notes

This section explains notes of the PPG.

Note the following when using the PPG:

■ PPG Operation Activation

1. The first load delays up to 2.5T after the activation trigger (T: Count clock). If an operation that loads a value to the down counter and another operation that decrements the counter occur at the same time, the first operation takes precedence (overwrites the down counter).



2. To activate the PPG, the timer operation enable bit (PCN:CNTE) must be set to "1" before or when PPG operation is enabled.

3. The PPG activation with a PSTR pin (PPG activation trigger) as a trigger is the same operation as the PPG activation using a software trigger or an external trigger (TRG pin).
Beforehand, it is necessary to set the register necessary for the PPG operation.

■ PPG is Operating

1. During the PPG operation, do not change any of the following: the mode selection bit (PCN:MDSE), the restart enable bit (PCN:RTRG), the counter clock selection bits (PCN:CKS1, CKS0), the trigger input edge selection bits (PCN:EGS1, EGS0), the interrupt factor selection bits (PCN:IRS1, IRS0), the activation trigger selection bits (GTRS:TSEL[6:0]), and the PPG output polarity selection bit (PCN:OSEL).
If any of the above bits is changed during the PPG operation, disable PPG operation before reconfiguring the register.

2. If the timer operation enable bit (PCN:CNTE) is set to "0" to disable the PPG during the PPG operation, the PPG down counter value will be maintained and the PPG output will be changed to "L" and will be stopped.
Thereafter, to restart the PPG, set the timer operation enable bit (PCN:CNTE) to "1" to enable the PPG and reload the cycle and the duty values by entering an activation trigger.

3. If the timer operation enable bit (PCN:CNTE) is set to "0" to disable the PPG, it takes 3 clocks in internal clock until the PPG output is stopped.

4. When the PPG output waveform selection (OWFS) is rewritten during the PPG communication operation, the setting will be reflected at the next cycle.

5. To change the PPG output mask (PCN.PGMS) from "1" to "0" to cancel mask during the PPG communication operation, perform the setting within the period between the beginning of cycle and the duty match.

■ Cycle Value (PCSR) and Duty (PDUT) Settings

1. When writing a cycle value (PCSR) and a duty value (PDUT), be sure to observe the sequence of (1) PCSR and (2) PDUT. Notes the following when rewriting the cycle value (PCSR) and duty value (PDUT):

(1) The cycle value (PCSR) and the duty value (PDUT) are fetched to the buffer when the duty value (PDUT) is written and will be transferred from the buffer to the counter when an activation trigger is generated or when a borrow occurs.

(2) If the cycle value (PCSR) or duty value (PDUT) is rewritten during the PPG operation, the new value will be effective on the output waveform at the next cycle after the duty value (PDUT) is rewritten.

(3) If only the cycle value (PCSR) needs to be rewritten, after the cycle value is reset, the duty value (PDUT), which is unchanged, must be reset in the order of (1) PCSR and (2) PDUT.

(4) The duty value (PDUT) may be freely rewritten.

2. When you set the PPG duty setting register (PDUT), use values smaller than that set to the PPG cycle setting register (PCSR).

3. When accessing the cycle setting register (PCSR) and duty setting register (PDUT) of the PPG, be sure to use word (16-bit) format. If these registers are accessed in byte format, the values are not written at an upper and lower bit positions.

■ GATE Function

1. Set the GATE function control register (GATEC) before PPG activation.
Do not change the GATE selection bit (STGR) and the polarity selection bit (EDGE) of the GATE function control register (GATEC) during the PPG operation.

2. It takes 4 clocks in internal clock until the PPG output is stopped after GATE signal is negated.

3. When the GATE function is valid (STGR = "1"), the restart enable (RTRG) is valid. The restart operation does not start even when other trigger is input.

4. If the GATE signal is changed from "1" to "0" (when EDG = "0") during the PPG operation, the PPG down counter value will be maintained and the PPG output will be changed to "L" and will be stopped. Thereafter, to start the PPG, set the GATE signal from "0" to "1" to reload the cycle and the duty values.

■ Interrupts

1. If the interrupt request flag is set to "0" when interrupt request flag (PCN:IRQF) is "1", the flag clear request is overwritten, and the interrupt request flag becomes "1".

2. The performance while the PPG output mask is set (PCN.PGMS="1") is shown below. The interrupt flag will not be set to "1" regardless of interrupt factor caused by duty match. The interrupt flag will be set to "1" because of interrupt factor caused by counter borrow occurrence. The interrupt flag will be set to "1" because of interrupt factor caused by triggers (software trigger, external trigger or trigger caused by GATE signal).

■ Start Delay Function

1. To activate the Start Delay mode, the timer operation enable bit (CNTE) and the Start Delay mode enable (STRD) must be set to "1" before or when PPG operation is enabled (activated).

2. When the Start Delay value (PSDR) is rewritten during the PPG operation in the Start Delay mode, the Start Delay value becomes effective after prohibiting operating once and generating the activation trigger. (The Start Delay value becomes effective with the activation trigger.)

3. The Start Delay setting period (PSDR) is waited again when the restart request is generated during the Start Delay operation (waiting time period). Moreover, the PPG waveform output is stopped and the Start Delay setting period is waited again when the restart is requested while outputting PPG waveform by the Start Delay mode enable (STRD)="1".

4. When the Start Delay mode enable (STRD) is set to "1", the Start Delay value cannot be set to "0". (A minimum setting of the Start Delay value is "1") Set the Start Delay mode enable (STRD) to "0" if the delay value is set to "0".

5. Be sure to prohibit operating once when "0" is written in the Start Delay mode enable (STRD) during the Start Delay period. Moreover, if the activation trigger is not generated, the Start Delay mode disable (STRD=0) does not become effective.

6. If the timer operation enable bit (CNTE) is set to "0" to disable the PPG during the Start Delay mode period, the PPG stops with its state (count and output level) maintained. (Refer to 5th particular of "6. Notes" for the return method.)

■ Timing Point Capture Function

1. The Timing Point Capture value setting (PTPC) has to set smaller than the cycle value (PCSR). When the value that is larger than cycle value (PCSR) is set, the A/D activation trigger or the Timing Point Capture match interrupt is not generated.

2. When the Start Delay mode enable (STRD) = "1" and the Timing Point Capture mode enable (TPC) = is set "1", neither the interrupt nor the A/D activation trigger by the Timing Point Capture value match during the Start Delay (waiting) are generated.

3. The value becomes effective at the next cycle after rewriting when the Timing Point Capture value (PTPC) is rewritten during the PPG operation.

4. When "0" is written in the Timing Point Capture mode enable (TPC) during the PPG operation, neither the interrupt by the Timing Point Capture value match nor the A/D activation trigger is generated. Make the setting according to the procedure when Timing Point Capture mode enable (TPC) is set again.

■ PPG Communication Mode Function

1. PPG communication is started by setting PPG communication enable (CMD), the cycle setting (PHCSR/PLCSR), the duty setting (PHDUT/PLDUT), the PPG communication mode data (PCMDDT), and the PPG communication data bit length (PCMDWD), and then setting the activation trigger at the end. Thus, PPC communication is started. Be sure to write the setting in the register when the PPG communication is activated. Also, set the other registers before the start triggers are set.
However, PPG communication will not be started if the activation triggers are generated before the settings above are not completed. Perform setting again after disabling the PPG communication (PCN2.CMD="0").
(similar when GATE function is used)

2. In the PPG communication mode, the following registers are valid or invalid.

Valid registers:

Software trigger (STRG)
Count clock selection (CKS1, CKS0)
PPG output mask selection (PGMS)
Trigger input selection (EGS1, EGS0)
Interrupt request enable (IREN)
Interrupt factor selection (IRS1, IRS0)
Interrupt request flag (IRQF)*1
PPG output polarity selection (OSEL)
PPG timer (PTMR)
GATE function control (GATEC)
PPG communication mode data reading selection (CMDSEL)
PPG communication mode enable (CMD)
Timing Point Capture interrupt (IRS2)
PPG communication data register Empty flag (REMP)*2
PPG communication data shift register Empty flag (SREMP)*2
High format cycle setting (PHCSR)
Low format cycle setting (PLCSR)
High format duty setting (PHDUT)
Low format duty setting (PLDUT)
Communication mode data setting (PCMDDT)
Communication mode data bit length setting (PCMDWD)
Low format pulse polarity selection (LFPR)
High format pulse polarity selection (HFPR)

Invalid registers:

Timer operation enable (CNTE)
Mode selection (MDSE)
Restart enable (RTRG)
PPG output waveform selection (OWFS)
Interrupt request flag (IRQF)
PPG cycle setting (PCSR)
PPG duty setting (PDUT)
Timing Point Capture enable (TPC)
Start Delay enable (STRD)
Start Delay value setting (PSDR)
Timing Point Capture value setting (PTPC)

(*1) IRS[2:0]=000b to 100b of the interrupt selection cannot be set during the PPG communication; however, the registers are enabled.

(*2) Cannot be set because this is a read only register.

3. During the PPG communication operation, do not change any of the following: the count clock selection (CKS1, CKS0), the interrupt selection (IRS2 to IRS0), the trigger input edge selection (EGS1, EGS0), the PPG output polarity selection (OSEL), the GATE function enable (STGR), the activation effective edge selection (EDGE), the PPG communication mode data reading selection (CMDSEL), the High/Low format pulse selection (HFPR/LFPR) and the communication mode data bit length setting (PCMDWD). If any of the above bits is changed during the PPG communication operation, disable PPG communication operation before reconfiguring the register.

4. Notes the following when rewriting the cycle value (PHCSR/PLCSR) and the duty value (PHDUT/PLDUT) in the PPG communication mode.

(1) The cycle value (PHCSR/PLCSR) and the duty value (PHDUT/PLDUT) will be transferred from the register to the counter when an activation trigger is generated or when a borrow occurs. Therefore, if the cycle (PHCSR/PLCSR) or duty value (PHDUT/PLDUT) is rewritten during the PPG communication operation, the new value will be effective on the output waveform at the next cycle after the borrow occurs

(2) The duty value (PHDUT/PLDUT) must be equal to or smaller than the cycle value (PHCSR/PLCSR).

If the duty value is set larger than the cycle value (PHCSR/PLCSR), disable PPG communication operation before changing the duty value (PHDUT/PLDUT) to a smaller value. If PPG communication operation is not disabled, the following will occur:

- The output level will be "H" or "L" depending on [the cycle value (PHCSR/PLCSR) < the duty value (PHDUT/PLDUT)] setting. ("H" or "L" is selected via the PPG output polarity selection setting.)

(3) If the cycle value (PHCSR/PLCSR) and the duty value (PHDUT/PLDUT) are set to the same value, or if the duty value (PHDUT/PLDUT) set to "0", the following will occur:

- When setting as cycle = duty:
  If the polarity is normal (OSEL=0), "H" output.
  If the polarity is inverted (OSEL=1), "L" output.
- When setting as duty = 0:
  If the polarity is normal (OSEL=0), "H" is output for 1 count clock cycle.
  If the polarity is inverted (OSEL=1), "L" is output for 1 count clock cycle.

5. When accessing the High/Low format cycle setting register (PHCSR/PLCSR) and the duty setting register (PHDUT/PLDUT) of the PPG communication mode, be sure to use 16-bit format. If these registers are accessed in byte format, the values are not written at an upper and lower bit positions.

6. When the PPG communication is continuously executed, rewriting PPG communication mode data (PCMDDT) is needed during the PPG communication operation. (Even if the same value is written, the PPG communication is executed.)

However, after the bit length according to PCMDWD (PPG communication mode data bit length) setting is output, the next setting data is output.

7. When the PPG output mask selection (PGMS) is rewritten during the PPG communication operation, the setting will be reflected at the next cycle.

Chapter 18: Watchdog Timer

This chapter explains the watchdog timer.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Usage Example

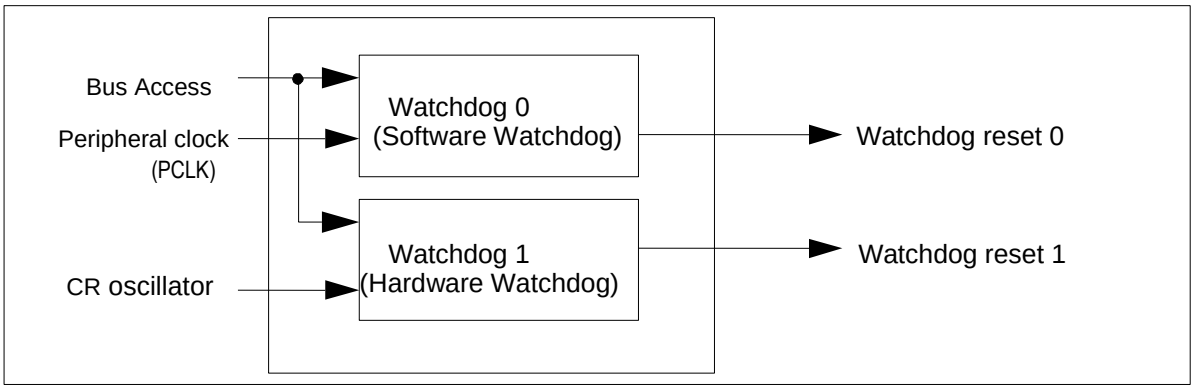
Code : FR81S10_WDT-1v1-91520-2-E

1. Overview

This section gives an overview of the watchdog timer.

This device has two watchdog timers that can detect both the states of software and hardware running out of control, and these watchdog timers can generate reset requests.

Figure 1-1 Block Diagram (schematic)



2. Features

This section explains features of the watchdog timer.

2.1. Watchdog Timer 0 (Software Watchdog)

2.2. Watchdog Timer 1 (Hardware Watchdog)

2.1. Watchdog Timer 0 (Software Watchdog)

This section explains features of the watchdog timer 0.

- Stop mode detection function
  Able to detect the transition to watch mode or stop mode and generate a reset request
- Watchdog timer clear
  The timer is cleared by operation initialization reset or by writing the inverse value of the value previously written to the clear register
- Illegal write detection function
  If the incorrect value is written to the clear register, a reset request is generated.
- Watchdog timer period

The period can be selected from among sixteen choices of the peripheral clock (PCLK) × (2^9 to 2^{24}) cycles

- Count stop conditions
  The count stops while the CPU is stopped
- To set the lower limit value of the timer count of the watchdog timer.
  The value can be selected from among sixteen choices of the peripheral clock (PCLK) × (2^8 to 2^{23}) cycles.
- Monitoring the watchdog timer window and generating a reset request.
  If the clear register is written below the lower limit value of the timer count of the watchdog timer, the watchdog timer generates a reset request.

2.2. Watchdog Timer 1 (Hardware Watchdog)

This section explains features of the watchdog timer 1.

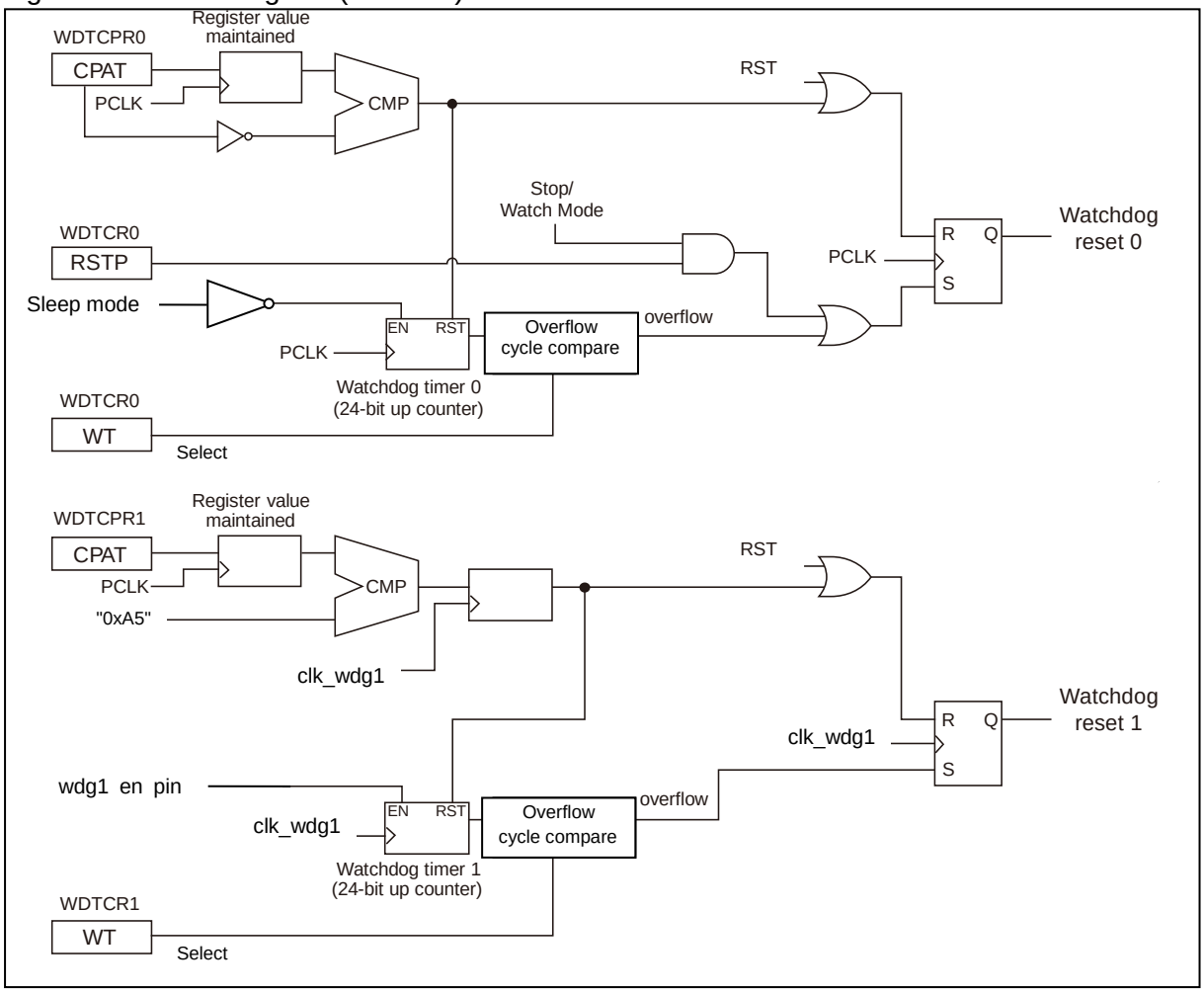
This timer is driven by the clock generated by the built-in CR oscillator circuit immediately after the reset is released. For information on settings (calibration) of the oscillator, see "CHAPTER: RTC/WDT1 (CALIBRATION)".

- 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

3. Configuration

This section shows the configuration of the watchdog timer.

Figure 3-1 Block Diagram (Detailed)



4. Registers

This section explains the registers of the watchdog timer.

Table 4-1 Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0038	WDTECR0	Reserved			Watchdog timer 0 extended configuration register
0x003C	WDTCR0	WDTCPR0	WDTCR1	WDTCPR1	Watchdog timer 0 control register Watchdog timer 0 clear register Watchdog timer 1 cycle information register Watchdog timer 1 clear register

4.1. Watchdog Timer 0 Control Register : WDTCR0 (WatchDog Timer 0 Configuration Register)

The bit configuration of the watchdog control register 0 is shown.

This register configures each of the settings of the watchdog timer 0.

Writing to this register is invalid after the watchdog timer 0 is activated.

■ WDTCR0 : Address 003C$_H$ (Access: Byte, Half-word, Word)

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

[bit7] Reserved: (Reserved bit)
Be sure to write "0" to this bit. The read value is "0".

[bit6] RSTP (Reset by SToP) : Stop mode detection reset enable
This bit configures whether a reset signal is generated or not when a transition to watch mode or stop mode is detected while the watchdog timer 0 is operating. When this bit is enabled, the watchdog timer reset 0 occurs when the CPU switches to watch mode or stop mode. When this bit is not enabled, the watchdog timer 0 is paused when the CPU switches to watch mode or stop mode, and the count stops until the CPU recovers from watch mode or stop mode.

RSTP	Stop mode detection
0	Not detected (initial value)
1	Generates a reset signal when detected

Writing to this bit after the watchdog timer 0 is activated is invalid.

[bit5, bit4] Reserved: (Reserved bits)

Be sure to write "0" to these bits. The read value is "0".

[bit3 to bit0] WT[3:0] (Watchdog Timer interval) : Watchdog timer cycle selection

These bits configure the number of cycles of timer interval starting from when the watchdog timer 0 was last cleared to when a watchdog reset 0 is issued. Details are shown as follows.

WT[3:0]	The 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

After the watchdog timer 0 is activated writing to this bit is invalid.

The watchdog timer 0 does not count while the CPU is not operating.

Counting is performed while the CPU is operating even if DMA transfer is being performed.

4.2. Watchdog Timer 0 Clear Register : WDTCPR0 (WatchDog Timer Clear Pattern Register 0)

The bit configuration of the watchdog timer 0 clear register is shown.

This register activates or clears (delays issue of a reset signal) the watchdog timer 0.

■ WDTCPR0 : Address 003D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CPAT[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit7 to bit0] CPAT[7:0] (Clear PATtern) : Watchdog Timer 0 clear

The watchdog timer 0 is activated by the first write to this register after the reset is released. The watchdog timer is cleared after being activated by writing a value with all of the bits inverted from the previously written value. If a value other than the inverse value of the previously written value is written, the watchdog reset 0 is issued at that time.

The value read out from this register is always "0x00" regardless of the value written.

4.3. Watchdog Timer 0 Extended Configuration Register : WDTECR0 (Watchdog Timer Extended Configuration Register 0)

The bit configuration of the watchdog timer 0 Extended Configuration Register is shown.

This register configures the settings for window watching function of the watchdog timer 0.

Writing to this register is invalid after the watchdog timer 0 is activated.

■ WDTECR0 : Address 0038$_H$ (Access: Byte, Half-word, Word)

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

[bit7 to bit5] Reserved: (Reserved bits)

Be sure to write "0" to this bit. The read value is "0".

[bit4] WTWE (Watchdog Timer Window Enable) : Watchdog Timer Window Function Enable

This bit controls the window function of the watchdog timer 0. When the bit WTWE is set to "1" the window function becomes enabled.
The initial value of this bit is "0". (The window function is invalid.)

WTWE	Window function enabled
0	Window function is invalid (initial value)
1	Window function is valid

[bit3 to bit0] WTLI[3:0] (Watchdog Timer Lower Interval) : Selection of the lower limit of watchdog timer

These bits configure the lower limit of the interval starting from when the watchdog timer 0 is cleared to when it is cleared next time. When the window function is valid, if a request for clearing the watchdog timer 0 comes before a lower limit of timer shown below, a watchdog reset signal is issued.

WTLI[3:0]	The Lower Limit of the Watchdog Timer
0000	PCLK (Peripheral Clock) × 2^8 cycles
0001	PCLK × 2^9 cycles
0010	PCLK × 2^{10} cycles
0011	PCLK × 2^{11} cycles
0100	PCLK × 2^{12} cycles
0101	PCLK × 2^{13} cycles
0110	PCLK × 2^{14} cycles
0111	PCLK × 2^{15} cycles
1000	PCLK × 2^{16} cycles
1001	PCLK × 2^{17} cycles
1010	PCLK × 2^{18} cycles
1011	PCLK × 2^{19} cycles
1100	PCLK × 2^{20} cycles
1101	PCLK × 2^{21} cycles
1110	PCLK × 2^{22} cycles
1111	PCLK × 2^{23} cycles

Set the watchdog timer below the period specified with WTCR0.WT[3:0]. If a period larger than that specified with WTCR0.WT[3:0] is set, a reset signal is generated. This is because the watchdog timer is satisfied with the condition to be cleared below the lower limit of the window even though the timer is cleared before overflow.

4.4. Watchdog Timer 1 Cycle information Register : WDTCR1 (WatchDog Timer Cycle information Register 1)

The bit configuration of the watchdog timer 1 cycle information register is shown.

This register configures each of the settings of watchdog timer 1.

■ WDTCR1 : Address 003E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				WT[3:0]			
Initial value	0	0	0	0	0	1	1	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R1,WX	R0,WX

This register cannot be written again.

[bit7 to bit4] Reserved: (Reserved bits)
The value "0" is always read. Writing to these bits has no influence on operation.

[bit3 to bit0] WT[3:0] (Watchdog Timer interval) : Watchdog timer cycle selection
These bits configure the number of cycles of timer interval starting from when the watchdog timer 1 was last cleared to when a watchdog reset 1 is issued. The cycle is fixed to 2^{15} cycles. Writing to these bits are invalid

WT[3:0]	Watchdog timer 1 cycle
0110	CR oscillator × 2^{15} cycles (initial value, fixed)

4.5. Watchdog Timer 1 Clear Register : WDTCPR1 (WatchDog Timer Clear Pattern Register 1)

The bit configuration of the watchdog timer 1 clear register is shown.

This register clears watchdog timer 1 (delays issue of a reset signal).

■ WDTCPR1 : Address 003F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CPAT[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit7 to bit0] CPAT[7:0] (Clear PATtern) : Watchdog timer 1 clear
Watchdog timer 1 activates after the reset is released. The watchdog timer is cleared after being activated by writing "0xA5". When a value other than "0xA5" is written, the watchdog reset 1 is issued at that time. The value read out from

this register is always "0x00"regardless of the value written.

5. Operation

This section explains operation of the watchdog timer.

5.1. Software Watchdog Function

5.2. Hardware Watchdog Function

5.1. Software Watchdog Function

This section explains the software watchdog function

5.1.1. Settings

5.1.2. Activation

5.1.3. Operation

5.1.1. Settings

This section explains settings of the software watchdog function.

Before activating the watchdog timer 0, set bits 3 to 0: WT[3:0] of the register WDTCR0 in order to select the period starting from clearing the watchdog timer to issuing the reset request.

Since the watchdog timer 0 counts only when the CPU is operating, set the period on the basis of the number of program steps and the clock division setting.

Before activating the watchdog timer 0, set bit6: RSTP of the register WDTCR0 in order to select whether or not to generate a reset signal when a transition to watch mode or stop mode is detected.

- When RSTP="0", the timer stops in watch mode or stop mode.
- When RSTP="1", a reset signal is generated as soon as the CPU enters watch mode or stop mode.

If the device is used in watch mode or stop mode, set RSTP="0". Writing to the RSTP bit is invalid after the watchdog timer 0 is activated.

5.1.2. Activation

This section explains activation of the software watchdog function.

The watchdog timer 0 is activated by the first write of any data to the register WDTCPR0 after reset.

There is no restriction on the data written.

The value "0x00" is always read out from the register WDTCPR0 regardless of any data written.

5.1.3. Operation

This section explains operation of the software watchdog function.

The operation of the watchdog timer 0 after activation is explained.

Counting Conditions

The watchdog timer 0 counts the rising edges of the peripheral clock (PCLK) while the CPU is operating.

DMA transfer does not influence the watchdog timer 0 to count.

As in sleep mode, the watchdog timer 0 stops counting only while the CPU is being stopped. Since sampling of operating state of the CPU is done by the peripheral clock, a change in the operating state of the CPU occurring within the period of the peripheral clock is ignored.

When the watchdog timer 0 is connected with ICE, the timer stops counting under the following conditions:

- In emulator mode
- In the debug interface functions, if the watchdog reset suppression function is enabled.

Under any conditions mentioned above, when the watchdog timer 0 stops counting it pauses without clearing the counter. Hence, when the watchdog timer 0 resumes counting the timer will continue counting from the previous count.

Because the peripheral clock stops during the oscillation stabilization wait time of the source clock, the watchdog timer 0 also stops counting.

Clearing the Timer

Once the watchdog timer 0 is activated, the timer must be cleared before the timer period has elapses.

Clearing the watchdog timer 0 is performed by writing data to the register WDTCPR0. These data written must be the inverted values of all bits of the WDTCPR0 that was written previously.

When the watchdog timer 0 is activated with the set value "0x55", for example, written to the register WDTCPR0, the timer is cleared in the following way:

- After activation of the watchdog timer 0, the set value should be written alternately like "0xAA" then "0x55" then "0xAA" then "0x55".

Since the read value of the register WDTCPR0 is always "0x00", the previously written value cannot be determined by reading WDTCPR0. For this reason, if the previously written value cannot be stored in other location, write to the register two times consecutively in a single clear.

When the window function is effective during the watching period, clear the timer within a period of time while the counter can be cleared effectively.

Reset Request Generation

The watchdog timer 0 generates a watchdog reset request under the following conditions:

- An overflow of the configured watchdog timer cycle occurs.
- There is a transition to watch mode or to stop mode while stop mode detection reset is enabled.
- A value, other than the inverted value of the value which is previously written, is written to the clear register.
- Writing to the clear register within the lower limit of the watching period of the window function.

5.2. Hardware Watchdog Function

This section explains operation of the hardware watchdog function.

5.2.1. Settings

5.2.2. Activation

5.2.3. Operation

5.2.1. Settings

This section explains settings of the hardware watchdog function.

The values set to those bits from bit3 to bit0:WT[3:0] of the register WDTCR1 of the watchdog timer 1 are fixed with hardware.

5.2.2. Activation

This section explains activation of the hardware watchdog function.

The watchdog timer 1 is activated immediately after the reset is released.

5.2.3. Operation

This section explains operation of the hardware watchdog function.

The operation of the watchdog timer 1 after activation is explained.

Counting conditions

The watchdog timer 1 counts the rising edges of the CR oscillation.

When the watchdog timer 1 is connected with ICE, the timer stops counting under the following conditions:

- In emulator mode
- In the debug interface functions, if the watchdog reset suppression function is enabled.

The watchdog timer 1 stops counting in sleep mode, watch mode, stop mode, and during the oscillation stabilization wait time recovering from standby mode.

Clearing the timer

Once the watchdog timer 1 is activated, the timer must be cleared before the timer period has elapses.

The watchdog timer 1 is cleared when the value "0xA5" is written to the register WDTCPR1.

Reset Request Generation

The watchdog timer 1 generates a watchdog reset request under the following conditions:

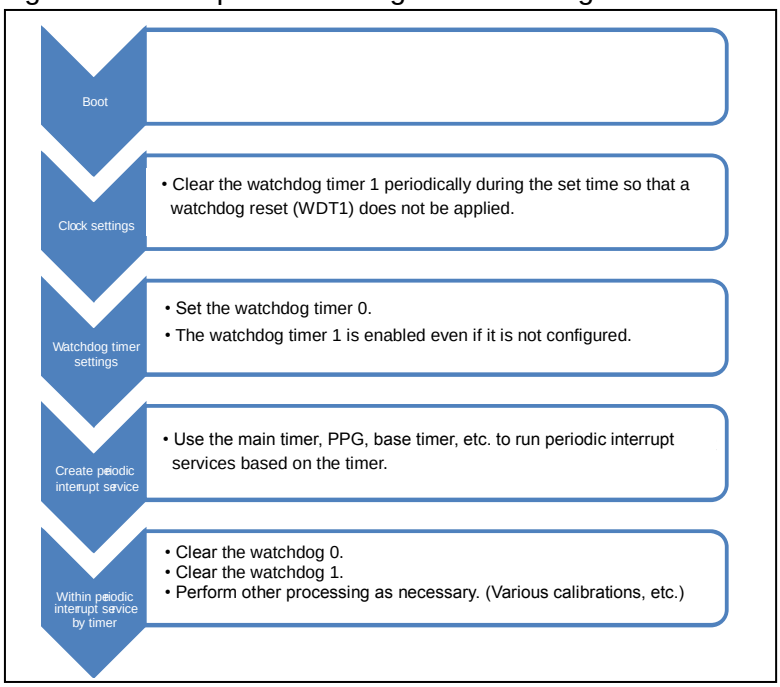
- An overflow of the watchdog timer cycle occurs.
- A value other than "0xA5" is written to the register WDTCPR1.

6. Usage Example

This section gives an example of how the watchdog timer is used.

This example shows how to clear the watchdog timer.

Figure 6-1 Example of Clearing the Watchdog Timers



Chapter 19: Base Timer



This chapter explains the base timer.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FM10-3v1-91520-3-E

1. Overview

This section explains the overview of the base timer.

This series includes the base timer for max 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

The base timer function is not provided in MB91F52xB(64pin).

Only ch.1 is available in MB91F52xD (80pin) and MB91F52xF (100pin).

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:

2.1. 16/32-bit Reload Timer

2.2. 16-bit PWM Timer

2.3. 16/32-bit PWC Timer

2.4. 16-bit PPG Timer

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 eight internal (peripheral) clocks and three external clocks (ECK signals).

  · Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.
  · External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

- **Activation trigger**
  One of the following can be selected:

  · Software trigger
  · External event: Rising edge, falling edge, or both edges
  · 16/32-bit reload timer reactivation: The 16/32-bit reload timer can be reactivated when an activation trigger is detected during counting.

- **Interrupt request**
  An interrupt request can be generated in one of the following events:

  · IRQ0: When an underflow occurs
  · IRQ1: When a 16/32-bit reload timer activation trigger is detected

2.2. 16-bit PWM Timer

This section explains the 16-bit PWM timer of the base timer.

The 16-bit PWM timer, PWM standing for Pulse Width Modulator, produces a desired waveform at an external pin when a duty ratio of the pulse width is specified.

- **I/O mode**
  You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.

- **Operation mode**
  You can select one of the following two:

  · Reload mode: In this mode, when the 16-bit down counter underflows, the preset cycle is reloaded to allow the timer to restart counting.
  · One-shot mode: Once the 16-bit down counter underflows, the counter will no longer count.

- **Count clock**
  You can select one of eight internal (peripheral) clocks and three external clocks (ECK signals).

  · Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.

- External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

Activation trigger

One of the following can be selected:

- Software trigger
- Three external events: (Rising edge, falling edge, or both edges detection)

16-bit PWM timer reactivation

The 16-bit PWM timer can be reactivated when an activation trigger is detected during counting.

Output waveform

The output signal from the external pin can be fixed at the "L" or "H" level.

Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0 : When an underflow occurs or counting is performed up to a preset value (duty)
- IRQ1 : When a 16-bit PWM timer activation trigger is detected

2.3. 16/32-bit PWC Timer

This section explains the 16/32-bit PWC timer of the base timer.

The 16/32-bit PWC timer, PWC standing for Pulse Width Counter, is used to measure pulse widths or cycles.

I/O mode

You can select a signal (waveform) I/O operation using the base timer I/O selection function.

Timer mode

You can run multiple timers for individual channels and can combine 16-bit PWC timers for 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 eight types.

- Clocks obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, and 2048.

Measurement mode

You can select one of the following five options relating to the pulse width and cycle to be measured:

- "H" pulse width: Duration in which the input signal is maintained at the "H" level
- "L" pulse width: Duration in which the input signal is maintained at the "L" level
- Rising edge interval: Period from the detection of a rising edge to the detection of the next rising edge
- Falling edge interval: Period from the detection of a falling edge to the detection of the next falling edge
- Edge-to-edge pulse width: The width between consecutive input edges is one of the following:
  - Period from the detection of a rising edge to the detection of the falling edge
  - Period from the detection of a falling edge to the detection of the rising edge

● 16/32-bit PWC timer reactivation

The 16/32-bit PWC timer can be reactivated when an activation trigger is detected during counting.

● Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0 : When an overflow occurs
- IRQ1 : When measurement ends

2.4. 16-bit PPG Timer

This section explains the 16-bit PPG timer of the base timer.

The 16-bit PPG timer, PPG standing for Programmable Pulse Generator, is a timer that generates a waveform with a desired pulse width.

● I/O mode

You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.

● Operation mode

You can select one of the following two:

- 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 eight internal (peripheral) clocks and three external clocks (ECK signals).

- Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.
- External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

● Activation trigger

One of the following can be selected:

- Software trigger
- Three external events: (Rising edge, falling edge, or both edges detection)

● 16-bit PPG timer reactivation

The 16-bit PPG timer can be reactivated when an activation trigger is detected during counting.

- **Interrupt request**

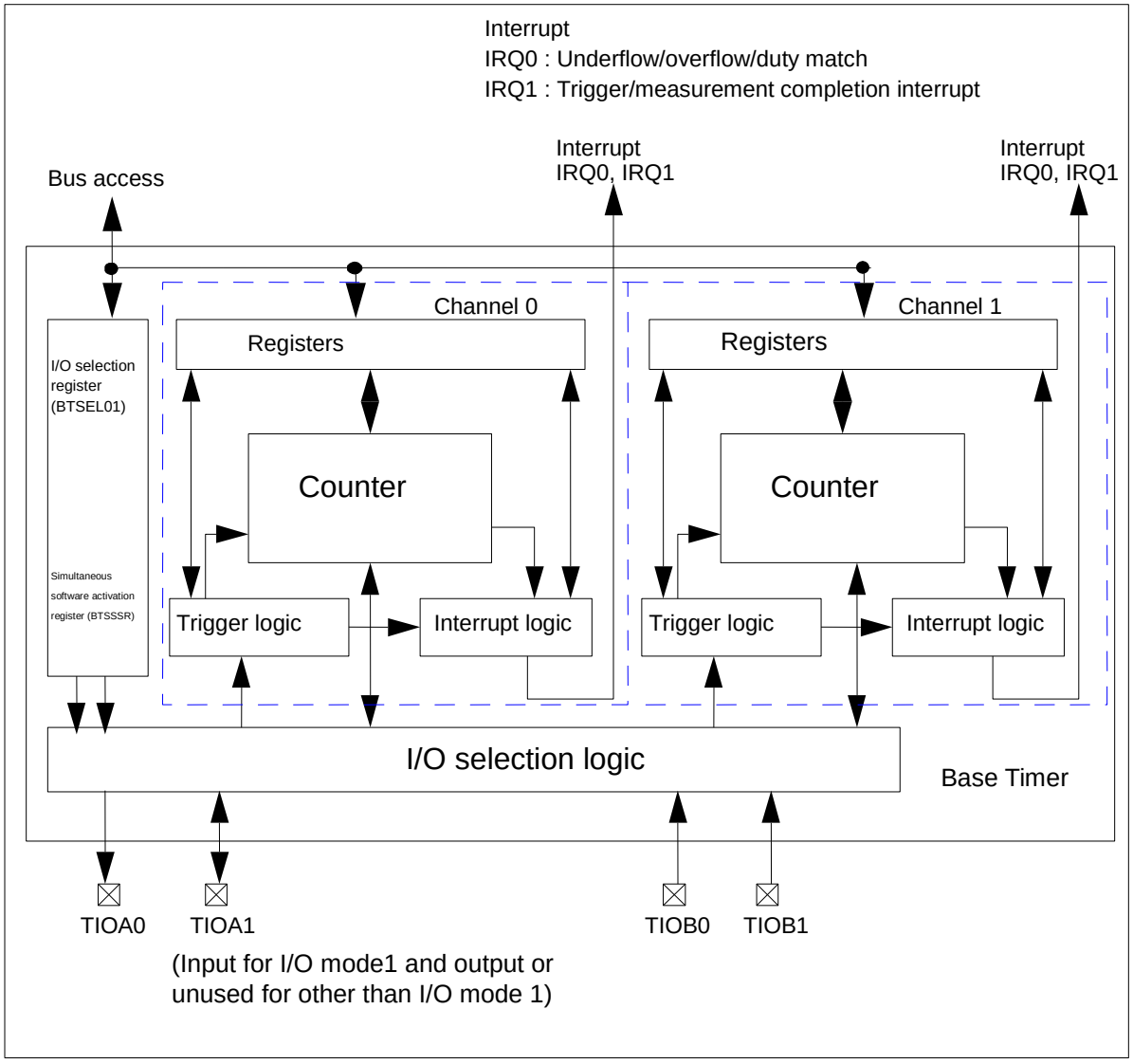
  An interrupt request can be generated in one of the following events:

  - IRQ0 : When an underflow occurs based on the value of the base timer x H width setting reload register (BTxPRLH).
  - IRQ1 : When a 16-bit PPG timer activation trigger is detected.

3. Configuration

This section explains the configuration of the base timer.

Figure 3-1 Block Diagram (Overview)



4. Registers

This section explains registers of the base timer.

■ List of Base Addresses (Base_addr) and External Pins

Table 4-1 Table of Base Addresses (Base_addr) and External Pins

Channel number	Base address	External pin *			
		MB91F52xB	MB91F52xD MB91F52xF	MB91F52xJ	MB91F52xK MB91F52xL
0	0x0080	No	No	TIOA0_0/TIOA0_1, TIOB0_1	TIOA0_0/TIOA0_1, TIOB0_0/TIOB0_1
1	0x0090		TIOA1_1, TIOB1_1	TIOA1_1, TIOB1_1	TIOA1_0/TIOA1_1, TIOB1_0/TIOB1_1

* : TIOA0, TIOA1, TIOB0, and TIOB1 are assigned according to the BTSEL01 register setting, but the setting without external pins is disabled.

■ Registers Map

Table 4-2 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0080	[Common] BT0TMR		[Common] BT0TMCR		[Common] Timer register 0 [Common] Control register 0
0x0084	[Common] BT0TMCR 2	[Reload timer] BT0STC [PWM] BT0STC [PPG] BT0STC [PWC] BT0STC	Reserved		[Common] Control register 20 [Reload timer] Status control register 0 [PWM] Status control register 0 [PPG] Status control register 0 [PWC] Status control register 0
0x0088	[Reload timer] BT0PCSR [PWM] BT0PCSR [PPG] BT0PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT0PDUT [PPG] BT0PRLH [PWC] BT0DTBF		[Reload timer] Cycle setting register 0 [PWM] Cycle setting register 0 [PPG] L width setting reload register 0 [PWM] Duty setting register 0 [PPG] H width setting reload register 0 [PWC] Data buffer register 0
0x008C	Reserved				

Address	Registers				Register function
	+0	+1	+2	+3	
0x0090	[Common] BT1TMR		[Common] BT1TMCR		[Common] Timer register 1 [Common] Control register 1
0x0094	[Common] BT1TMCR 2	[Reload timer] BT1STC [PWM] BT1STC [PPG] BT1STC [PWC] BT1STC	Reserved		[Common] Control register 21 [Reload timer] Status control register 1 [PWM] Status control register 1 [PPG] Status control register 1 [PWC] Status control register 1
0x0098	[Reload timer] BT1PCSR [PWM] BT1PCSR [PPG] BT1PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT1PDUT [PPG] BT1PRLH [PWC] BT1DTBF		[Reload timer] Cycle setting register 1 [PWM] Cycle setting register 1 [PPG] L width setting reload register 1 [PWM] Duty setting register 1 [PPG] H width setting reload register 1 [PWC] Data buffer register 1
0x009C	BTSEL01	Reserved	BTSSSR		I/O selection register Simultaneous software activation register

4.1. Common Registers

This section explains the common registers of the base timer.

The registers described here are common to various operations.

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 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.

■ BTxTMR : Address Base_addr + 00_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

Note:

These registers must be accessed in 16-bit mode.

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 variously configure and stop the base timer and issue software triggers.

■ BTxTMCR : Address Base_addr + 02_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	CKS[2:0]			[PWM - 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,W0 *3	R/W	R/W	R/W	R/W R0,WX *1	R/W R0,WX *1	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[Reload timer - PWC] T32 [Others] Reserved	FMD[2:0]			[Reload timer - PWM - PPG] OSEL [Others] Reserved	MDSE	CTEN	STRG
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W R0,W0 *1 R0,W0 *2	R/W	R/W	R/W	R/W R/W0 *1	R/W	R,W	R0,W0 *1

*1: Attribute assumed for "Reserved"
*2: Attribute assumed for a 32-bit timer serving an odd-number channel
*3: Attribute assumed for a 32-bit timer serving an odd-number channel or for a 16/32-bit PWC timer

■ BTxTMCR2 : Address Base_addr + 04_H (Access: Byte)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							CKS3
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

Notes:

- If you need to change the FMD[2:0] setting, once reset it to FMD[2:0] = 000, and then set FMD[2:0] to the desired value.
- Reserved bits must be set to "0".
- If you want to set bits of these registers except for the software trigger (STRG) bit, proceed as follows:
  1. Once stop operation by writing FMD[2:0] = 000 or CTEN = 0.
  2. Write desired values to the timer function selection bits (FMD[2:0]) and other bits.
- When writing to the software trigger bit (STRG), be careful not to clear other bits.
- Since FMD[2:0] = 000 specifies reset mode, you cannot set other bits when setting FMD[2:0] = 000.
- These registers must be accessed in 16-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15] Reserved
Write 0 to this bit.

[BTxTMCR2:bit8, BTxTMCR:bit14 to bit12] CKS[3:0] (ClocK Select) : Count clock selection bits
This bit selects a count clock.

CKS[3:0]	Description	
	Clock source	Description
0000	Internal clock (Peripheral clock (PCLK))	1 division
0001		4 division
0010		16 division
0011		128 division
0100		256 division
0101	[Reload timer/PWM/PPG] external clock (ECK signal) [PWC] Setting is prohibited	Rising edge
0110		Falling edge
0111		Both edges
1000	Internal clock (Peripheral clock (PCLK))	512 division
1001		1024 division
1010		2048 division
Other	Setting is prohibited.	

In the PWC mode, settings of 0101, 0110, and 0111 are prohibited.

[PWM/PPG] [bit11] RTGEN (Restart by TriGger ENable) : Restart enable bit

If "1" is written to the STRG bit or an external activation trigger (TGIN signal) is detected, this bit sets whether or not to recount the value of cycle setting register (BTxPCSR)/L width setting reload register (BTxPRLL) by reloading it to the 16-bit down counter.

RTGEN	Description of operation
0	Does not reactivate
1	Reactivates

[PWM/PPG] [bit10] PMSK (Pulse MaSK) : Pulse output mask bit

This bit selects a level of waveform to output (TOUT signal) from the followings:

- Normal output : Output the waveform output from the 16-bit PWM/PPG timer without modification.
- Fixed output : Output a sequence of "L" level or "H" level signals regardless of the settings of cycle or duty.

PMSK	Description
0	Normal output
1	Fixed output

If the fixed output is selected by writing "1" to this bit, the level being output will vary depending on the settings of the OSEL bit.

- If OSEL=0 : "L" level will be output.
- If OSEL=1 : "H" level will be output.

[Reload timer/PWM/PPG] [bit9, bit8] EGS[1:0] (EdGe Select) : Trigger input selection bits

These bits select an effective edge for the external activation trigger (TGIN) signal.

EGS[1:0]	Description
00	Trigger input has no effect on the operation
01	Rising edge
10	Falling edge
11	Both edges

[PWC] [bit10 to bit8] EGS[2:0] (EdGe Select) : Measurement mode selection bits

These bits select a measurement mode.

EGS[2:0]	Description
000	"H" pulse width measurement: Duration in which the input signal is maintained at the "H" level
001	Rising edge interval measurement: Time from the detection of a rising edge to the detection of the next rising edge
010	Falling edge interval measurement: Time from the detection of a falling edge to the detection of the next falling edge
011	Edge-to-edge pulse width measurement: The width between consecutive input edges is either:(1) or (2). (1) Time from the detection of a rising edge to the detection of the falling edge (2) Time from the detection of a falling edge to the detection of the rising edge
100	"L" pulse width measurement: Duration in which the input signal is maintained at the "L" level(Time from the detection of a falling edge to the detection of the rising edge)

EGS[2:0]	Description
101 110 111	Setting is prohibited

[Reload timer/PWC] [bit7] T32 (Timer 32bit) : 32-bit timer selection bit

This bit selects whether to run the 16/32-bit timer individually by each channel or use the 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 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

This bit enables/disables the counter operation.

CTEN	Description	
	Read	Write
0	Stopped	This bit becomes "0".
1	Operation enabled	This bit becomes "1".

Note:

When a falling edge is output from the even-number channel during timer operation in I/O mode 4 and I/O mode 6, this bit, which is an odd-number channel bit, is cleared to 0.

[bit0] STRG (Software TRiGger) : Software trigger bit

Functions as a trigger for timer activation, etc.

The read value at PWC is "0". Write "0" in this bit at PWC.

STRG	Description
0	No effect on the operation
1	Issues a trigger.

Notes:

- When writing to this bit, be careful not to clear other bits.
- When writing to CTEN and FMD[2:0] simultaneously, a trigger is issued as soon as operation is enabled.

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 set the I/O mode of ch.0 and ch.1 for the base timer.

■ BTSEL01 : Address 009C$_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SEL01[3:0]			
Initial value	1	1	1	1	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

Notes:

- These registers must be accessed in 8-bit mode.
- Rewrite this register after setting the FMD2 to FMD0 bits of the base timer x the timer control register (BTxTMCR) to the base timer reset mode (FMD2 to FMD0 = 000).

[bit3 to bit0] SEL01[3:0] (SELect) : ch.0/ch.1 I/O selection bits

These bits set the I/O mode of ch.0 and ch.1 for the base timer.

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

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 5-2 Wiring Diagram of Each I/O Mode (2) ".

SSSR0/1	Description
0	No effect on the operation.
1	"1" pulse is applied to the timer input, and then the corresponding channel is activated.

4.2. Registers for 16/32-bit Reload Timer

This section explains registers for 16/32-bit reload timer.

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.

■ **BTxSTC : Address Base_addr + 05_H (Access: Byte)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1),W	R0,W0	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable) : Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when an activation trigger for 16/32-bit reload timer has been detected (TGIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable) : Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the down counter underflows (UDIR = 1).

TGIE/UDIE	Description
0	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

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.

■ BTxPCSR : Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16/32-bit reload timer (FMD2 to FMD0 = 011) using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data) : Data bits

These registers with a buffer set the cycle for the 16/32-bit reload timer. The down counter counts down from the value set to these registers.

The value set to these registers is loaded to the 16-bit down counter in the following cases:

- When the 16/32-bit reload timer is started

- When the down counter underflows

The following values are set to these registers when two channels of a 16-bit reload timer are cascaded and used as the 32-bit reload timer.

- Value of even-number channel cycle setting register (BTxPCSR) : Value of lower 16-bit
- Value of odd-number channel cycle setting register (BTxPCSR) : Value of upper 16-bit

For this reason, in the 32-bit timer mode, write values into these registers in the following order.

1. Odd-number channel base timer x cycle setting register (BTxPCSR)
2. Even-number channel base timer x cycle setting register (BTxPCSR)

4.3. Registers for 16-bit PWM Timer

This section explains registers for 16-bit PWM timer.

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.

■ BTxSTC : Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	DTIE	UDIE	Reserved	TGIR	DTIR	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R(RM1),W	R(RM1),W	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR, DTIR, and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable) : Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when a 16-bit PWM timer activation trigger is detected (TGIR = 1).

[bit5] DTIE (DuTy Interrupt Enable) : Duty match interrupt request enable bit

This bit sets whether or not to generate a duty match interrupt request when the value of the 16-bit down counter matches the value of the base timer x duty setting register (BTxPDUT) (DTIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable) : Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the down counter underflows (UDIR = 1).

TGIE/DTIE/UDIE	Description
0	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.

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.

■ BTxPCSR : Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- Be sure to rewrite the duty setting register (BTxPDUT) when these registers are rewritten.
- Do not set a value smaller than the value set to the duty setting register (BTxPDUT).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data) : Data bits

These registers with a buffer set the cycle for the 16-bit PWM timer. The 16-bit down counter counts down from the value set to these registers. When the counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.

These registers have a buffer and thus can be rewritten during counting. The value set to these registers is loaded to the 16-bit down counter in the following cases:

- When the 16-bit PWM timer is activated
- When the down counter underflows

When the same value is set to these registers and the base timer x duty setting register (BTxPDUT), the level of the output signal (TOUT) can be fixed. The output signal level is as follows according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR):

- OSEL=0: "H" level
- OSEL=1: "L" level

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.

■ 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

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- Do not set the value higher than the value set to the cycle setting register (BTxPCSR) when these registers are rewritten.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data) : Data bits

These registers with a buffer set the duty for the 16-bit PWM timer. When the 16-bit down counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.

These registers have a buffer and thus can be rewritten during counting.

If the 16-bit down counter underflows, the buffer value will be transferred.

When the same value is set to these registers and the base timer x cycle setting register (BTxPCSR), the level of the output signal (TOUT) can be fixed. The output signal level is as follows according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR):

- OSEL=0: All "H" level
- OSEL=1: All "L" level

4.4. Registers for 16-bit PPG Timer

This section explains registers for 16-bit PPG timer.

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.

■ BTxSTC : Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1),W	R0,W0	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable) : Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when a 16-bit PPG timer activation trigger is detected (TGIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable) : Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the base timer x H width setting reload register (BTxPRLH) completed counting down and the counter underflows (UDIR = 1).

TGIE/UDIE	Description
0	Disabled.
1	Enabled.

[bit2] TGIR (TriGger Interrupt Register) : Trigger interrupt request flag bit

This bit indicates that a 16-bit PPG timer activation trigger is detected. When this bit is "1" and the TGIE bit is set to "1", a trigger interrupt request is generated.

[bit0] UDIR (UnDerflow Interrupt Register) : Underflow interrupt request flag bit

This bit indicates that the base timer x H width setting reload register (BTxPRLH) completed counting down and an underflow occurred. An underflow will occur if the register attempts counting down when the 16-bit down counter value is "0000_H". When this bit is "1" and the UDIE bit is set to "1", an underflow interrupt request is generated.

TGIR/UDIR	Read	Write
0	No trigger detection/underflow occurred.	This bit is cleared.
1	Trigger detection/underflow occurred.	No effect on the operation.

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.

■ BTxPRLL : Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data) : Data bits

These registers set the default level for the signal output from the 16-bit PPG timer. When the 16-bit down counter completes counting down the value set to these registers, the level of the output waveform (TOUT) will be inverted. Setting these registers and the base timer x H width setting reload register (BTxPRLH) determines the widths of "L" level and "H" level for the output signal. The signal level width set to these registers depends on the setting of the OSEL bit of the timer control register (BTxTMCR) as follows:

- OSEL=0: "L" level width
- OSEL=1: "H" level width

The value set to these registers is loaded to the 16-bit down counter when a 16-bit PPG timer activation trigger is detected or when the base timer x H width setting reload register (BTxPRLH) completed counting values and underflows.

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.

■ BTxPRLH : Address Base_addr + $0A_H$ (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data) : Data bits

These registers with a buffer set the width of signal level output when the L width setting reload register (BTxPRLL) completes counting values. When the 16-bit down counter completes counting down the value set to these registers, the signal level of the output waveform (TOUT) will be inverted.

Setting these registers and the base timer x L width setting reload register (BTxPRLL) determines the widths of "L" level and "H" level for the output signal. The signal level width set to these registers depends on the setting of the OSEL bit of the base timer x timer control register (BTxTMCR) as follows:

- OSEL = 0: "H" level width
- OSEL = 1: "L" level width

These registers have a buffer and thus can be rewritten during counting. These registers transfer values at the following timing.

- Transfer to the buffer
  - When a 16-bit PPG timer activation trigger is detected
  - When the base timer x H width setting reload register (BTxPRLH) completes counting down values and underflows
- Transfer to the 16-bit down counter
  - When counting down from the value of the base timer x L width setting reload register (BTxPRLL) is completed.

For rewriting timing, see "■Write Timing " in "5.6.3 Operation in Reload Mode".

4.5. 16/32-bit PWC Timer Register

This section explains registers for 16/32-bit PWC timer.

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.

■ **BTxSTC : Address Base_addr + 05_H (Access: Byte)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ERR	EDIE	Reserved	OVIE	Reserved	EDIR	Reserved	OVIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R(RM1), W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to OVIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD= 000).

[bit7] ERR (ERRor) : Error flag bit

This bit indicates that the next measurement is completed before the measurement result is read from the data buffer register (BTxDTBF) in the continuous measurement mode and the measurement result has been overwritten by the new value. The old value is discarded. This bit is cleared to "0" when a value is read from the data buffer register (BTxDTBF).

ERR	Description
0	The measurement result has not been overwritten.
1	The measurement result has been overwritten.

[bit6] EDIE (EnD Interrupt Enable) : Measurement completion interrupt request enable bit

This bit sets whether or not to generate a measurement completion interrupt request when the measurement of the 16/32-bit PWC timer is completed (EDIR = 1).

[bit4] OVIE (OVerflow Interrupt Enable) : Overflow interrupt request enable bit

This bit sets whether or not to generate an overflow interrupt request when the up counter overflows (OVIR = 1).

EDIE/OVIE	Description
0	Disabled
1	Enabled

[bit2] EDIR (EnD Interrupt Register) : Measurement completion interrupt request flag bit

This bit indicates that the measurement of the 16/32-bit PWC timer is completed. When this bit is "1" and the EDIE bit is set to "1", a measurement completion interrupt request is generated. This bit is cleared when the measurement result (BTxDTBF) is read out.

[bit0] OVIR (OVerflow Interrupt Register) : Overflow interrupt request flag bit

This bit indicates that the up counter value has changed from "$FFFF_H$" to "0000_H" and an overflow occurred. When this bit is "1" and the OVIE bit is set to "1", an overflow interrupt request is generated. This bit is cleared when "0" is written.

EDIR/OVIR	Read	Write
0	No measurement completion/overflow occurred.	(EDIR) No effect on the operation. (OVIR) This bit is cleared.
1	Measurement completion/overflow occurred.	No effect on the operation.

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 used as the 32-bit PWC timer.

- Value of even-number channel data buffer register (BTxDTBF): Value of lower 16-bit
- Value of odd-number channel data buffer register (BTxDTBF): Value of upper 16-bit

In the 32-bit timer mode, read values from these registers in the following order.

1. Even-channel data buffer register (BTxDTBF)
2. Odd-channel data buffer register (BTxDTBF)

5. Operation

This section explains the operation of the base timer.

5.1 Selection of Timer Function

5.2 I/O Allocation

5.3 32-bit Mode Operation

5.4 16/32-bit Reload Timer Operation

5.5 16-bit PWM Timer Operation

5.6 16-bit PPG Timer Operation

5.7 16/32-bit PWC Timer Operation

5.1. Selection of Timer Function

This section explains selection of the timer function.

Select the timer function for BTxTMCR.FMD[2:0].

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(*) of the output signal from the even-number channel and stops with the falling edge(*).

- **I/O mode 5**
  Simultaneous software activation mode

  More than one channels are started by the software at the same time in this mode.

- **I/O mode 6**
  Software activation timer activation/stop mode

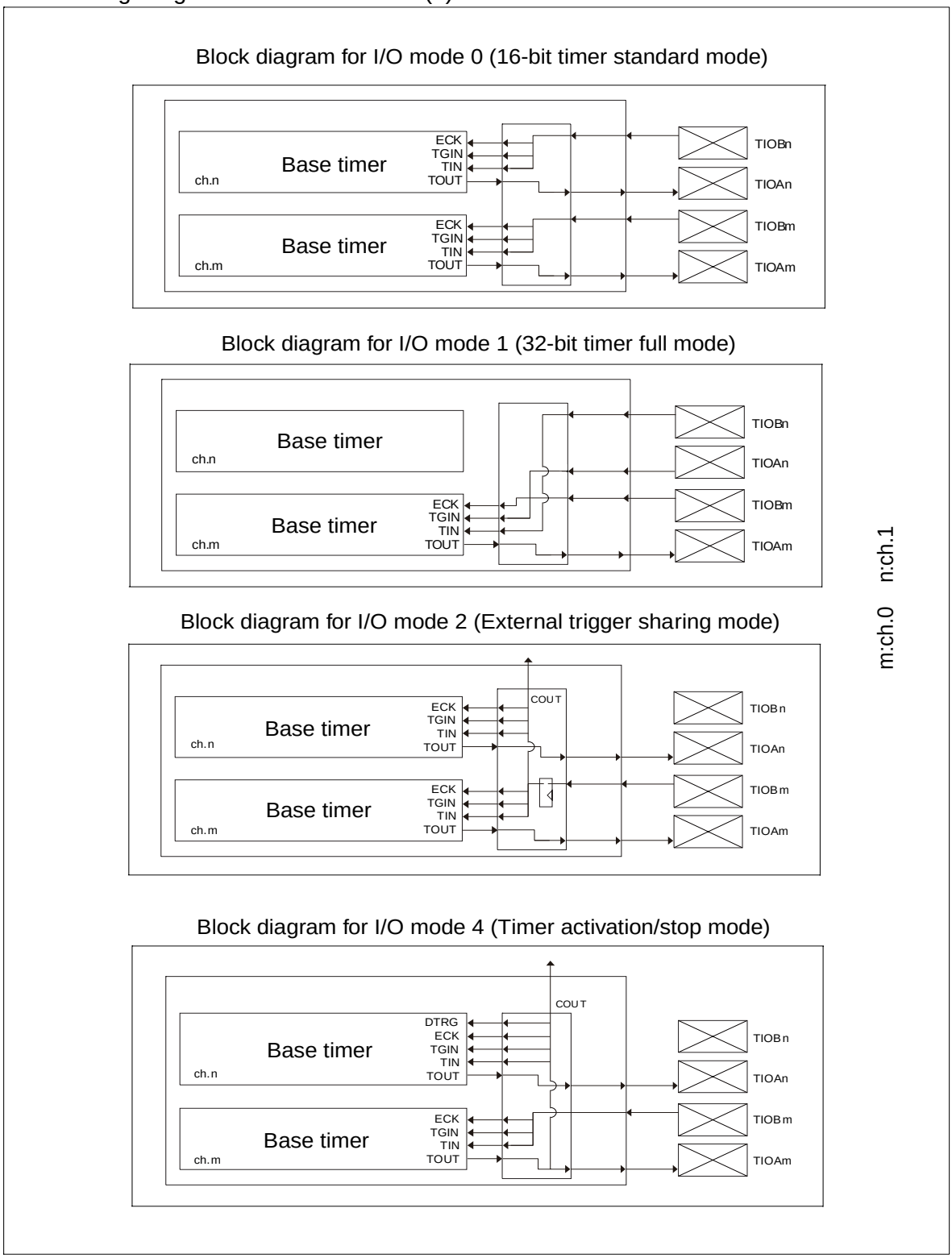
  Activation/stop of the odd-number channel is controlled by the even-number channel in this mode. The even-number channel is started by the software. The odd-number channel is started with the rising edge(*) of the output signal from the even-number channel and stops with the falling edge(*).

- **I/O mode 7**
  Timer activation mode

  Activation of the odd-number channel is controlled by the even-number channel in this mode. The odd-number channel is started with the rising edge(*) of the output signal from the even-number channel.

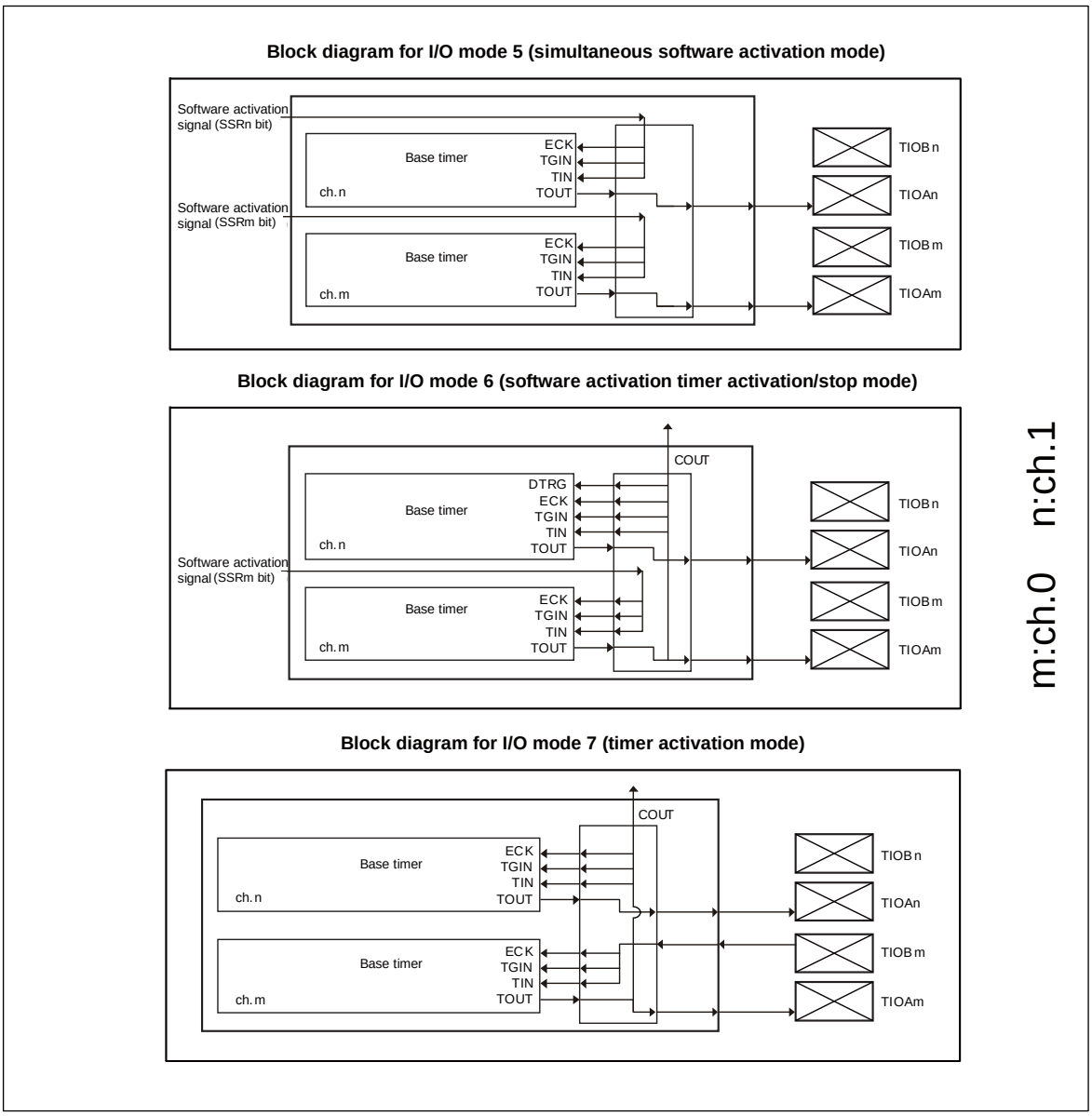
(*): Make a setting using the trigger input selection bit (BTxTMCR.EGS).

Figure 5-1 Wiring Diagram of Each I/O Mode (1)



m: Channel 0 n: Channel 1

Figure 5-2 Wiring Diagram of Each I/O Mode (2)



m: Channel 0 n: Channel 1

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.

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.

5.3.2. 32-bit Mode Setting

This section explains the 32-bit mode setting.

First, set "000" to the FMD bits of the BTxTMCR register of the even-number channel to reset to the reset mode, then select the reload timer or PWC timer and set the operation as in the 16-bit mode. While doing so, set to the 32-bit mode by writing "1" to the T32 bit of the BTxTMCR register. Leave the T32 bit of the odd-number channel "0". You do not have to set the reset mode. For the reload timer, set the upper 16-bit reload values of the 32-bit to the cycle setting register of the odd-number channel, then set the lower 16-bit reload values to the cycle setting register of the even-number channel.

The transition to the 32-bit mode is reflected immediately after the writing to the T32 bit. Thus, setting change for both channels must be done when the counting is stopped.

To transit from the 32-bit mode to the 16-bit mode, set "000" to the FMD bits of the BTxTMCR register of the even-number channel to reset both the even-number and odd-number channels, and make a setting in the 16-bit mode for each channel.

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 5-11 Configuration in 32-bit Timer Mode" and Figure 5-29 Configuration in 32-bit Timer Mode"

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 5-3 Block Diagram (16-bit Reload Timer Operation)

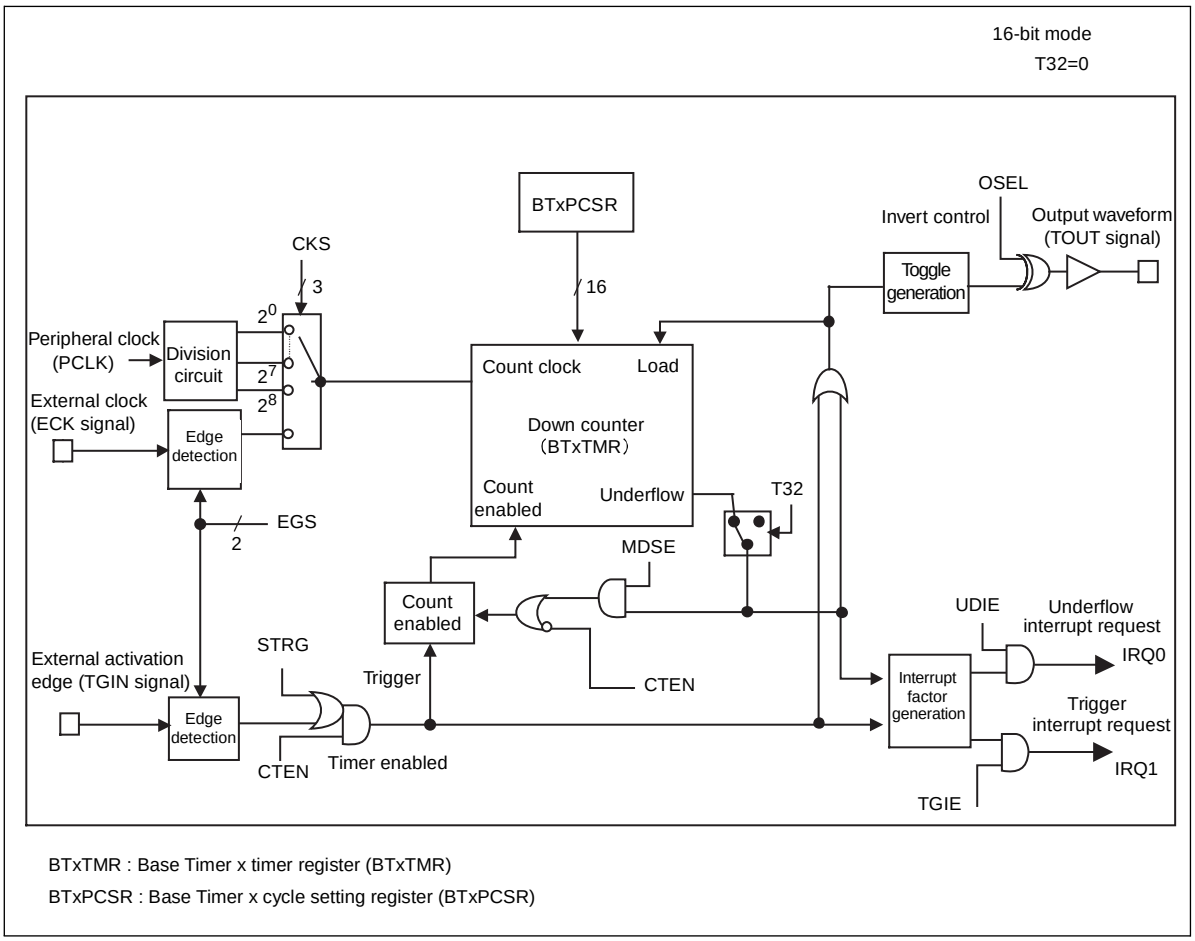
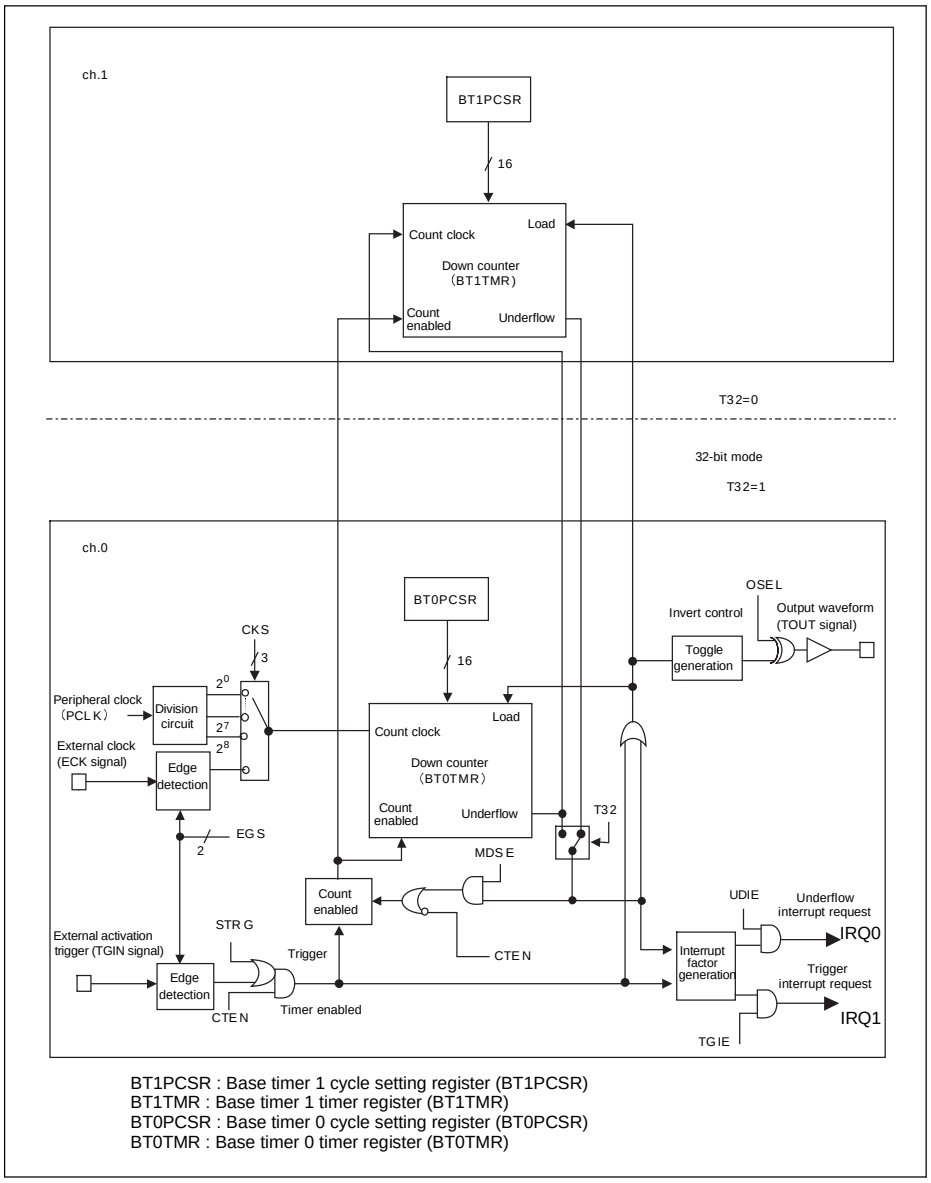


Figure 5-4 Block Diagram (32-bit Reload Timer Operation)



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.

5.4.2. Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

■ Overview

In this mode, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded every time an underflow occurs to ensure that countdown is continued. To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0"(MDSE=0).

■ Operation

● Activation

Activate the 16/32-bit reload timer with the following procedure:

1. Permit 16/32-bit reload timer operation by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1"(CTEN=1).
   The 16/32-bit reload timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01). See "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 5-5 and Figure 5-6 show the count start timing.

Figure 5-5 Count Start Timing (Software Trigger)

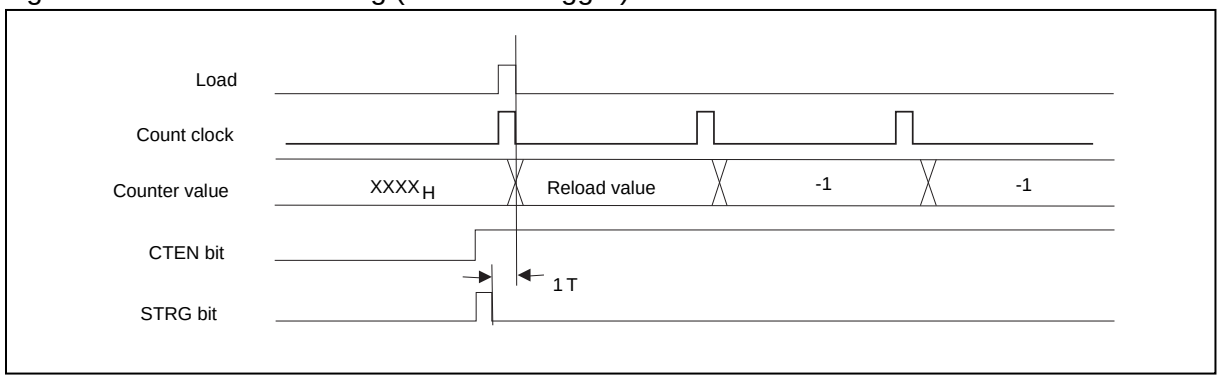
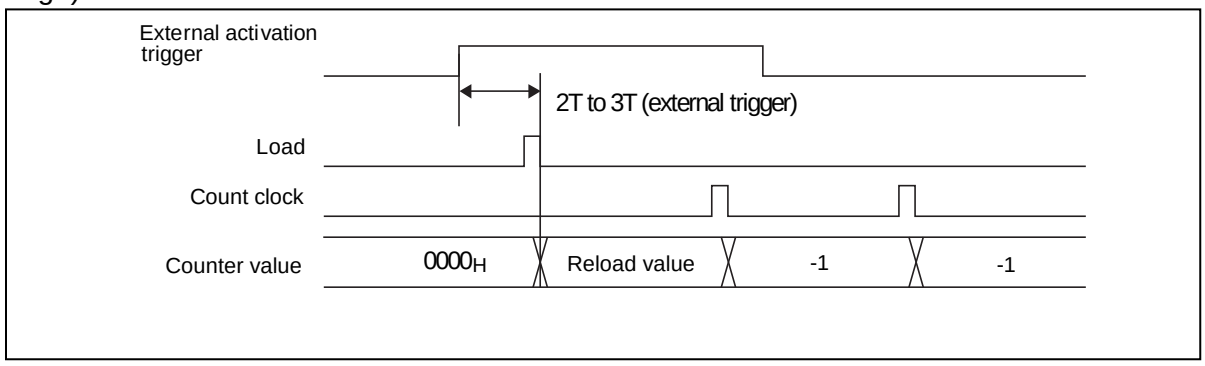


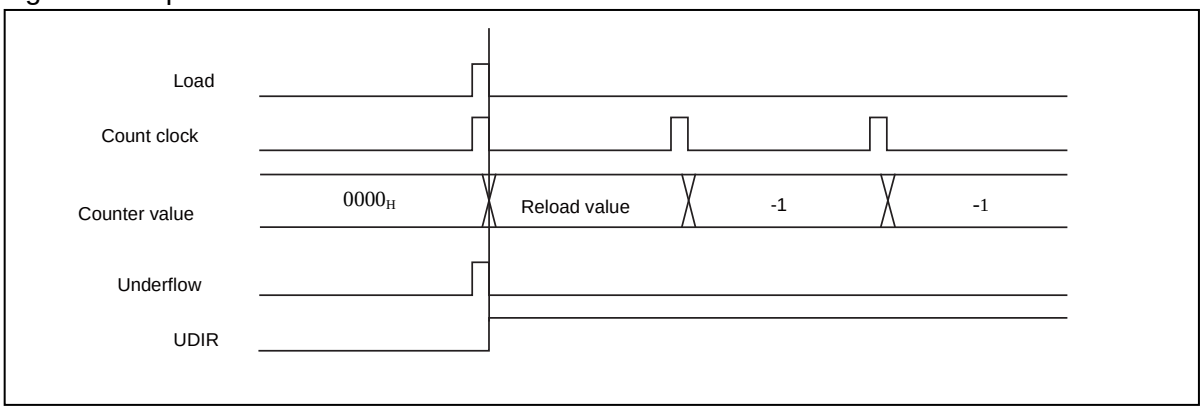
Figure 5-6 Count Start Timing (External Activation Trigger (TGIN Signal), Effective Edge = Rising Edge)



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 "5.2 I/O Allocation".
When the down counter underflows after attempting to count down further from the value of "0000_H", the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is reloaded to the down counter, which continues to count down. If an underflow occurs, the UDIR bit of the base timer x status control register (BTxSTC) changes to "1". At this time, an underflow interrupt request occurs if the UDIE bit is set to "1". Figure 5-7 shows the operation in case of an underflow.

Figure 5-7 Operation in Case of an Underflow



■ Output Waveform

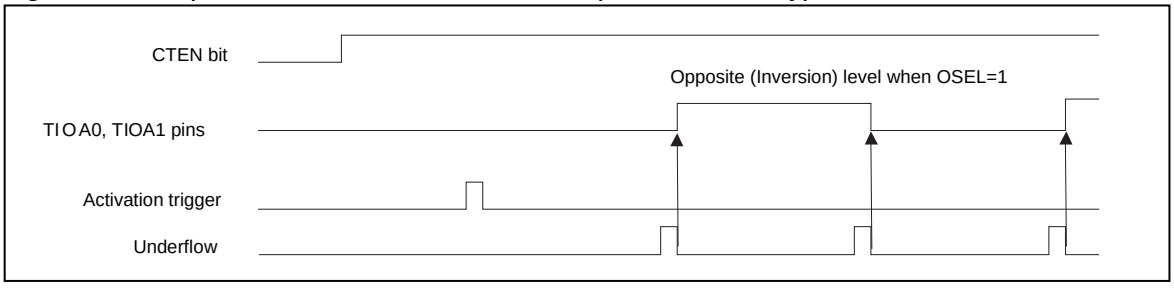
The waveform (TOUT signal) of the 16/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

Table 5-1 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 5-8 shows the output waveform in reload mode.

Figure 5-8 Output Waveform in Reload Mode (Normal Polarity)



5.4.3. Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

■ Overview

In this mode, the counter will no longer count down once an underflow occurs.

To use this mode, set one-shot mode by setting the MDSE bit of the base timer x timer control register (BTxTMCR) to "1"(MDSE=1).

■ Operation

● Activation

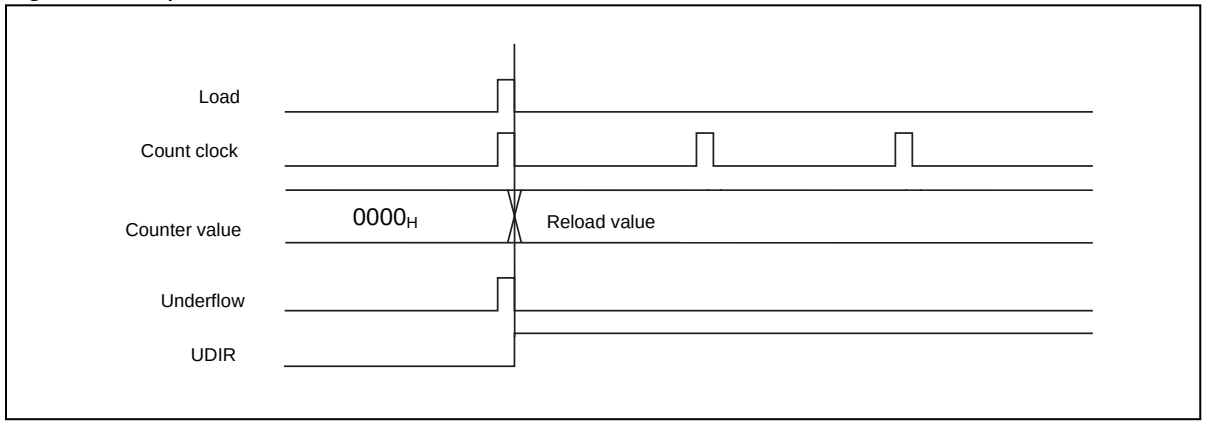
The same operation as in reload mode. See "■Operation" in "5.4.2 Operation in Reload Mode".

● Counting Operation

The operation is the same as in reload mode until an underflow occurs. See "■Operation". 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 5-9 shows the operation in case of an underflow.

Figure 5-9 Operation in Case of an Underflow



■ Output Waveform

The waveform (TOUT signal) of the 16/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

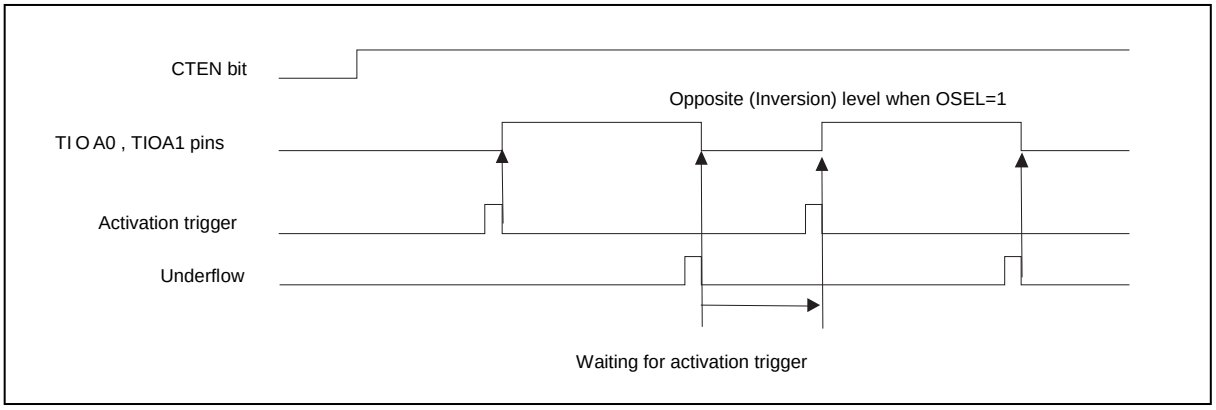
Table 5-2 shows the correspondence between output polarities and output waveforms.

Table 5-2 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 5-10 shows the output waveform in one-shot mode.

Figure 5-10 Output Waveform in One-shot Mode (Normal Polarity)



5.4.4. 32-bit Timer Mode Operation

This section explains the 32-bit timer mode operation.

This section explains the setting and operation for cascading 2 channels of a 16-bit reload timer and using them as a 32-bit reload timer.

■ Overview

Using the T32 bit of the base timer x timer control register (BTxTMCR), 2 channels of a 16-bit reload timer can be cascaded and used as a 32-bit reload timer. In this mode, the even-number channel corresponds to the lower 16-bit operation, and the odd-number channel corresponds to the upper 16-bit operation. Therefore, set the reload values in the order of the upper 16 bits (odd-number channels) → the lower 16 bits (even-number channels) and read the down counter values in the order of the lower 16 bits (even-number channels) → the upper 16 bits (odd-number channels).

■ Setting Procedure (Example)

To set 32-bit timer mode, set the T32 bit of the base timer x timer control register (BTxTMCR) of even-number channels to "1" and the T32 bit of the base timer x timer control register (BTxTMCR) of the odd-number channels to "0". When setting 32-bit timer mode, set the registers using the procedure shown below. 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). (T32=1)
3. Set a reload value in the upper 16 bits in the base timer 1 cycle setting register (BT1PCSR).
4. Set a reload value in the lower 16 bits in the base timer 0 cycle setting register (BT0PCSR).

Notes:

- Rewrite the T32 bit while the operation of both of the even-number and odd-number channels is stopped. Whether the counting operation is stopped can be checked by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "0"(CTEN=0).

- A reload value in the base timer x cycle setting register (BTxPCSR) must be set in the order of the odd-number → even-number channels.

■ Operation

In 32-bit timer mode, the counting operation is basically the same as in 16-bit timer mode.

However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the following registers for the odd-number channels.

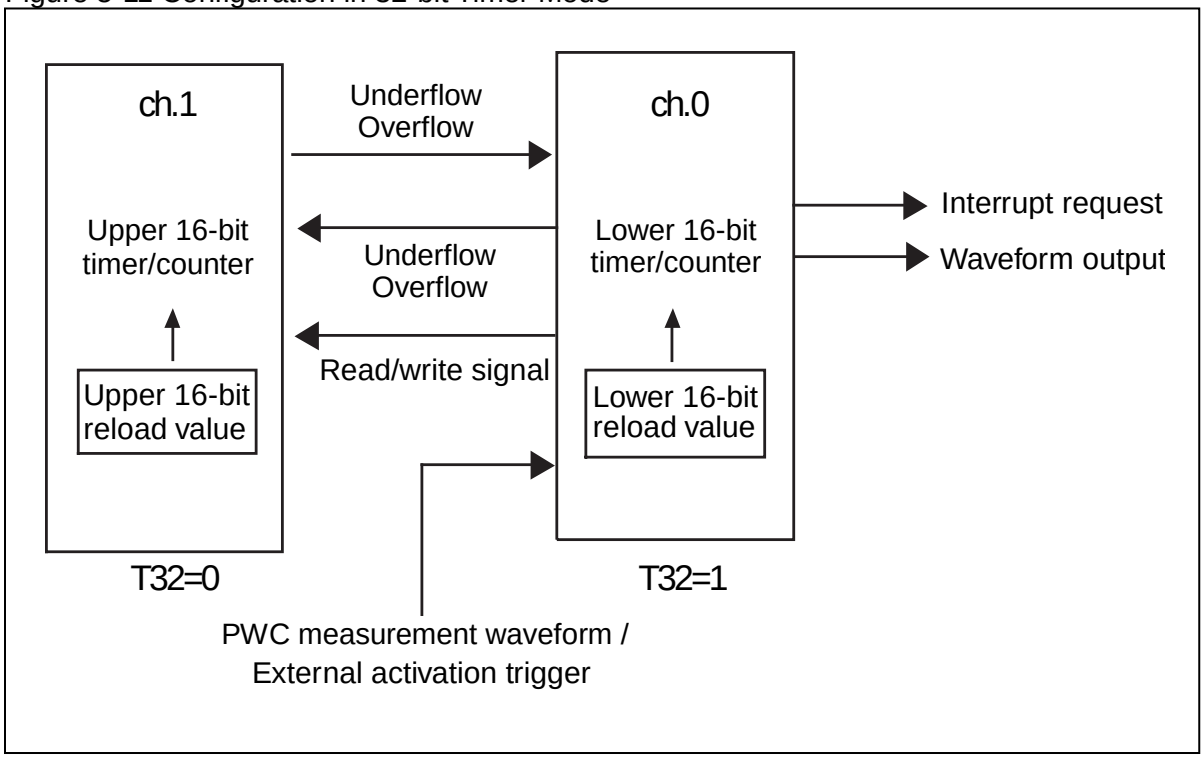
- Base timer x timer control register (BTxTMCR)
- Base timer x status control register (BTxSTC)

This section explains the counting in the 32-bit timer mode.

1. When the 32-bit reload timer activates, the values in the odd-number channel base timer x cycle setting register (BTxPCSR) and the even-number channel base timer x cycle setting register (BTxPCSR) (lower 16-bit) are loaded to the down counter.
2. The down counter starts counting as a 32-bit counter with the even-number channels serving as the lower 16-bit and the odd-number channels as the upper 16-bit.
3. When the down counter underflows, the UDIR bit of the base timer x timer control register (BTxTMCR) of the even-number channels changes to "1".

Figure 5-11 shows the channel configuration in 32-bit timer mode.

Figure 5-11 Configuration in 32-bit Timer Mode



Notes:

- The value of the down counter can be checked by reading the base timer x timer register (BTxTMR). In the 32-bit timer mode, it must be read in the order of the lower 16-bit (even-number channel) → upper 16-bit (odd-number channel).
- In 32-bit timer mode, the operation of the 32-bit reload timer conforms to the settings of the even-number channels. Therefore, activation triggers and interrupt requests from even-number channels are valid. The output signal (TOUT) from an odd-number channel pin is fixed to "L" level.

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 5-3 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 "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 "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

5.4.6. Precautions for Using this Device

This section explains precautions for using this device.

Note the following when using the 16/32-bit reload timer:

■ Notes on Program Setting

- Change the following bits of the base timer x timer control register (BTxTMCR) after stopping the 16-bit down counter by resetting CTEN bit to "0"(CTEN=0).
  - CKS2 to CKS0 bits
  - EGS1 and EGS0 bits
  - T32 bit
  - FMD2 to FMD0 bits
  - MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the timer control register (BTxTMCR) are set to "000" to select reset mode.
  Before the base timer function or T32 bit can be changed, the base timer must be reset once. Except when rewriting the status of FMD2 to FMD0 bits or T32 bit of the timer control register (BTxTMCR) after a reset, be sure to set the FMD2 to FMD0 bits to "000" to select the reset mode. Then, rewrite the status of these bits.

■ Notes on Operations

- If the count timing of the down counter and the load timing occur at the same time, the load operation is given precedence.
- If a 16/32-bit reload timer activation trigger is detected when counting ends in one-shot mode, the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is loaded to the 16-bit down counter, which begins counting. A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

■ Note on Interrupts

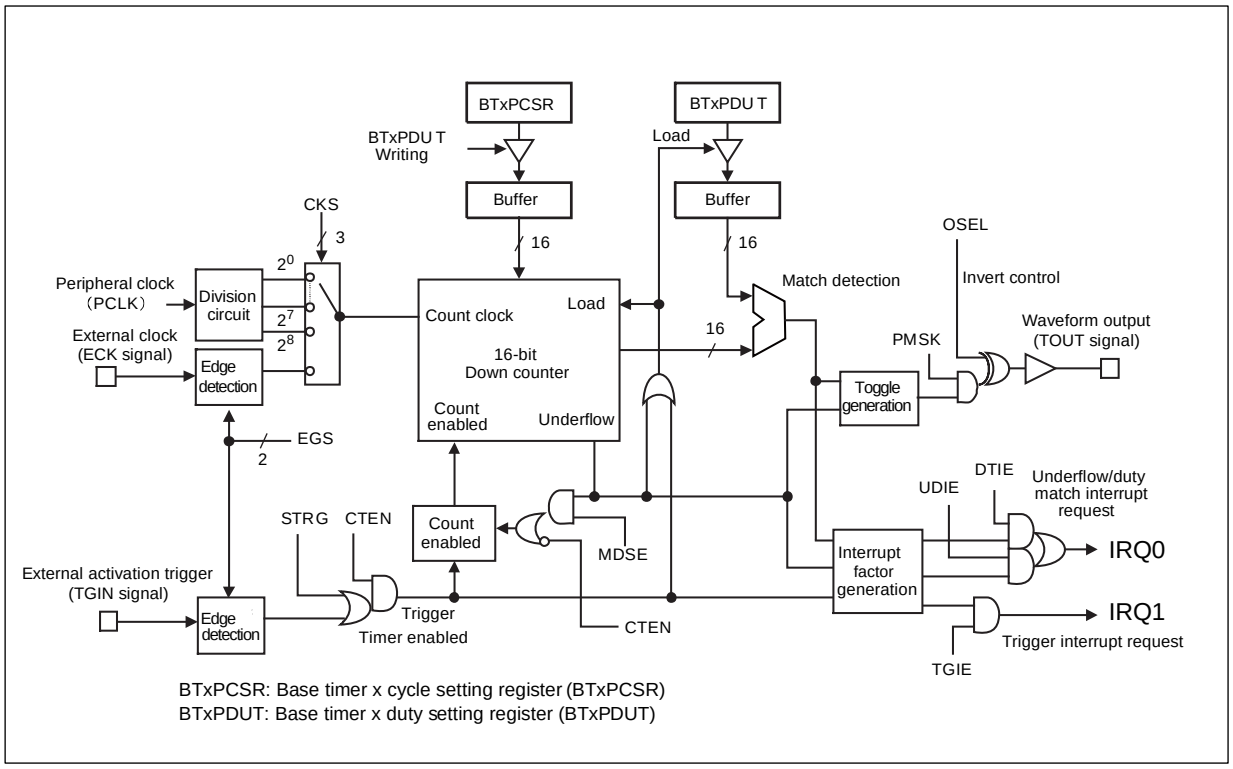
If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

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 5-12 Block Diagram (16-bit PWM Timer Operation)



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.

5.5.2. Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

■ Overview

In this mode, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded every time an underflow occurs to ensure that countdown is continued. To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0"(MDSE=0).

■ Operation

● Activation

Activate the 16-bit PWM timer with the following procedure:

1. Permit the 16-bit PWM timer operation by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1"(CTEN=1).
   - The 16-bit PWM timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

The 16-bit down counter starts decreasing from the value set in the base timer x cycle setting register (BTxPCSR).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01).
- 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:

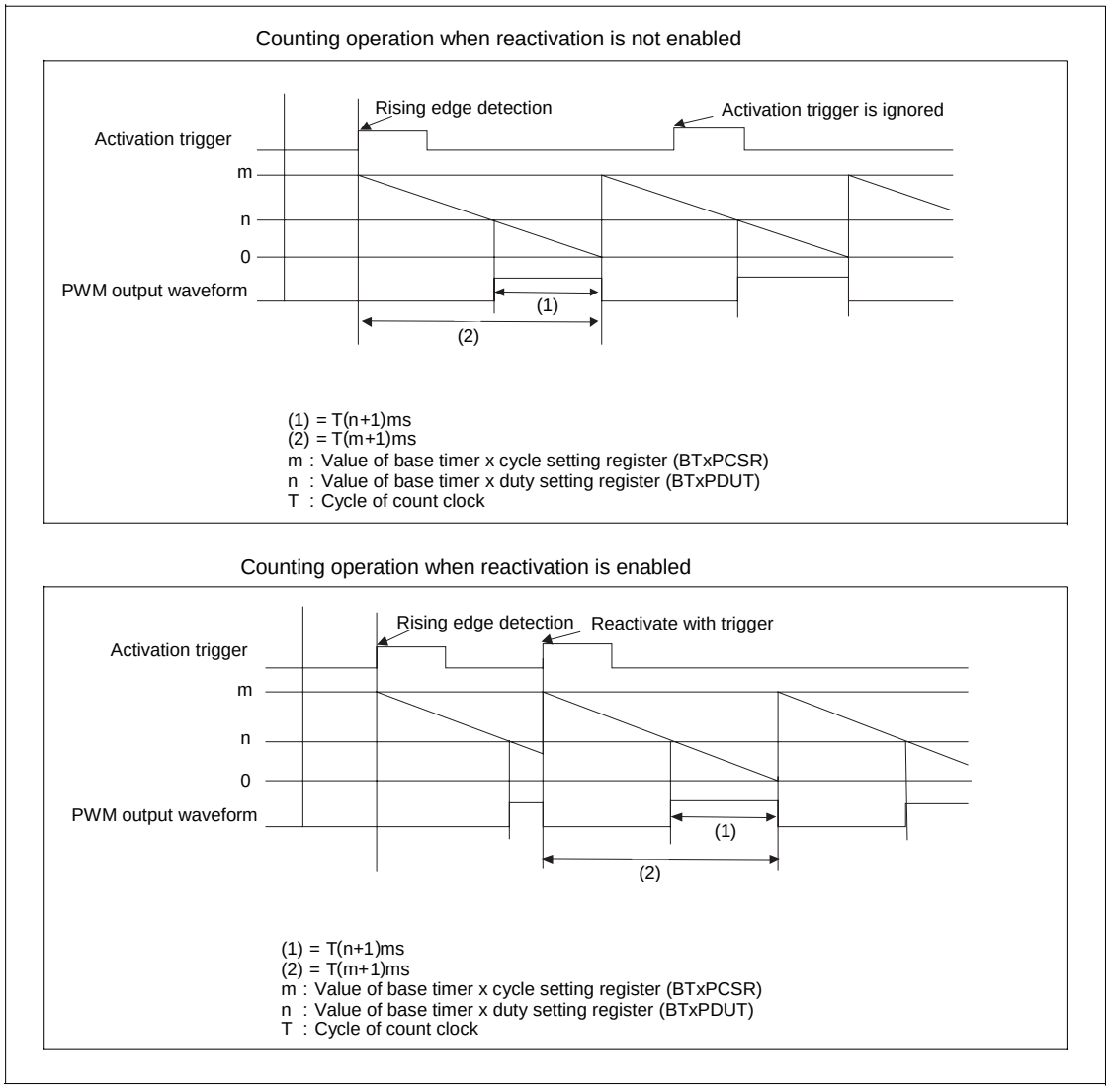
- The DTIR bit of the status control register (BTxSTC) changes to "1".
- The level of the output signal (TOUT) is inverted.
- Countdown is continued. Later, when the 16-bit down counter underflows, the operation is performed as follows:
- The UDIR bit of the status control register (BTxSTC) changes to "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 5-13 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

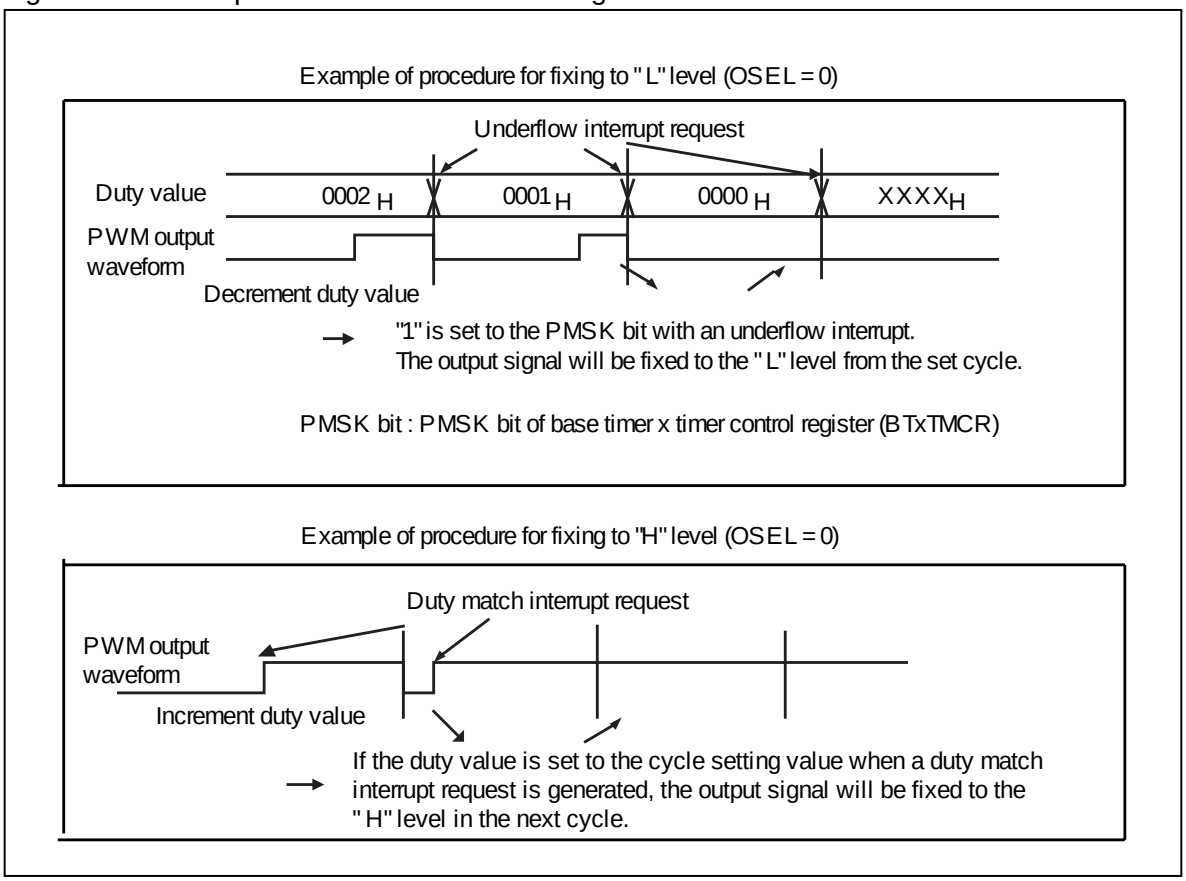
● Inverted polarity (OSEL = 1)

- 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 5-14 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

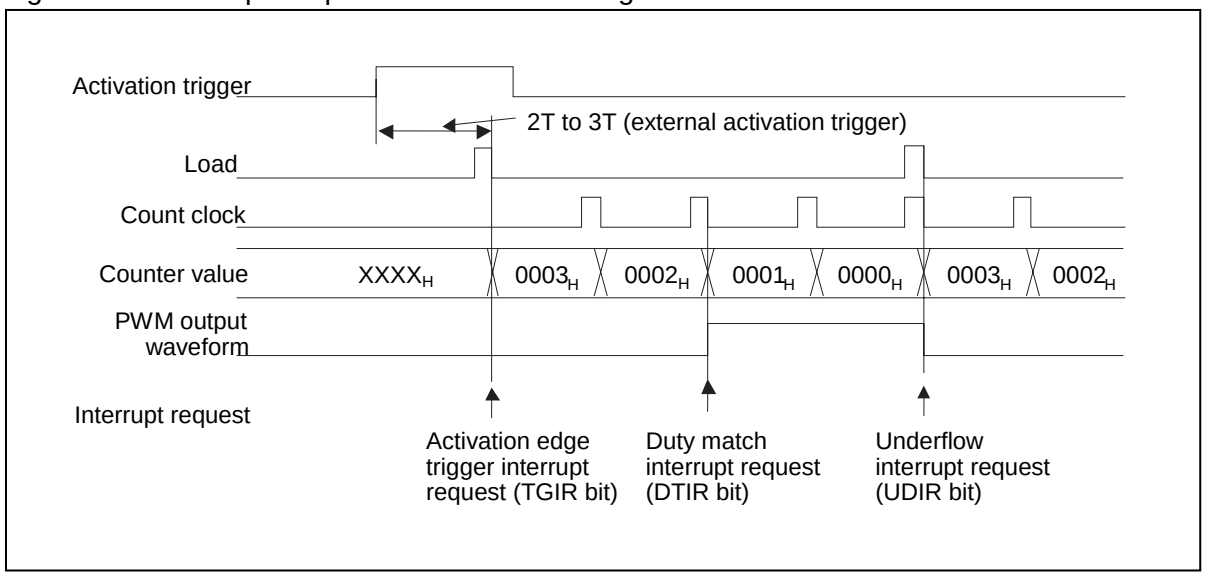
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- 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 5-15 Interrupt Request Generation Timing Chart



5.5.3. Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

■ Counting Operation

In this mode, counting stops if an underflow occurs when the value of the 16-bit down counter changes from the value set in the cycle setting register (BTxPCSR) to "FFFF$_H$".

To use this mode, set one-shot mode by setting the MDSE bit of the timer control register (BTxTMCR) to "1"(MDSE=1).

● Activation

It is the same operation as in reload mode. See "■Operation" in the section entitled "5.5.2 Operation in Reload Mode".

● Counting Operation

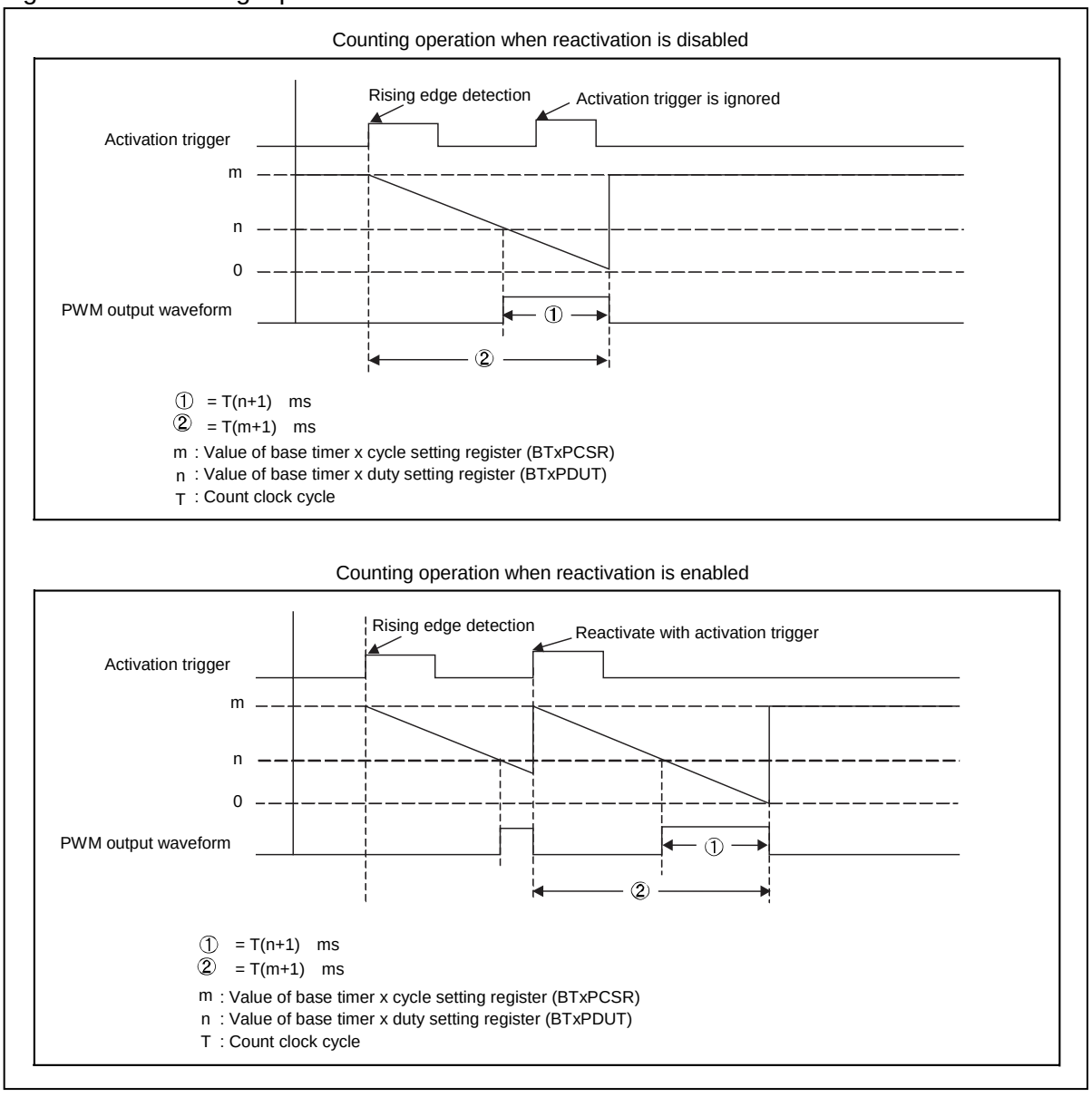
When an activation trigger is input, the 16-bit down counter, in synchronization with the count clock, starts decreasing from the value set in the cycle setting register (BTxPCSR). When the value of the 16-bit down counter matches the value of the duty setting register (BTxPDUT), the operation is performed 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 5-16 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 "5.5.2 Operation in Reload Mode".

■ Interrupt Generation Timing

It is the same operation as in reload mode. See "■Interrupt Generation Timing" in "5.5.2 Operation in Reload Mode".

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 5-4 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 "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 "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

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.

■ Note on Interrupts

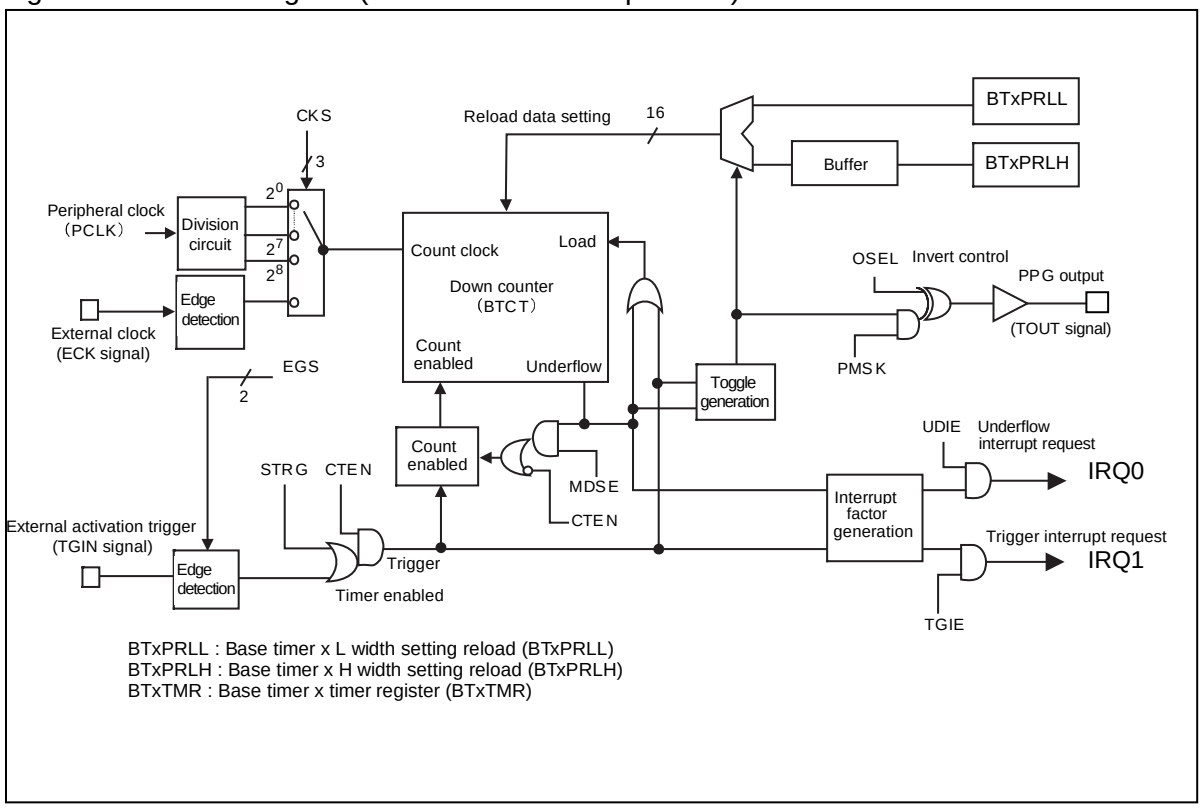
If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

5.6. 16-bit PPG Timer Operation

This section explains the 16-bit PPG timer operation.

This section explains the operation performed when the base timer included in this series is used as the 16-bit PPG timer. Examples of procedures for setting various operating conditions are also provided.

Figure 5-17 Block Diagram (16-bit PPG Timer Operation)



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.

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.

5.6.3. Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

■ Overview

In this mode, the values set in the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) are alternately reloaded to the down counter to ensure that the down counter continues to count down. A desired pulse width can be output continuously by rewriting the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) each time an underflow interrupt request is issued.

To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0"(MDSE=0).

■ Operation

● Activation

Activate the 16-bit PPG timer with the following procedure:

1. Permit the 16-bit PPG timer operation by setting the CTEN bit of the timer control register (BTxTMCR) to "1"(CTEN=1). The 16-bit PPG timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01).
- 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).

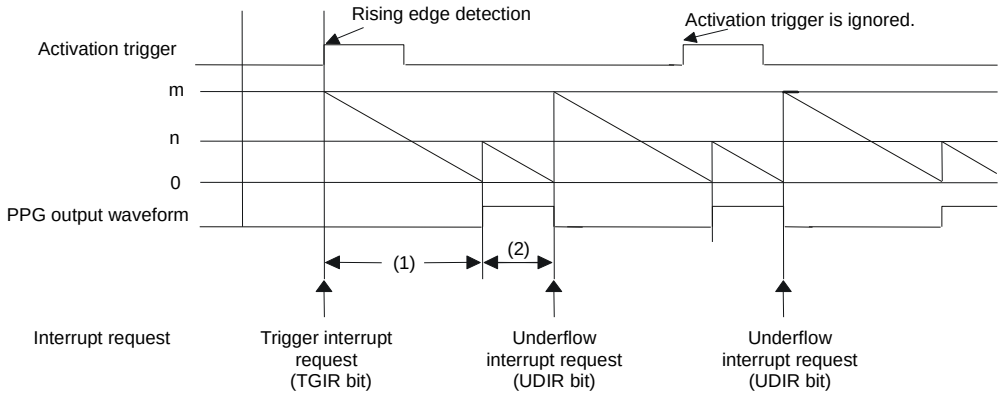
1. The value set in the L width setting reload register (BTxPRLL) is transferred to the 16-bit down counter and the value set in the base timer x H width setting reload register (BTxPRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting 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 5-18 Example of Counting Operation in Reload Mode

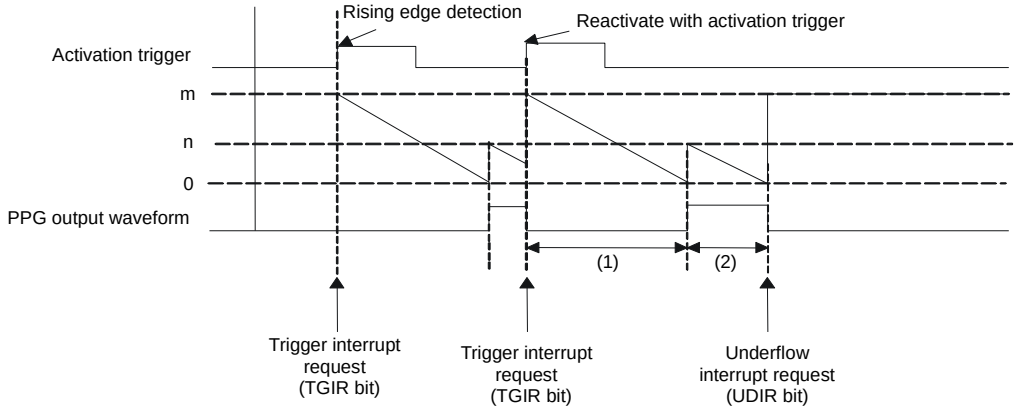
Counting operation when reactivation is disabled



(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

Note:

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:

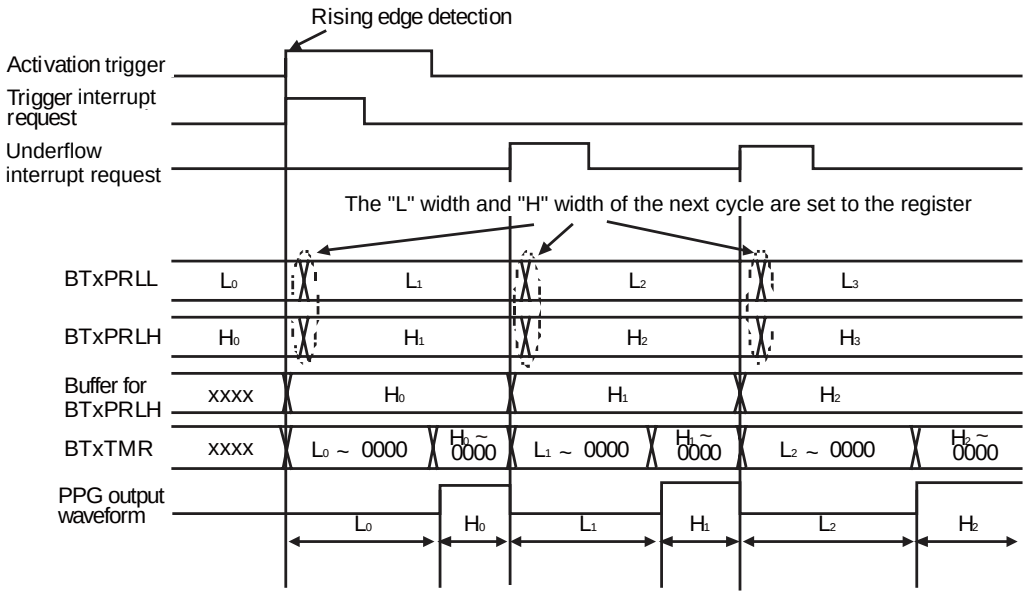
- An activation trigger is detected.
- An underflow occurs after counting down from the value of the base timer x H width setting reload register (BTxPRLH) is completed.

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 5-19 Write Timing



■ Interrupt Generation Timing

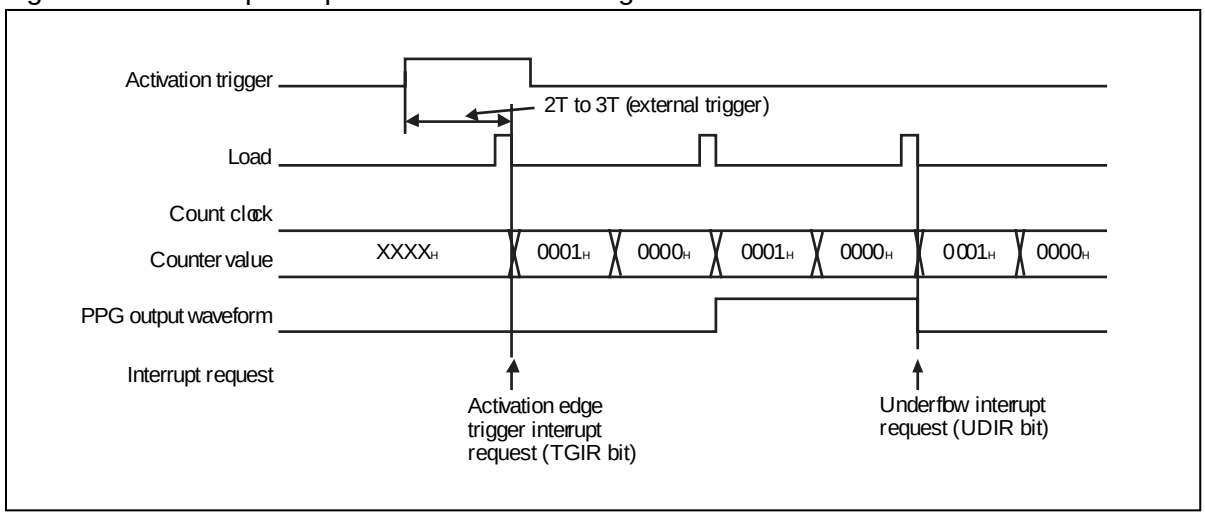
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- 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 5-20 Interrupt Request Generation Timing Chart



5.6.4. Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

■ Counting Operation

● Activation

It is the same operation as in reload mode. See "■Operation" in "5.6.3 Operation in Reload Mode".

● Counting Operation

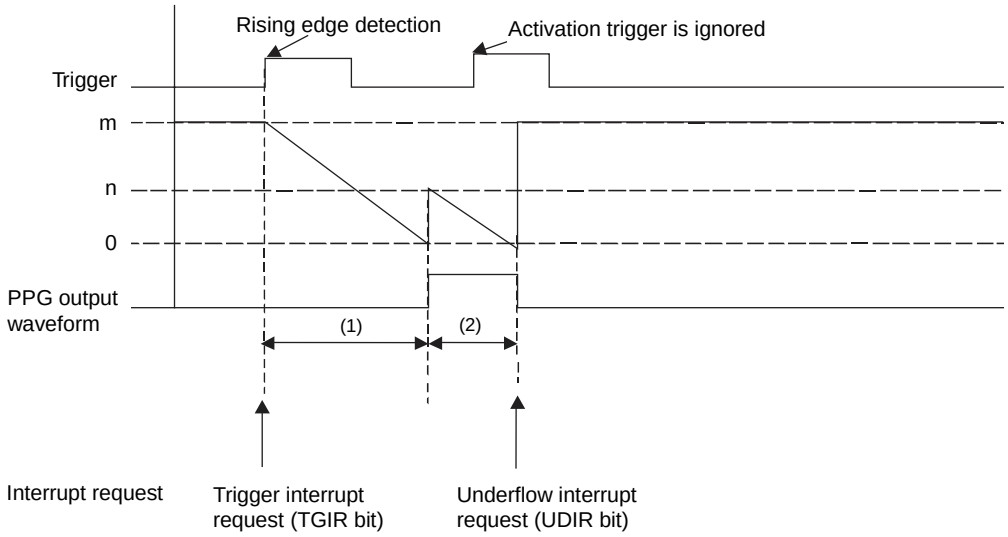
Counting operation initiated by the entry of an activation trigger is explained below, using an example where the OSEL bit of the timer control register (BTxTMCR) is set for normal polarity (OSEL = 0).

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).

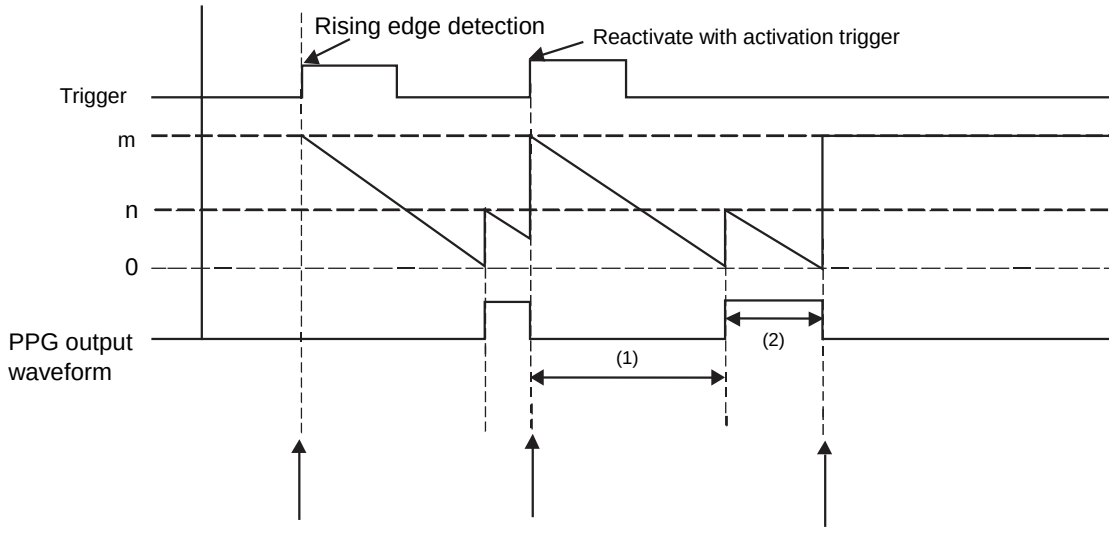
- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.
- If reactivation is permitted (RTGEN =1): The TGIR bit of the status control register (BTxSTC) changes to "1". In addition, the value of L width setting reload register (BTxPRLL) is reloaded to the 16-bit down counter, which starts counting.

Figure 5-21 Example of Counting Operation If Reactivation Is Not 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

Figure 5-22 Example of Counting Operation If 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 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 "5.6.3 Operation in Reload Mode".

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 5-5 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.
- Set an interrupt level corresponding to the interrupt vector number, using interrupt control registers (ICR00 to ICR47). For information on interrupt level setting, see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

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.

■ Note on Interrupts

If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

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 5-23 Block Diagram (16-bit PWC Timer Operation)

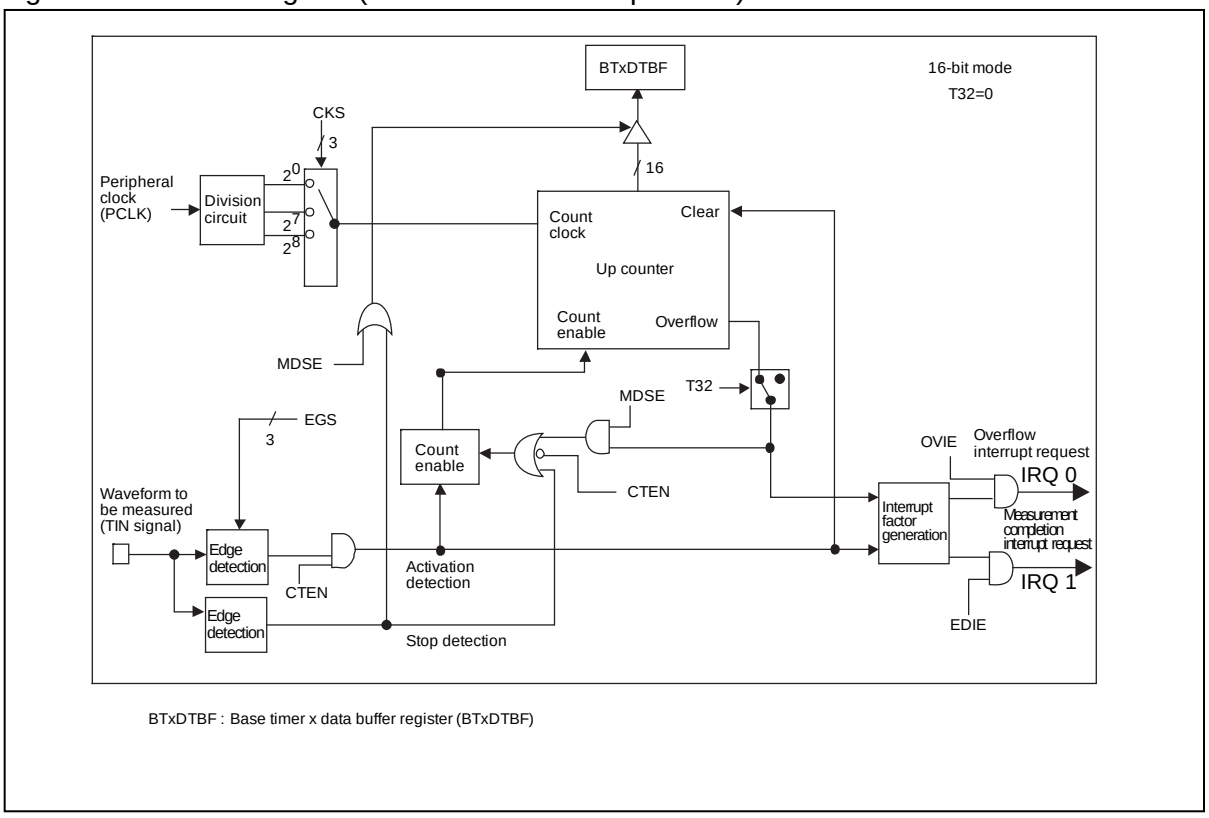
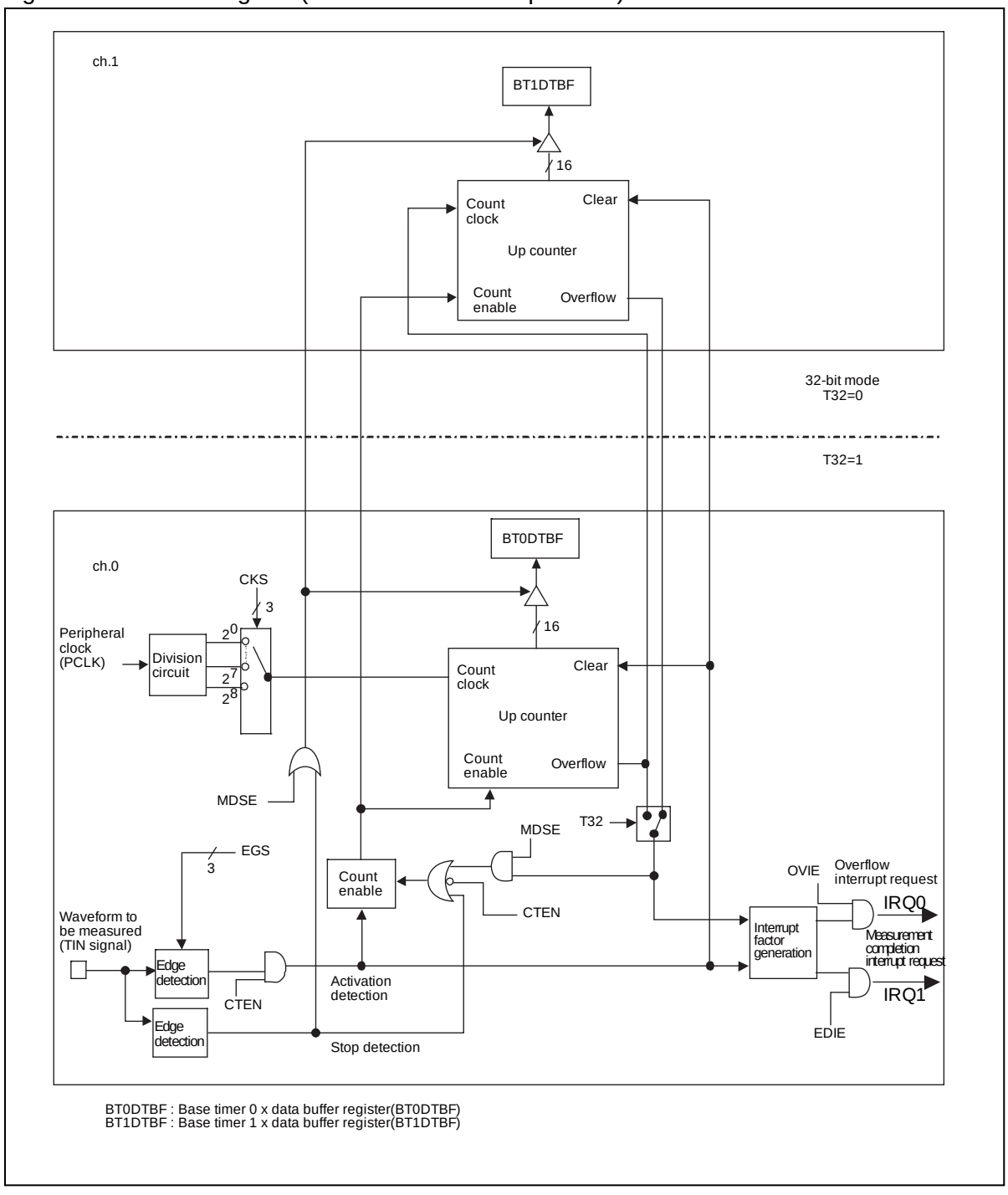


Figure 5-24 Block Diagram (32-bit PWC Timer Operation)



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 "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 "5.7.3 32-bit Timer Mode Operation" for details on the operation in 32-bit timer mode.

Note:

The T32 bit setting differs between odd-number and even-number channels when the 32-bit timer mode is selected. For details, see "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).

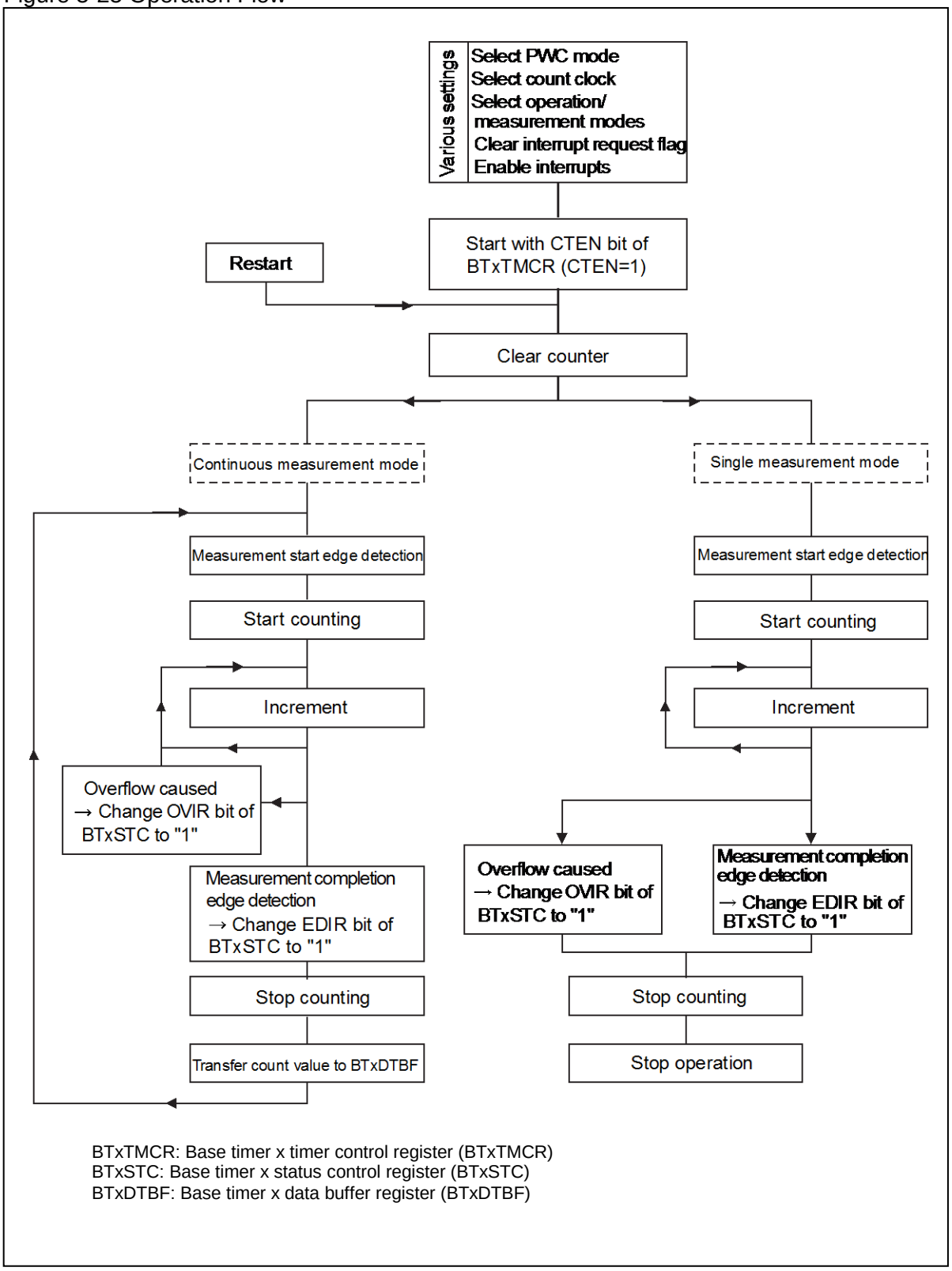
- Continuous measurement mode (MDSE = 0): In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
- Single measurement mode (MDSE = 1): In this mode, measurement is conducted only once. Differences between the single and continuous measurement modes are listed on the table below.

Table 5-6 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 5-25 shows the standard operation flow.

Figure 5-25 Operation Flow



BTxTMCR: Base timer x timer control register (BTxTMCR)
BTxSTC: Base timer x status control register (BTxSTC)
BTxDTBF: Base timer x data buffer register (BTxDTBF)

Note:

In the continuous measurement mode, if the next measurement is completed before the measurement result has been read from the data buffer register (BTxDTBF), the value being held by the data buffer register (BTxDTBF) is overwritten by the new value. The old value is discarded. If it has occurred, the ERR bit of the status control register (BTxSTC) changes to "1". This ERR bit is cleared to "0" when a value is read from the base timer x data buffer register (BTxDTBF).

■ Measurement Mode

Either of the following five modes can be selected using EGS2 to EGS0 bits of the timer control register (BTxTMCR).

Figure 5-26 Measurement Modes and their Explanation 1

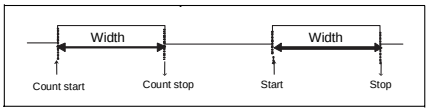
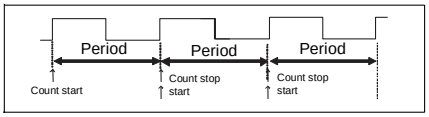
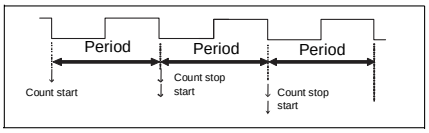
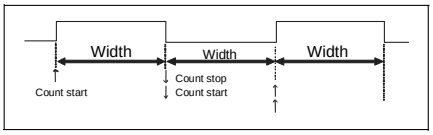
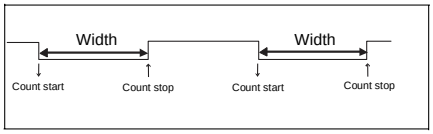
Measurement mode (EGS2 to EGS0)	Measurement description
Measurement of H pulse width (EGS2 to EGS0=000)	The width of the period which the "H" level signal is being input is measured.  Count (measurement) start: at rising edge detection Count (measurement) stop: at falling edge detection
Measurement of the cycle between rising edges (EGS2 to EGS0=001)	The cycle from the rising edge detection to the next rising edge detection is measured.  Count (measurement) start: at rising edge detection Count (measurement) stop: at rising edge detection
Measurement of the cycle between falling edges (EGS2 to EGS0=010)	The cycle from the falling edge detection to the next falling edge detection is measured.  Count (measurement) start: at falling edge detection Count (measurement) stop: at falling edge detection

Figure 5-27 Measurement Modes and their Explanation 2

Measurement mode (EGS2 to EGS0)	Measurement description
Measurement of the pulse width between all edges (EGS2 to EGS0=011)	The width between the edges input continuously is measured. •From rising edge detection to falling edge detection •From falling edge detection to rising edge detection  Count (measurement) start: at edge detection Count (measurement) stop: at edge detection
Measurement of L pulse width (EGS2 to EGS0=100)	The width of the period during which the "L" level signal being input is measured.  Count (measurement) start: at falling edge detection Count (measurement) stop: at rising edge detection

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 5-26 Measurement Modes and their Explanation 1" and "Figure 5-27 Measurement Modes and their Explanation 2".

■ Activation

Activate the 16/32-bit PWC timer with the following procedure:

- Enable the 16/32-bit PWC timer operation by setting the CTEN bit of the timer control register (BTxTMCR) to "1"(CTEN=1). The counter value is cleared to "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).

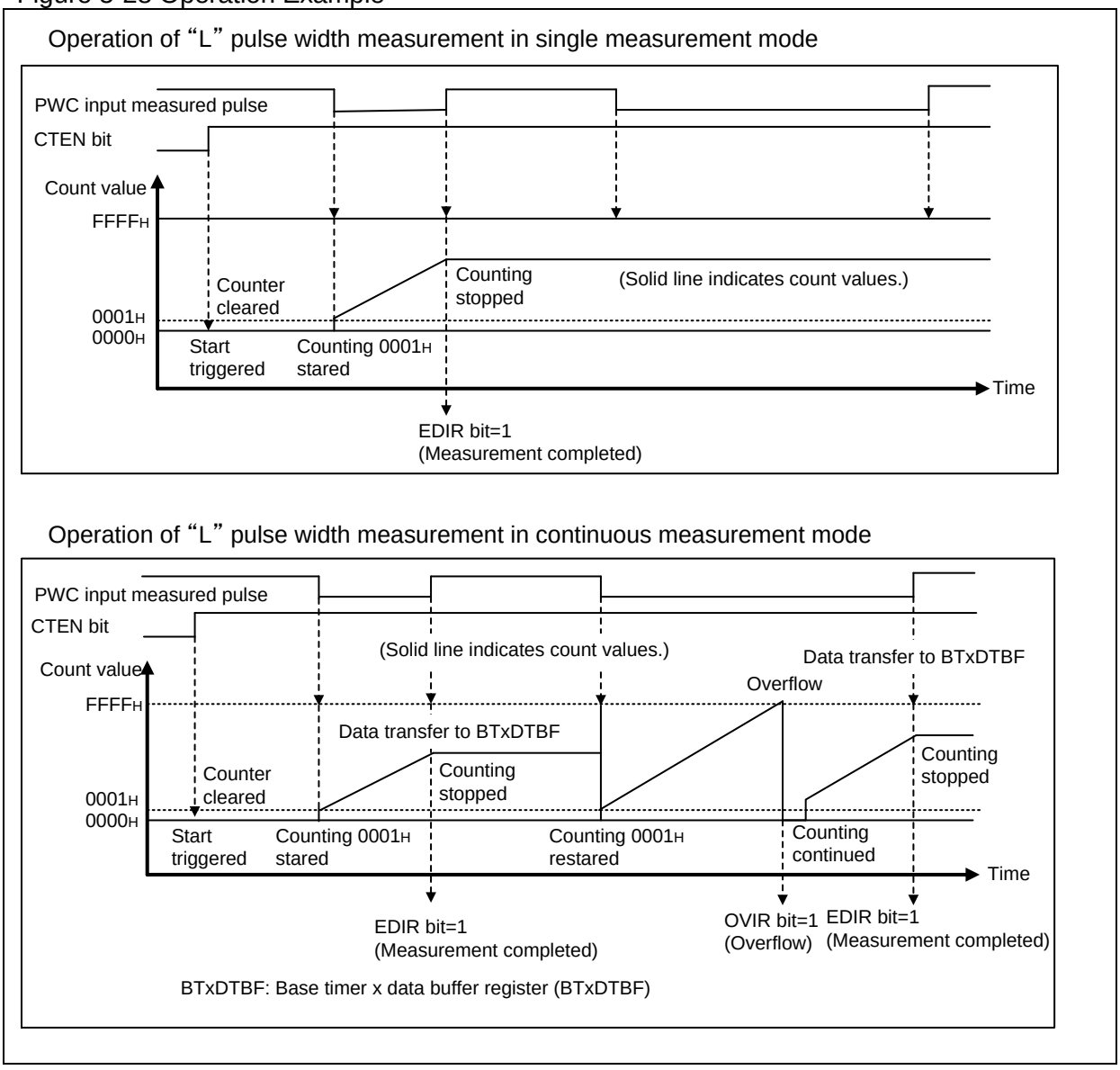
Operation in continuous measurement mode

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 5-28 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

5.7.3. 32-bit Timer Mode Operation

This section explains the 32-bit timer mode operation.

This section explains the setting and operation for cascading 2 channels of a 16-bit PWC timer and using them as a 32-bit PWC timer.

■ Overview

Using the T32 bit of the timer control register (BTxTMCR), 2 channels of a 16-bit PWC timer can be cascaded and used as a 32-bit PWC timer.

In this mode, the even-number channel corresponds to the lower 16-bit operation, and the odd-number channel corresponds to the upper 16-bit operation. Therefore, the up counter must be read in the order of the lower 16 bits (even-number channel) → the upper 16 bits (odd-number channel).

■ Setting Procedure (Example)

To select the 32-bit timer mode, set the T32 bit of the base timer x timer control register (BTxTMCR) of the even-number channel to "1". Also, set the T32 bit of the odd-number channel to "0". When setting 32-bit timer mode, set the registers using the procedure shown below.

The register setting differs between even-number and odd-number channels. In this example, channel 0 and channel 1 are connected by cascading.

1. Specify ch.0 to reset mode by setting FMD2 to FMD0 bits of the base timer 0 timer control register (BT0TMCR). (FMD2 to FMD0 = 000)
2. Set FMD2 to FMD0 of the timer control registers (BT0TMCR, BT1TMCR) of both base timer ch.0 and ch.1 to "100" (16/32 bit PWC timer) and, at the same time, set the T32 bits of the timer control registers (BT0TMCR, BI1TMCR) to "1" and "0", respectively, to set 32-bit timer mode. (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-number and odd-number 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 following registers for the odd-number channels.

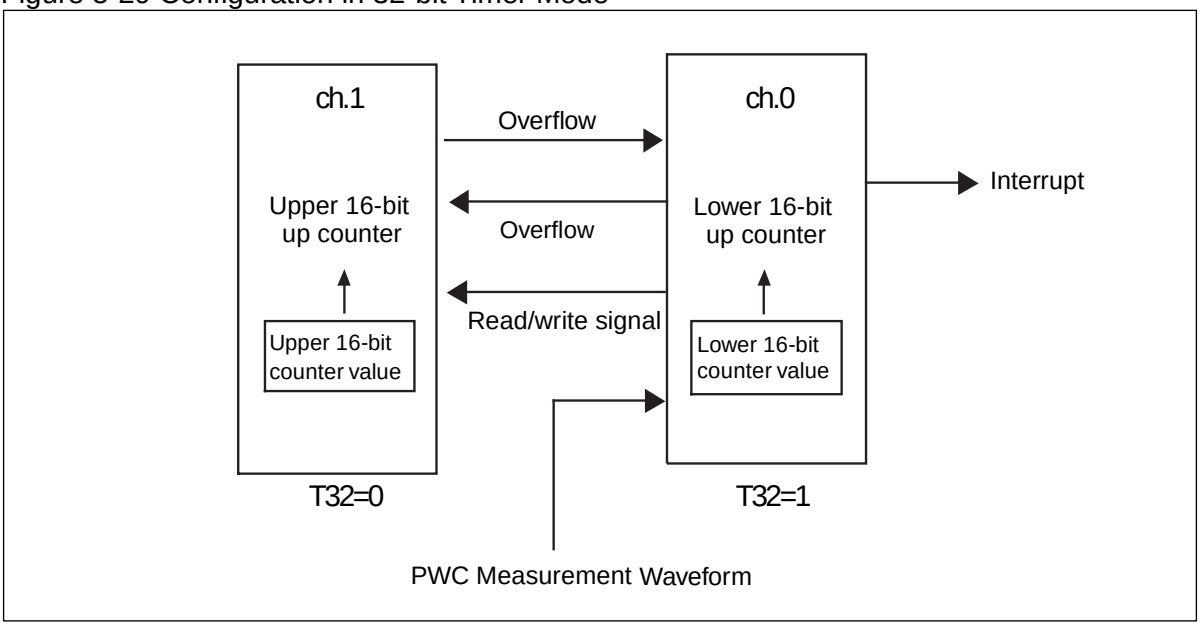
- Base timer x timer control register (BTxTMCR)
- Base timer x status control register (BTxSTC)

This section explains the counting in the 32-bit timer mode.

1. If the 16/32-bit PWC timer operation is enabled using the CTEN bit of the timer control register (BTxTMCR) (by setting CTEN = 1) of the even-number channel, the 32-bit PWC timer starts.
2. When a measurement start edge is detected in the input signal (TIN), the counting starts.
3. The up counter starts counting as a 32-bit counter with the even-number channel serving as the lower 16 bits and the odd-number channel as the upper 16 bits.
4. When a measurement end edge is detected in the input signal (TIN signal), the lower 16-bit data of the up counter value is stored in the data buffer register (BTxDTBF) of the even-number channel, and the upper 16-bit data of the up counter value is stored in the data buffer register (DTxDTBF) of the odd-number channel.

The channel configuration in 32-bit timer mode is shown below.

Figure 5-29 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-number 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-number channel is effective.

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 5-7 Interrupt Occurrence Conditions

Interrupt request	Interrupt request flag	Permission of interrupt request	Interrupt request clear
Overflow interrupt request	BTxSTC:OVIR=1	BTxSTC:OVIE=1	Set the OVIR bit of BTxSTC to "0".
Measurement end interrupt request	BTxSTC:EDIR=1	BTxSTC:EDIE=1	Read BTxDTBF

Notes:

- Once the generation of an interrupt request is enabled while the interrupt request flag is "1", an interrupt request will be issued when the interrupt is enabled.
- To enable the generation of an interrupt request, perform one of the following operations:
  - Clear the current interrupt request before enabling the generation of an interrupt request.
  - Clear the current interrupt request when enabling the interrupt.
- Either clear the current interrupt request after disabling the generation of an interrupt request or clear the current interrupt request within the interrupt processing routine.
- For interrupt vector numbers used for issuing an interrupt request, see "List of 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 "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

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-number channel is made valid. The timer setting of odd-number channel is ignored.
- The input operation of measurement waveforms varies depending on the base timer I/O selection function.

■ Notes on Interrupts

- If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".
- If a detection of measurement end edge and a reactivation of 16/32-bit PWC timer occur simultaneously, the following may result. In such case, set the interrupt control correctly by considering the operation of the interrupt request flag.
  - Pulse width single measurement mode: The timer reactivates and waits for a measurement start edge. Also, the measurement end interrupt request flag (EDIR) is set to "1".
  - Pulse width continuous measurement mode: The timer reactivates and waits for a measurement start edge. The measurement end interrupt request flag (EDIR) is set to "1", and the currently measured result is transferred to the data buffer register (BTxDTBF).

Chapter 20: Reload Timer

This chapter explains the reload timer.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Application Note

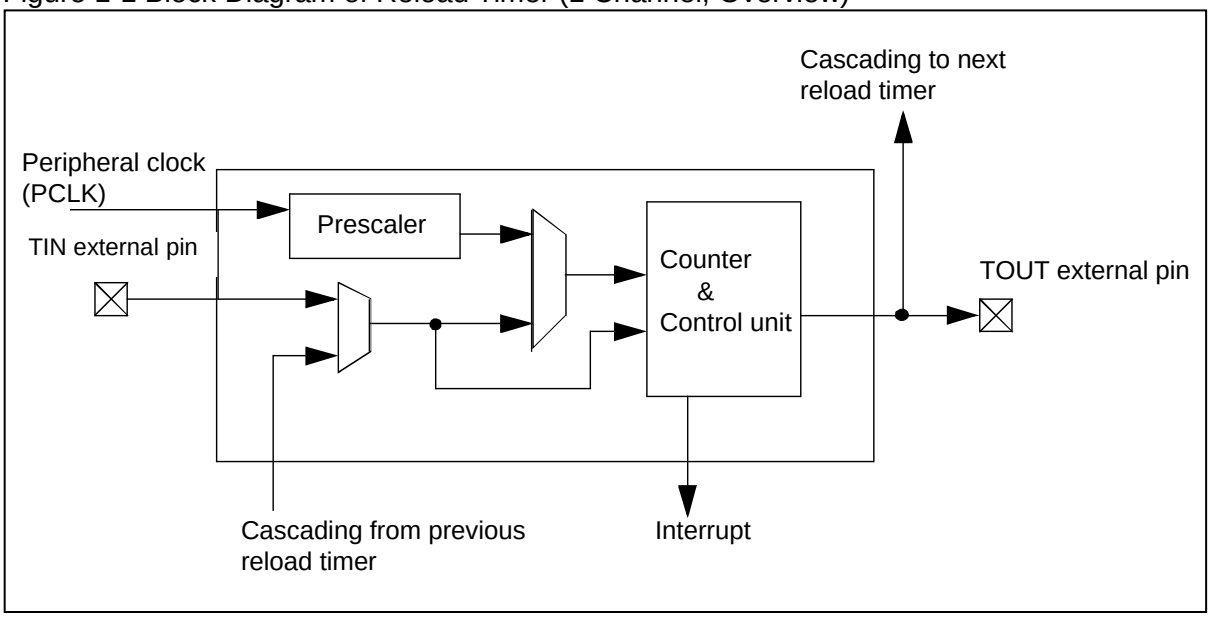
Code : FR81S10_RLT-1v1-91520-2-E

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 1-1 Block Diagram of Reload Timer (1 Channel, Overview)



The numbers of available channels are shown below.

MB91F52xB (64pin) : 7
MB91F52xD (80pin) : 7
MB91F52xF (100pin) : 8
MB91F52xJ (120pin) : 8
MB91F52xK (144pin) : 8
MB91F52xL (176pin) : 8

2. Features

This section explains features of the reload timer.

An 8-channel reload timer is installed in this series.

- 7channels : MB91F52xB, MB91F52xD
- 8channels : MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL

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 (TIN) ×1
- External toggle output (TOUT) ×1
- Control register ×1
- Count comparator ×1

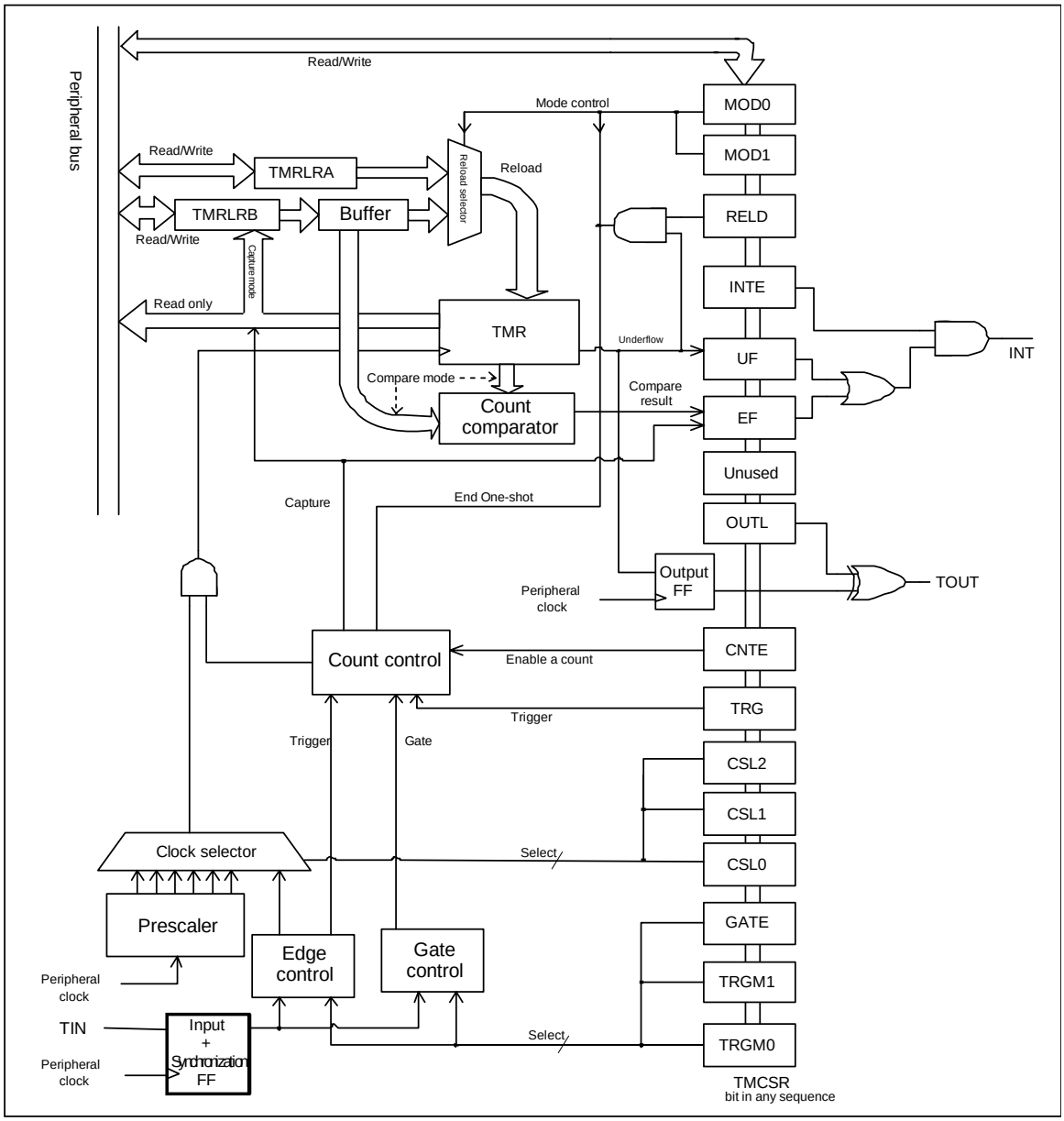
This timer, equipped with the interval timer mode/event counter mode described below, can be used for the following purposes and functions by setting the registers:

- 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. Use ch.6 output for ch.7 input.

3. Configuration

This section explains the configuration of the reload timer.

Figure 3-1 Block Diagram of Reload Timer (1 Channel, Details)



4. Registers

This section explains registers of the reload timer.

■ Table of Base Address (Base_addr), External Pins

Table 4-1 Table of Base Address (Base_addr), External Pins

Channel	Base_addr	External pin (TOUT output, TIN input)					
		MB91 F52xB	MB91 F52xD	MB91 F52xJ	MB91 F52xL	MB91 F52xK	MB91 F52xL
0	0x0060	TOT0_0	TOT0_0	TOT0_0	TOT0_0	TOT0_0/ TOT0_1	TOT0_0/ TOT0_1
		TIN0_2	TIN0_2	TIN0_2	TIN0_0/ TIN0_2	TIN0_0/ TIN0_2	TIN0_0/ TIN0_1/ TIN0_2
1	0x0100	TOT1_2	TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_1/ TOT1_2
		TIN1_0	TIN1_0	TIN1_0	TIN1_0	TIN1_0	TIN1_0/ TIN1_1
2	0x0108	none	none	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1
				TIN2_0	TIN2_0	TIN2_0/ TIN2_1	TIN2_0/ TIN2_1
3	0x0110	TOT3_0	TOT3_0	TOT3_0	TOT3_0	TOT3_0/ TOT3_1	TOT3_0/ TOT3_1
		TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_1/ TIN3_2	TIN3_0/ TIN3_1/ TIN3_2
4	0x01D8	TOT4_0	TOT4_0	TOT4_0	TOT4_0	TOT4_0/ TOT4_1	TOT4_0/ TOT4_1
		TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1
5	0x01F0	TOT5_0	TOT5_0	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1
		TIN5_0	TIN5_0	TIN5_0	TIN5_0/ TIN5_1	TIN5_0/ TIN5_1	TIN5_0/ TIN5_1
6	0x01F8	TOT6_1	TOT6_1	TOT6_0/ TOT6_1	TOT6_0/ TOT6_1	TOT6_0/ TOT6_1	TOT6_0/ TOT6_1
		TIN6_1	TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1
7	0x0068	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1
		TIN7_0	TIN7_0	TIN7_0	TIN7_0	TIN7_0/ TIN7_1	TIN7_0/ TIN7_1

■ Registers Map

Table 4-2 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x01D8	TMRLRA4		TMR4		16-bit timer reload register A4 16-bit timer register 4
0x01DC	TMRLRB4		TMCSR4		16-bit timer reload register B4 Control status register 4
0x01F0	TMRLRA5		TMR5		16-bit timer reload register A5 16-bit timer register 5
0x01F4	TMRLRB5		TMCSR5		16-bit timer reload register B5 Control status register 5
0x01F8	TMRLRA6		TMR6		16-bit timer reload register A6 16-bit timer register 6
0x01FC	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
0x0068	TMRLRA7		TMR7		16-bit timer reload register A7 16-bit timer register 7
0x006C	TMRLRB7		TMCSR7		16-bit timer reload register B7 Control status register 7

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 to bit8 and bit6 to bit4 and write counter operation enabling by writing CNTE= "1" simultaneously. It is also possible to rewrite bit15 to bit8, bit6 to bit4 and write operation disabling by writing CNTE= "0" simultaneously.

■ TMCSR : Address Base_addr + 06_H (Access : Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MOD[1:0]		TRGM[1:0]		CSL[2:0]			GATE
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EF	Reserved	OUTL	RELD	INTE	UF	CNTE	TRG
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R,W	R,W	R,W	R/W	R(RM1),W	R/W	R0,W

[bit15, bit14] MOD [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) : TIN Input mode selection bits

These bits control input pin functions. The functions of the interval timer mode differ from those of the event counter mode.

[Interval timer mode, trigger input (bit8:GATE = "0")]

Select an effective external edge which can be a reload trigger through TIN input in the following manner:

TRGM[1:0]	TIN effective external edge
00	No external trigger detection (initial value)
01	Rising edge
10	Falling edge
11	Both edges

[Interval timer mode, gate input (bit8:GATE bit= "1")]

Select the pin level which enables the counter during TIN input in the following manner:

TRGM[1:0]	TIN effective level
x0	Counted only during the input period for TIN pin "L" (initial value)
x1	Counted only during the input period for TIN pin "H"

[Effective edge setting at the event counter mode]

In the event counter mode, select an edge for external event detection in the following manner:
Every time an external event is detected, the counter value is decreased. When an external event is selected, the setting of the bit8:GATE bit becomes invalid.

TRGM[1:0]	Count target edge
00	Reserved
01	Rising edge
10	Falling edge
11	Both edges

[bit11 to bit9] CSL[2:0] (Count source SeLect) : Count source selection bits

These bits specify the count source. Select a count source from the internal clock (peripheral clock (PCLK)) and the external event (TIN input) specified following: When the event counter mode is set, set the count effective edge using bit13, bit12:TRGM[1:0].

CSL[2:0]	Count source	Operation mode
000	Division of the peripheral clock frequency by 2 (initial value)	Interval timer mode
001	Division of the peripheral clock frequency by 4	
010	Division of the peripheral clock frequency by 8	
011	Division of the peripheral clock frequency by 16	
100	Division of the peripheral clock frequency by 32	
101	Division of the peripheral clock frequency by 64	
110	Cascade mode (ch.0: TIN0, ch.1:TOUT0, ch.2:TOUT1, ch.3:TOUT2, ch.4: TIN4, ch.5:TOUT4, ch.6:TOUT5, ch.7:TOUT6)	Event counter mode
111	External event (TIN input)	

[bit8] GATE (GATE input enable) : Gate input enabling bit

This bit controls the functions of the input pin (TIN) of (bit11 to bit9:CSL[2:0]=000 to 101) at the interval timer mode specified following.

GATE	TIN input pin 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. * (initial value)
1	Reload mode	Counter underflow occurs. At the same time, the contents of the reload register are loaded to the counter to continue count operation.

* : However, the dual one-shot function reloads TMRLRB at the same time as TMRLRA underflow and continues counting. After that, count operation stops at the same time as TMRLRB underflow.

[bit3] INTE (INTerrupt Enable) : Interrupt request enabling bit

This bit controls an interrupt request in case of underflow/compare match (event counter mode)/capture specified following:

INTE	Description of operation
0	Interrupt disabled (no interrupt is generated even if the UF/EF bit is set.) (initial value)
1	Interrupt enabled (an interrupt request is generated if the UF/EF bit is set.)

[bit2] UF (Under flow Flag) : Underflow flag

This flag indicates that underflow has occurred when the counter value is decreased from 0x0000.

Set factor	Counter underflow occurrence
Clear factor	Writing "0" to this bit or reset.

[bit1] CNTE (timer CouNTer Enable) : Timer count enabling bit

This bit controls the operation of the timer as follows:

CNTE	Description of operation
0	Operation disabled (initial value)
1	Operation enabled (waiting for activation trigger)

[bit0] TRG (software TRiGger) : Software trigger bit

This bit generates a timer software trigger. If a software trigger is generated, the contents of the reload register are loaded to the counter to initiate count operation.

TRG	Description of operation
Write "0"	No influence on the operation
Write "1"	A software trigger is generated.

When "0" is written into this bit, no influence on the operation. The read value is always "0".
Trigger input through this register is effective only when bit1:CNTE = "1".
Writing "1" into the TRG bit always generates an effective trigger if the timer is activated (bit1:CNTE= "1") in any operation mode.

4.2. 16-bit Timer Register : TMR (16bit TiMer Register)

The bit configuration of the 16-bit timer register is shown below.

This register can read the timer count value.

Always perform 16-bit access to this register.

■ TMR : Address Base_addr + 02_H (Access : Half-word)

	bit15	bit14	....	bit2	bit1	bit0
	TMR[15:0]					
Initial value	X	X	....	X	X	X
Attribute	R,WX	R,WX	....	R,WX	R,WX	R,WX

[bit15 to bit0] TMR (TiMeR) : 16-bit timer

This register can read the counter value of the 16-bit timer. The initial value is undefined.

4.3. 16-bit Timer Reload Register A, 16-bit Timer Reload Register B : TMRLRA, TMRLRB(16bit TiMer ReLoad Register A/B)

The bit configuration of 16-bit timer reload register A and 16-bit timer reload register B is shown below.

TMRLRA sets the count initial value.

TMRLRB applies different functions according to the operation mode.

Always perform 16-bit access to this register.

■ TMRLRA : Address Base_addr + 00_H (Access : Half-word)

	bit15	bit14	....	bit2	bit1	bit0
	TMRLRA[15:0]					
Initial value	X	X	....	X	X	X
Attribute	R/W	R/W	....	R/W	R/W	R/W

■ TMRLRB : Address Base_addr + 04_H (Access : Half-word)

	bit15	bit14	....	bit2	bit1	bit0
	TMRLRB[15:0]					
Initial value	X	X	....	X	X	X
Attribute	R,W	R,W	....	R,W	R,W	R,W

[bit15 to bit0] TMRLRA (TiMer ReLoad Register A) : 16-bit reload setting register A

[bit15 to bit0] TMRLRB (TiMer ReLoad Register B) : 16-bit reload setting register B

The TMRLRA register holds the count initial value. The TMRLRA can be used in all mode regardless of the bit15, bit14:MOD[1:0] setting in the TMCSR register.

The TMRLRB is to be used based on the bit15, bit14:MOD[1:0] setting in the TMCSR register specified following:

Mode	MOD[1:0]	TMRLRB 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 output is shown in the table below.

MOD[1:0]	Mode	RELD	OUTL	TOUT output	
				H width	L width
00	Single	0	0	TMRLRA+1	-
			1	-	TMRLRA+1
		1	0	TMRLRA+1	
			1		
01	Dual	0	0	TMRLRB+1	TMRLRA+1
			1	TMRLRA+1	TMRLRB+1
		1	0	TMRLRB+1	TMRLRA+1
			1	TMRLRA+1	TMRLRB+1
10	Compare	0	0	See the explanation below.*	
			1		
		1	0		
			1		
11	Capture	0	0	TMRLRA+1	-
			1	-	TMRLRA+1
		1	0	TMRLRA+1	
			1		

*: H width and L width are as follows in the compare mode:

- When TMRLRB < TMRLRA
  (OUTL=0) "L" width of TMRLRA-TMRLRB + 1, "H" width of TMRLRB
  (OUTL=1) "H" width of TMRLRA-TMRLRB + 1, "L" width of TMRLRB

- When TMRLRB = 0
  (OUTL=0) "L" output fixed
  (OUTL=1) "H" output fixed

- When TMRLRB > TMRLRA
  (OUTL=0) "H" output fixed
  (OUTL=1) "L" output fixed

- When TMRLRB = TMRLRA
  (OUTL=0) "L" output of 1 cycle, "H" width of TMRLRB

(OUTL=1) "H" output of 1 cycle, "L" width of TMRLRB

The following formula represents the TOUT output time (TOUT) when the register is used as the single mode and dual mode in the interval timer mode:

TOUT = (Setting value of this register + 1) × count source cycle

* : The formula described above is effective only in the interval timer mode.

5. Operation

This section explains the operation of the reload timer.

5.1 Setting

5.2 Operation Procedure

5.3 Operations of Each Counter

5.4 Cascade Input

5.5 Priority of Concurrent Operations

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}).

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 5-1 List of Count Source

CSL[2:0]	Count source	Operation mode
000	Division of the peripheral clock frequency by 2 (initial value)	Interval timer mode
001	Division of the peripheral clock frequency by 4	
010	Division of the peripheral clock frequency by 8	
011	Division of the peripheral clock frequency by 16	
100	Division of the peripheral clock frequency by 32	
101	Division of the peripheral clock frequency by 64	
110	Cascade mode (ch.0:TIN0, ch.1:TOUT0, ch.2:TOUT1, ch.3:TOUT2, ch.4: TIN4, ch.5:TOUT4, ch.6:TOUT5, ch.7:TOUT6)	Event counter mode
111	External event (TIN input)	

5.1.2. Timer Underflow cycle

The timer underflow cycle is shown below.

Underflow is defined as counter down-counting from 0x0000. Set the time (cycle) to underflow occurrence since timer count operation start in the reload register (TMRLRA/TMRLRB). After loading to the reload register, underflow takes place if the count value reaches "reload register setting value + 1" count. The timer underflow cycle, TUF, in the interval timer mode can be represented as follows:

TUF = Peripheral clock (PCLK) cycle × prescaler division value (2 to 64) ×
(Reload register value (TMRLRA/B) + 1)

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 TIN pin.

The TIN pin is used as a count source in the event counter mode. Hence, a software trigger is always used.
In the interval timer mode, settings are made in the TMCSR register.

5.1.4. Gate

The gate of the reload timer is shown below.

When configuring gate input (TMCSR.GATE = "1") in the interval timer mode, it is possible to stop counter down counting using the TIN external pin.

Table 5-2 TIN Effective Level

TRGM[0]	TIN Effective Level
0	Counted only during the input period for TIN pin "L" (initial value)
1	Counted only during the input period for TIN pin "H"

5.1.5. Counter Operation Selection

The counter operation selection is shown below.

Select the operation in case of counter underflow using the mode selection bits (bit15, bit14:MOD[1:0] of the TMCSR register) and the reload operation enabling bit (bit4:RELD of the TMCSR register). For details of operation in each mode, see the section of each counter operation.

Table 5-3 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 "5.3.3Dual 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

5.1.6. TOUT Pin Level Setting

The TOUT Pin level setting is shown below.

Set pin output polarity using bit5:OUTL bit in the TMCSR register.

The relationships between events and the TOUT pin in each function are as follows:

A/B of the UF (underflow) section below indicates whether down counting underflow has occurred with a value when loading TMRLRA data or TMRLRB data. CMP (compare-match) shows the timing of down counting from TMRLRB = TMR.

Figure 5-1 TOUT Output Change in Each Event (1 / 3)

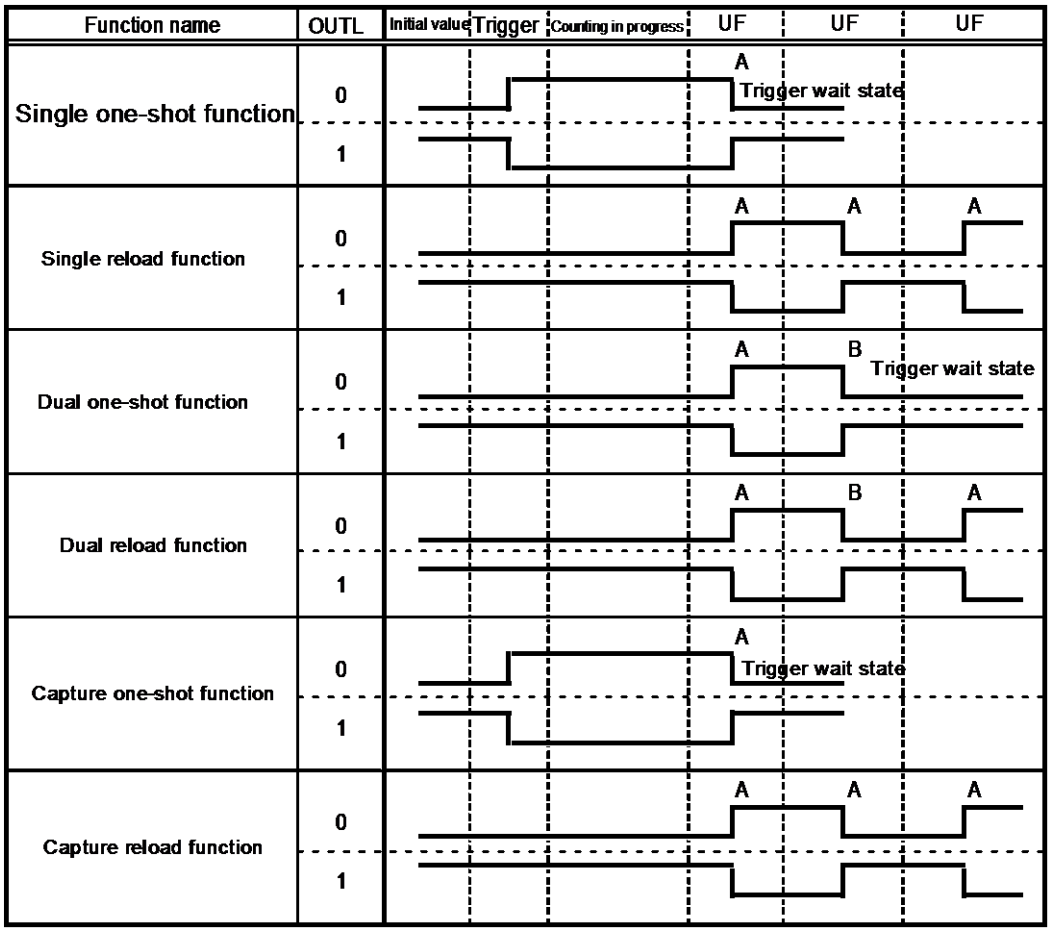


Figure 5-2 TOUT Output Change in Each Event (2 / 3)

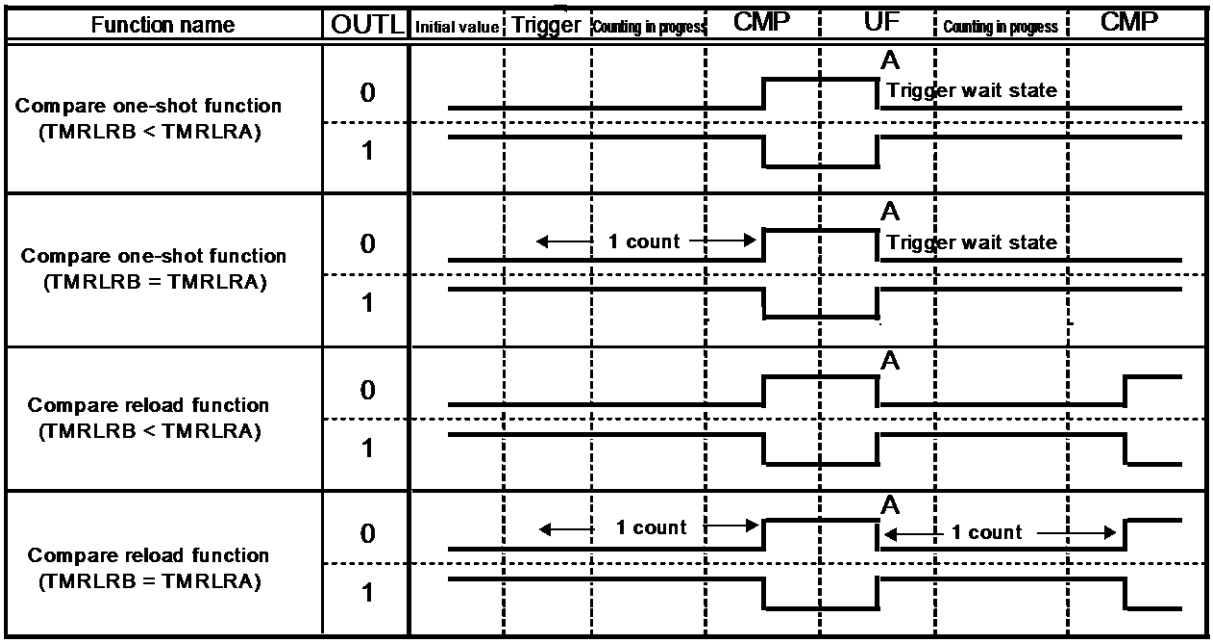
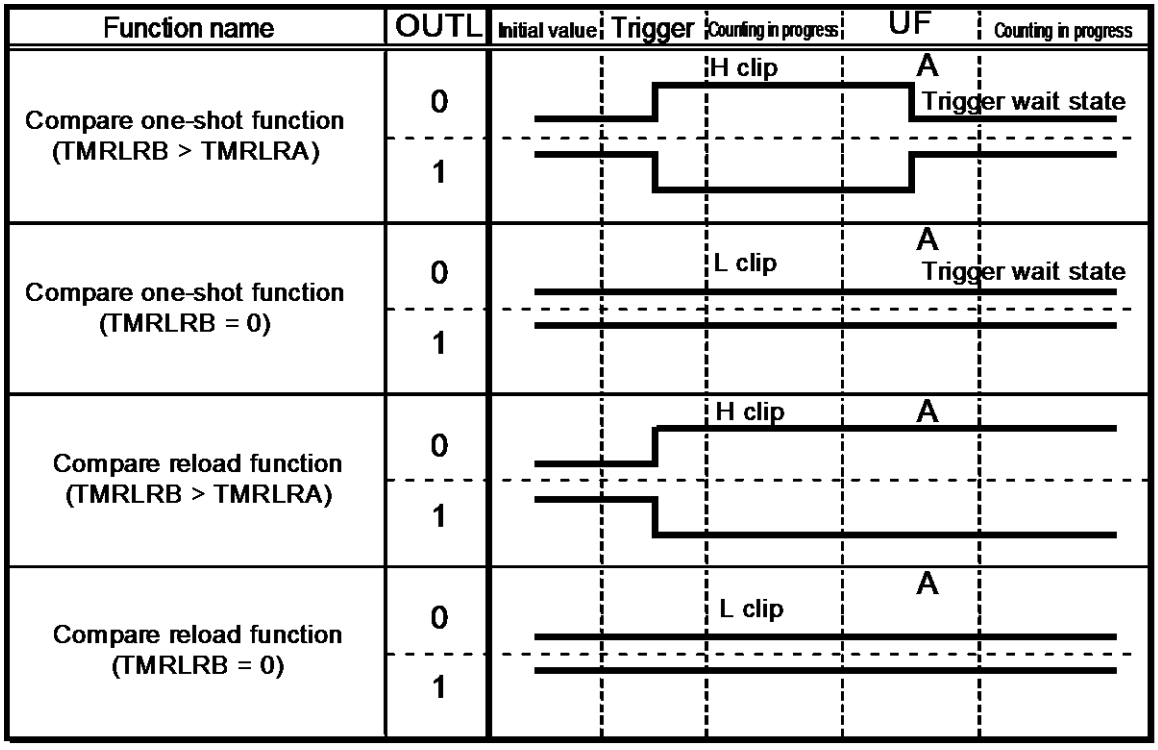


Figure 5-3 TOUT Output Change in Each Event (3 / 3)



5.2. Operation Procedure

Operation procedures are shown.

5.2.1. Activation

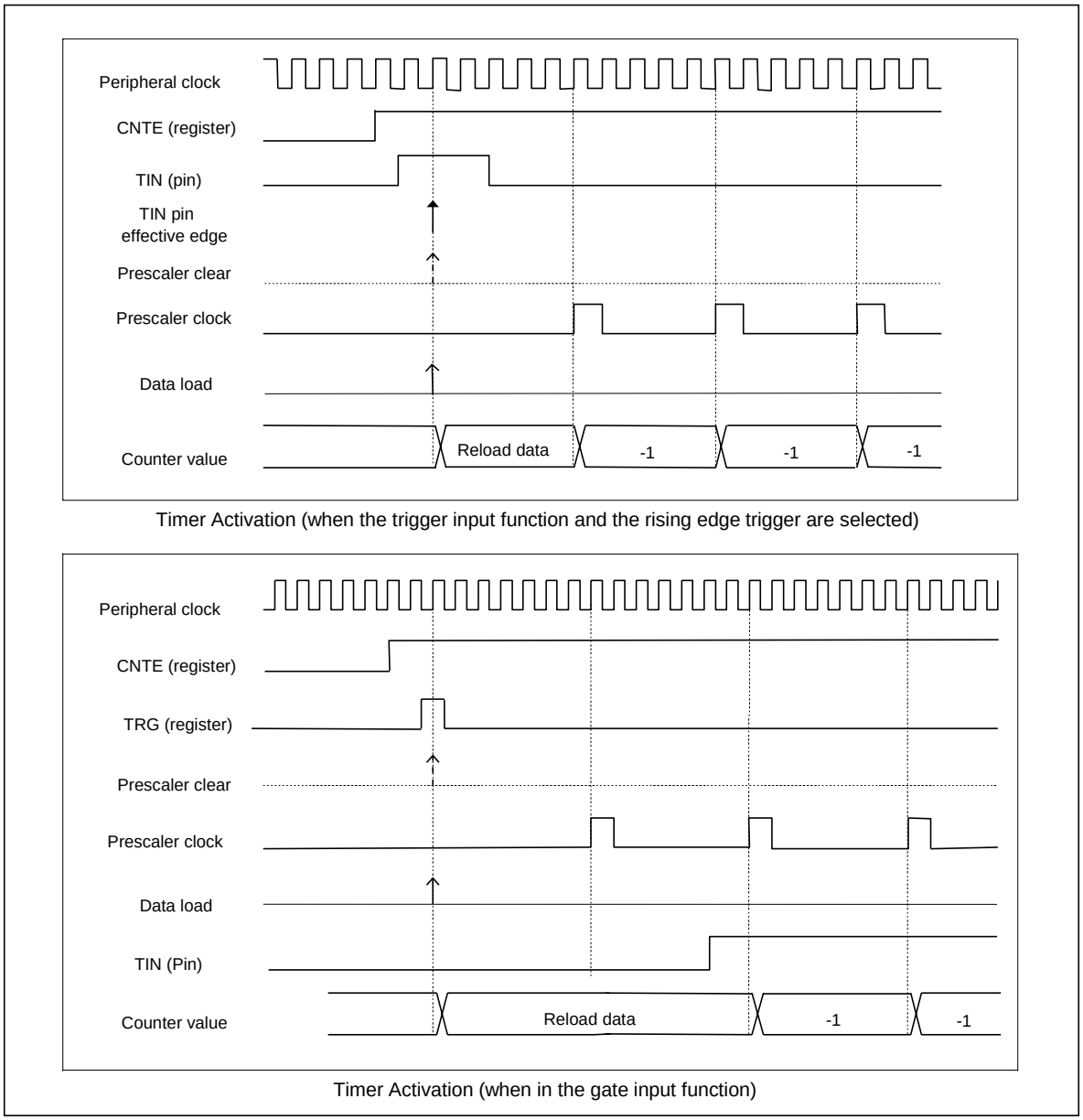
Activation is shown below.

Writing "1" into the bit1:CNTE bit of the TMCSR register changes the counter state to activation trigger waiting.

- TIN input during trigger input functioning
  If writing "1" to the bit0:TRG bit of the TMCSR register or inputting external trigger through TIN input takes place during activation trigger waiting, the prescaler will be cleared and the timer will load a value from the reload register to start down count operation. For TIN, input pulse of 2 × T (T indicates the peripheral clock (PCLK) cycle) or more.

- TIN input during gate input functioning
  If writing "1" to the bit0:TRG bit of the TMCSR register during activation trigger waiting, the prescaler will be cleared and the timer will load a value from the reload register and change the state to effective input polarity waiting. If there is any gate input with effective polarity from TIN input in the effective input polarity waiting, the timer initiates down count operation. For TIN, input pulse of 2 × T (T indicates the peripheral clock (PCLK) cycle) or more.

Figure 5-4 Timer Activation



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:

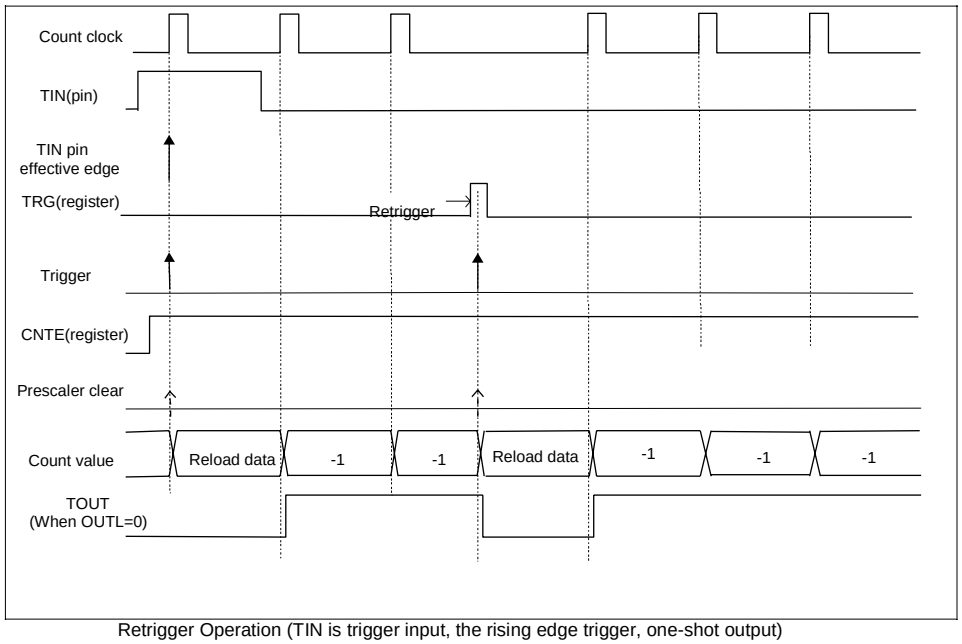
1. Initialize TOUT
2. Load the reload register value to the counter
3. Clear the 6-bit prescaler
4. Continue counting

Only in the capture mode, retrigger generation transfers a value being counted to the TMRLRB to set the EF bit of the TMCSR register.

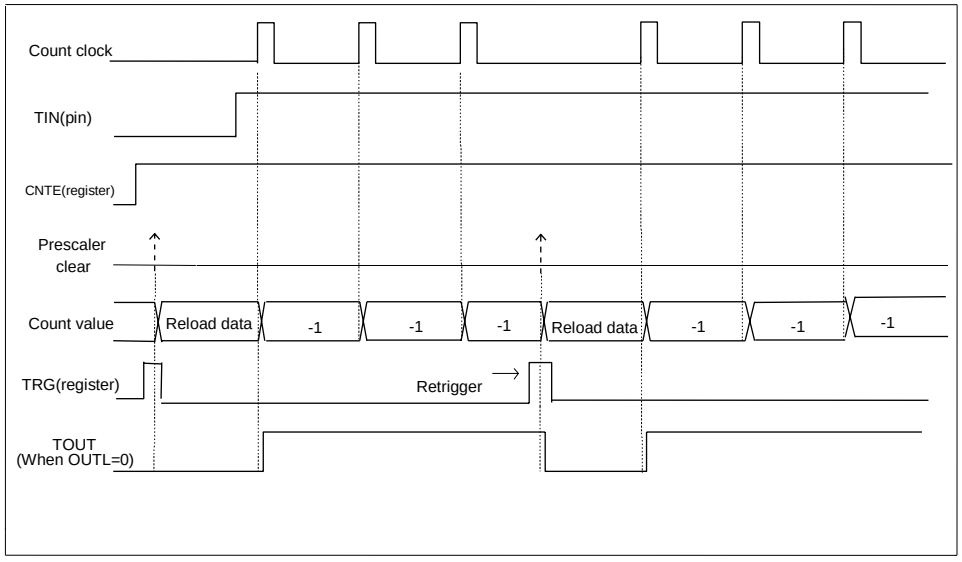
Note:

TOUT is not initialized in the one shot mode at retrigger.

Figure 5-5 Retrigger Operation



Retrigger Operation (TIN is trigger input, the rising edge trigger, one-shot output)



Retrigger Operation (TIN is gate input, count in H level, one-shot output)

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.

5.2.4. Generation of Interrupt Requests

Generation of interrupt requests is shown below.

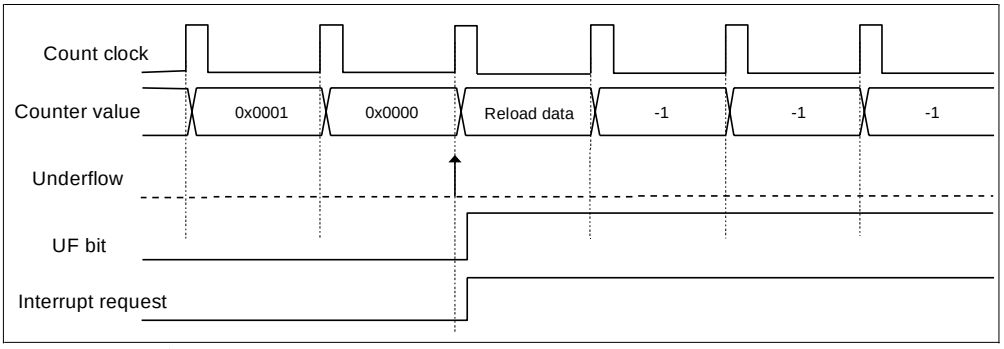
When bit3:INTE bit of the TMCSR register is "1", if bit2:UF bit/bit7:EF bit are set, an interrupt request is generated. In interval timer mode, the UF bit and the EF bit will be set under the following conditions.

- 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 5-6 Example of UF Interrupt Request Output Operation



UF interrupt request output operation (bit4:RELD= "1" and bit3:INTE="1" of TMCSR register)

5.2.5. Concurrent Operation of Register Write and Timer Activation

The concurrent operation of register write and timer activation is shown below.

The following table shows the operation when a register write by a user and the timer operation occurred simultaneously.

Table 5-4 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)

5.3. Operations of Each Counter

Operations of each counter are shown.

5.3.1. Single One-shot Operation

The single one-shot operation is shown below.

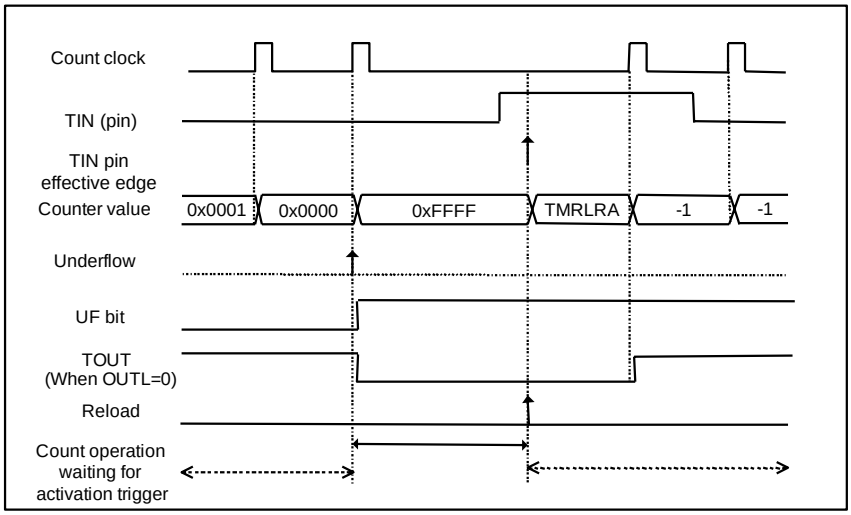
When bit15, 14: MOD[1:0] = "00" and bit4: RELD of the TMCSR register = "0", the timer performs single one-shot operation triggered by an underflow occurrence, that stops at 0xFFFF.

In the single one-shot configuration, if an underflow occurs, the following operation will be performed.

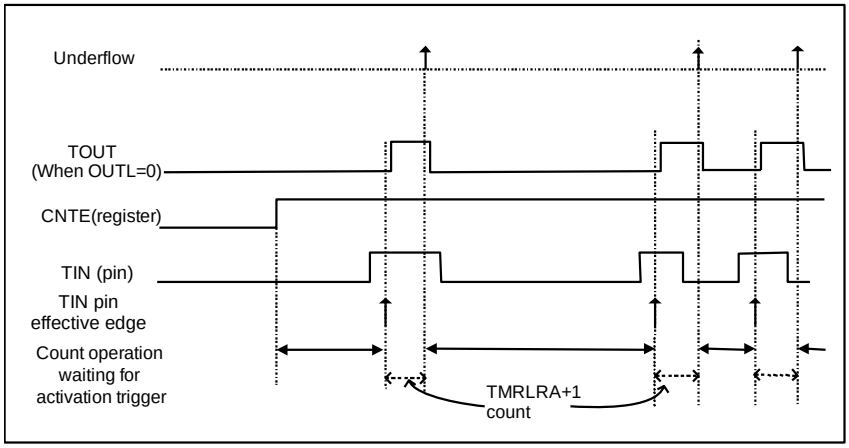
- Sets the UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Stops the count with 0xFFFF.
- Initializes TOUT output.
- Timer is waiting for a trigger.

For the single one-shot timer, TMRLRA turns to the initial value of the counter when a reload took place. TMRLRB is not used.

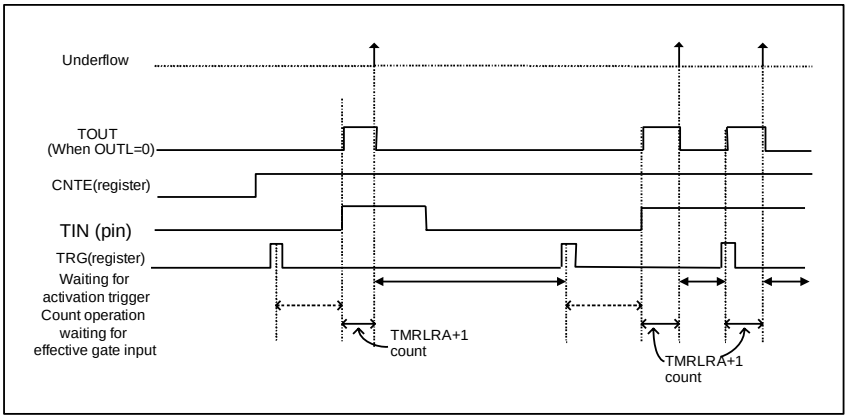
Figure 5-7 Single One-shot Operation



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)

5.3.2. Single Reload Operation

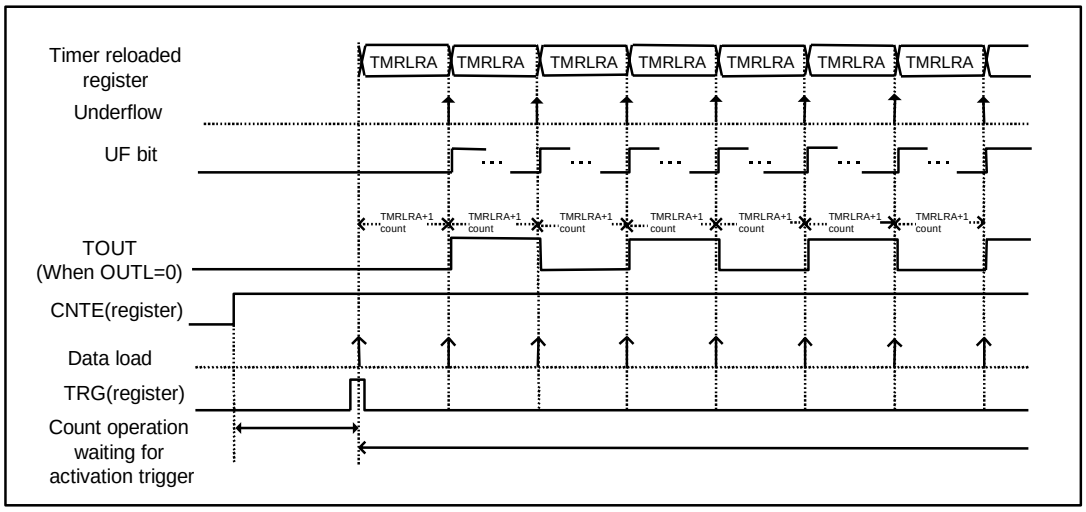
The single reload operation is shown below.

When bit15, 14: MOD[1:0] = "00" and bit4: RELD of the TMCSR register = "1", the single reload operation will be performed.

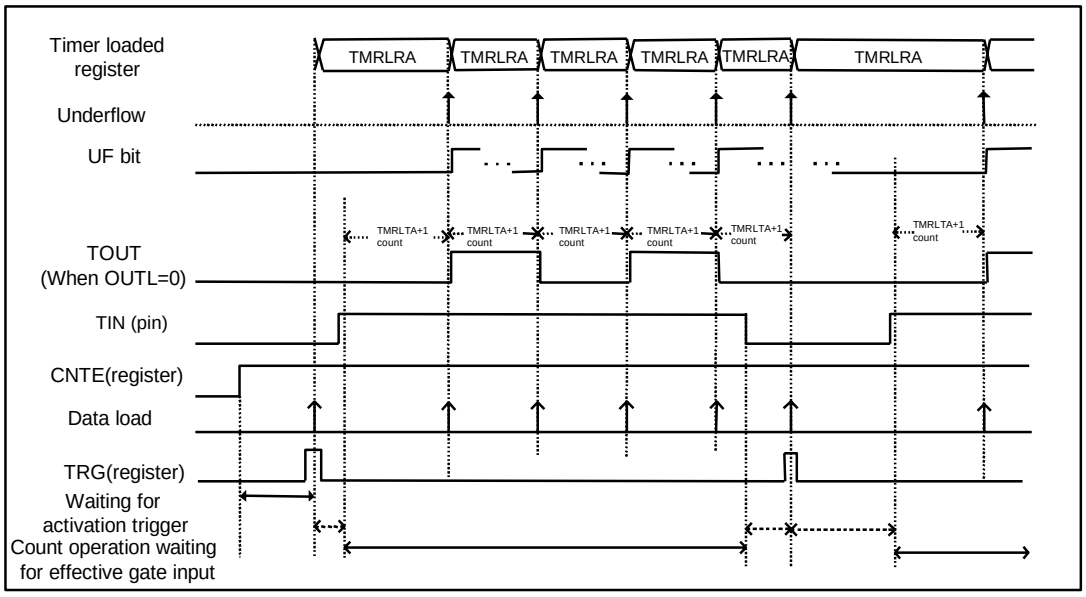
In single reload operation, a value will be loaded from TMRLRA to the timer by trigger input, a down count (decrementing the count) will start. When an underflow occurs, the value is reloaded from TMRLRA again and the down count operation continues. The value of TMRLRA represents the time the timer will 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 5-8 Single Reload Operation



Single reload function (GATE="0": trigger input)



Single reload function (GATE="1": gate input, TRGM: H input interval count)

5.3.3. Dual One-shot Operation

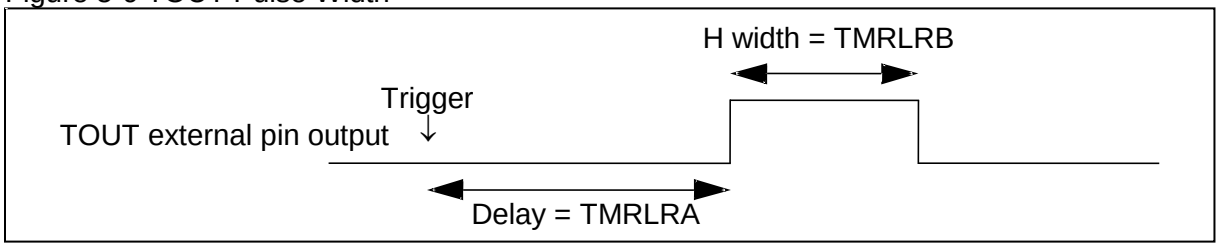
The dual one-shot operation is shown below.

When bit15, 14: MOD[1:0] = "01" and bit4: RELD 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, and the down count is executed for each of the loaded values. The counter stops when the second underflow occurs.

When bit5: OUTL of the TMCSR register = "0" , the value of TMRLRA represents the time interval between a timer activation (TOUT output is in L level) to a toggling of the TOUT output to "H", and the value of TMRLRB represents the time interval of H width of the TOUT output.

Figure 5-9 TOUT Pulse Width



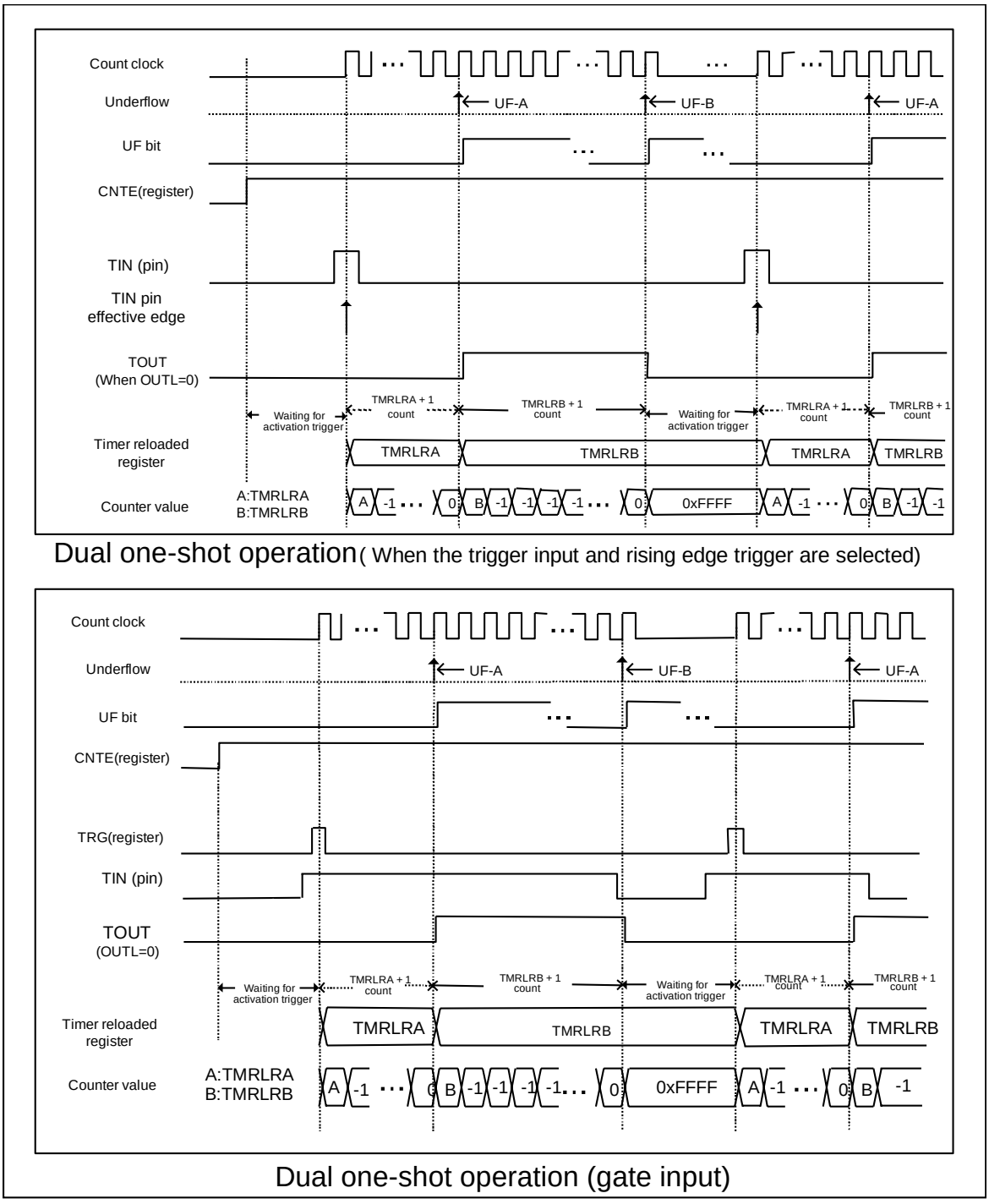
When the first underflow occurs (UF-A), the following operation will take place.

- 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 5-10 Dual One-shot Operation



Dual one-shot operation (When the trigger input and rising edge trigger are selected)

Dual one-shot operation (gate input)

5.3.4. Dual Reload Operation

The dual one-shot operation is shown below.

When bit15, 14: MOD[1:0] = "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 of the TMCSR register = "0", the value of TMRLRA represents the time interval between a timer activation (TOUT output is in L level) to a toggling of the TOUT output to "H", and the value of TMRLRB represents the time interval of H width of the TOUT output.

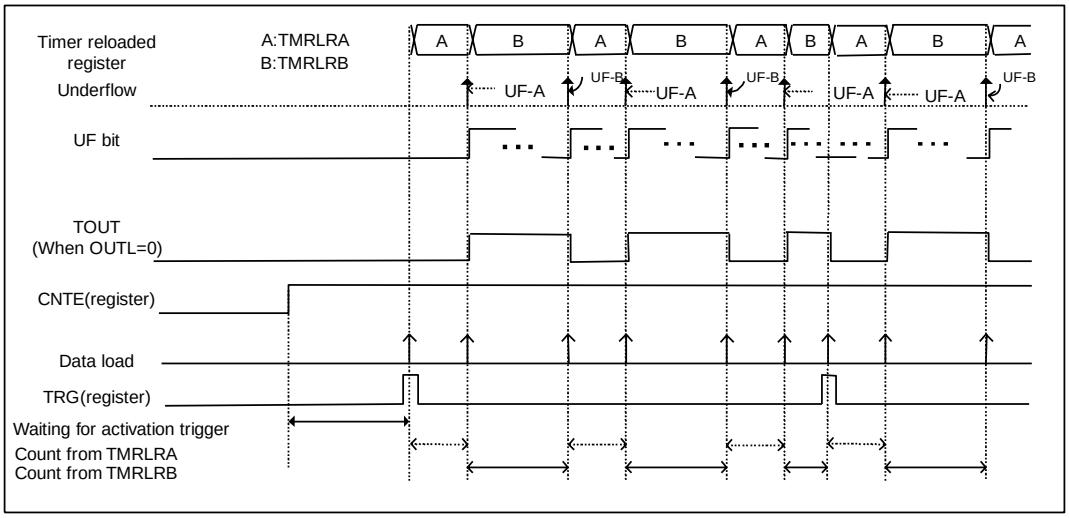
If an underflow (UF-A) occurs at the down count after loading a value from the TMRLRA, the following operation will be performed.

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRB to the counter.
- Inverts TOUT output.
- Starts a down count from TMRLRB.

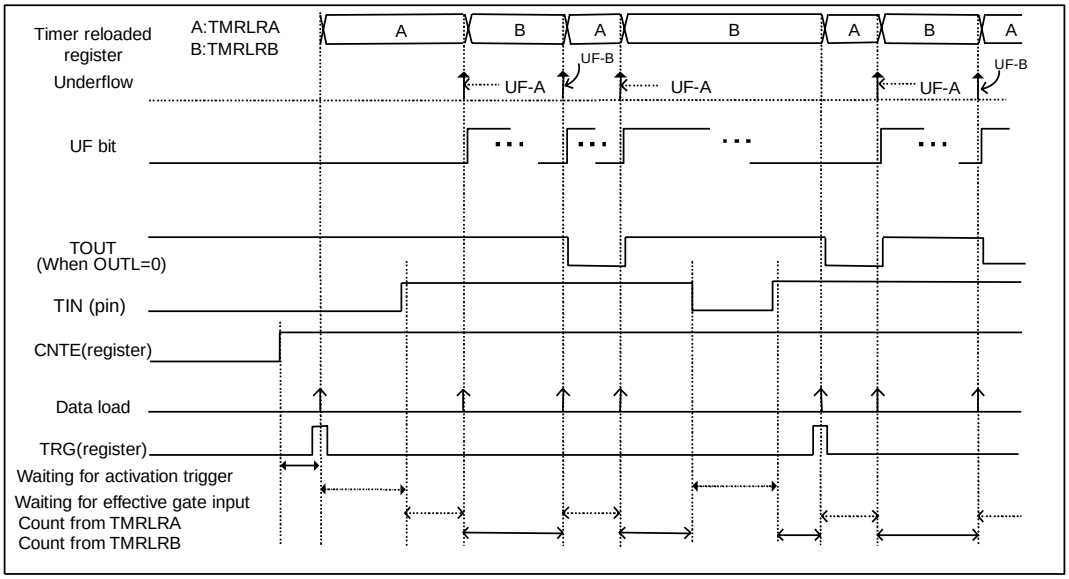
If an underflow (UF-B) occurs at the down count after loading a value from the TMRLRB, the following operation will be performed.

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRA to the counter.
- Inverts TOUT output.
- Starts a down count from TMRLRA.

Figure 5-11 Dual Reload Operation



Dual Reload function (GATE= "0" : trigger input)



Dual Reload function (GATE= "1" : gate input, H input interval count)

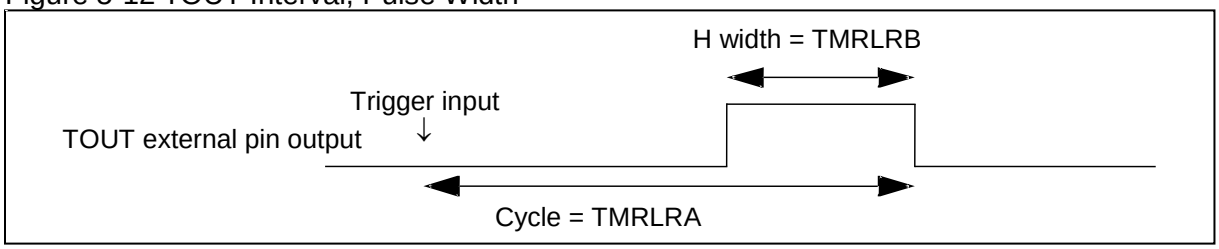
5.3.5. Compare One-shot Operation

The compare one-shot operation is shown below.

When bit15, 14: MOD[1:0] = "10" and bit4: RELD in 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 compare match (TMR = TMRLRB), the TOUT output will be inverted. When an underflow occurs, count operations stopped, TOUT output is initialized, and the timer go into the activation trigger wait state.

The value of TMRLRA indicates the time interval between the activation of a timer and the end of it and the value of TMRLRB indicates the counter value when an output of the H width of TOUT output starts. When OUTL="0" and TMR < TMRLRB, the TOUT output will become the "H level".

Figure 5-12 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.

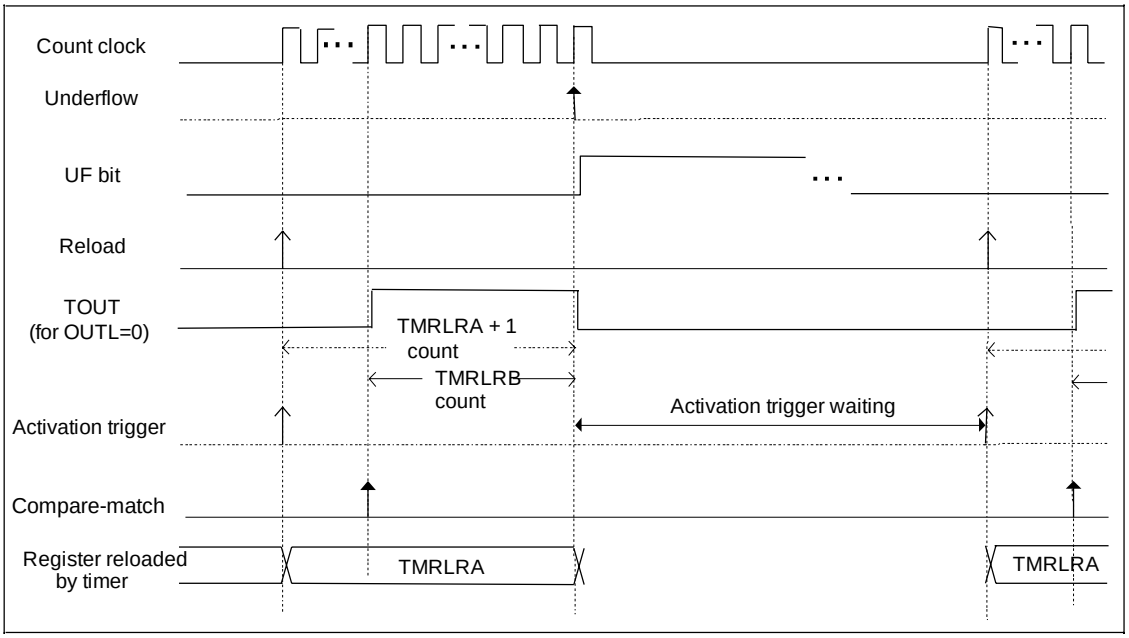
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- The timer stops with 0xFFFF.
- Timer is waiting for an activation trigger.

The operation of the compare function changes depending on the setting relation between TMRLRA and TMRLRB.

Figure 5-13 Compare One-shot Operation (1 / 2)

• Sets TMRLRB < TMRLRA

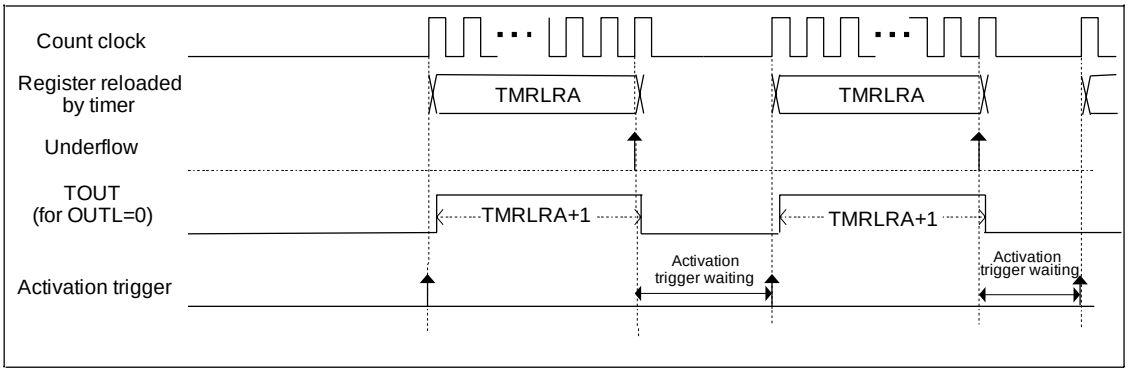
When the register relation is as described above, the TOUT output is the "L" level until TMR and TMRLRB match after loading to the timer. When down counting from the compare match (TMR=TMRLRB), the TOUT output is "H" level until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT output will be initialized. Then, the timer will stop counting operation and turn into the activation trigger waiting state (for OUTL="0").



Compare one-shot function (TMRLRB < TMRLRA)

• Sets TMRLRB > TMRLRA

When the register relation is as described above, the TOUT output is the "H" level between an activation trigger generation and an underflow occurrence because TMR is already smaller than TMRLRB after loading to the timer. When an underflow occurs, the timer will turn into the activation trigger waiting state and the TOUT output will be the "L" level (for OUTL="0").

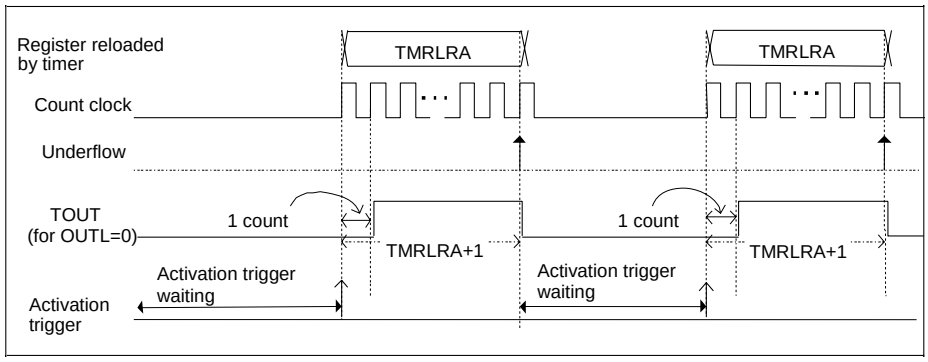


Compare one-shot function (TMRLRB > TMRLRA)

Figure 5-14 Compare One-shot Operation (2 / 2)

• Sets TMRLRB = TMRLRA

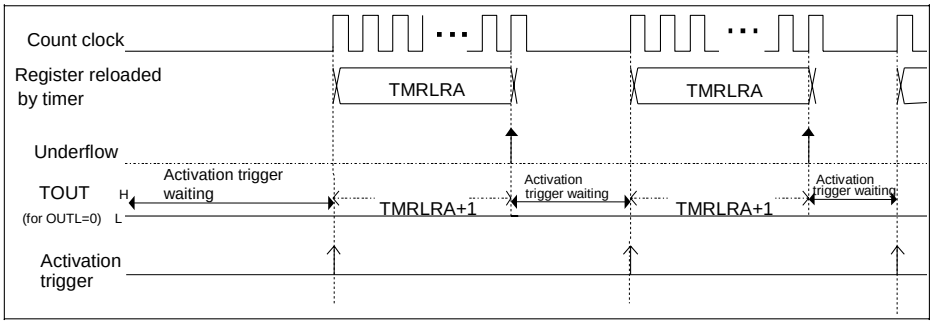
When the register relation is as described above, After loading to the timer, TMR will become smaller than TMRLRB after 1 count. Thus the TOUT output is the "L" level for 1 down count and then the "H" level until an underflow occurs. When an underflow occurs, the timer will turn into the activation trigger waiting state and the TOUT output will be the "L" level (for OUTL="0").



Compare one-shot function (TMRLRB=TMRLRA)

• Sets TMRLRB = 0

When the register relation is as described above, the TOUT output is the "L" level between down count start and an underflow occurrence because TMR will not become smaller than TMRLRB. The level will remain to be "L" even when an underflow occurs (for OUTL="0").



Compare one-shot function (TMRLRB="0")

5.3.6. Compare Reload Operation

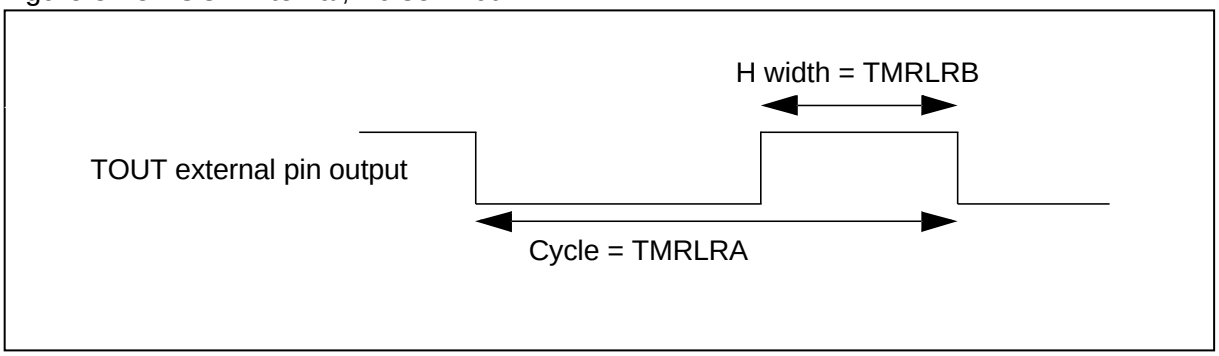
The compare reload operation is shown below.

When bit15, 14: MOD[1:0] = "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 match (TMR = TMRLRB) is detected, a down count starts and the TOUT output will be inverted. When an underflow occurs, the compare reload operation will be performed, in which a value is loaded from TMRLRA again and the down count operation starts. A load onto the counter starts from TMRLRA.

The value of TMRLRA indicates the counter interval from a timer activation until a reload and the value of TMRLRB indicates the "H level width" after the TOUT output inverted from "L level output" to "H level output".

When TMR + 1 = TMRLRB, TOUT output will invert to the "H level" (when OUTL= "0").

Figure 5-15 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.
- 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.

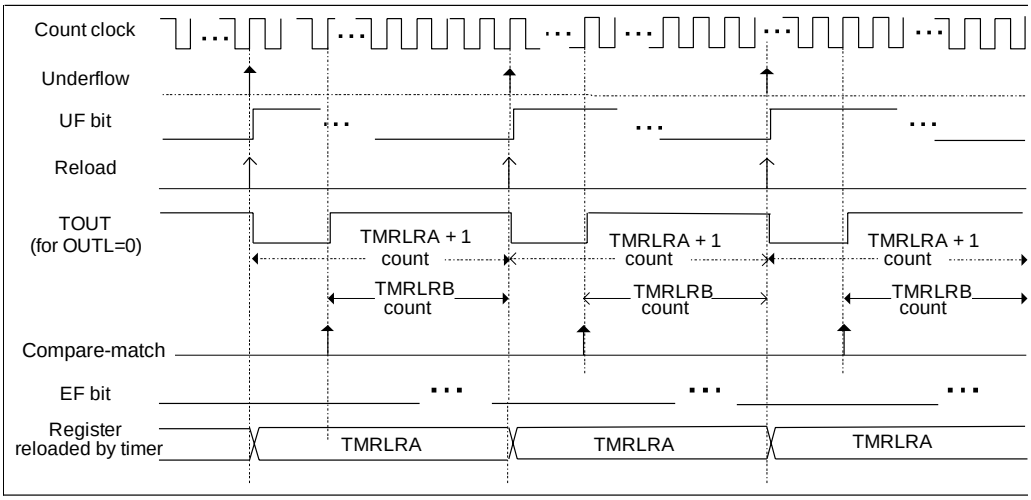
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- Reloads a value from TMRLRA.
- The timer continues to count.

The operation of a compare feature depends on the relationship between TMRLRA and TMRLRB.

Figure 5-16 Compare Reload Operation (1 / 2)

• Sets TMRLRB < TMRLRA

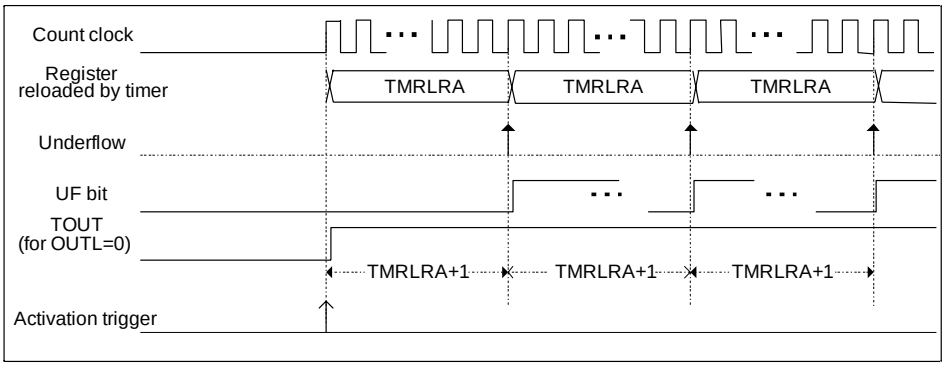
When the register relation is as described above, the TOUT output is the "L" level until TMR and TMRLRB match after loading to the timer. When down counting from the compare match (TMR=TMRLRB), the TOUT output is "H" level until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT output will be initialized. When an under flow occurs, the timer will reload from TMRLRA and continue counting operation (for OUTL="0").



Compare reload function (TMRLRB < TMRLRA) trigger input

• Sets TMRLRB > TMRLRA

When the register relation is as described above, the TOUT output is the "H" level after an activation trigger is generated and an underflow occurs because TMR is always smaller than TMRLRB. The level will remain to be "H" even when an underflow occurs. When an underflow occurs, the timer will load from TMRLRA and continue counting operation (for OUTL="0").

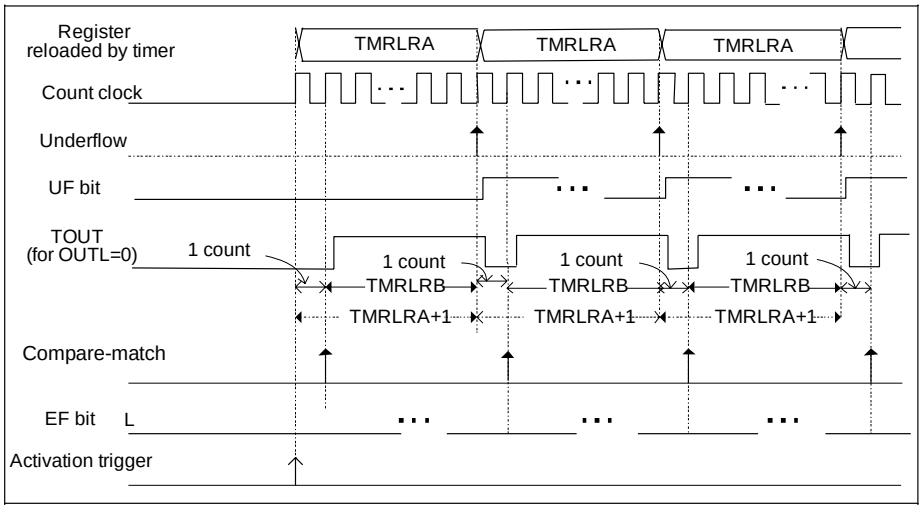


Compare reload function (TMRLRB > TMRLRA) trigger input

Figure 5-17 Compare Reload Operation (2 / 2)

• Sets TMRLRB = TMRLRA

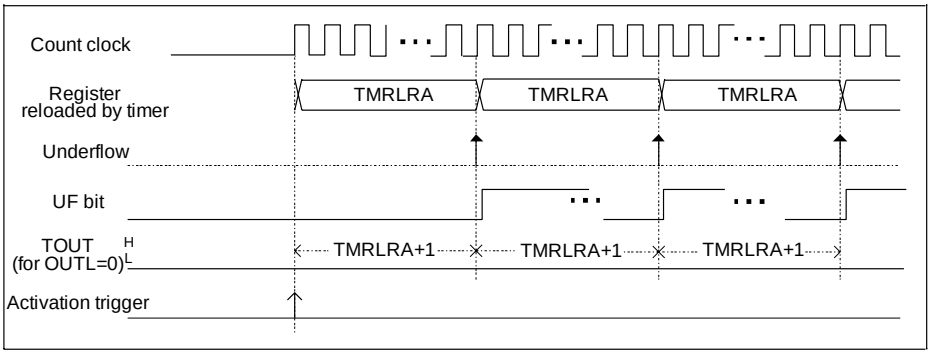
When the register relation is as described above, 1 count up after loading to the timer makes TMR become smaller than TMRLRB. Thus the TOUT output is the "L" level for 1 down count and then the "H" level until an underflow occurs. When an underflow occurs, the timer will reload from TMRLRA and continue counting operation. The TOUT output will remain to be the L level. (For OUTL= "0")



Compare reload function (TMRLRB = TMRLRA) trigger input

• Sets TMRLRB = 0

When the register relation is as described above, the TOUT output is the "L" level between down count start and an underflow occurrence after loading to the timer because TMR will not become smaller than TMRLRB. The level will remain to be "L" even when an underflow occurs.



Compare reload function (TMRLRB = "0") trigger input

5.3.7. Capture Mode

The capture mode is shown below.

When bit15, 14: 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 TIN input as the gate input (when bit8:GATE= "1" of the TMCSR register), generate a retrigger by bit0:TRG of the TMCSR register.

In a mode other than capture, a capture will not be performed at a retrigger. The EF interrupt will also not be generated.

The timer operation and the TOUT output will be the same for the single one-shot feature and the single reload feature.

Note:

TOUT is not initialized in the one shot mode at retrigger.

Figure 5-18 Operation of Capture

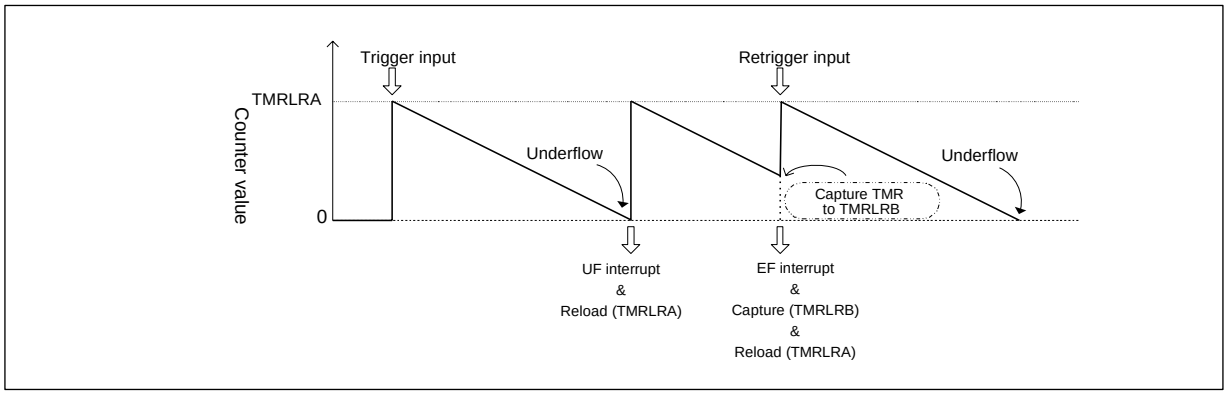


Figure 5-19 Flowchart of Trigger Input Features in Interval Timer Mode

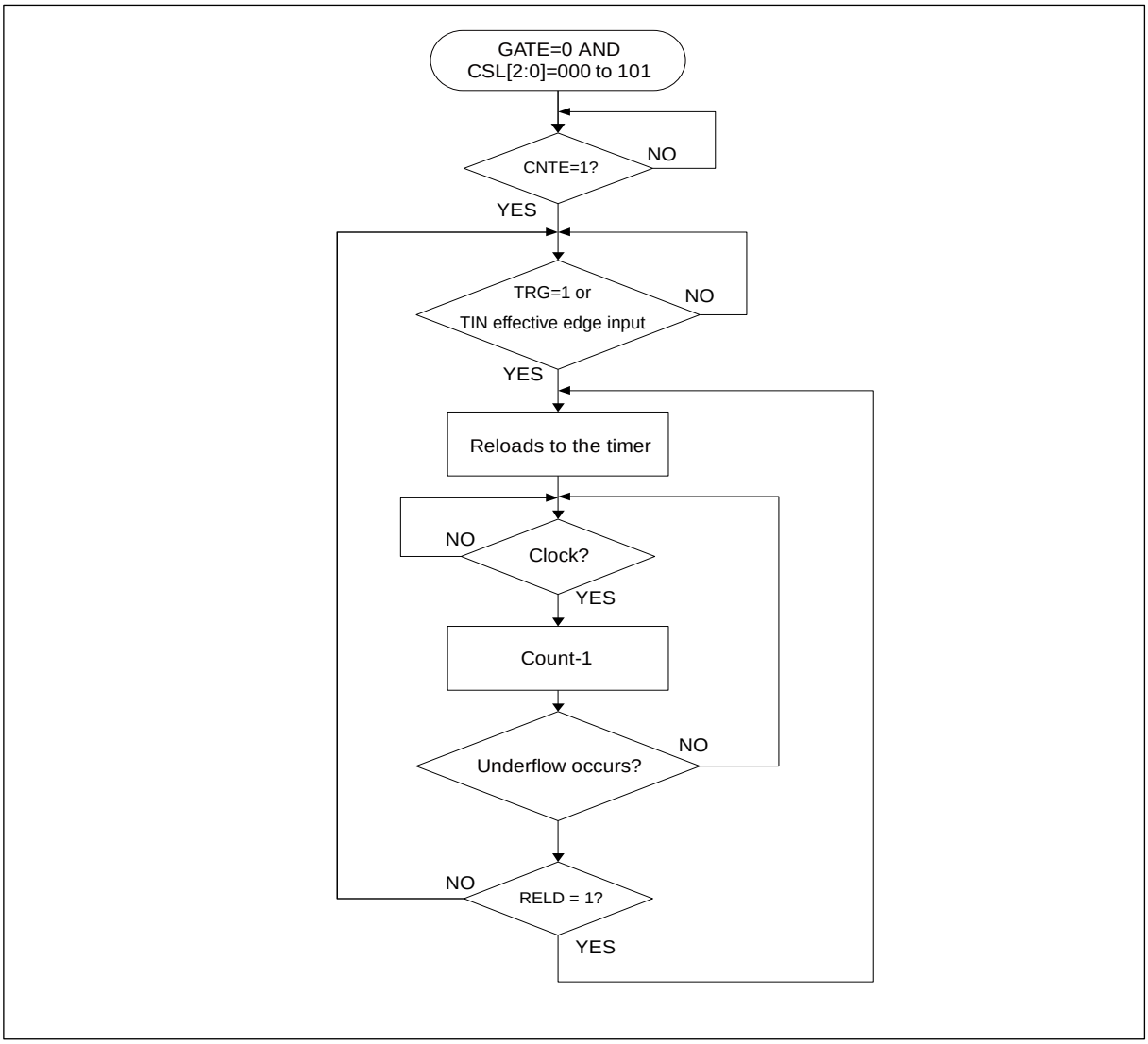
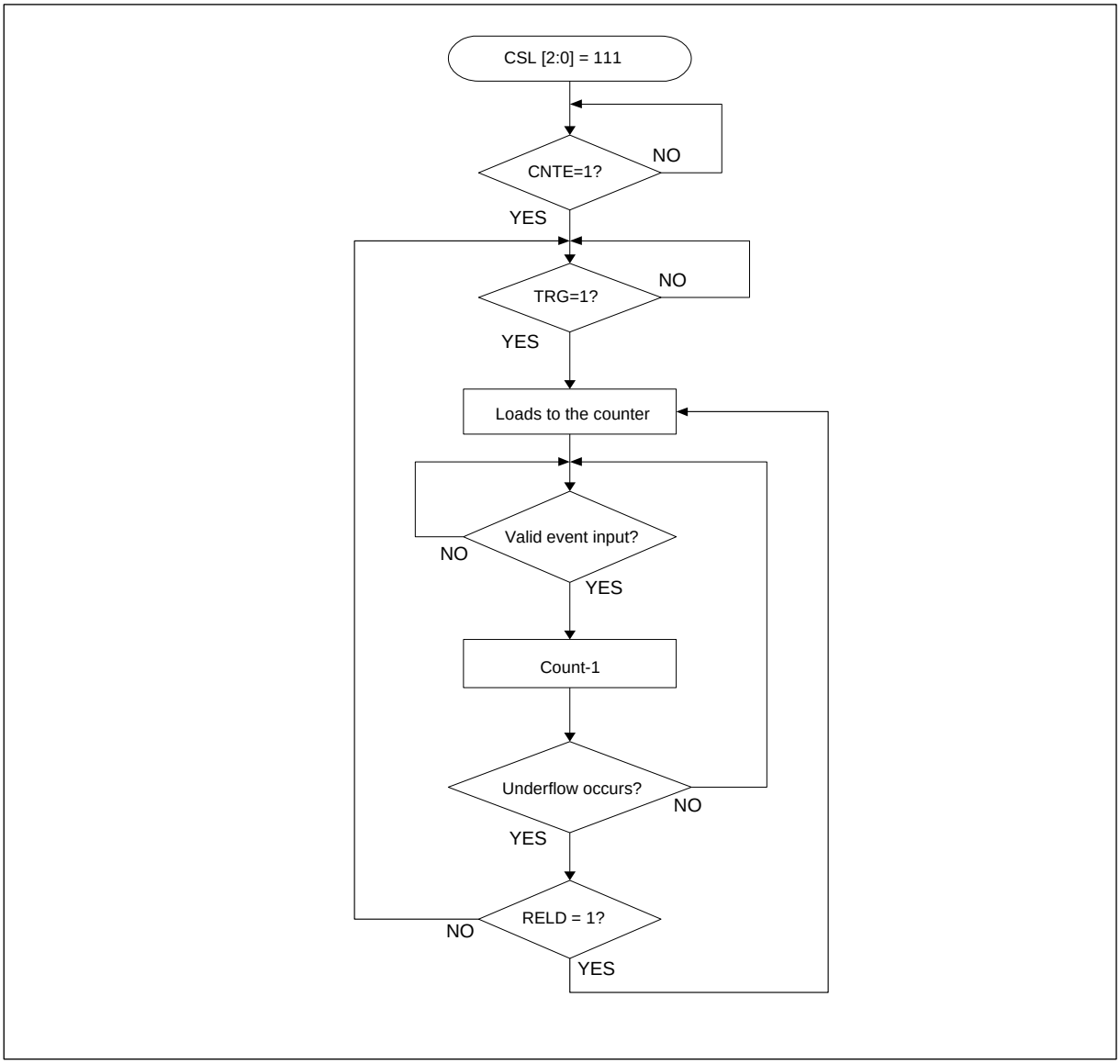


Figure 5-20 Flowchart in Event Counter Mode



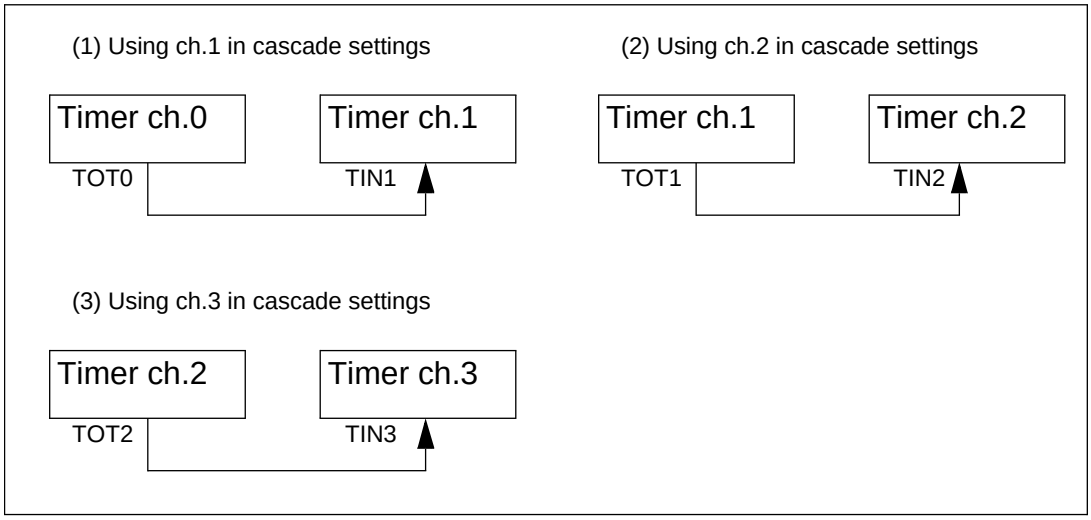
5.4. Cascade Input

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 (TIN1), ch.1 output (TOUT1) for the input for ch.2 (TIN2), and ch.2 output (TOUT2) for the input for ch.3 (TIN3).

ch.4 to ch.7 are also similar to the above.

Figure 5-21 Timer Input/Output in Cascade Input Configuration



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.

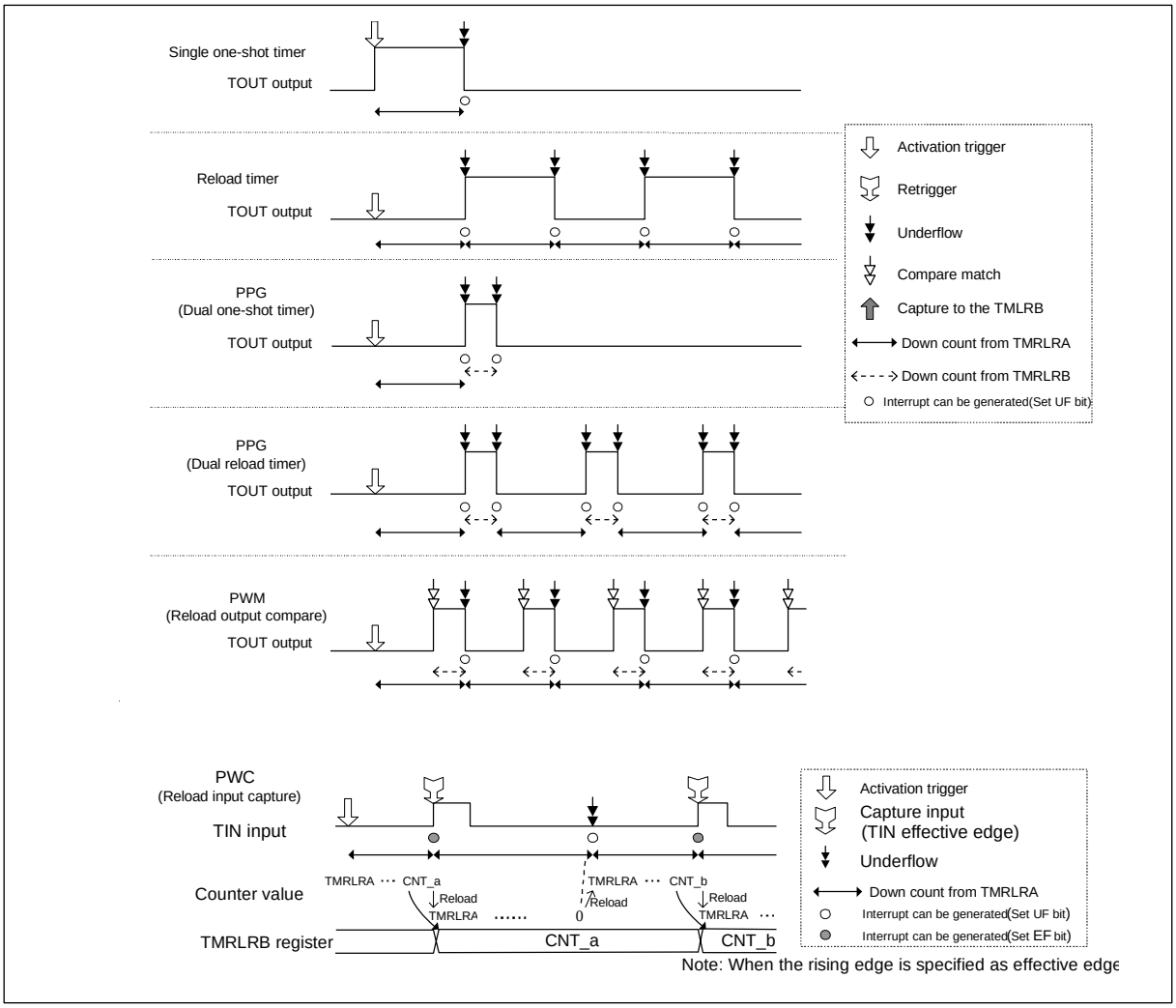
1. Setting flag by the timer operation
2. Writing to a register for a clear of flag to the UF bit/EF bit

6. Application Note

This section explains an application note concerning the register for the reload timer.

This section shows the typical functions which can be realized with this timer.

Figure 6-1 Example



Following are some configurations for use of example figure above.

Table 6-1 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	-

6.1. Single One-shot Timer

The single one-shot timer is shown below.

The single one-shot timer loads a value from the TMRLRA register onto the counter and starts to decrement the counter (down count operation) when a trigger is input. When an underflow occurs, the counting stops.

The TOUT pin outputs the "H level" in counting and when an underflow occurs it will output the "L level". (When OUTL= "0")

[Configuration] To use this timer as a single one-shot timer, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	00	*1	0	-	*2	0	*3	-	1	S	

S :Use at timer activation
-:Does not influence operation
*1:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*2:TOUT output polarity setting
OUTL=0------Initial value L=> Count starts H=> Underflow occurs L
OUTL=1------Initial value H=> Count starts L=> Underflow occurs H
*3:Interrupt request enable setting

INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	1	-	*3	0	*4	-	1	S	

S :Use at timer activation
-:Does not influence operation
*1: TIN effective level setting
TRGM[1:0]=x0------Count only for L input interval
TRGM[1:0]=x1------Count only for H input interval
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*3:TOUT output polarity setting
OUTL= 0------Initial value L=> Count starts H=> Underflow occurs L
OUTL= 1------Initial value H=> Count starts L=> Underflow occurs H
*4:Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

3. When using TIN input as a trigger input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	0	-	*3	0	*4	-	1	S	

S :Use at timer activation
-:Does not influence operation
*1: TIN effective level setting
TRGM[1:0]= 00------Does not detect external trigger edge
TRGM[1:0]= 01------Rising edge
TRGM[1:0]= 10------Falling edge
TRGM[1:0]= 11------Both edges
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3:OUT output polarity setting

OUTL= 0------Initial value L=> Count starts H=> Underflow occurs L
OUTL= 1------Initial value H=> Count starts L=> Underflow occurs H

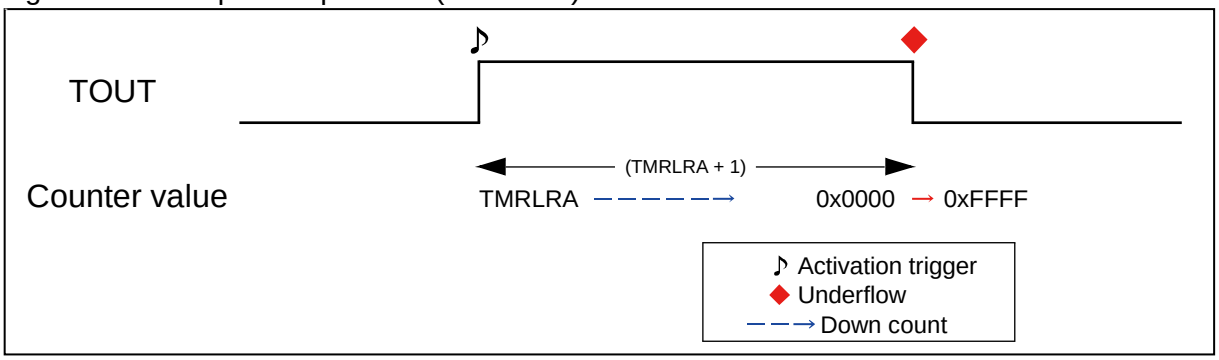
*4:Interrupt request enable setting

INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)
- Input an effective level when you use TIN pin input as the gate input

Figure 6-2 Example of Operation (OUTL = 0)



6.2. Reload Timer

The reload timer is shown below.

The reload timer loads from the TMRLRA register onto the counter and repeats the down count operation each time underflow occurs. The TOUT outputs the "L level" while the count is ongoing from the activation trigger to the occurrence of the first underflow. The output is inverted each time an underflow occurs, and the TOUT outputs "H level" with the occurrence of the first underflow. When a retrigger occurs, TOUT output returns to its initial value. (When OUTL= "0")

[Configuration] To use the timer as the reload timer, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	00	*1	0	-	*2	1	*3	-	1	S	

S :Use at timer activation

-:Does not influence operation
*1:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*2:TOUT output polarity setting
OUTL=0------Initial value L=> Count starts L=> Invert whenever an underflow occurs
OUTL=1------Initial value H=> Count starts H=> Invert whenever an underflow occurs
*3:Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	1	-	*3	1	*4	-	1	S	

S :Use at timer activation
-:Does not influence operation
*1: TIN effective level setting
TRGM[1:0]=x0------Count only for TIN=L input interval
TRGM[1:0]=x1------Count only for TIN=H input interval
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*3:OUT output polarity setting
OUTL=0------Initial value L=> Count starts L=> Invert whenever an underflow occurs
OUTL=1------Initial value H=> Count starts H=> Invert whenever an underflow occurs
*4:Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

3. When using TIN input as a trigger input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	0	-	*3	1	*4	-	1	S	

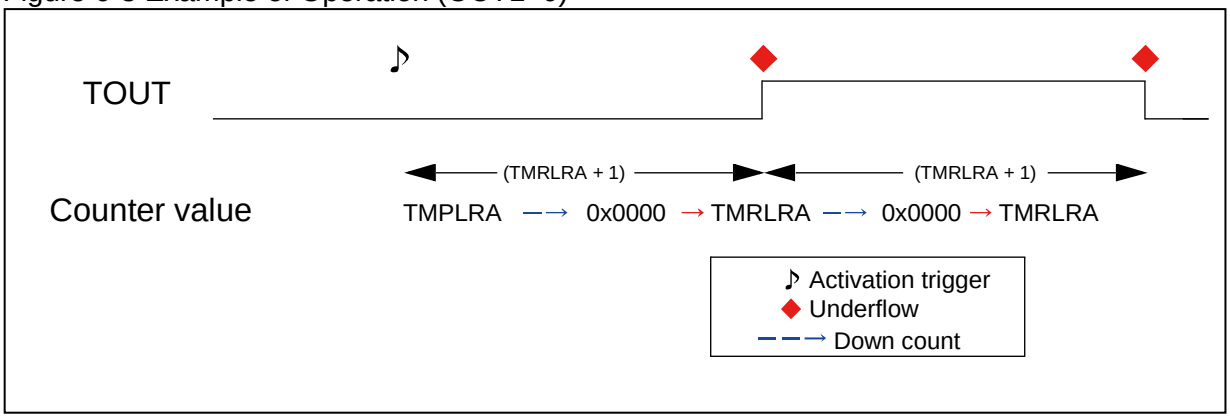
S :Use at timer activation

-:Does not influence operation
*1: TIN effective edge setting
TRGM[1:0]= 00------Does not detect external trigger edge
TRGM[1:0]= 01------Rising edge
TRGM[1:0]= 10------Falling edge
TRGM[1:0]= 11------Both edges
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*3:OUT output polarity setting
OUTL= 0------Initial value L=> Count starts L=> Invert whenever an underflow occurs
OUTL= 1------Initial value H=> Count starts H=> Invert whenever an underflow occurs
*4:Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)
- Input an effective level when you use TIN pin input as the gate input

Figure 6-3 Example of Operation (OUTL=0)



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 TIN input is not used

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	00	*1	0	-	*2	*3	*4	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)
(B): The reload value at an underflow caused by the count from the TMRLRA value
S :Use at timer activation
-:Does not influence operation
* 1:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
* 2:TOUT output polarity setting
OUTL= 0------
Initial value L => Count L from TMRLRA => H when an underflow occurs => Count H from TMRLRB => L when an underflow occurs
OUTL= 1------

Initial value H => Count H from TMRLRA => L when an underflow occurs =>
Count L from TMRLRB => H when an underflow occurs

*3:Reload setting when an underflow occurs
RELD= 0------One-shot mode
RELD= 1------Reload mode
*4:Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	*1	*2	1	-	*3	*4	*5	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)
(B): The reload value at an underflow caused by the count from the TMRLRA value
S :Use at timer activation
-:Does not influence operation
* 1: TIN effective level setting
TRGM[1:0]= x0------Count only for TIN=L input interval
TRGM[1:0]= x1------Count only for TIN=H input interval
* 2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
* 3:TOUT output polarity setting
OUTL= 0------
Initial value L=> Count L from TMRLRA => H when an underflow occurs =>
Count H from TMRLRB => L when an underflow occurs
OUTL= 1------
Initial value H=> Count H from TMRLRA => L when an underflow occurs =>
Count L from TMRLRB => H when an underflow occurs
*4:Reload setting when an underflow occurs
RELD=0------One-shot mode
RELD=1------Reload mode
*5:Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

3. When using TIN input as a trigger input

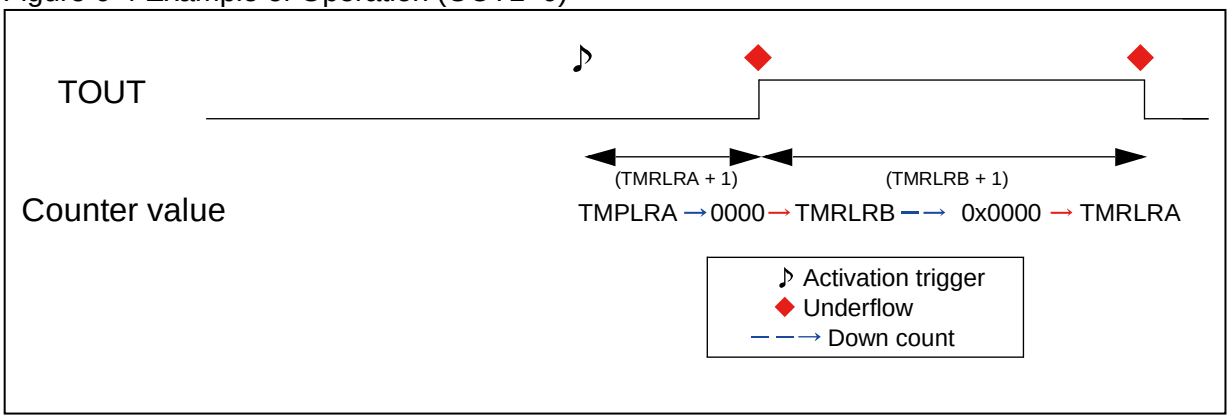
TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)
(B): The reload value at an underflow caused by the count from the TMRLRA value
S :Use at timer activation
-:Does not influence operation
*1: TIN effective edge setting
TRGM[1:0]= 00------Does not detect external trigger edge
TRGM[1:0]= 01------Rising edge
TRGM[1:0]= 10------Falling edge
TRGM[1:0]= 11------Both edges
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*3:TOUT output polarity setting
OUTL= 0------Initial value L=> Count L from TMRLRA => Invert whenever an underflow occurs
OUTL= 1------Initial value H=> Count H from TMRLRA => Invert whenever an underflow occurs
*4:Reload setting when an underflow occurs
RELD=0------One-shot mode
RELD=1------Reload mode
*5:Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)
- Input an effective level when you use TIN pin input as the gate input

Figure 6-4 Example of Operation (OUTL=0)



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 TIN input is not used

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	0	*1	0	-	*2	*3	*4	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)
(B): Set the value to compare to the counter value (TMRLRB < TMRLRA) *5
S :Use at timer activation
-:Does not influence operation
*1:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*2:TOUT output polarity setting
OUTL= 0------Initial value L=> Count L from TMRLRA => H, the counter value is smaller than TMRLRB
OUTL= 1------Initial value H=> Count H from TMRLRA => L, the counter value is smaller than TMRLRB

*3:Reload setting when an underflow occurs
RELD= 0------One-shot mode
RELD= 1------Reload mode

*4:Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

*5:To use TOUT output with L clip output, set to TMRLRB = "0".
To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

2. When using TIN input as a gate input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	*1	*2	1	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)
(B): Set the value to compare to the counter value(TMRLRB < TMRLRA) *6
S :Use at timer activation
-:Does not influence operation

*1: TIN effective level setting
TRGM[1:0]= x0------Count only for TRGM=L input interval
TRGM[1:0]= x1------Count only for TRGM=H input interval

*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3:TOUT output polarity setting
OUTL= 0------Initial value L=> Count L from TMRLRA => H, the counter value is smaller than TMRLRB
OUTL= 1------Initial value H=> Count H from TMRLRA => L, the counter value is smaller than TMRLRB

*4:Reload setting when an underflow occurs
RELD=0------One-shot mode
RELD=1------Reload mode

*5:Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

*6:To use TOUT output with L clip output, set to TMRLRB = "0".
To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

3. When using TIN input as a trigger input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)

(B): Set the value to compare to the counter value (TMRLRB < TMRLRA) *6

S :Use at timer activation

-:Does not influence operation

*1: TIN effective edge setting

TRGM[1:0]= 00------Does not detect external trigger edge
TRGM[1:0]= 01------Rising edge
TRGM[1:0]= 10------Falling edge
TRGM[1:0]= 11------Both edges

*2:Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3:TOUT output polarity setting

OUTL= 0------Initial value L=> Count L from TMRLRA => H, the counter value is smaller than TMRLRB
OUTL= 1------Initial value H=> Count H from TMRLRA => L, the counter value is smaller than TMRLRB

*4:Reload setting when an underflow occurs

RELD= 0------One-shot mode
RELD= 1------Reload mode

*5:Interrupt request enable setting

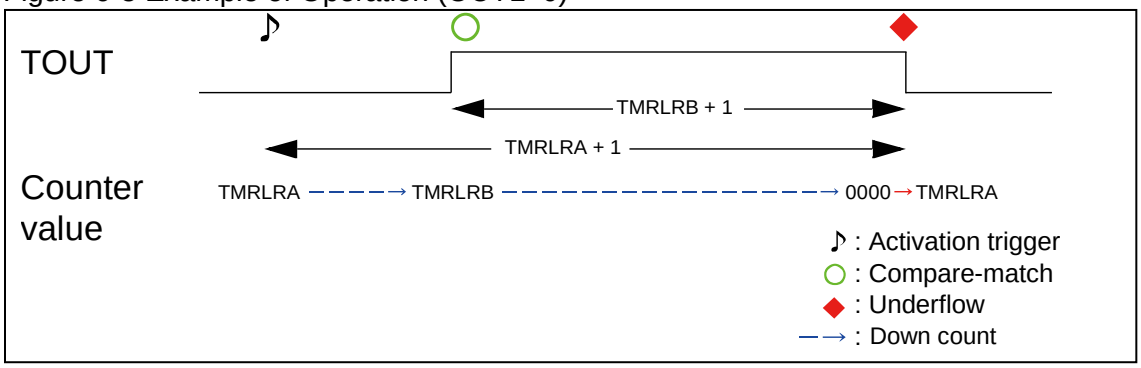
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

*6:To use TOUT output with L clip output, set to TMRLRB = "0".
To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to TRG bit or an input of effective external edge from TIN pin)
- Input an effective level when you use TIN pin input as the gate input

Figure 6-5 Example of Operation (OUTL=0)



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		

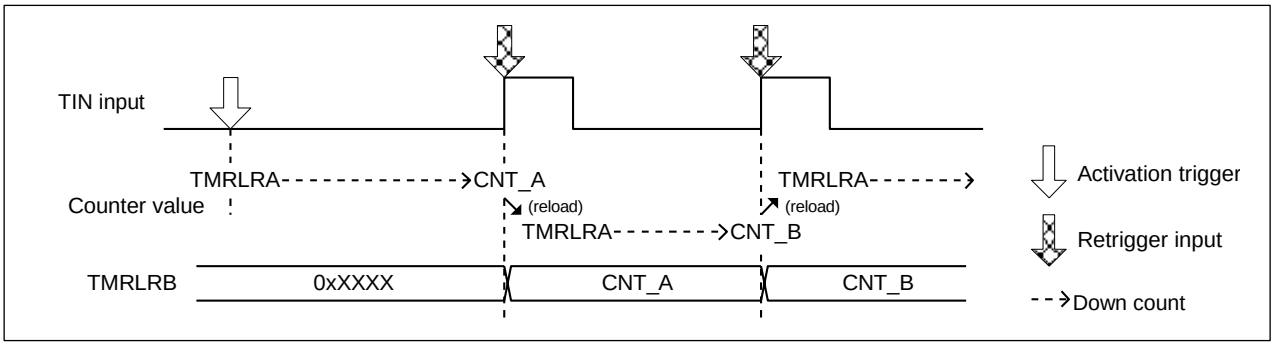
(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)
(B): Count value when a trigger occurs during the count operation
S :Use at timer activation
-:Does not influence operation
*1: TIN effective edge setting
TRGM[1:0]= 00------Does not detect external trigger edge
TRGM[1:0]= 01------Rising edge
TRGM[1:0]= 10------Falling edge
TRGM[1:0]= 11------Both edges
*2:Count clock division setting
CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2
CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8
CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16
CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32
CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64
*3:TOUT output polarity setting
OUTL= 0------Initial value L=> Count L from TMRLRA => Invert whenever an underflow occurs
OUTL= 1------Initial value H=> Count H from TMRLRA => Invert whenever an underflow occurs

*4:Reload setting when an underflow occurs
RELD= 0------One-shot mode
RELD= 1------Reload mode
*5:Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to TRG bit or an input of effective external edge from TIN pin) While down counting, the counter value will be captured onto the TMRLRB whenever a trigger input occurs. The time interval between edges of the triggers to input will be obtained by the following formula.
  T = (The set value for TMRLRA - The captured value for TMRLRB) × Peripheral clock (PCLK) cycle × Division ratio set with CSL

Figure 6-6 Example of Operation (TRGM=01)



Chapter 21: 32-Bit Free-Run Timer

This chapter explains the 32-bit free-run timer.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Setting
7. Q&A
8. Sample Program
9. Notes

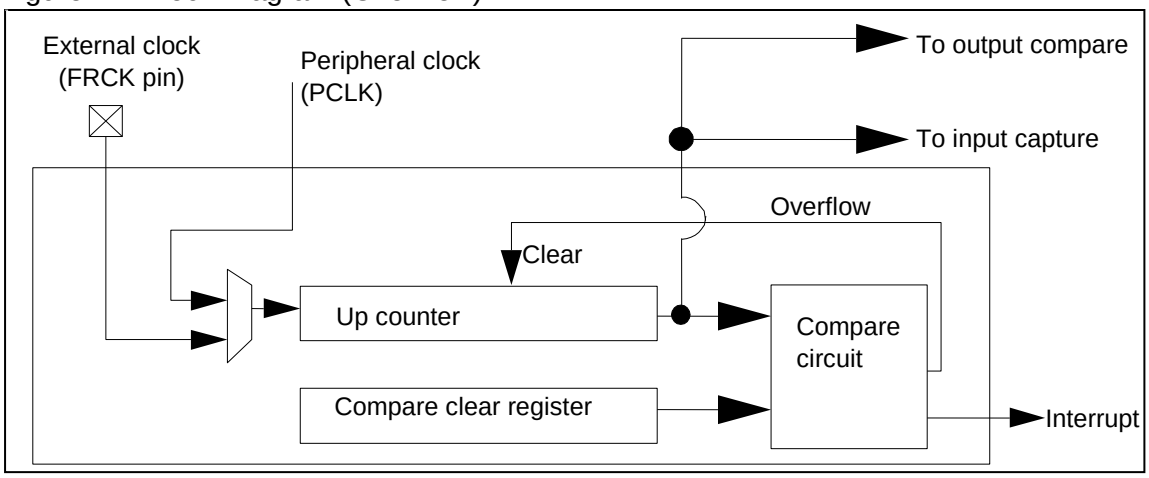
Code : FG61-1v0-91520-2-E

1. Overview

This section explains the overview of the 32-bit free-run timer.

The 32-bit 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 1-1 Block Diagram (Overview)



The numbers of channels available from external clocks are shown below.

MB91F52xB (64pin) : 1
MB91F52xD (80pin) : 2
MB91F52xF (100pin) : 3
MB91F52xJ (120pin) : 3
MB91F52xK (144pin) : 3
MB91F52xL (176pin) : 3

2. Features

This section explains the features of the 32-bit free-run timer.

2.1 Functions of the 32-bit Free-run Timer

2.2 Functions of the Free-run Timer Selector

2.1. Functions of the 32-bit Free-run Timer

The functions of the 32-bit free-run timer is shown.

- Format : 32-bit up counter
- Number of units : 3
- 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 "0000_0000_H".
- It is used as the reference count for 32-bit output compare and 32-bit input capture.

2.2. Functions of the Free-run Timer Selector

The functions of the free-run timer selector is shown.

The allocation of the free-run timer can be selected from output compare and the input capture according to the free-run timer selection register.

3. Configuration

This section explains configuration of the free-run timer.

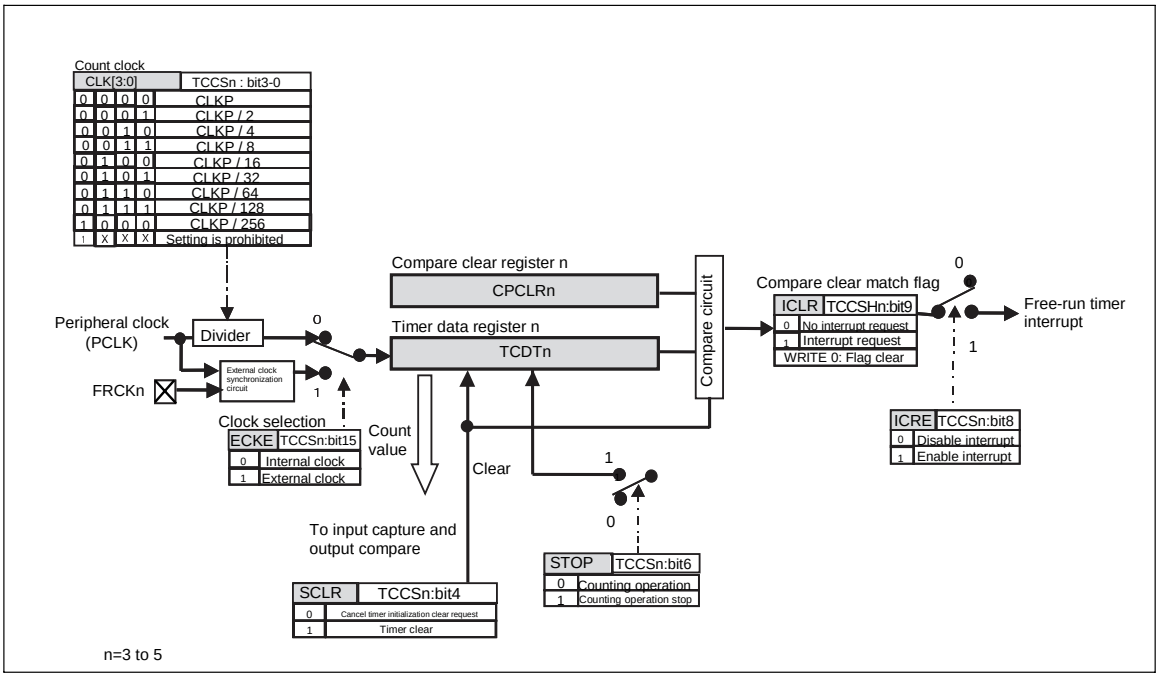
3.1 Configuration Diagram of the 32-bit Free-run Timer

3.2 Configuration Diagram of the Free-run Timer Selector

3.1. Configuration Diagram of the 32-bit Free-run Timer

The configuration diagram of the 32-bit free-run timer is shown.

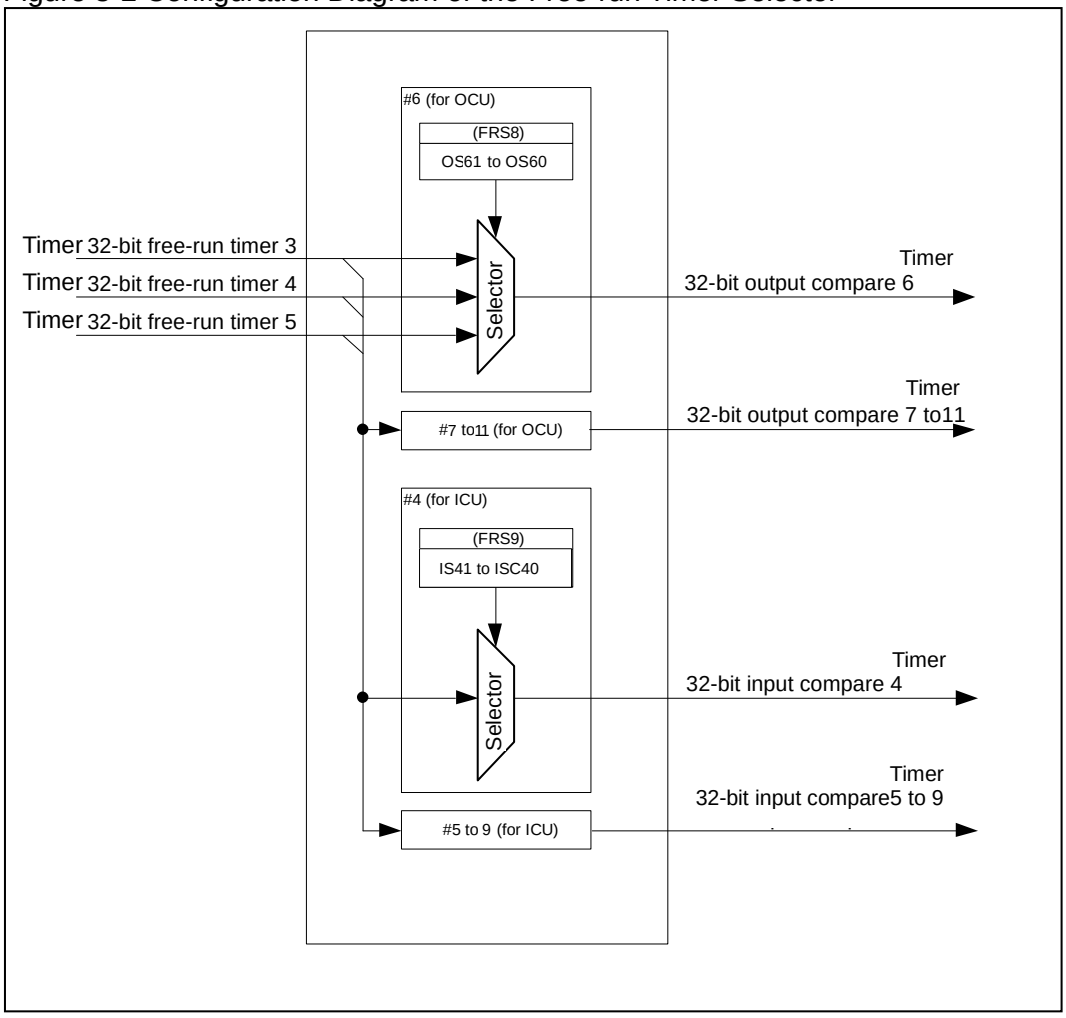
Figure 3-1 Configuration Diagram of the 32-bit free-run timer (only one channel)



3.2. Configuration Diagram of the Free-run Timer Selector

The configuration diagram of the free-run timer selector is shown.

Figure 3-2 Configuration Diagram of the Free-run Timer Selector



4. Registers

This section explains the registers of the free-run timer.

Table 4-1 Base Address (Base_Addr) and External Pin Table

Channel	Base_addr	External pin (FRCK)			
		MB91F52xB	MB91F52xD	MB91F52xF MB91F52xJ MB91F52xK	MB91F52xL
3	0x0240	none	none	FRCK3_0	FRCK3_0/FRCK3_1
4	0x024C	FRCK4_0	FRCK4_0	FRCK4_0	FRCK4_0/FRCK4_1
5	0x0FA0	none	FRCK5_0	FRCK5_0	FRCK5_0/FRCK5_1

Table 4-2 Registers Map of 32-bit Free-run Timer

Address	Registers				Register function
	+0	+1	+2	+3	
0x0240	CPCLR3				Compare clear register 3
0x0244	TCDT3				Timer data register 3
0x0248	TCCSH3	TCCSL3	Reserved		Timer control register (Upper Bit) 3 Timer control register (Lower Bit) 3
0x024C	CPCLR4				Compare clear register 4
0x0250	TCDT4				Timer data register 4
0x0254	TCCSH4	TCCSL4	Reserved		Timer control register (Upper Bit) 4 Timer control register (Lower Bit) 4
0x0FA0	CPCLR5				Compare clear register 5
0x0FA4	TCDT5				Timer data register 5
0x0FA8	TCCSH5	TCCSL5	Reserved		Timer control register (Upper Bit) 5 Timer control register (Lower Bit) 5

Table 4-3 Registers Map of Free-run Timer Selector

Address	Registers				Register function
	+0	+1	+2	+3	
0x0070	Reserved	FRS8			Free-run timer selection register 8
0x0074	Reserved	FRS9			Free-run timer selection register 9

4.1. Registers of the 32-bit Free-run Timer

The registers of the 32-bit free-run timer is shown.

4.1.1. Timer Control Register (Upper Bit) : TCCSH

The bit configuration of the timer control register (Upper bit) is shown.

This register controls the operation of the free-run timer.

■ TCCSH3-5 (Free-run timer 3-5): 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, 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:

Change for the count clock selection bit 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". Writing to these bits has no effect on 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.

4.1.2. Timer Control Register (Lower Bit) : TCCSL

The bit configuration of timer control register (Lower bit) is shown.

This register controls the operation of the free-run timer.

■ TCCSL3-5 (Free-run timer 3-5): 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". Writing to this bit has no effect on 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". Writing to this bit has no effect on operation.

[bit4] SCLR : Timer clear

SCLR	State	
	Read	Write
0	The read value is always "0".	Writing "0" has no meaning.
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} =16MHz	F_{PCLK} =8MHz	F_{PCLK} =4MHz	F_{PCLK} =1MHz
0	0	0	0	$1/F_{PCLK}$	62.5ns	125ns	0.25µs	1µs
0	0	0	1	$2/F_{PCLK}$	125ns	0.25µs	0.5µs	2µs
0	0	1	0	$4/F_{PCLK}$	0.25µs	0.5µs	1µs	4µs
0	0	1	1	$8/F_{PCLK}$	0.5µs	1µs	2µs	8µs
0	1	0	0	$16/F_{PCLK}$	1µs	2µs	4µs	16µs
0	1	0	1	$32/F_{PCLK}$	2µs	4µs	8µs	32µs
0	1	1	0	$64/F_{PCLK}$	4µs	8µs	16µs	64µs
0	1	1	1	$128/F_{PCLK}$	8µs	16µs	32µs	128µs
1	0	0	0	$256/F_{PCLK}$	16µs	32µs	64µs	256µs

CLK3	CLK2	CLK1	CLK0	Clock frequency selection (F_{PCLK}: Peripheral clock (PCLK))				
				Count clock	F_{PCLK} =16MHz	F_{PCLK} =8MHz	F_{PCLK} =4MHz	F_{PCLK} =1MHz
Other settings prohibited				-	-	-	-	-

- 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".

4.1.3. Compare Clear Register : CPCLR

The bit configuration of the compare clear register is shown.

Compare clear register is a 32-bit register to be used for comparison with the free-run timer.

■ CPCLR3-5 (Free-run timer 3-5): Address Base_Addr+00_H (Access: Word)

bit	31															0
	CL[31:0]															
Initial value	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.

4.1.4. Timer Data Register : TCDT

The bit configuration of the timer data register is shown.

The timer data register reads the count value of the 32-bit free-run timer.

■ TCDT3-5 (Free-run timer 3-5): Address Base_addr+04$_H$ (Access: Word)

bit	31							0
	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							

[bit31 to bit0] T[31:0] :

- 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.

4.2. Registers of the Free-run Timer Selector

The registers of the free-run timer selector is shown.

4.2.1. Free-run Timer Selection Register : FRS

The bit configuration of the free-run timer selection register is shown.

This register controls the operation of the free-run timer.

■ FRS8: Address 0071$_H$ (Access: Byte, Half-word, Word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-	-	OS111	OS110	-	-	OS101	OS100
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	-	OS91	OS90	-	-	OS81	OS80
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	OS71	OS70	-	-	OS61	OS60
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

[bit23, bit22] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit21, bit20] OS111, OS110 : Free-run timer selector for output compare 11

OS111	OS110	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 11.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19, bit18] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit17, bit16] OS101, OS100 : Free-run timer selector for output compare 10

OS101	OS100	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 10.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15, bit14] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit13, bit12] OS91, OS90 : Free-run timer selector for output compare 9

OS91	OS90	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 9.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11, bit10] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit9, bit8] OS81, OS80 : Free-run timer selector for output compare 8

OS81	OS80	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 8.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7, bit6] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit5, bit4] OS71, OS70 : Free-run timer selector for output compare 7

OS71	OS70	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 7.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3, bit2] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit1, bit0] OS61, OS60 : Free-run timer selector for output compare 6

OS61	OS60	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the output compare 6.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

■ FRS9: Address 0075_H (Access: Byte, Half-word, Word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-	-	IS91	IS90	-	-	IS81	IS80
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	-	IS71	IS70	-	-	IS61	IS60
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	IS51	IS50	-	-	IS41	IS40
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

.

[bit23, bit22] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit21, bit20] IS91, IS90 : Free-run timer selector for input capture 9

IS91	IS90	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 9.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19, bit18] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit17, bit16] IS81, IS80 : Free-run timer selector for input capture 8

IS81	IS80	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 8.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15, bit14] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit13, bit12] IS71, IS70 : Free-run timer selector for input capture 7

IS71	IS70	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 7.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11, bit10] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit9, bit8] IS61, IS60 : Free-run timer selector for input capture 6

IS61	IS60	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 6.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7, bit6] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit5, bit4] IS51, IS50 : Free-run timer selector for input capture 5

IS51	IS50	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 5.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3, bit2] : Undefined

The read value is always "0". Writing to these bits has no effect on operation.

[bit1, bit0] IS41, IS40 : Free-run timer selector for input capture 4

IS41	IS40	Function
0	0	Free-run timer 3
0	1	Free-run timer 4
1	0	Free-run timer 5
1	1	Setting prohibited (operation is not guaranteed)

These bits configure the free-run timer assigned to the input capture 4.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

5. Operation

This section explains the operations of the free-run timer.

5.1 Operation of the 32-bit Free-run Timer

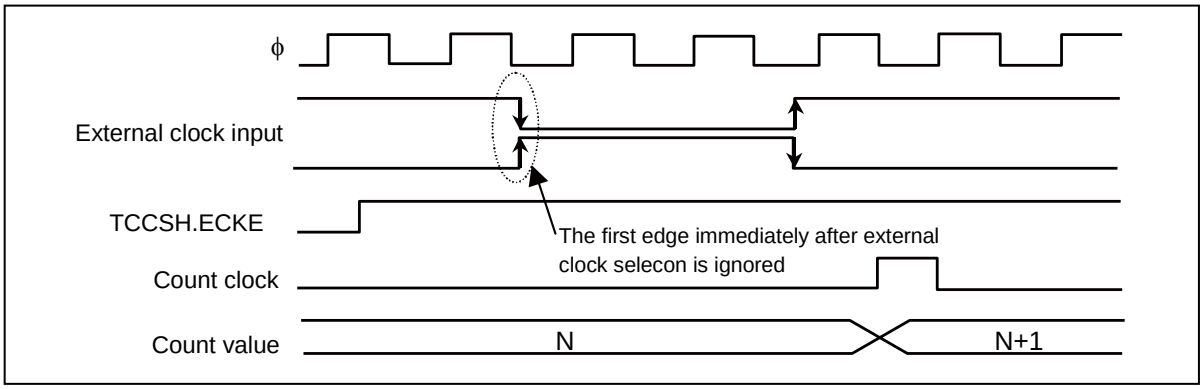
5.2 Operation of the 32-bit Free-run Timer Selector

5.1. Operation of the 32-bit Free-run Timer

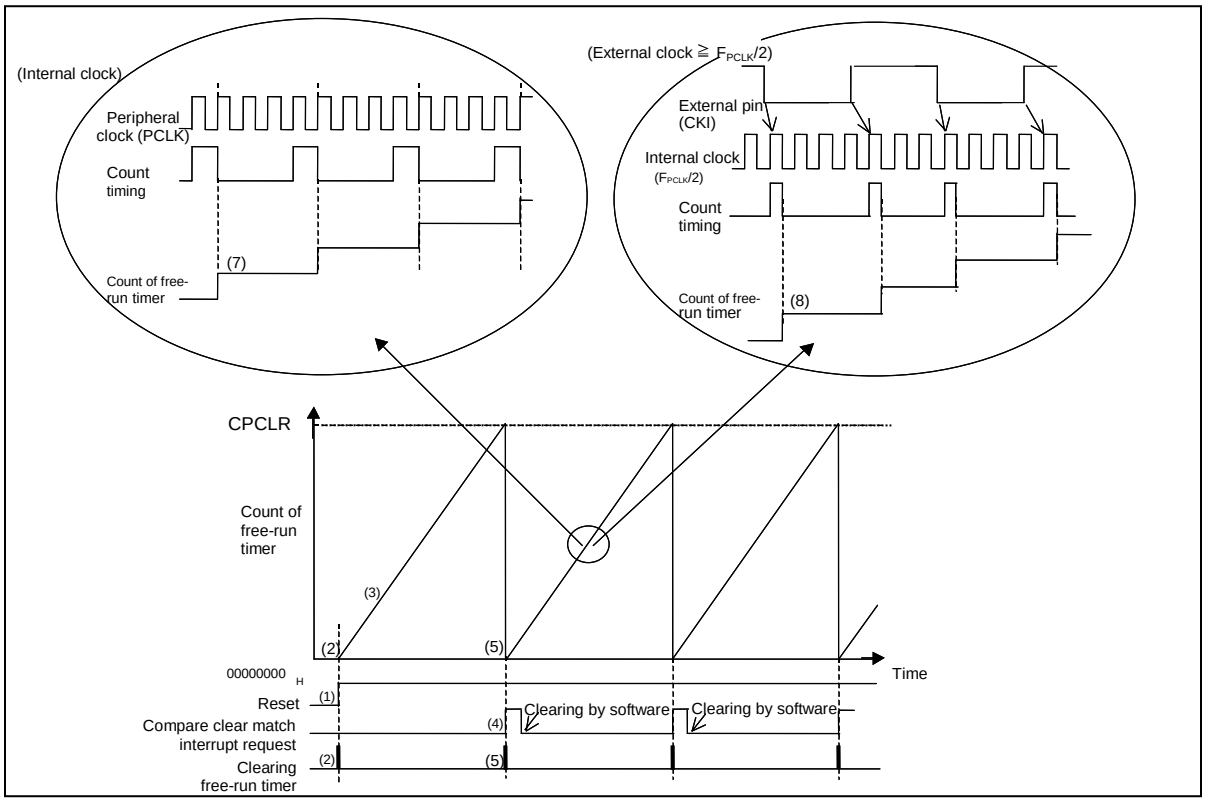
This section shows the operations of the 32-bit free-run timer.

5.1.1. Count Operation

This section shows the count operation.



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".



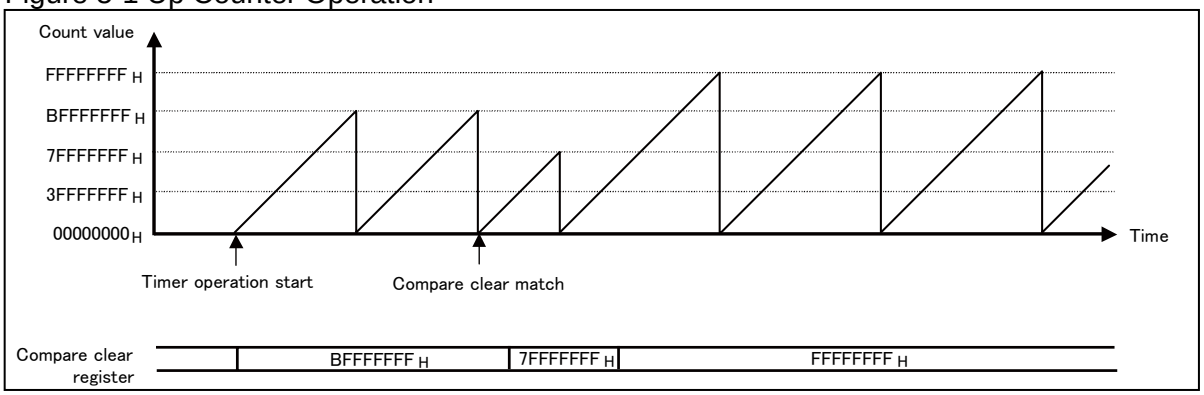
(1) Reset
(2) Clearing of the free-run timer by reset (Count value "0000_0000_H")
(3) Count up operation by the free-run timer
(4) Compare clear match of the free-run timer and interrupt generation
(5) Clearing of the free-run timer by compare clear match (Count value "0000_0000_H")
(6) Repetition of 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.

5.1.2. Counting Up

This section shows counting up.

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 "0000_0000_H" and start counting up again.

Figure 5-1 Up Counter Operation



5.1.3. Timer Clear

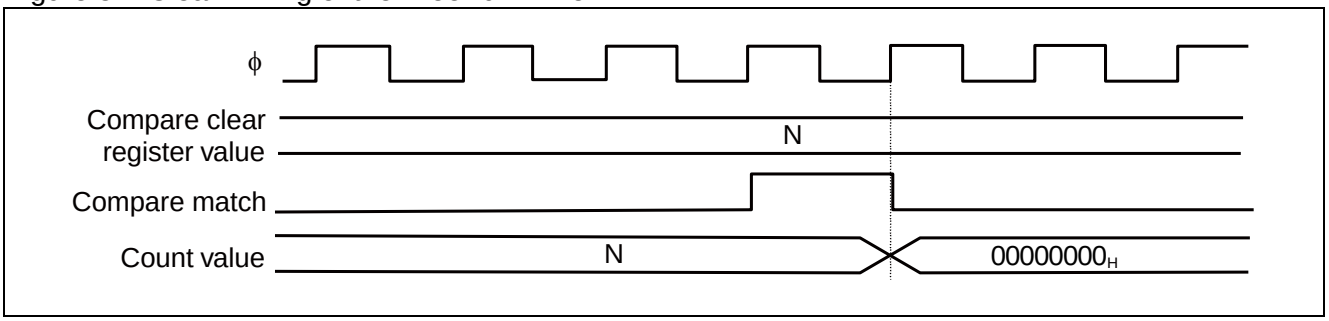
This section shows timer clear.

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 "0000_0000$_H$" is written to the TCDT register while it is in stop
- When it has been reset.

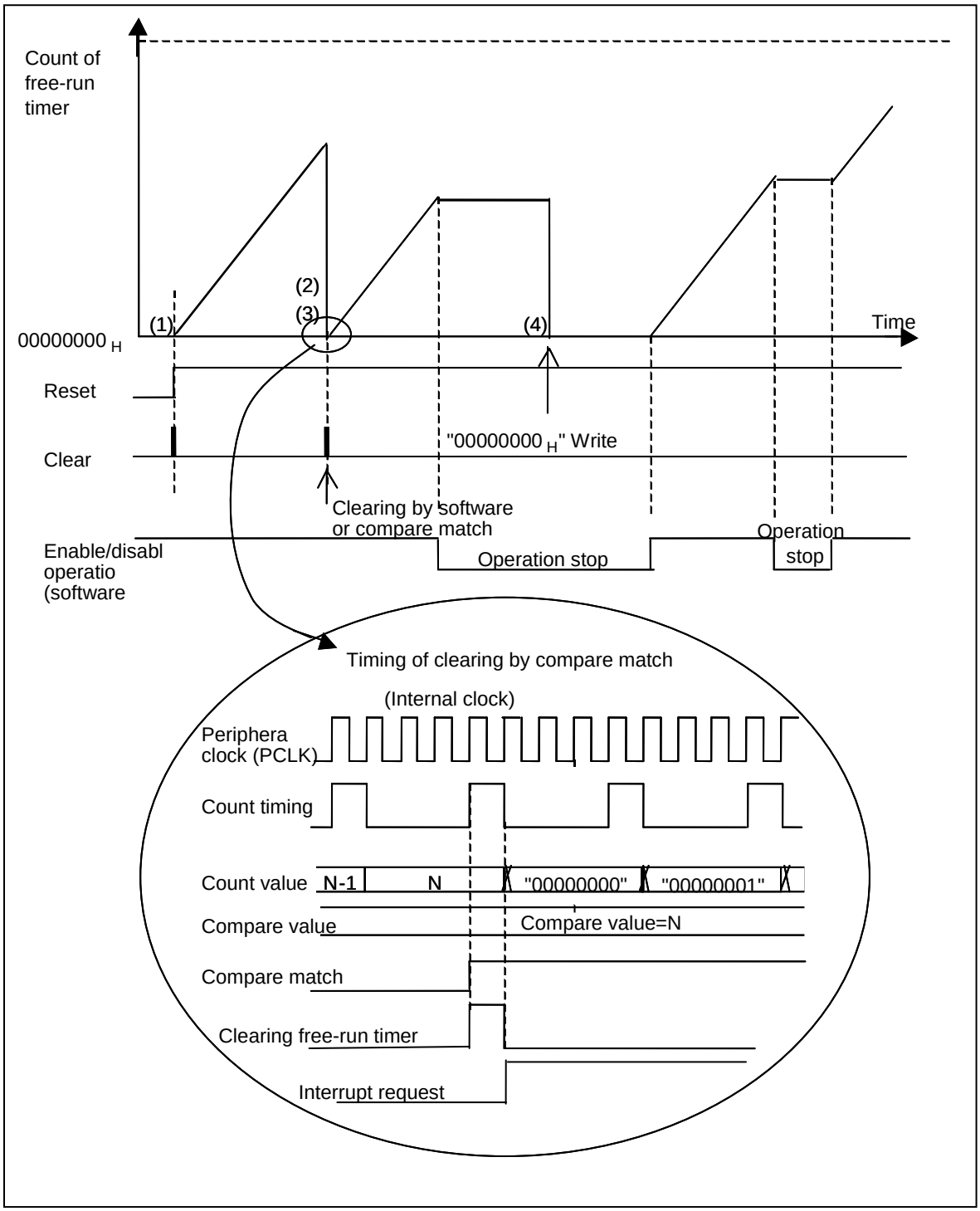
The counter will be cleared as soon as it has been reset. When there is a match with the compare clear register, the counter will be cleared in synchronization with the count timing.

Figure 5-2 Clear Timing of the Free-run Timer



5.1.4. Each Clear Operations of the Free-run Timer

This section shows 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 "0000_0000$_H$" is written to the TCDT register while it is in stop

5.1.5. Timer Interrupt

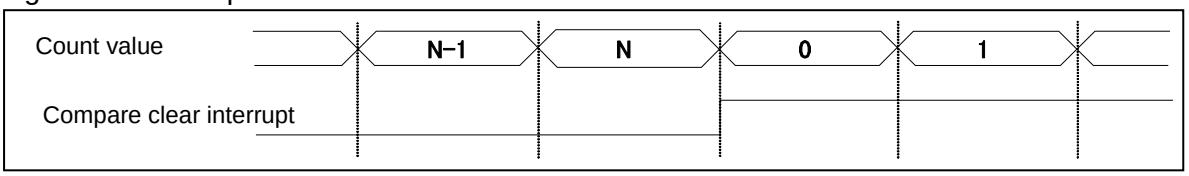
This section shows timer interrupt.

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 5-3 Interrupt



5.2. Operation of the 32-bit Free-run Timer Selector

This section shows the operations of the 32-bit free-run timer selector.

32-bit free-run timer selector is used to set the free-run timer input of 32-bit output compare and 32-bit input capture.

This series consists of 32-bit free-run timer (3 channels), 32-bit output compare (6 channels) and 32-bit input capture (6 channels). The free-run timer used in the register setting shown in the following tables can be selected.

Table 5-1 Table for Registers

Resource	Register	Remarks
OCU6	FRS8.OS6[1:0]	32-bit output compare
OCU7	FRS8.OS7[1:0]	
OCU8	FRS8.OS8[1:0]	
OCU9	FRS8.OS9[1:0]	
OCU10	FRS8.OS10[1:0]	
OCU11	FRS8.OS11[1:0]	
ICU4	FRS9.IS4[1:0]	32-bit input capture
ICU5	FRS9.IS5[1:0]	
ICU6	FRS9.IS6[1:0]	
ICU7	FRS9.IS7[1:0]	
ICU8	FRS9.IS8[1:0]	
ICU9	FRS9.IS9[1:0]	

Table 5-2 Table for Setting Values of Registers

Setting Value	Free-run Timer
00_B	FRT3 (Initial Value)
01_B	FRT4
10_B	FRT5
11_B	Setting prohibited (operation is not guaranteed)

Note:

Before configuring the free-run timer selection register, make sure to verify that the free-run timer is inactive.

6. Setting

This section explains setting of the free-run timer.

Table 6-1 Settings Required for Using the Free-run Timer

Configuration	Register to be configured	Setting method
Timer initialization condition setting	Timer control registers (TCCSH3 to TCCSH5) (TCCSL3 to TCCSL5)	See 7.4.
Count clock setting Internal clock selection		See 7.1.
External clock selection		See 7.2.
Count operation start		See 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 6-2 Settings Required for Performing Free-run Timer Interrupt

Configuration	Register to be configured	Setting method
Free-run timer interrupt vector, Free-run timer interrupt level setting	See "CHAPTER: INTERRUPT CONTROL".	See 7.5.
Free-run timer interrupt setting Interrupt request clear Interrupt request enable	Timer control registers (TCCSH3 to TCCSH5)	See 7.6

Table 6-3 Settings Required for Stopping the Free-run Timer

Configuration	Register to be configured	Setting method
Free-run timer stop bit setting	Timer control registers (TCCSL3 to TCCSL5)	See 7.7.

7. Q&A

This section explains Q&A of the free-run timer.

7.1 How to Select Internal Clock Dividers

7.2 How to Select the External Clock

7.3 How to Enable/Disable the Count Operation of the Free-run Timer

7.4 How to Clear the Free-run Timer

7.5 About Interrupt Related Registers

7.6 How to Enable Compare Clear Interrupt

7.7 How to Stop the Free-run Timer Operation

7.1. How to Select Internal Clock Dividers

This section shows how to select internal clock dividers.

There are nine types of internal clock dividers. You can configure it using the clock selection bits (TCCSHn.ECKE [n=3 to 5]) and count clock bits (TCCSLn.CLK[3:0] [n=3 to 5]).

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".

7.2. How to Select the External Clock

This section shows how to select the external clock.

You can configure it using the clock selection bits (TCCSHn.ECKE [n=3 to 5]), data direction bits and port function bits.

To set to external clock input	Configuration		Pin	Pulse width (H width, L width)
Free-run timer 3	Set the clock selection bit (ECKE) to "1".	Set the FRCK3 pin for peripheral input. (See "CHAPTER: I/O PORTS".)	FRCK3	$4/F_{PCLK}$ or higher
Free-run timer 4		Set the FRCK4 pin for peripheral input. (See "CHAPTER: I/O PORTS".)	FRCK4	
Free-run timer 5		Set the FRCK5 pin for peripheral input. (See "CHAPTER: I/O PORTS".)	FRCK5	

7.3. How to Enable/Disable the Count Operation of the Free-run Timer

This section shows how to enable/disable the count operation of the free-run timer.

Set the count operation bits (TCCSLn.STOP [n=3 to 5]).

Operation	Count operation bit (STOP)
To operate the free-run timer	Set "0".
To stop the free-run timer	Set "1".

7.4. How to Clear the Free-run Timer

This section shows how to clear the free-run timer.

You can clear the free-run timer using the following method.

- Set using the clear bits (TCCSLn.SCLR [n=3 to 5]).

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 "0000_0000_H" while the free-run timer is inactive.
  If "0000_0000_H" is written while the free-run timer is inactive, the count value will be "0000_0000_H".
- Overflow of the free-run timer will result in the count value returning to "0000_0000_H".
- It will be cleared if there is a match with the compare clear register.

7.5. About Interrupt Related Registers

This section shows 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 "4. Table 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 3	#51 Address: $0F_FF30_H$	Interrupt level register (ICR35) Address: 0_0463_H
Free-run timer 4	#50 Address: $0F_FF34_H$	Interrupt level register (ICR34) Address: 0_0462_H
Free-run timer 5	#51 Address: $0F_FF30_H$	Interrupt level register (ICR35) Address: 0_0463_H

Since interrupt request flags (TCCSHn.ICLR [n=3 to 5]) will not be cleared automatically, clear the flags using software before returning from interrupt processing. (Write "0" to the ICLR bit)

7.6. How to Enable Compare Clear Interrupt

This section shows how to enable compare clear interrupt.

Enable interrupt request, interrupt request flag

Interrupt enable setting can be performed using interrupt request enable bits (TCCSHn.ICRE [n=3 to 5]).

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 bits (TCCSHn.ICLR [n=3 to 5]).

Operation	Compare clear interrupt flag bit (ICLR)
Interrupt request clear	Write "0".

7.7. How to Stop the Free-run Timer Operation

This section shows how to stop the free-run timer operation.

See "7.3. How to Enable/Disable the Count Operation of the Free-run Timer".

8. Sample Program

This section explains sample program of the free-run timer.

Setting procedure example 1

Free-run timer 3, Clock=PCLK/2^6,
Count the number of compare matches using interrupt processing.

< Initial setting>

-Free-run timer ch.3 control — Register name.Bit name

Control register setting Clock selection>>	TCCSH3/TCCSL3 .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	TCDT3

-Interrupt-related — Register name.Bit name

Interrupt level setting	ICR35
I flag setting	(CCR)

-Variable setting

<Activation>

-Free-run timer ch.3 activation — Register name.Bit name

Count operation activation	TCCSL3 .STOP

<Interrupt >

-Interrupt processing — Register name.Bit name

Clearing of interrupt request flag	TCCSH3.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_TIMER3_sample(void)
{
        FREERUN3_initial();
        FREERUN3_start();
}

void FREERUN3_initial(void)
{
        IO_TCCS3.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           Initialization of SCLR free-run timer value
                                                       (no) */
                                 /* bit3-0 = 0001      CLK3-0 Count clock PCLK/2=32MHz/2 */

        IO_TCDT3 = 0x0000;       /* Initialization of timer data value */

        IO_ICR[35].byte = 0x10;  /* Free-run timer 3 interrupt level setting (any value) */
        __EI();                  /* Interrupt enabled */
        count = 0;
}

void FREERUN3_start(void)
{
        IO_TCCSL3.bit.STOP = 0;  /* bit6 = 0 STOP count enabled */
}

__interrupt void
FREE_RUN_TIMER3_int(void)
{
        IO_TCCSH3.bit.ICLR = 0;  /* bit9 = 0 Clearing of ICLR compare match flag */

        count++;
}

Specification of interrupt routine required in vector table
#pragma intvect FREE_RUN_TIMER3_int 50
```

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 "0000_0000_H".
- A software clear (TCCSL.SCLR=1) clears the counter in the following cycle when a 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 "0000_0000_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.

Chapter 22: 32-Bit Output Compare

This chapter explains the 32-bit output compare.

1. Overview
2. Features
3. Configuration Diagram
4. Registers
5. Operation
6. Setting
7. Q&A
8. Sample Program
9. Notes

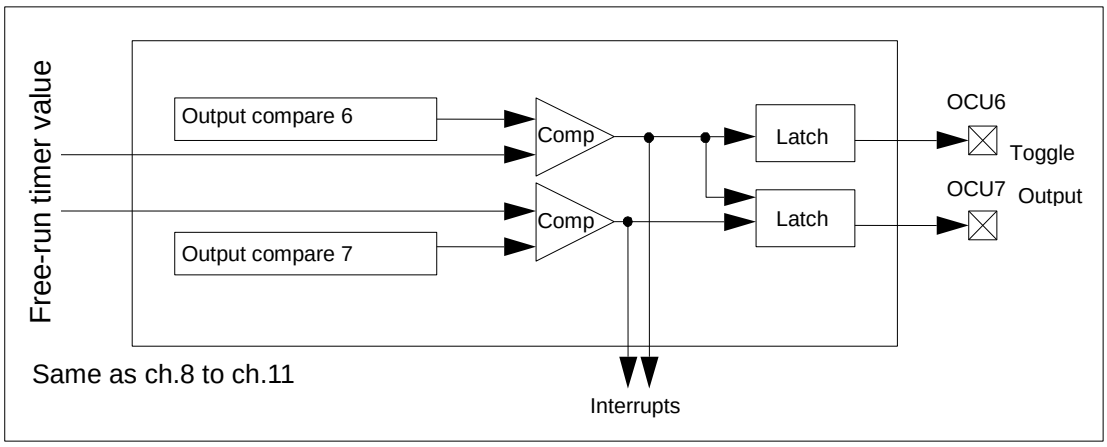
Code : BIP009-1v1-91520-2-E

1. Overview

This section explains the overview of the 32-bit output compare.

The output compare consists of a 32-bit compare register, a compare output latch, a compare control register, and an output control register. When the 32-bit free-run timer value matches the compare register value, the output level is inverted or the H/L level is output and an interrupt also can be generated.

Figure 1-1 Block Diagram (Overview)



The numbers of available external output pins are shown below.

MB91F52xB (64pin) : 4
MB91F52xD (80pin) : 4
MB91F52xF (100pin) : 6
MB91F52xJ (120pin) : 6
MB91F52xK (144pin) : 6
MB91F52xL (176pin) : 6

2. Features

This section explains the features of the 32-bit output compare.

Figure 2-1 Output Waveform

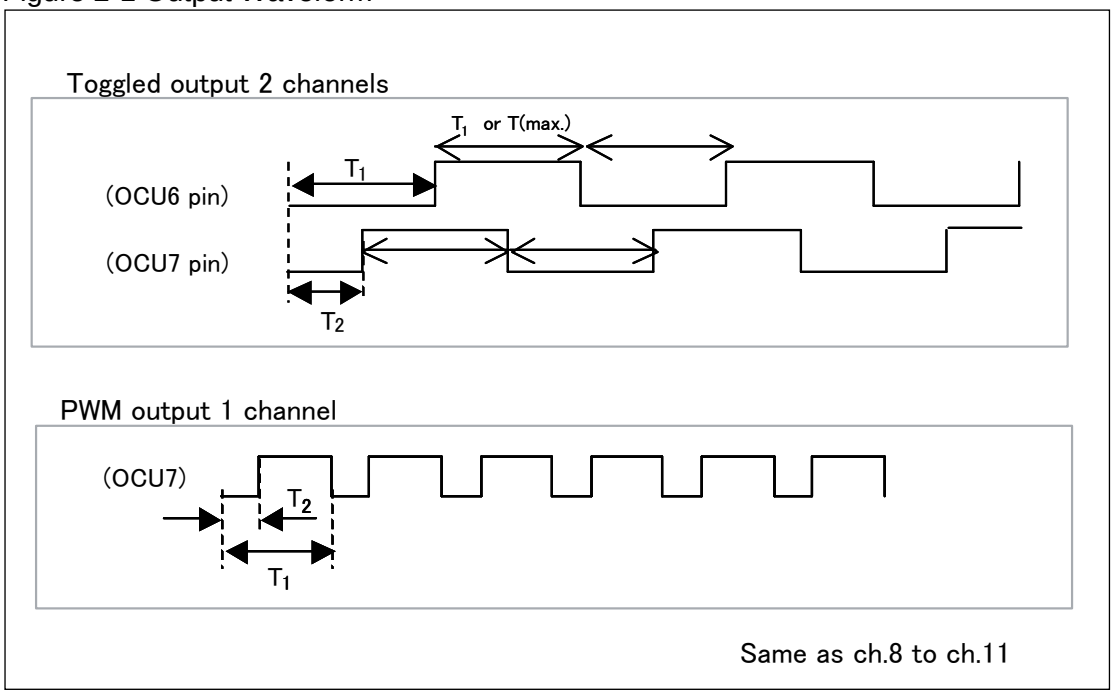
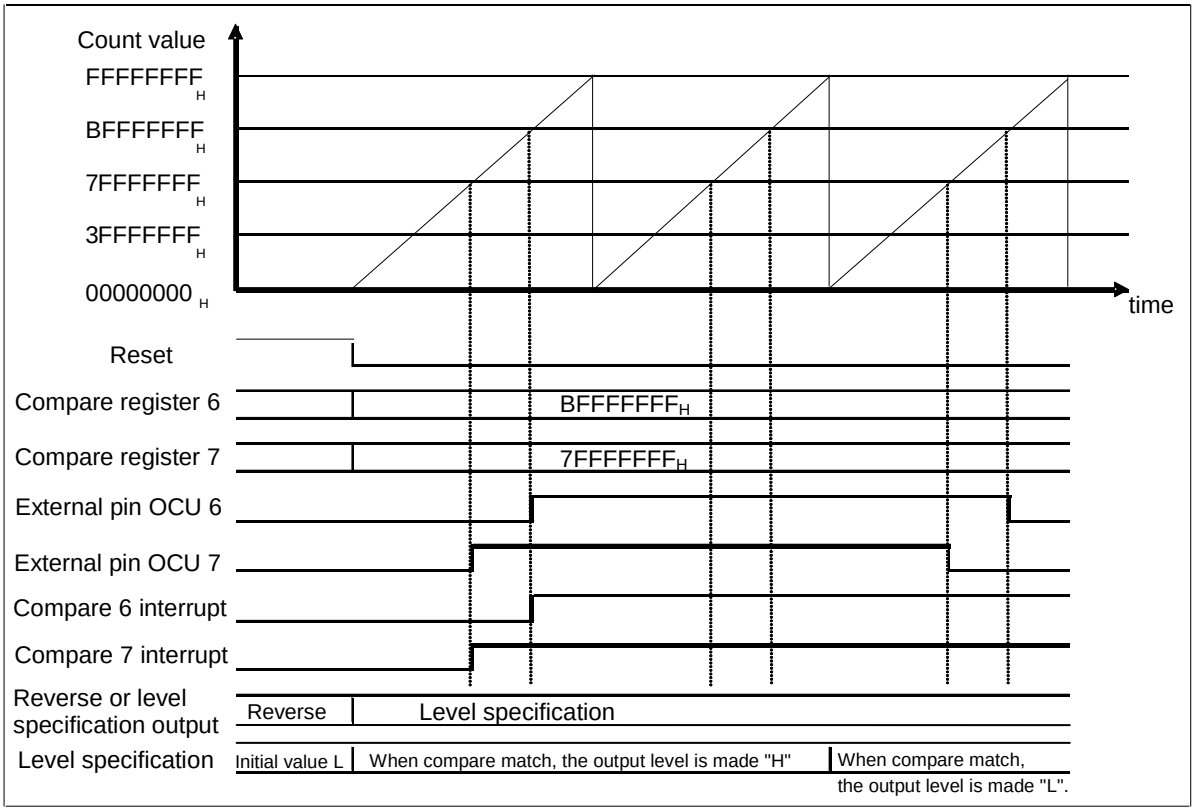


Figure 2-2 Output Level specification Waveform



- Type : 32-bit compare register × 4 + compare circuit
- Corresponding timer : Free-run timer is used
  Number of units 6 channels
- Operation by compare match
  - Pin output value invert (toggle output) or signal output of H/L level specified
  - 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:

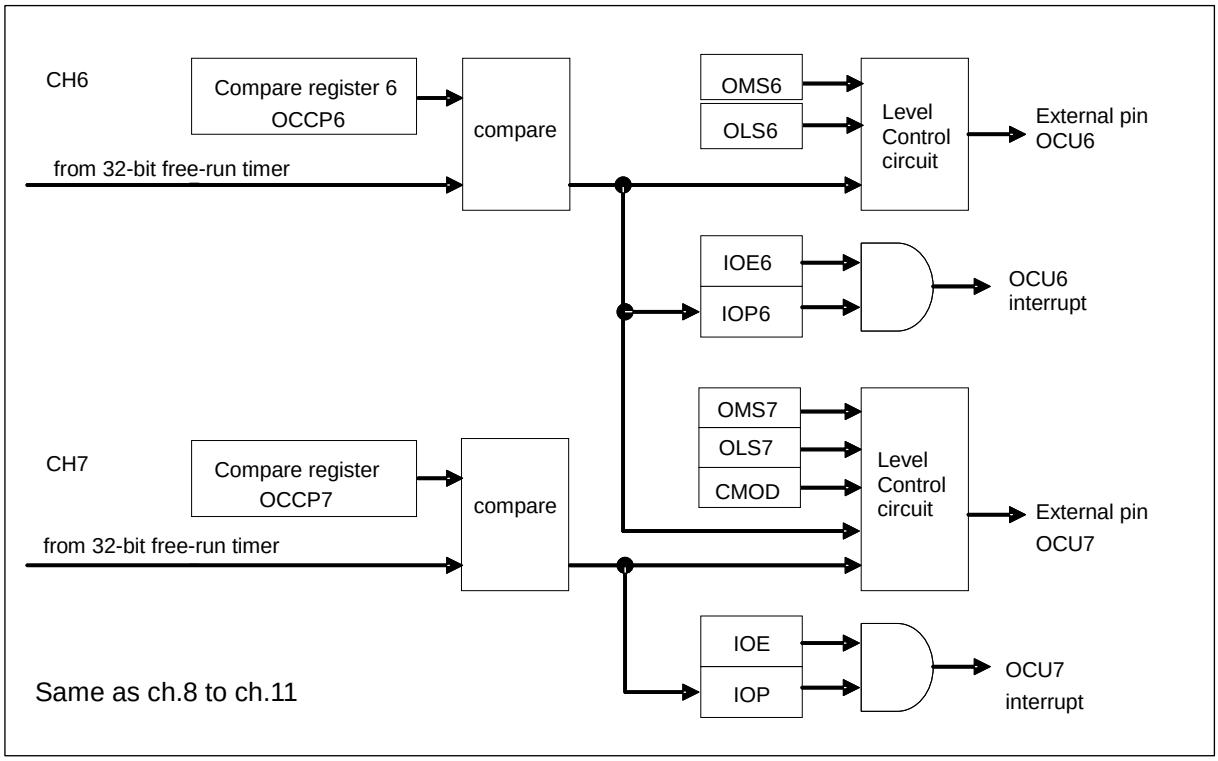
The setting of the peripheral clock (PCLK) divided by 1 is prohibited.

- 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.
  - 6 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 OCU7, 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.

3. Configuration Diagram

This section explains the configuration diagram of the 32-bit output compare.

Figure 3-1 Configuration Diagram (Detail)



4. Registers

This section explains the registers of the 32-bit output compare.

Table 4-1 Table of Base_addr and External Pins

Channel	Base_addr	External pin (OCU output)			
		MB91F52xB MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK MB91F52xL
6	0x0120	none	OCU6_1	OCU6_0/ OCU6_1	OCU6_0/ OCU6_1
7	0x0120	OCU7_1	OCU7_1	OCU7_1	OCU7_0/ OCU7_1
8	0x012C	OCU8_0/ OCU8_1	OCU8_0/ OCU8_1	OCU8_0/ OCU8_1	OCU8_0/ OCU8_1
9	0x012C	OCU9_1	OCU9_1	OCU9_1	OCU9_0/ OCU9_1
10	0x0F90	none	OCU10_1	OCU10_1	OCU10_0/ OCU10_1
11	0x0F90	OCU11_1	OCU11_1	OCU11_0/ OCU11_1	OCU11_0/ OCU11_1

Table 4-2 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0120	OCCP6				Compare register 6
0x0124	OCCP7				Compare register 7
0x0128	Reserved	Reserved	OCSH67	OCSL67	Output control register 67 upper Output control register 67 lower
0x012C	OCCP8				Compare register 8
0x0130	OCCP9				Compare register 9
0x0134	Reserved	Reserved	OCSH89	OCSL89	Output control register 89 upper Output control register 89 lower
0x0F90	OCCP10				Compare register 10
0x0F94	OCCP11				Compare register 11
0x0F98	Reserved	Reserved	OCSH1011	OCSL1011	Output control register 1011 upper Output control register 1011 lower

Address	Registers				Register function
	+0	+1	+2	+3	
0x0078	Reserved			OCLS67	Output level control register 67
0x007C	Reserved			OCLS89	Output level control register 89
0x0F9C	Reserved			OCLS1011	Output level control register 1011

4.1. Output Control Register (Upper Bit) : OCSH

The bit configuration of the output control register (Upper bit) is shown below.

Compare control register (OCSH) controls compare output (OCU pin) level, output enable, output level invert mode, compare operation enable, compare match interrupt enable, and compare match interrupt flag.

This register is to control operations of the output compare.
x: Channel number 6, 8, and 10.
y: Channel number 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 has no effect on operation.

[bit12] CMOD : Output level switch mode

CMOD	Operating mode
0	Independent operation (OCU6 to OCU11 pins output level invert operation is independent.) OCU6, 8, 10 pins: When the free-run timer value corresponds to the compare register 6, 8, 10 (OCCP6, 8, 10) value, the output is inverted. OCU7, 9, 11 pins: When the free-run timer value corresponds to the compare register 7, 9, 11 (OCCP7, 9, 11) value, the output is inverted. The comparison target free-run timer is selected by FRS8 register.

CMOD	Operating mode
1	Coordinated operation OCU6, 8, 10 pins: When the free-run timer value corresponds to the compare register 6, 8, 10 (OCCP6, 8, 10), the output is inverted. OCU7, 9, 11 pins: When the free-run timer value corresponds to either the compare register (6 or 7), (8 or 9), (10 or 11), the output is inverted. The comparison target free-run timer is selected by FRS8 register.

When the compare register 6 and 7 have the same value, the operation is the same one as when only one compare register is used. Same as the compare register 8 to 11.

[bit11, bit10] : Reserved

The read value is always "0".
Write "0" to these bits.

[bit9] OTD : Pin level setting (Output compare y)

[bit8] OTD : Pin level setting (Output compare x)

This bit specifies the pin output level (initial value) when OCU pins output is enabled.

OTD	Operation	
	Read	Write
0	OCU pins output	OCU pins output level (initial value) is set to "L".
1		OCU pins output level (initial value) is set to "H".

- When OCU pins output is performed, the setting of a general-purpose port is required.
- *Writing to these bits is enabled when the compare operation is stopped (OCSL.CSTx or CSTy="0"). The setting should be performed after the compare operation is stopped.
- With the reading operation, the output compare pin output is read.

4.2. Output Control Register (Lower Bit) : OCSL

The bit configuration of the output control register (Lower bit) is shown below.

Compare control register (OCSL) controls compare output (OCU pin) level, output enable, output level invert mode, compare operation enable, compare match interrupt enable, and compare match interrupt flag.

This register is to control operations of the output compare.
x: Channel number 6, 8, and 10.
y: Channel number 7, 9, and 11.

■ OCSLxy (Output compare xy): Address Base_addr+0x0B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IOPy	IOPx	IOEy	IOEx	Reserved	Reserved	CSTy	CSTx
Initial value	0	0	0	0	1	1	0	0
Attribute	R(RM1),W	R(RM1),W	R/W	R/W	R1,W1	R1,W1	R/W	R/W

[bit7] IOP : Interrupt request flag (output compare y)
[bit6] IOP : Interrupt request flag (output compare x)

IOP	State	
	Read	Write
0	No compare match interrupt occurs for the compare register.	Flag (IOP) is cleared.
1	Compare match interrupt occurs for the compare register.	No effect on operations

- This bit is an interrupt flag that indicates whether the value of the compare register matched that of the free-run timer.
- 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".
- If a read-modify-write (RMW) instruction is executed, "1" is always read.

[bit5] IOE : Interrupt request enable (Output compare y)
[bit4] IOE : Interrupt request enable (Output compare x)

IOE	State
0	Output compare interrupt request is disabled.
1	Output compare interrupt request is enabled.

- This bit enables 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] Reserved

The read value is always "1".
Write "1" to these bits.

[bit1] CST : Operation enable (Output compare y)
[bit0] CST : 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 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 is also stopped.

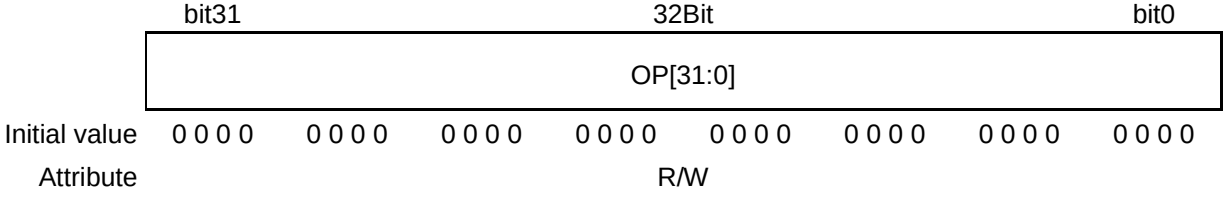
4.3. Compare Register : OCCP

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 6, 8, and 10.
y: Channel number 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)



- 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 changed.

- For access to this register, use a word access instruction.

4.4. Output Level Control Register : OCLS

The bit configuration of the output level control register is shown below.

This register controls compare output (OUT pin) operation mode and compare output level.
x: Channel number 6, 8, 10
y: Channel number 7, 9, 11

■ OCLS67 (Output compare 67): Address 007B$_H$ (Access: Byte, Half-word, Word)

■ OCLS89 (Output compare 89): Address 007F$_H$ (Access: Byte, Half-word, Word)

■ OCLS1011 (Output compare 1011): Address 0F9F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	OLSy	OLSx	OMSy	OMSx
Initial value	X	X	X	X	0	0	0	0
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] : Undefined
The read value is undefined. Writing has no effect on operation.

[bit3] OLS : Output level specification (Output compare y)
[bit2] OLS : Output level specification (Output compare x)

OLS	Operation
0	When compare match, the output level is made "L".
1	When compare match, the output level is made "H".

- This bit specifies the pin output level of the compare register.
- When output mode selection bit(OMS) is "1", the pin output outputs the level specified by this bit when the free-run timer is corresponding to the compare register.
- When output mode selection bit(OMS) is "0", the pin output is inverted regardless of this bit value.

[bit1] OMS : Output mode selection (Output compare y)
[bit0] OMS : Output mode selection (Output compare x)

OMS	Operation
0	When compare match, the output level is reversed.
1	When compare match, the level specified by output level specification bit(OLS) is

	output.

This bit specifies the operation of the output pin when the free-run timer is corresponding to the compare register.

5. Operation

This section explains the operations of the 32-bit output compare.

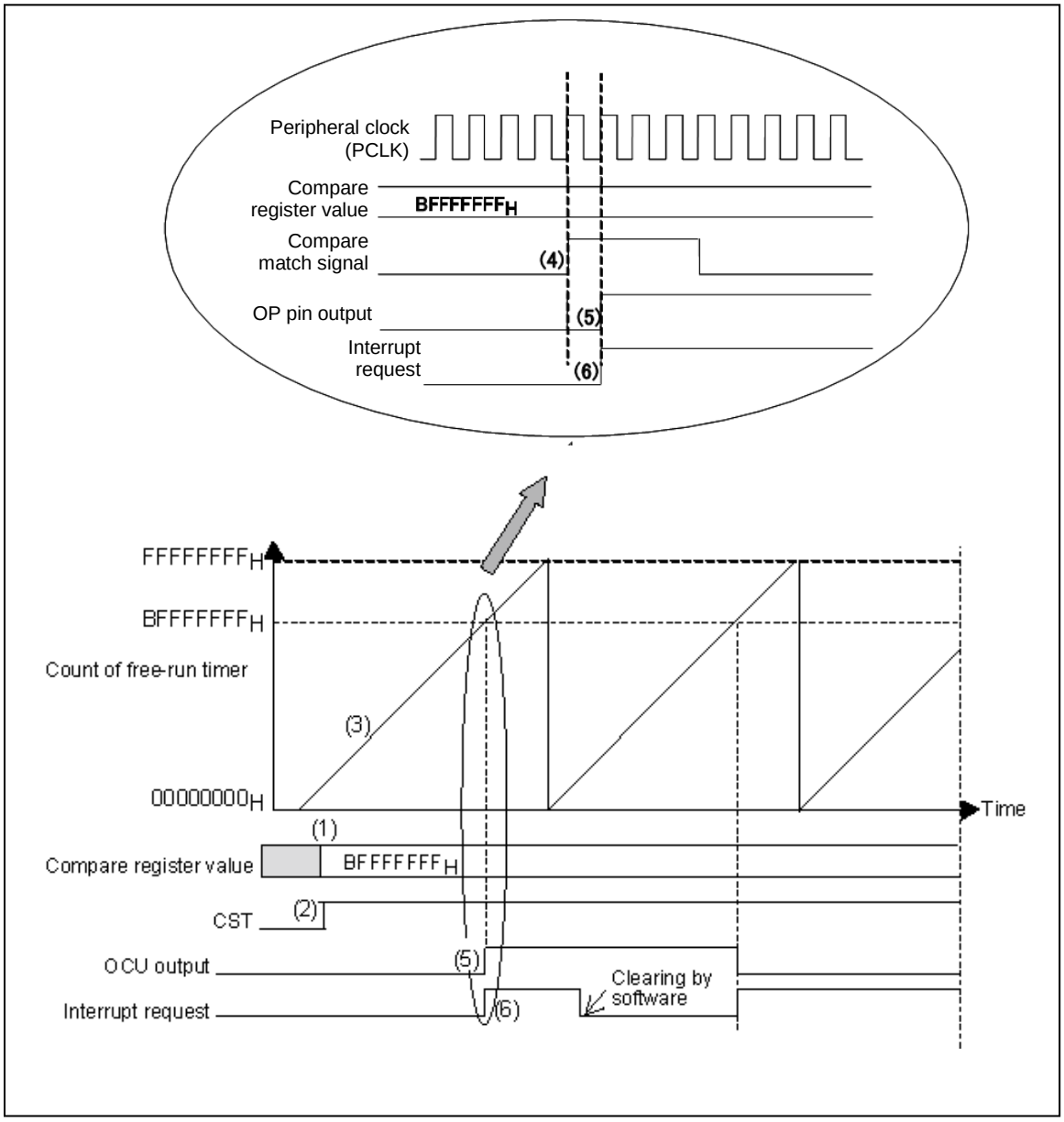
5.1 Output Compare Output (Independent Invert) CMOD = "0"

5.2 Output Compare Output (Coordinated Invert) CMOD = "1"

5.3 Output Compare Operation Timing

5.1. Output Compare Output (Independent Invert) CMOD = "0"

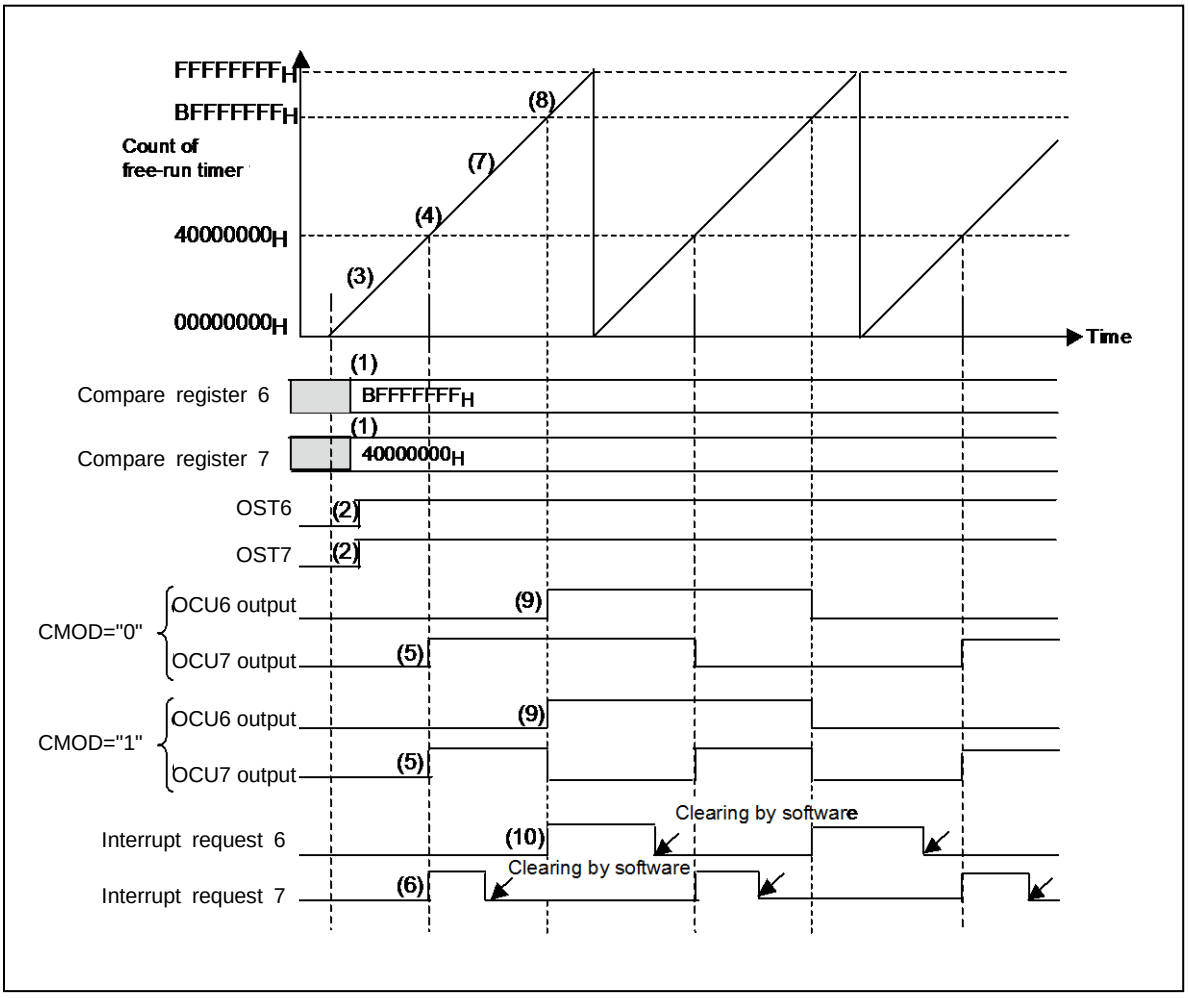
This section shows the output compare output (independent invert) CMOD="0".



(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.

5.2. Output Compare Output (Coordinated Invert) CMOD = "1"

This section shows the output compare output (coordinated invert) CMOD="1".



(1) Values of Compare 6 and Compare 7 are set.
(2) Compare operation is enabled.
(3) Free-run timer count up
(4) Compare 7 match
(5) OCU7 output level is inverted.
(6) Compare 7 match interrupt
(7) Free-run timer count up
(8) Compare 6 match
(9) OCU6 output level is inverted.
When CMOD = "1", OCU7 output level is also inverted.
(10) Compare 6 match interrupt

5.3. Output Compare Operation Timing

This section shows the output compare operation timing.

With the use of two pairs of compare registers, the output level can be changed. (For CMOD = "1")

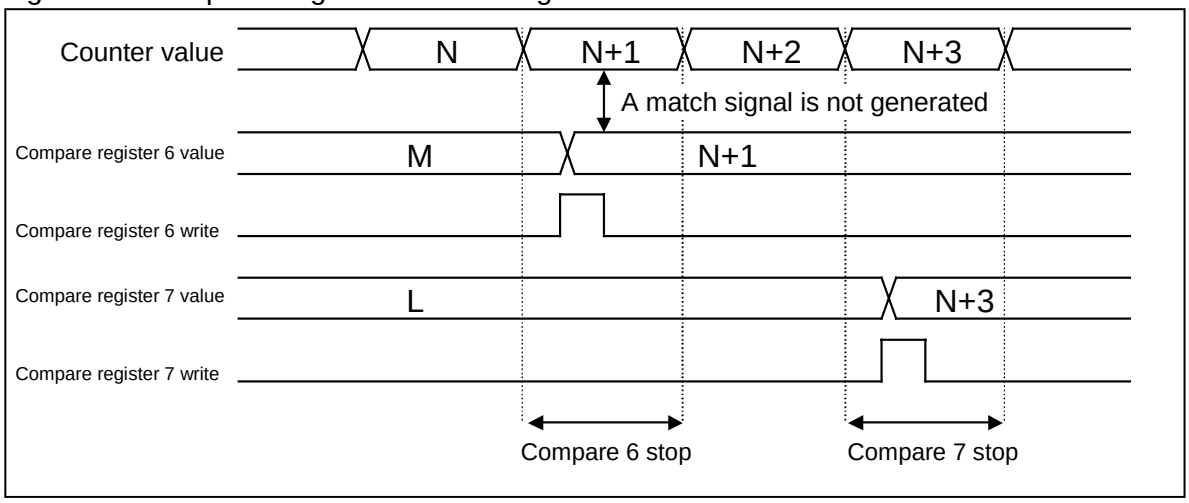
The output compare can invert the output as well as generate an interrupt when the free-run timer value matches the specified compare register value and a compare match signal is generated. The output invert timing on compare match is synchronized with the counter count timing.

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.

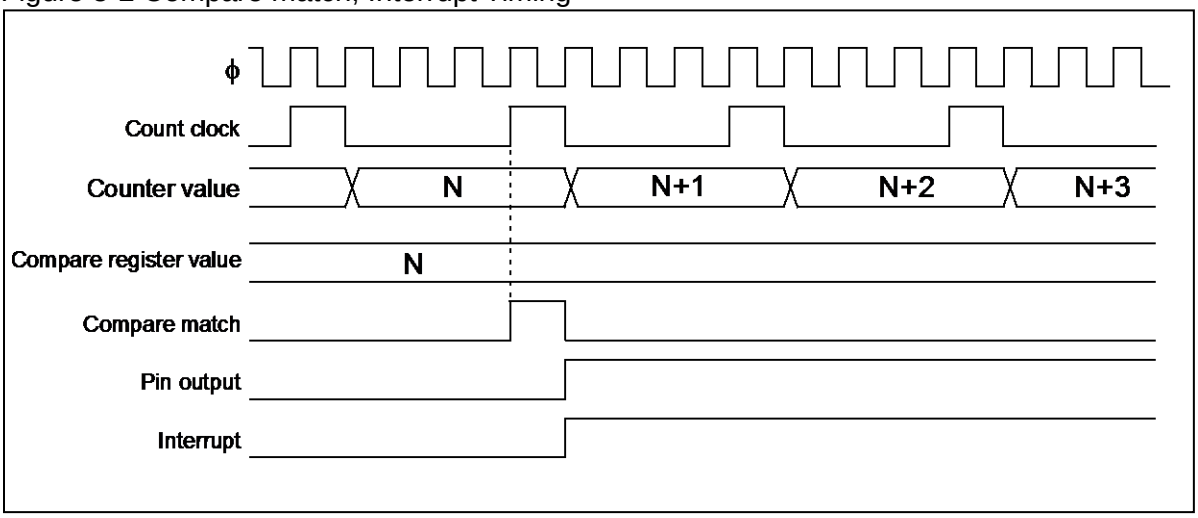
Figure 5-1 Compare Register Write Timing



5.3.2. Compare Match, Interrupt

Compare match, interrupt are shown below.

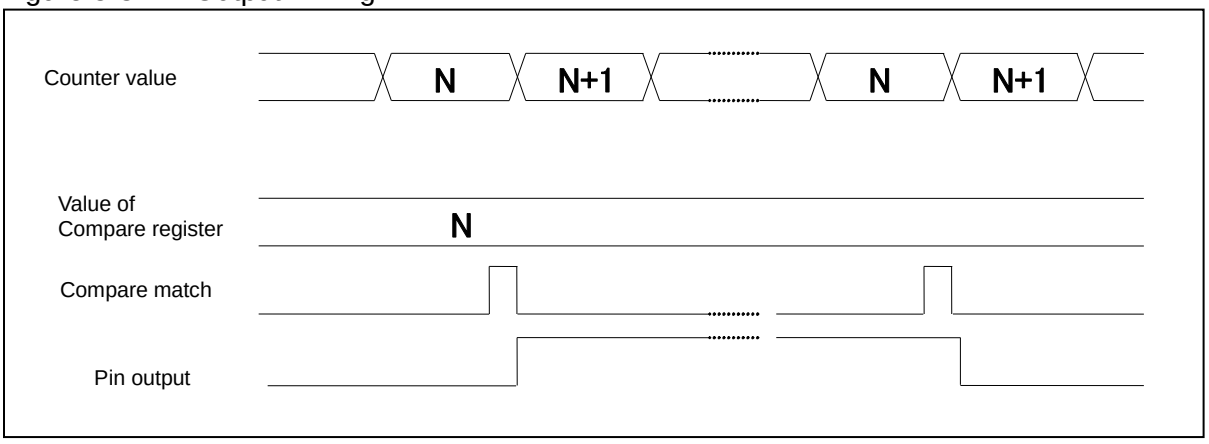
Figure 5-2 Compare match, Interrupt Timing



5.3.3. Pin Output

This section shows the pin output.

Figure 5-3 Pin Output Timing



6. Setting

This section explains settings of the 32-bit output compare.

Table 6-1 Configuration Necessary for Use of Output Compare

Configuration	Register to be configured	Setting Method
Setting of the free-run timer	See "CHAPTER: 32-BIT FREE-RUN TIMER".	-
Setting of the compare value	Compare register: (OCCPx)	See 7.1.
Setting of the compare mode	Output control register (OCSHxx, OCSLxx)	See 7.2.
Compare operation stop		See 7.3.
Setting of the compare pin output initial level		See 7.4.
Setting of OCU6, OCU7 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: 32-BIT FREE-RUN TIMER ".	See 7.6.
Compare operation enable (activation)	Output control register (OCSLxx)	See 7.7.
Operation mode selection	Output level control register (OCLS)	See 7.12.

Table 6-2 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 7.8.
Setting of output compare interrupt Interrupt request clear Interrupt request enable	Output control register (OCSHxx, OCSLxx)	See 7.10.

7. Q&A

This section explains Q&A of the 32-bit output compare.

7.1 How to Set the Compare Value

7.2 How to Set the Compare Mode (Example with OCU7)

7.3 How to Enable/Disable the Compare Operation (Example with OCU6, 7)

7.4 How to Set the Compare Pin Output Initial Level (Example with OCU6, 7)

7.5 How to Set the Compare Pin OCU6-OCU7 for Output

7.6 How to Clear the Free-run Timer

7.7 How to Enable the Compare Operation (Example with OCU6, 7)

7.8 Interrupt Related Register

7.9 Interrupt Type

7.10 How to Enable the Interrupt

7.11 Calculation Method for the Compare Value

7.12 How to Set the Operation Mode

7.1. How to Set the Compare Value

This section shows how to set the compare value.

Write the compare value to the compare register OCCPx.

7.2. How to Set the Compare Mode (Example with OCU7)

This section shows how to set the compare mode.

Set with the compare mode bit (OCSH67.CMOD)

Operation	Compare mode bit
To invert the OCU7 pin output when the free-run timer value matches the compare register 7 (OCCP7)	Set (OCSH67.CMOD) to "0".
To invert the OCU7 pin output when the free-run timer value matches either the compare register 6 (OCCP6) or the compare register 7 (OCCP7)	Set (OCSH67.CMOD) to "1".

Regardless of the CMOD bit, the operation is as follows:

Regardless of the compare mode bit (OCSH67.CMOD) setting, the OCU6 output is inverted when the free-run timer value matches the compare register 6 (OCCP6).

7.3. How to Enable/Disable the Compare Operation (Example with OCU6, 7)

This section shows how to enable/disable the compare operation.

Set the compare operation enable bit (OCSL67.CST6), (OCSL67.CST7).

Operation	Compare	Compare operation enable bit
To stop (disable) the compare operation	Compare 6	Set (OCSL67.CST6) to "0".
	Compare 7	Set (OCSL67.CST7) to "0".
To enable the compare operation	Compare 6	Set (OCSL67.CST6) to "1".
	Compare 7	Set (OCSL67.CST7) to "1".

7.4. How to Set the Compare Pin Output Initial Level (Example with OCU6, 7)

This section shows how to set the compare pin output initial level.

Set the compare pin output specification bit (OCSH67.OTD6), (OCSH67.OTD7).

Operation	Compare pin output specification bit
To set the compare 6 pin to "L"	Set (OCSH67.OTD6) to "0".
To set the compare 6 pin to "H"	Set (OCSH67.OTD6) to "1".
To set the compare 7 pin to "L"	Set (OCSH67.OTD7) to "0".
To set the compare 7 pin to "H	Set (OCSH67.OTD7) to "1".

7.5. How to Set the Compare Pin OCU6-OCU7 for Output

This section shows how to set the compare pin OCU6-OCU7 for output.

Set the pin for peripheral output. For setting method, see "CHAPTER: I/O PORTS".

7.6. How to Clear the Free-run Timer

This section shows 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: 32-BIT FREE-RUN TIMER".

7.7. How to Enable the Compare Operation (Example with OCU6, 7)

This section explains how to enable the compare operation.

Set the compare operation enable bit (OCSL67.CST6, OCSL67.CST7).

See "7.3 How to Enable/Disable the Compare Operation (Example with OCU6, 7)".

7.8. Interrupt Related Register

This section shows the interrupt related register.

Both the output compare interrupt vector and the output compare interrupt level are set.

The relation among the output compare channel, interrupt level, and interrupt vector is shown in the table below:

For the interrupt level and interrupt vector, see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

Channel	Interrupt Vector (Default)	Interrupt Level Setting Bit (ICR[4:0])
Output compare 6/7/10/11	#58 Address: $0FFF14_H$	Interrupt level register (ICR42) Address: $0046A_H$
Output compare 8/9	#59 Address: $0FFF10_H$	Interrupt level register (ICR43) Address: $0046B_H$

The interrupt request flag (OCSLxy.IOPx, OCSLxy.IOPy x=6, 8, 10 y=7, 9, 11) are not cleared automatically. Before recovering from the interrupt process, write "0" to each bit to clear with software.

7.9. Interrupt Type

This section shows the interrupt type.

The interrupt has one type only. It is generated by a compare match.

7.10. How to Enable the Interrupt

This section shows how to enable the interrupt.

Configure the interrupt request enable bit (OCSLxy.IOEx, OCSLxy.IOEy x=6, 8, 10 y=7, 9, 11) for the interrupt enable setting.

Operation	Interrupt request enable bit (OCSLxy.IOEx, OCSLxy.IOEy x=6, 8, 10 y=7, 9, 11)
To disable interrupt	Set "0".
To enable input	Set "1".

Set the interrupt request flag bit (OCSLxy.IOPx, OCSLxy.IOPy x=6, 8, 10 y=7, 9, 11) for the interrupt request clear.

Operation	Interrupt request flag bit (OCSLxy.IOPx, OCSLxy.IOPy x=6, 8, 10 y=7, 9, 11)
To clear interrupt request	Write "0".

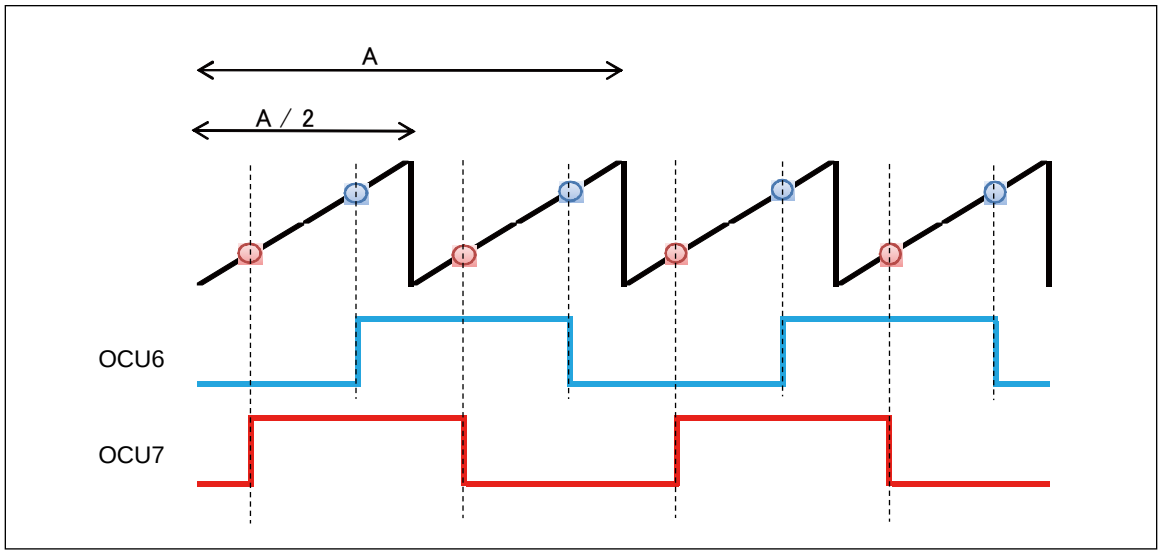
7.11. Calculation Method for the Compare Value

This section shows the calculation method for the compare value.

7.11.1. Toggle Output Pulse

This section shows the toggle output pulse.

(Example) To calculate a two-phase pulse with OCU6, 7, cycle A, and one-fourth phase difference



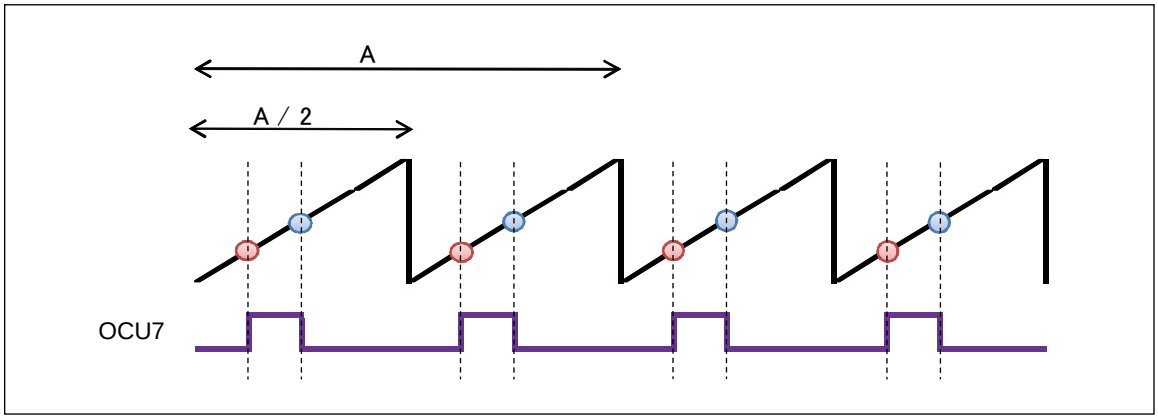
Formula:

Set as follows.
Set Free-run Timer Compare Clear Value = (A/2)-1
Compare Register 6 value = (A/2x3/4)-1
Compare Register 7 value = (A/2x1/4)-1
OCSH67.CMOD = 0

7.11.2. PWM Output

This section shows the PWM output.

(Example) To calculate the PWM with OCU6, 7, cycle A, and duty 1/4



Formula:

Set as follows.
Set Free-run Timer Compare Clear Value = (A/2)-1
Compare Register 6 value = (A/2x1/2)-1
Compare Register 7 value = (A/2x1/4)-1
OCSH67.CMOD = 1

7.12. How to Set the Operation Mode

This section shows how to set operation mode.

Use the OMS bit to specify the operation of the output pin when the free-run timer is corresponding to the compare register.

OCLS.OMS=0 : compare match, the output level is reversed.
OCLS.OMS=1 : compare match, the level specified by output level specification bit (OLS) is output.

8. Sample Program

This section explains a sample program for 32-bit output compare.

Configuration procedure example 1

.2 channels independent output
Compare operation (7FFF, BFFF)
Compare not cleared for interrupt occurrence

1. Initial setting

- Free-run timer ch.4 control	Register name.bit name
Setting of control register	TCCSH4, TCCSL4
Clock selection>>	.ECKE
Compare interrupt request flag>>	.ICLR
Compare interrupt request enable>>	.ICRE
Counting Operation>>	.STOP
TCDT clear>>	.SCLR
Count clock>>	.CLK3-0
Setting of the timer data value	TCDT4

- Port	Register name.bit name
Port OCU6 output setting	See "CHAPTER: I/O PORTS"
Port OCU7 output setting	

- Output compare control	Register name.bit name
Free-run timer selection	OCFS67
Setting of control register	OCSH67, OCSL67
Pin output level invert operation>>	.CMOD
Pin output level specification>>	.OTD7, OTD6
Interrupt request flag>>	.IOP7, IOP6
Interrupt request enable>>	.IOE7, IOE6
Operation enable setting>>	.CST7, CST6
Setting of compare value ch.6	OCCP6
Setting of compare value ch.7	OCCP7

- Interrupt relation	Register name.bit name
Setting of an interrupt level.	ICR42
Setting of I flag	(CCR)

2. Activation

- Output compare activation	Register name.bit name
Interrupt control	OCSL67.IOE7 OCSL67.IOE6
Compare operation activation	OCSL67.CST7 OCSL67.CST6

- Free-run timer ch.4 activation	Register name.bit name
Counting operation activation	TCCSL4.STOP

3. Interrupt

- Interrupt process	Register name.bit name
Clearing of interrupt request flag	OCSL67.IOP6
(any process)	
.....	
Clearing of interrupt request flag	OCSL67.IOP7
(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 OUTPUT67_sample(void)
{
        freerun4_initial();
        OUTPUT67_initial();
        OUTPUT67_start();
        freerun4_start();
}

void freerun4_initial(void)
{
        IO_TCCS4.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_TCDT4 = 0x0000;              /* Timer data value initialization */
}

void OUTPUT67_initial(void)
{
        PORT_SETTING_OCU6_OUT();  /* Set the OCU6 pin for peripheral input. */
        PORT_SETTING_OCU7_OUT();  /* Set the OCU7 pin for peripheral input. */

        IO_OCFS67.hword = 0x0003;       /* Select the free-run timer 4. */
        IO_OCS67.hword = 0xEC0C;        /* Setting value =1110_1100_0000_1100 */
                                        /* bit15-13 = 111     Undefined bit*/
                                        /* bit12 = 0          CMOD ch.6, ch.7 level invert */
                                        /* bit11-10 = 11      Undefined bit*/
                                        /* bit9-8 = 00        OTD7, OTD6 Compare pin output L */
                                        /* bit7-6 = 00        IOP7, IOP6 Output compare no match */
                                        /* bit5-4 = 00        IOE7, IOE6 Output compare interrupt disable */
                                        /* bit3-2 = 11        Undefined bit*/
                                        /* bit1-0 = 00        CST7, CST6 Compare operation disable */
        IO_OCCP6 = BFFF                 /* Setting of compare register ch.6 */
        IO_OCCP7 = 7FFF                 /* Setting of compare register ch.7 */

        IO_ICR[42].byte = 0x10;         /* Output compare ch.6, ch.7 interrupt level setting (any value) */
        _EI();                          /* Interrupt enable */
}

void OUTPUT67_start(void)
{
        IO_OCS67.hword = 0xEC3C;        /* bit5-4 = 11 IOE7, IOE6 Output compare interrupt enable */
        IO_OCS67.hword = 0xEC3F;        /* bit1-0 = 11 CST7, CST6 Compare operation enable */
}

void freerun4_start(void)
{
        IO_TCCSL4.bit.STOP = 0;         /* bit4 = 0 STOP Counting enable */
}

__interrupt void INPUT67_int(void)
{
        IO_OCSL67.byte & = 0xBF;        /* bit6 = 0 IOP6 Clearing of interrupt flag */
        ……
}
__interrupt void INPUT67_int(void)
{
        IO_OCSL67.byte & = 0x7F;        /* bit7 = 0 IOP7 Clearing of interrupt flag */
        ……
}

Interrupt routine specification with the vector table is required.
#pragma intvect OUTPUT6_int 58
```

Configuration procedure example 2

.Compare for two pairs
Output of ch.4
Compare operation (7FFF, BFFF)
Compare is cleared with a cycle of a larger compare value.
Interrupt occurrence

1. Initial setting

- Free-run timer ch.4 control — Register name.bit name

Setting of control register	TCCSH4, TCCSL4
Clock selection>>	.ECKE
Compare interrupt request flag>>	.ICLR
Compare interrupt request enable>>	.ICRE
Counting Operation>>	.STOP
TCDT clear>>	.SCLR
Count clock>>	.CLK3-0
Setting of the timer data value	TCDT4

- Port — Register name.bit name

Port OCU7 output setting	See "CHAPTER: I/O PORTS ".

- Output compare control — Register name.bit name

Free-run timer selection	OCFS67
Setting of control register	OCSH67, OCSL67
Pin output level invert operation>>	.CMOD
Pin output level specification>>	.OTD7, OTD6
Interrupt request flag>>	.IOP7, IOP6
Interrupt request enable>>	.IOE7, IOE6
Operation enable setting>>	.CST7, CST6
Setting of the compare value ch.6	OCCP6
Setting of the compare value ch.7	OCCP7

- 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	OCSL67.IOE7 OCSL67.IOE6
Compare operation activation	OCSL67.CST7 OCSL67.CST6

- Free-run timer ch.4 activation — Register name.bit name

Counting operation activation	TCCSL4.STOP

3. Interrupt

- Interrupt process — Register name.bit name

Clearing of interrupt request flag	OCSL67.IOP6
(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 OUTPUT67_sample(void)
{
        freerun4_initial();
        OUTPUT67_initial();
        OUTPUT67_start();
        freerun4_start();
}

void freerun4_initial(void)
{
        IO_TCCS4.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_TCDT4 = 0x0000;          /* timer data value initialization */
}

void OUTPUT67_initial(void)
{
        PORT_SETTING_OCU7_OUT(); /* Set the OCU7 pin for peripheral input. */

        IO_OCFS67.hword = 0x0003;   /* Select the free-run timer 4. */
        IO_OCS67.hword = 0xEC0C;    /* Setting value =1110_1100_0000_1100 */
                                    /* bit15-13 = 111    Undefined bit */
                                    /* bit12 = 0         CMOD ch.6, ch.7 Level invert */
                                    /* bit11-10 = 11     Undefined bit */
                                    /* bit9-8 = 00       OTD7, OTD6 Compare pin output L*/
                                    /* bit7-6 = 00       IOP7, IOP6 Output compare no match */
                                    /* bit5-4 = 00       IOE7, IOE6 Output compare interrupt disable */
                                    /* bit3-2 = 11       Undefined bit*/
                                    /* bit1-0 = 00       CST7, CST6 Compare operation disable */
        IO_OCCP6 = BFFF             /* Setting of compare register ch.6 */
        IO_OCCP7 = 7FFF             /* Setting of compare register ch.7 */

        IO_ICR[42].byte = 0x10;     /* Output compare ch.6 interrupt level setting (any value) */
        IO_ICR[43].byte = 0x10;     /* Output compare ch.7 interrupt level setting (any value) */
        __EI();                     /* Interrupt enable */
}

void OUTPUT67_start(void)
{
        IO_OCS67.hword = 0xEC3C;    /* bit5-4 = 11 IOE7, IOE6 Output compare interrupt enable */
        IO_OCS67.hword = 0xEC3F;    /* bit1-0 = 11 CST7, CST6 Compare operation enable */
}

void freerun4_start(void)
{
        IO_TCCSL4.bit.STOP = 0;     /* bit4 = 0 STOP Counting enable */
}


__interrupt void INPUT0_int(void)
{
        IO_OCSL67.byte & = 0xBF;    /* bit6 = 0 IOP6 Clearing of interrupt flag */
        ......
        IO_OCSL67.byte & = 0x7F;    /* bit7 = 0 IOP7 Clearing of interrupt flag */
        ......
}
```

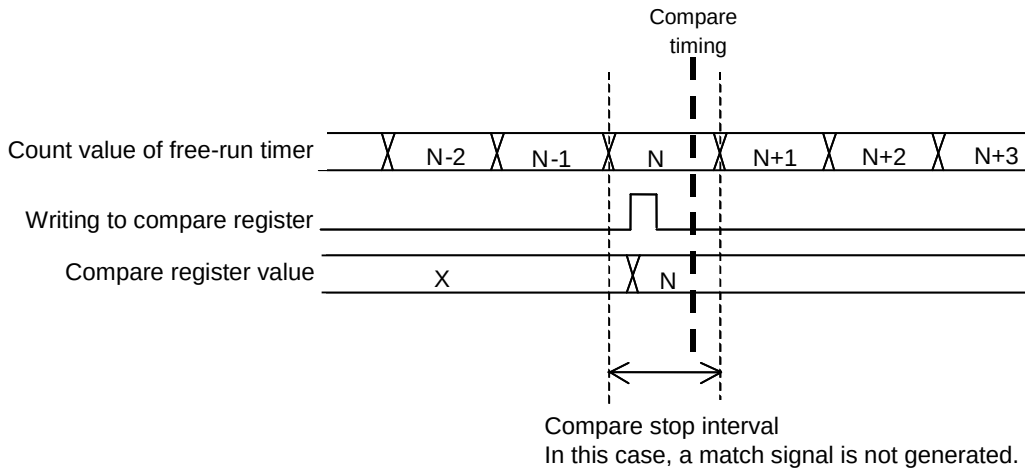
Interrupt routine specification with the vector table is required.
#pragma intvect OUTPUT6_int 58

9. Notes

This section explains the notes of the 32-bit 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 OCCP6 = OCCP7, when compare match occurs, the port inverts only once. (Similar in ch.8 to ch.11)
- When the output level of compare output pins (OCU6 to OCU11) is specified, first stop the compare operation, and then specify it.
- Because the 32-bit output compare is synchronized with the free-run timer, when the free-run timer is stopped, the compare operation is also stopped.
- When the compare mode bit is set to CMOD = "1" also, the interrupt operation occurs for each OCU6 and OCU7 independently. (The same is true for ch.8 to ch.11.)
- 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].
- About read-modify-write
  When the interrupt request flag bits (IOP6 to IOP11) are read with read-modify-write instruction, "1" is read.
- About interrupt
  Please clear the compare match interrupt request flag (IOP6 to IOP11) with "0" to return from the interrupt processing when "1" is set to compare match interrupt request flag (IOPx) of the compare control register, and the compare match interrupt request is permitted next (IOE6 to IOE11 ="1").

Chapter 23: 32-Bit Input Capture

This chapter explains the 32-bit input capture.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Setting
7. Q&A
8. Sample Program
9. Notes

Code : FIP008-1v1-91520-2-E

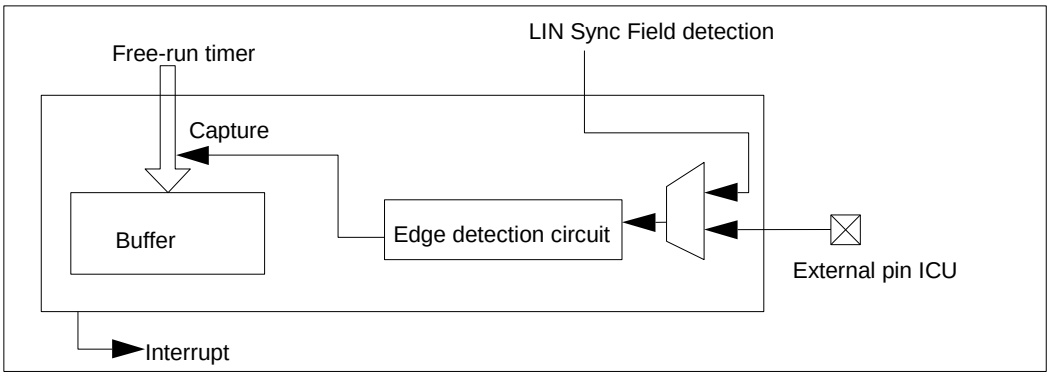
1. Overview

This section explains the overview of the 32-bit 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.

Moreover, the cycle and the pulse width of the input effective edge can be measured.

Figure 1-1 Block Diagram



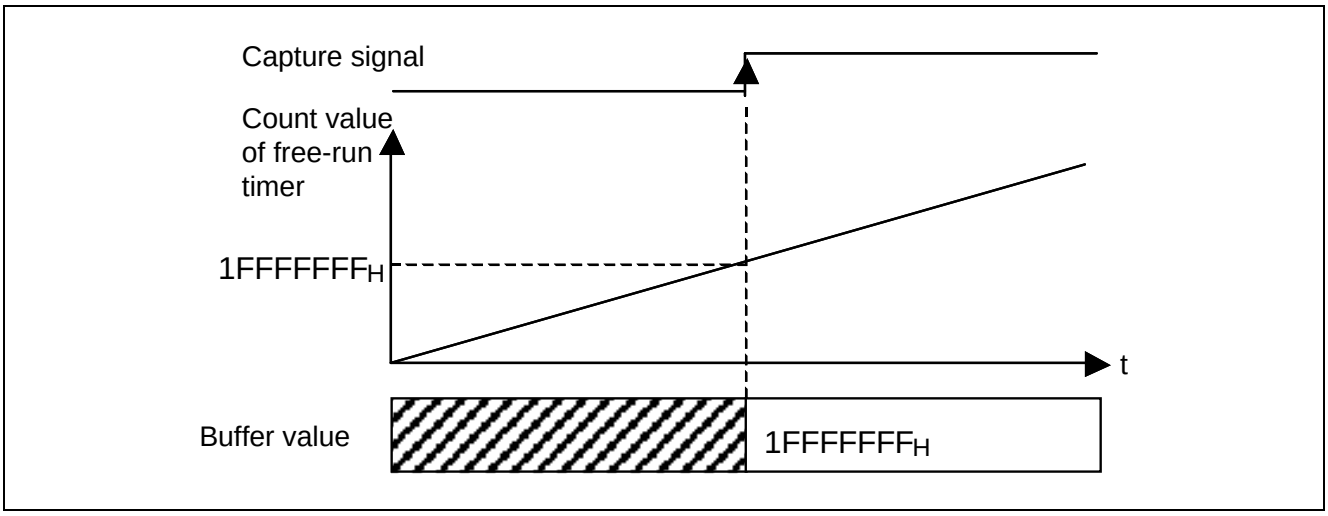
The numbers of available external input pins are shown below.

MB91F52xB (64pin) : 5
MB91F52xD (80pin) : 5
MB91F52xF (100pin) : 6
MB91F52xJ (120pin) : 6
MB91F52xK (144pin) : 6
MB91F52xL (176pin) : 6

2. Features

This section explains features of the 32-bit input capture.

- Format : Edge detection circuit + 32-bit buffer (capture register)
- Number of units : 6
- Edge detection : Rising/falling/both edges
- Interrupt : Edge detection interrupt
- Capture value : Timer count value (00000000_H to $FFFFFFFF_H$)
- Timer : Use free-run timer 3 to 5.
  See "CHAPTER: 32-BIT FREE-RUN TIMER" for the selection method.
- Count accuracy: Peripheral clocks (PCLK)/1, /2, /4, /8, /16, /32, /64, /128, /256 (count clock of the free-run timer)



Cycle and pulse width measurement function

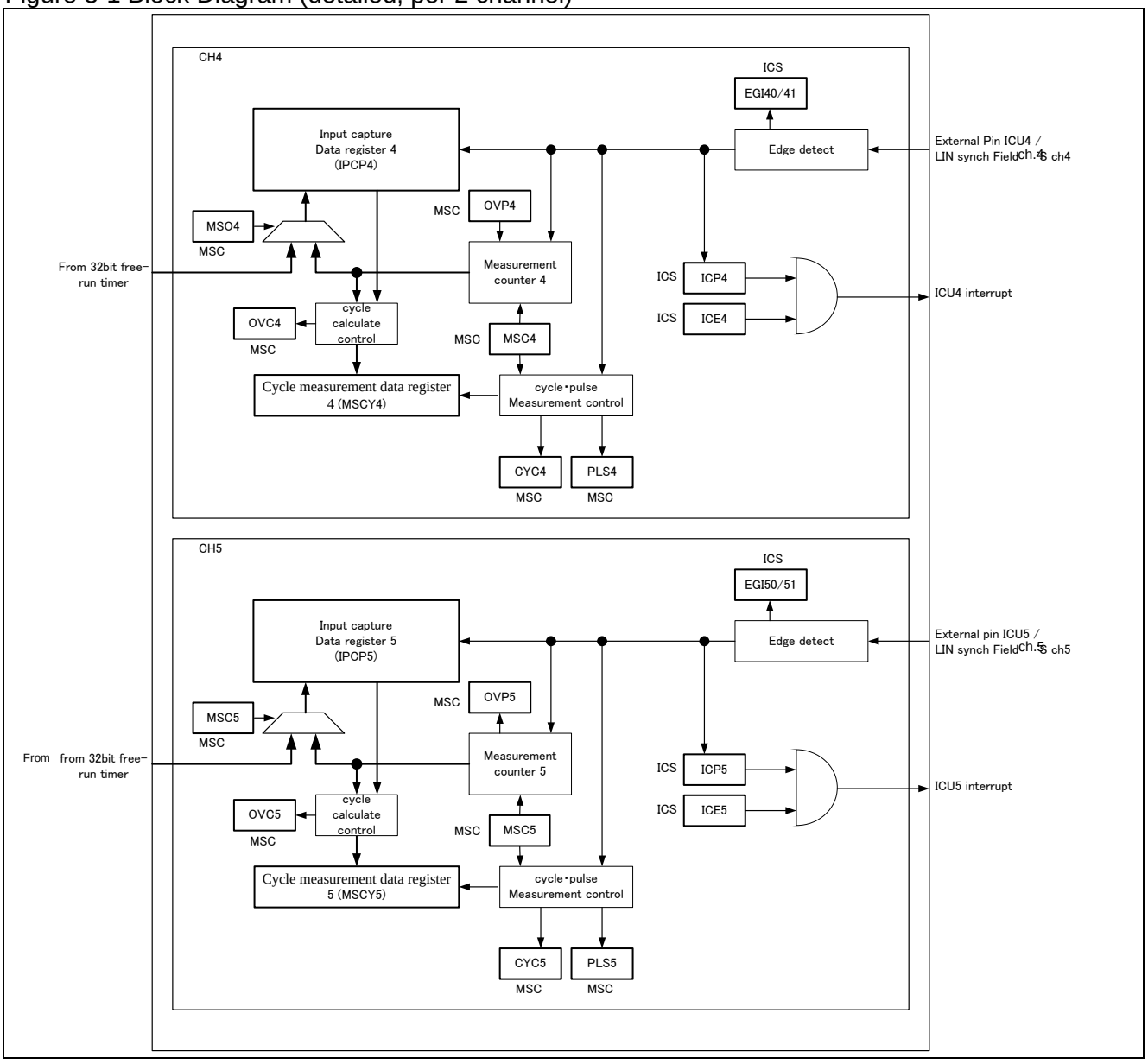
The cycle and the pulse width can be measured by the following settings.

- When setting of rising edge detection : Cycle from rising edge to rising edge
- When setting of falling edge detection : Cycle from falling edge to falling edge
- When setting of both edge detection : Cycle from rising edge to rising edge,
  Cycle from falling edge to falling edge,
  Pulse width from rising edge to falling edge,
  Pulse width from falling edge to rising edge

3. Configuration

This section explains the configuration of the 32-bit input capture.

Figure 3-1 Block Diagram (detailed; per 2 channel)



4. Registers

This section explains registers of the 32-bit input capture.

■ Table of Base Addresses (Base_addr) and External Pins

Table 4-1 Table of Base Addresses (Base_addr) and External Pins

Channel	Base_addr	External pin (ICU input)				
		MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK MB91F52xL
4	0x0FD0	ICU4_2	ICU4_2	ICU4_2	ICU4_1/ ICU4_2	ICU4_0/ ICU4_1/ ICU4_2
5	0x0FD0	ICU5_0/ ICU5_1	ICU5_0/ ICU5_1	ICU5_0/ ICU5_1	ICU5_0/ ICU5_1	ICU5_0/ ICU5_1
6	0x0FDC	ICU6_0	ICU6_0/ ICU6_1	ICU6_0/ ICU6_1	ICU6_0/ ICU6_1	ICU6_0/ ICU6_1
7	0x0FDC	none	none	ICU7_0/ ICU7_1	ICU7_0/ ICU7_1	ICU7_0/ ICU7_1
8	0x0FE8	ICU8_0/ ICU8_1	ICU8_0/ ICU8_1	ICU8_0/ ICU8_1	ICU8_0/ ICU8_1	ICU8_0/ ICU8_1
9	0x0FE8	ICU9_1	ICU9_1	ICU9_1	ICU9_0/ ICU9_1	ICU9_0/ ICU9_1

Table 4-2 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0FD0	IPCP4				Input capture data register 4
0x0FD4	IPCP5				Input capture data register 5
0x0FD8	Reserved	Reserved	LSYNS1	ICS45	LIN SYNCH FIELD switching register 1 Input capture control register 45
0x0FDC	IPCP6				Input capture data register 6
0x0FE0	IPCP7				Input capture data register 7
0x0FE4	Reserved	Reserved		ICS67	Input capture control register 67
0x0FE8	IPCP8				Input capture data register 8
0x0FEC	IPCP9				Input capture data register 9
0x0FF0	Reserved	Reserved		ICS89	Input capture control register 89
0x0118	MSCY4				Cycle measurement data register 4
0x011C	MSCY5				Cycle measurement data register 5
0x0F88	Reserved		MSCH45	MSCL45	Cycle and pulse width measurement control register 45
0x0F68	MSCY6				Cycle measurement data register 6
0x0F6C	MSCY7				Cycle measurement data register 7
0x0F8C	Reserved		MSCH67	MSCL67	Cycle and pulse width measurement control register 67
0x0FF4	MSCY8				Cycle measurement data register 8
0x0FF8	MSCY9				Cycle measurement data register 9
0x0FFC	Reserved		MSCH89	MSCL89	Cycle and pulse width measurement control register 89

4.1. Input Capture Data Register : IPCP

The bit configuration for the input capture data register is shown.

This register can hold and read the count value or the pulse width measurement data value of the free-run timer using a change in the input signal from the external source as a trigger.
x: Channel number 4, 6, 8
y: Channel number 5, 7, 9

■ 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							

[bit31 to bit0] CP[31:0] :
When MSCL.MSCx or MSCy is "0", this register indicates the value of free-run timer at the edge detection.
When MSCL.MSCx or MSCy is "1", this register indicates the value of the pulse width at the edge detection.

Note:

When accessing this register, use a word access instruction. No data can be written to this register.

4.2. Input Capture Control Register : ICS

The bit configuration the input capture control register is shown.

This register controls the input capture.
x: Channel number 4, 6, 8
y: Channel number 5, 7, 9

■ ICSxy (Input capture xy): 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] ICPn : 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

- 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").

Note: ICPn: n corresponds to the input capture channel numbers.

[bit5, bit4] ICEn : Input capture interrupt request enabled

ICEn	Operation
0	Interrupt disabled
1	Interrupt enabled

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".

Note: ICEn: n corresponds to the input capture channel numbers.

[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)

- These bits 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".

Note: EGn1, EGn0: n corresponds to the input capture channel numbers.

4.3. LIN SYNCH FIELD Switching Register : LSYNS

The bit configuration for the LIN SYNCH FIELD switching register is shown.

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 4-9): Address 0FDA$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LSYN91	LSYN90	LSYN81	LSYN80	LSYN7	LSYN6	LSYN5	LSYN4
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:

The input for the input capture must be switched while the capture is inactive (ICS.EG[n1:n0]= "00").

[bit7, bit6] LSYN91, LSYN90 : Input capture ch.9 input selection

LSYN91, LSYN90	Input selection
00	External pin input (ICU9)
01	LIN synch field detection signal input from the multi-function serial interface ch.10.
10	LIN synch field detection signal input from the multi-function serial interface ch.11.
11	Setting prohibited. (Operation is not guaranteed.)

[bit5, bit4] LSYN81, LSYN80 : Input capture ch.8 input selection

LSYN81, LSYN80	Input selection
00	External pin input (ICU8)
01	LIN synch field detection signal input from the multi-function serial interface ch.8.
10	LIN synch field detection signal input from the multi-function serial interface ch.9.
11	Setting prohibited. (Operation is not guaranteed.)

[bit3 to bit0] LSYN7 to LSYN4 : Input capture ch.4 to ch.7 input selection

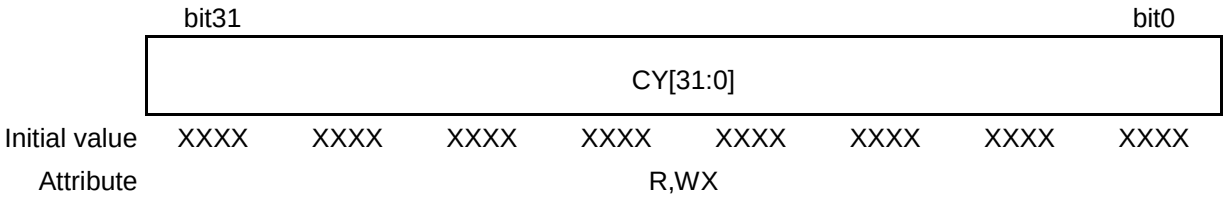
LSYNn (n=4 to 7)	Input selection
0	External pin input (ICUn)
1	LIN synch field detection signal input from the multi-function serial interface ch.n.

4.4. Cycle Measurement Data Register : MSCY

The cycle measurement data register is shown.

This register stores the measured cycle data value, when an effective edge of the corresponding external input pin signal was detected.

- **MSCY4 (Input capture 4): Address 0118_H (Access: Half-word, Word)**
- **MSCY5 (Input capture 5): Address $011C_H$ (Access: Half-word, Word)**
- **MSCY6 (Input capture 6): Address $0F68_H$ (Access: Half-word, Word)**
- **MSCY7 (Input capture 7): Address $0F6C_H$ (Access: Half-word, Word)**
- **MSCY8 (Input capture 8): Address $0FF4_H$ (Access: Half-word, Word)**
- **MSCY9 (Input capture 9): Address $0FF8_H$ (Access: Half-word, Word)**



When MSCL.MSCn(n=4 to 9) is "0", "0000_0000_H " is set in this register.
When MSCL.MSCn(n=4 to 9) is "1", the cycle value when the edge is detected is set in this register.

Note:

Please use the half-word or the word access instruction for this register. Moreover, data cannot be written in this register.

4.5. Cycle and Pulse Width Measurement Control Register (Upper bit) : MSCH

The bit configuration of the cycle and pulse width measurement control register (upper bit) is shown.

This register controls the input capture.
x: Channel number 4, 6, 8
y: Channel number 5, 7, 9

■ **MSCH45 (Input capture 45): Address $0F8A_H$ (Access: Byte, Half-word, Word)**

■ **MSCH67 (Input capture 67): Address $0F8E_H$ (Access: Byte, Half-word, Word)**

■ **MSCH89 (Input capture 89): Address $0FFE_H$ (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CYCy	CYCx	PLSy	PLSx	OVCy	OVCx	OVPy	OVPx
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] CYCn : Cycle measurement flag

CYCn	Explanation
0	Cycle data from falling edge to falling edge.
1	Cycle data from rising edge to rising edge.

These bits show that the data stored in the cycle measurement data register (MSCYn) is either rising cycle or falling cycle. Whenever an effective edge is detected and measured, these bits are updated.
(Note): CYCn : n corresponds to the channel number of the input capture.

[bit13, bit12] PLSn : Pulse width measurement flag

PLSn	Explanation
0	L pulse width.
1	H pulse width.

These bits show that the data stored in the input capture data register (IPCPn) is either H pulse width or L pulse width. Whenever an effective edge is detected and measured, these bits are updated.
Note: PLSn : n corresponds to the channel number of the input capture.

[bit11, bit10] OVCn : Cycle measurement over flag

OVCn	Explanation
0	The cycle data value is maximum value FFFF_FFFF$_H$ or less.
1	The cycle data value exceeds maximum value FFFF_FFFF$_H$.

These bits show that the data stored in the cycle measurement data register (MSCYn) have exceeded the maximum value. Whenever an effective edge is detected and measured, these bits are updated.
Note: OVCn : n corresponds to the channel number of the input capture.

[bit9, bit8] OVPn : Pulse width measurement over flag

OVPn	Explanation
0	The pulse width data value is maximum value FFFF_FFFF$_H$ or less.
1	The pulse width data value exceeds maximum value FFFF_FFFF$_H$.

These bits show that the data stored in the input capture data register (IPCPn) have exceeded the maximum value. Whenever an effective edge is detected and measured, these bits are updated.
Note: OVPn : n corresponds to the channel number of the input capture.

4.6. Cycle and Pulse Width Measurement Control Register (Lower bit) : MSCL

The configuration of the cycle and pulse width measurement control register (lower bit) is shown.

This register controls the input capture.
x: Channel number 4, 6, 8
y: Channel number 5, 7, 9

- **MSCL45 (Input capture 45): Address 0F8B$_H$ (Access: Byte, Half-word, Word)**
- **MSCL67 (Input capture 67): Address 0F8F$_H$ (Access: Byte, Half-word, Word)**
- **MSCL89 (Input capture 89): Address 0FFF$_H$ (Access: Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	MSCy	MSCx
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

[bit7 to bit2] : Undefined
The read value is always "1". Writing has no effect on operation.

[bit1, bit0] MSCn : Operation mode setting

MSCn	Explanation
0	Input capture operation
1	Measurement operation of cycle and pulse width

These bits select the operation mode when the edge of external input ICUn is detected.
Note: MSCn : n corresponds to the channel number of the input capture.

5. Operation

This section explains the operation of the 32-bit 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.

5.1 Capture and Interrupt Timings

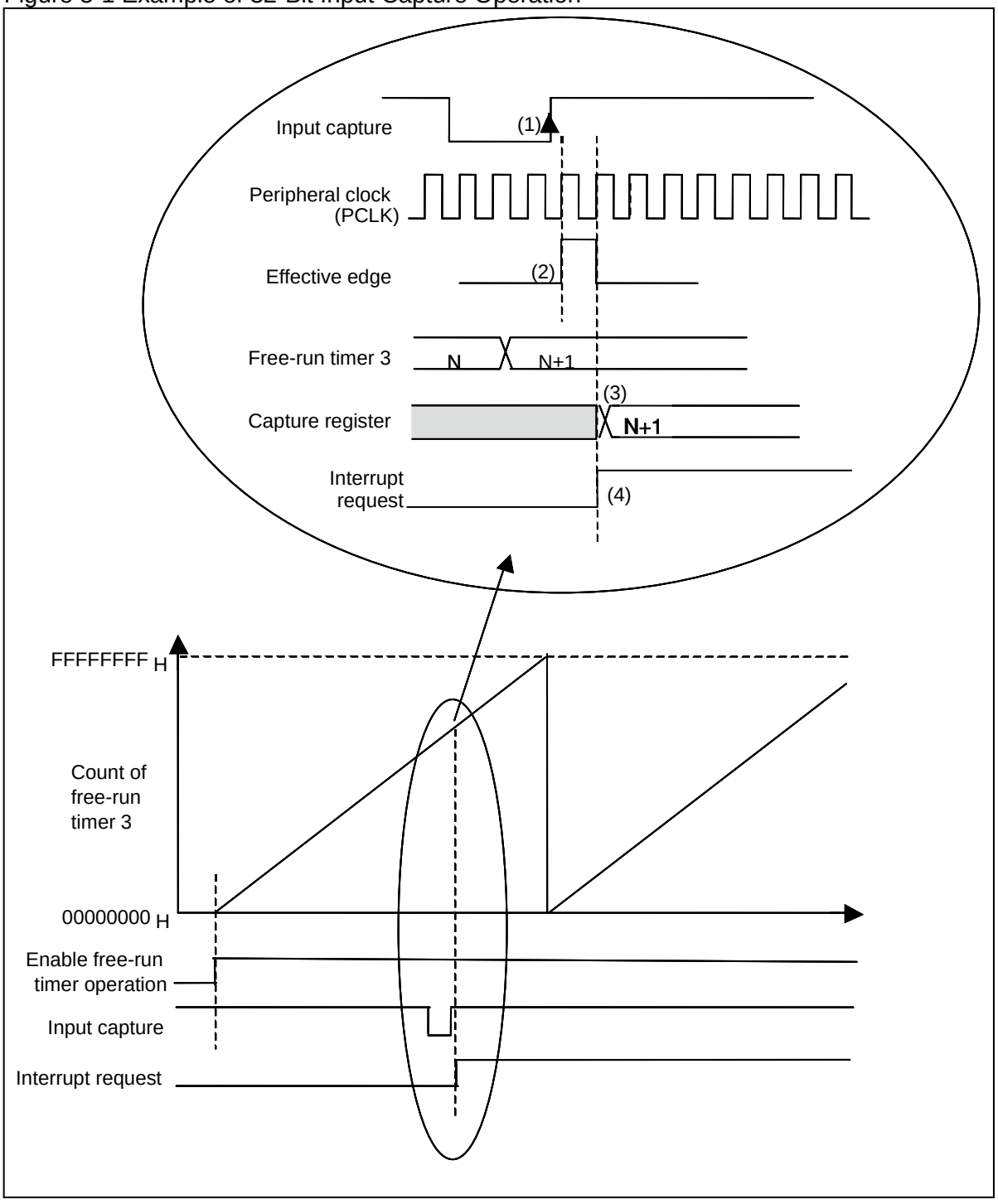
5.2 Edge Detection Specifications for Input Capture And Their Operations

5.3 Cycle and Pulse Width Measurement Operation

5.1. Capture and Interrupt Timings

This section shows capture and interrupts timings.

Figure 5-1 Example of 32-Bit Input Capture Operation

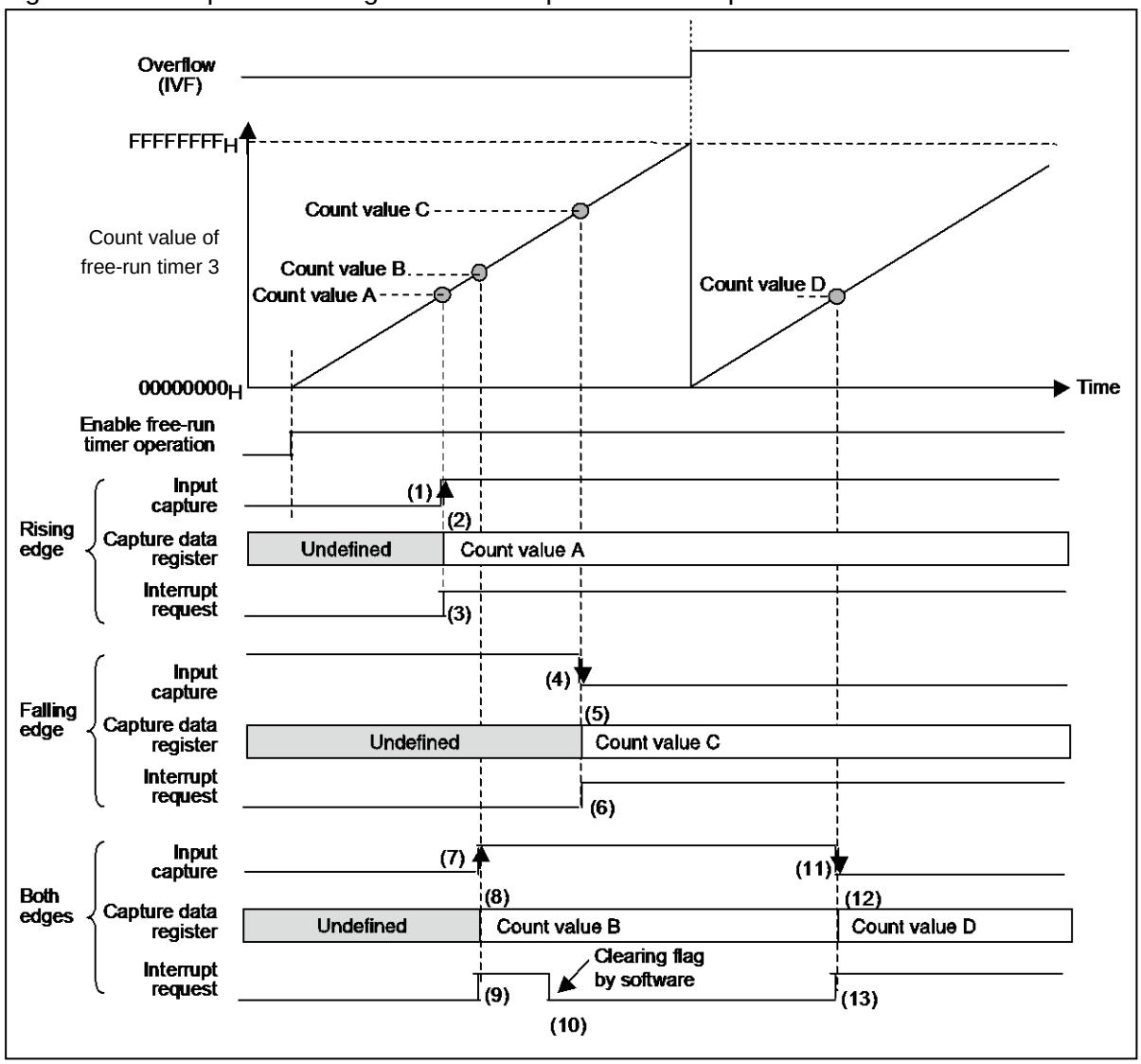


(1) Rising edge of the input signal
(2) Internal signal generated by edge detection (synchronized to the peripheral clock)
(3) Free-run timer value is recorded to the capture register (capture).
(4) Input capture interrupt is generated (ICP(4 to 9)="1").

5.2. Edge Detection Specifications for Input Capture And Their Operations

This section shows edge detection specifications for the input capture and their operations.

Figure 5-2 Example of the Edge Detection Specifications Operation



- 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 ((ICS45.ICP4), (ICS45.ICP5), (ICS67.ICP6), (ICS67. ICP7), ...) 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.

5.3. Cycle and Pulse Width Measurement Operation

This section shows the cycle and pulse width measurement operation.

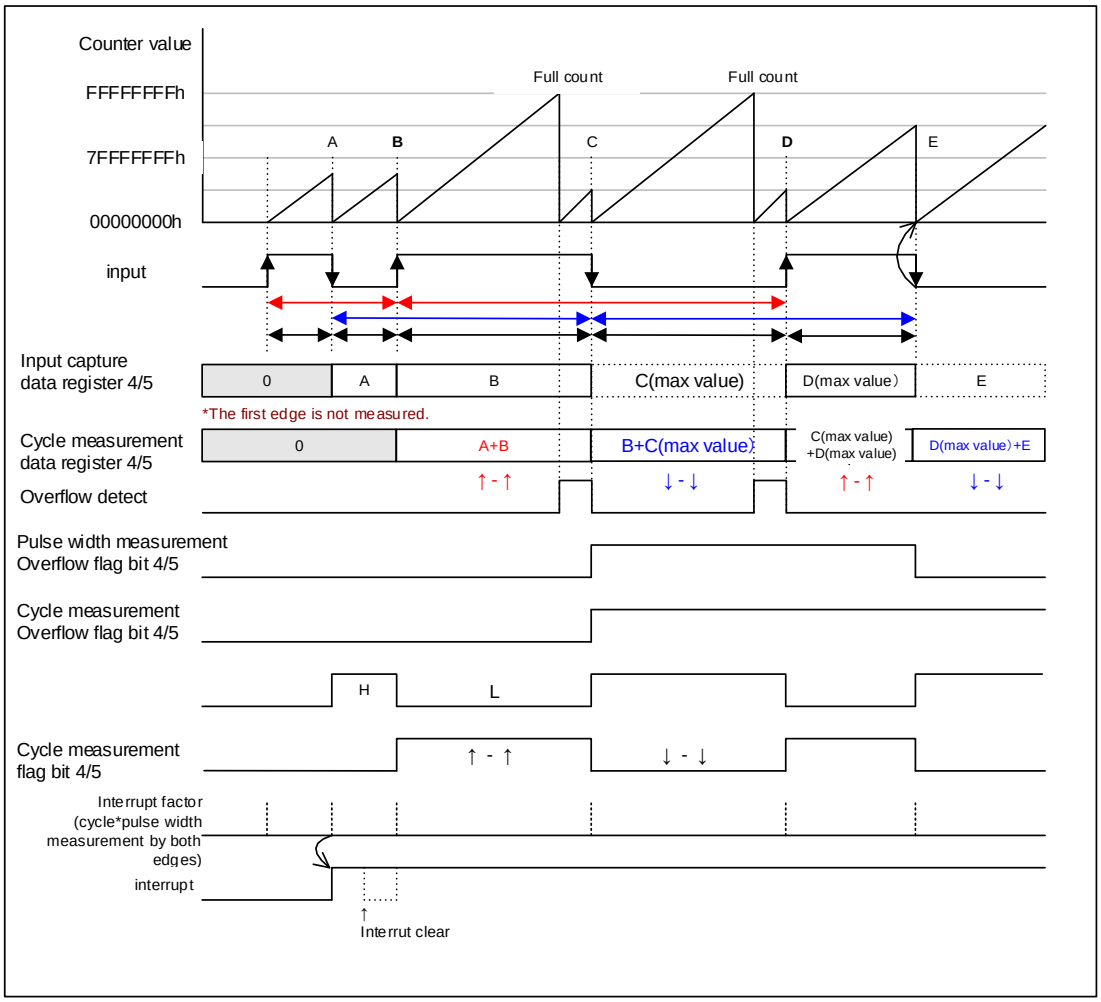
The edge of the external pin input is detected, and the cycle (rising or falling) and the pulse width (H or L) are measured with a counter clocked by the peripheral clock PCLK2.

When measuring, a measurement value is stored in the input capture data register (MSCYn: n=4 to 9) and the pulse width measurement data register (IPCPn: n=4 to 9). At the same time, the input capture is displayed that a cycle of measurement, a type of pulse width, and whether the measurement value exceeds the maximum value in the cycle and pulse width measurement control register (MSCHxy.CYCx/y, PLSx/y, OVCx/y, OVPx/y: x=4, 6, 8 y=5, 7, 9).

The maximum value of the cycle and pulse width is FFFF_FFFFh. When the maximum value is exceeded, the capture value of the counter is displayed as a measuring data. At the same time, the input capture is displayed that the measurement value exceeded the maximum value in the cycle measurement overflow flag (MSCH:OVCn: n=4 to 9) and the pulse width measurement overflow flag (MSCH:OVPn: n=4 to 9) of the cycle and pulse width measurement control register.

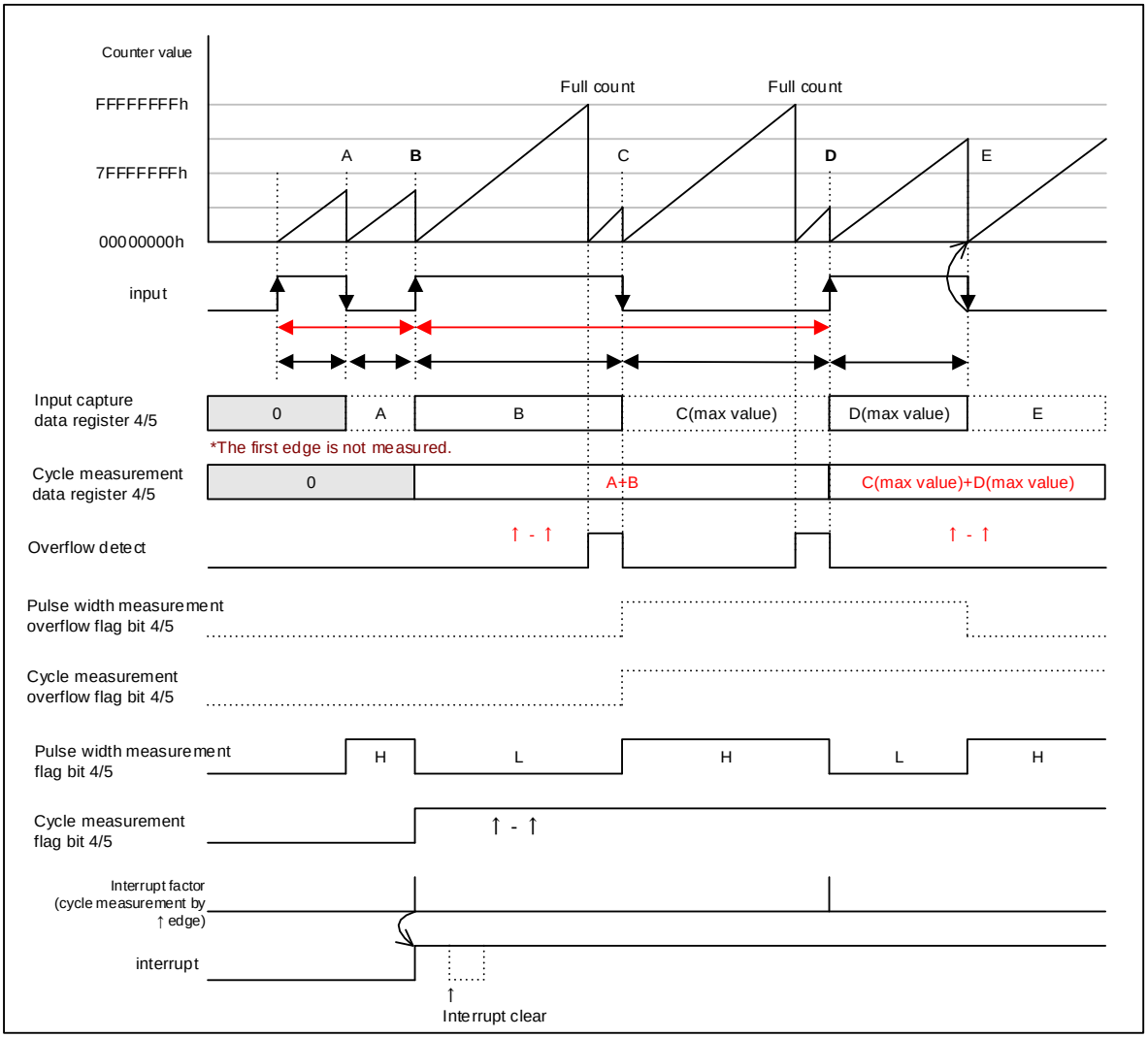
After the measurement operation starts, the measurement is started by cycle or pulse width from the first edge.

Figure 5-3 Example of the Cycle and Pulse Width Measurement Operation (The both edges are specified).



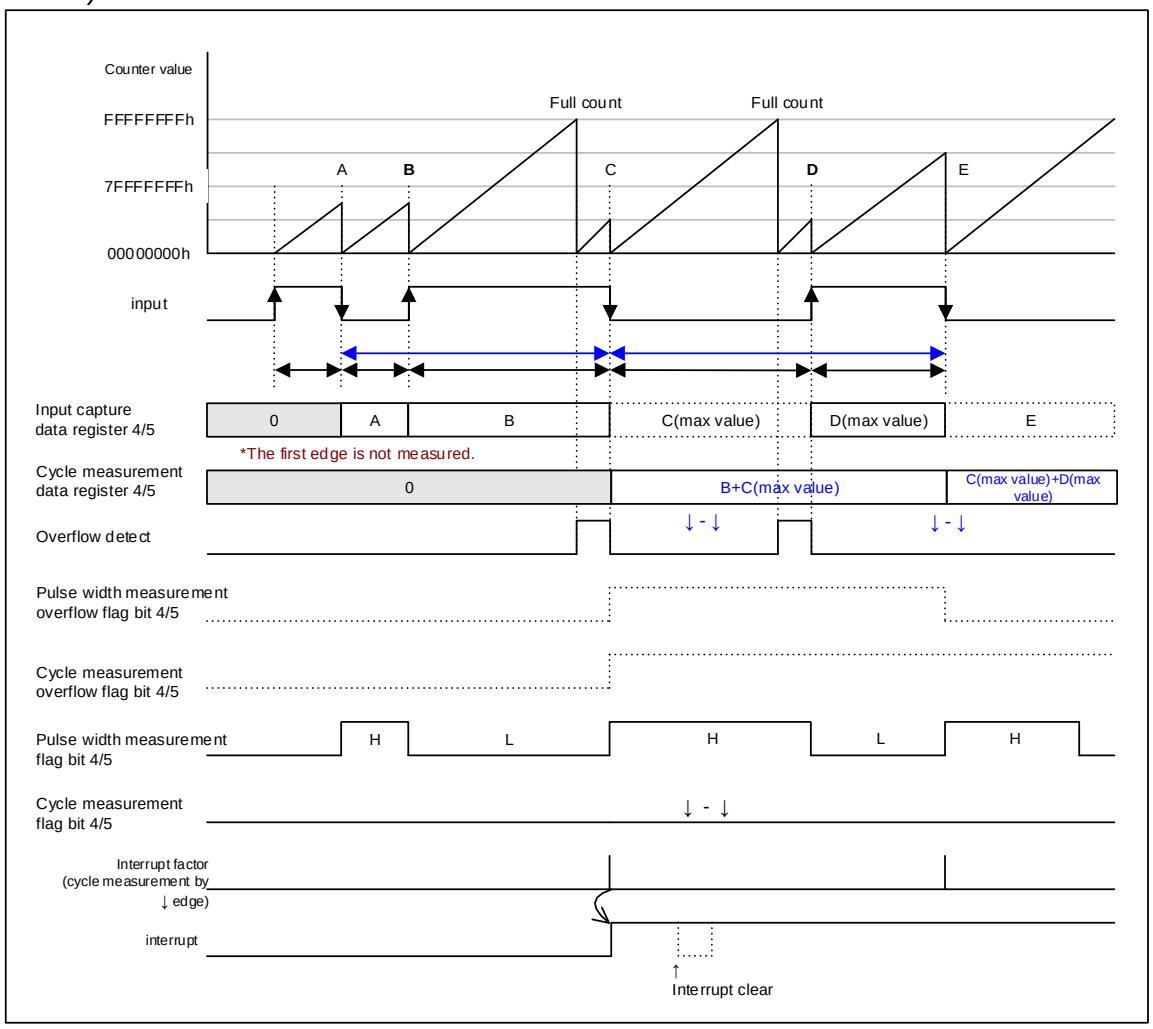
When both edges are specified, cycle of from rising edge to rising edge and from falling edge to falling edge, pulse width of from rising edge to falling edge and from falling edge to rising edge are measured.

Figure 5-4 Example of the Cycle and Pulse Width Measurement Operation (The rising edge is specified).



When the rising edge is specified, cycle of from rising edge to rising edge is measured.
At this time, interrupt is not output though pulse width of from rising edge to falling edge and from falling edge to rising edge is stored in input capture data register (IPCPn).

Figure 5-5 Example of the Cycle and Pulse Width Measurement Operation (The falling edge is specified).



When the falling edge is specified, cycle of from falling edge to falling edge is measured.
At this time, interrupt is not output though pulse width of from rising edge to falling edge and from falling edge to rising edge is stored in input capture data register (IPCPn).

6. Setting

This section explains setting of the 32-bit input capture.

Table 6-1 Settings Required for Using Input Capture

Configuration	Register to be configured	Setting method
Free-run timer setting	See "CHAPTER: 32-BIT FREE-RUN TIMER".	—
Free-run timer activation		
Setting for switching inputs between input pins ICU4 to ICU9 and input capture	If the linkage function for multi-function serial interface is used: LIN SYNCH FIELD switching register (LSYNS1) External input: Settings of the LIN SYNCH FIELD switching register (LSYNS1), Setting of ICU4 to ICU9 pins (See "CHAPTER: I/O PORTS").	See 7.2.
Effective edge polarity selection for external input	Input capture control registers (ICS45), (ICS67), (ICS89)	See 7.1.
Operation mode setting	Operation mode setting bit (MSCn) is set.	See 7.7.

Table 6-2 Settings Required for Performing Input Capture Interrupt

Configuration	Register to be configured	Setting method
Input capture interrupt vector and input capture interrupt level settings	See "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".	See 7.3.
Input capture interrupt setting Interrupt request clear Interrupt request enable	Input capture control registers (ICS45), (ICS67), (ICS89)	See 7.5.

7. Q&A

This section explains Q&A of the 32-bit input capture.

7.1 Effective Edge Polarity of External Input: Types and How to Select Them

7.2 How to Enable External Input Pins (ICU4 to ICU9)

7.3 About Interrupt Related Registers

7.4 About Interrupt Types

7.5 How to Enable Interrupt

7.6 How to Measure the Pulse Width of the Input Signal

7.7 How to Set the Operation Mode

7.1. Effective Edge Polarity of External Input: Types and How to Select Them

This section shows 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 (ICS45.EG[41:40]), (ICS45:EG[51:50]), (ICS67.EG[61:60]), (ICS67:EG[71:70]), (ICS89.EG[81:80]), (ICS89:EG[91:90]).

Operation	Effective edge polarity bits of the external input (EG[n1:n0] n=4 to9)
To select rising edge	Select "01".
To select falling edge	Select "10".
To select both edges	Select "11".

7.2. How to Enable External Input Pins (ICU4 to ICU9)

This section shows how to enable setting of external input pins (ICU4 to ICU9).

Set the LSYNS1 register for external pin input. Also, set the ICU pin for peripheral input. For information on the setting method of peripheral input, see "CHAPTER: I/O PORTS".

7.3. About Interrupt Related Registers

This section shows interrupt related registers.

Input capture interrupt vector and input capture interrupt level settings

See "Table of Interrupt Vector" in "APPENDIX" for interrupt number.

Set interrupt level by the ICR register. For details of the interrupt levels, see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

Interrupt request flags ((ICS45.ICP4), (ICS45.ICP5), (ICS67.ICP6), (ICS67.ICP7), (ICS89.ICP8), (ICS89.ICP9)) are not cleared automatically. Therefore, clear the input capture interrupt request flags by writing "0" using software before returning from interrupt processing.

7.4. About Interrupt Types

This section shows interrupt types.

There are 2 types of interrupts.

When the input capture operates:	The interrupt is generated by an edge detection of the input signal.
When the cycle and pulse width measurement operates:	The interrupt is generated by an edge detection of the input signal, and completion of measurement of cycle or pulse width.

7.5. How to Enable Interrupt

This section shows how to enable interrupt.

Set enable interrupt request and interrupt request flag.
You can configure the interrupt enable setting using the following interrupt request enable bits:
(ICS45.ICE4), (ICS45.ICE5), (ICS67.ICE6), (ICS67.ICE7), (ICS89.ICE8), (ICS89.ICE9)

Operation	Interrupt request enable bits (ICE4 to ICE9)
Interrupt disabled	Set "0".
Interrupt enabled	Set "1".

You can clear the interrupt request using the following interrupt request flags:
(ICS45.ICP4), (ICS45.ICP5), (ICS67.ICP6), (ICS67.ICP7), (ICS89.ICP8), (ICS89.ICP9)

Operation	Interrupt request flag bits (ICP4 to ICP9)
Interrupt request clear	Write "0".

7.6. How to Measure the Pulse Width of the Input Signal

This section shows how to measure the pulse width of the input signal.

The setting example in ch.4 is shown.

(1) Set the operation mode to the cycle measurement mode. (MSCL.MSC4: 1)
(2) Specify both edges for the edge detection. (ICSL.EG41-40: 11_B)
(3) Set enable to the interrupt request bit. (ICSL.ICE4: 1)
(4) Clear the interrupt request flag. (ICSL.ICP4: 0)

The cycle measurement data is stored to MSCY4 and the pulse width measurement data is stored in input capture data register IPCP4.

7.7. How to Set the Operation Mode

This section shows how to set the operation mode.

You can select the operation mode when the edge is detected using the MSCn bit.

MSCL.MSCn=0: Input capture operation
MSCL.MSCn=1: Measurement operation of cycle and pulse width

8. Sample Program

This section explains the sample program of the 32-bit input capture.

Setting procedure example 1

Detect the rising edge of the pulse for input to ICU4 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.3 control — Register name.Bit.name

Control register setting	TCCSH3/TCCSL3
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	TCDT3

-Port — Register name.Bit name

Port ICU4 input setting	See "CHAPTER: I/O PORTS".

-Input capture control — Register name.Bit name

Control register setting	ICS45
Interrupt request flag>>	.ICP5, ICP4
Interrupt request enabled>>	.ICE5, ICE4
ch.5 Effective edge polarity selection>>	.EG51, EG50
ch.4 Effective edge polarity selection>>	.EG41, EG40

-Interrupt-related — Register name.Bit name

Sets an interrupt level.	ICR36
I flag setting	(CCR)

-Variable setting

2. Activation

-Input capture ch.4 activation — Register name.Bit name

Interrupt control	ICS45.ICE4

-Free-run timer ch.3 activation — Register name.Bit name

Count operation activation	TCCSL3.STOP

3. Interrupt

-Interrupt processing — Register name.Bit name

Clearing of interrupt request flag	ICS45.ICP4
(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: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

Program example 1

```
void INPUT0_sample_1(void)
{
        freerun0_initial();
        INPUT4_initial();
        INPUT4_start();
        freerun0_start();
}

void freerun0_initial(void)
{
        IO_TCCS3.word = 0x0041;     /* Setting value=0000_0000_0100_0001 */
                                    /* bit15 = 0       ECKE internal clock source */
                                    /* bit14 to 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 to 0 = 0001 */
        IO_TCDT3 = 0x0000;          /* Initialization of timer data value */
}

void INPUT4_initial(void)
{
        PORT_SETTING_ICU4_IN();  /* Set the ICU0 pin for peripheral input. */

        IO_ICS45.byte = 0x01;       /* Setting value=0000_0001 */
                                    /* bit7 to 6 = 00 ICP5, 4, 0 Interrupt request flag clear */
                                    /* bit5 to 4 = 00 ICE5, 4, 0 Interrupt disabled */
                                    /* bit3 to 2 = 00 EG51, EG50 ch.5 No edge detected */
                                    /* bit1 to 0 = 01 EG41, EG40 ch.4 Rising edge detected */

        IO_ICR[36].byte = 0x10;     /* Input capture ch.4 interrupt level setting (any value) */
        __EI();                     /* Interrupt enabled */
        count = 0;
}

void INPUT4_start(void)
{
        IO_ICS45.bit.ICE4 = 1;      /* bit4 = 1 ICE45 ch.4 Interrupt enabled */
}

void freerun3_start(void)
{
        IO_TCCSL3.bit.STOP = 0;     /* bit6 = 0 STOP count enabled */
}

__interrupt void INPUT4_int(void)
{
        IO_ICS45.bit.ICP4 = 0;      /* bit6 = 0 Clearing of ICP4 effective edge detection flag */
} count++;

Specification of interrupt routine required in vector table
#pragma intvect INPUT4_int 52
```

9. Notes

This section explains notes of the 32-bit input capture.

- **Input capture data register**

  The input capture register value is undefined after a reset.

  Reading of the input capture data register must be performed in word (32-bit mode) access.

- **Cycle measurement data register**

  Reading of the cycle measurement data register must be performed in word (32-bit mode) access.

- **Read-modify-write**

  The input capture interrupt request bits (ICP4 to ICP9) are "1" when read using a read-modify-write.

- **Notes when interrupt is processed**

  - It is necessary to clear the interrupt request flag (ICPn) to "0" to return from the interrupt processing, when "1" is set to interrupt request flag (ICPn) of input capture control register (ICS), and the interrupt request is set to enable (ICS.ICEn=1).
  - When the cycle and pulse width measurement operates, the edge of the external input pin (ICUn) is detected while the interrupt routine is being processed, and the cycle and the pulse width are measured, measured latest information is shown in the cycle measurement data register (MSCYn) and the input capture data register (IPCPn).

Chapter 24: 16-Bit Free-Run Timer



This chapter explains the 16-bit free-run timer.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS17-1v0-91520-2-E

1. Overview

This section is explains the overview of 16-bit free-run timers.

The free-run timer consists of one free-run timer simultaneous activation, three 16-bit free-run timers (1ch each and total of 3ch), and one free-run timer selector.

2. Features

This section is explains the features of 16-bit free-run timers.

■ Function of the Free-run Timer Simultaneous Activation

- Of the three 16-bit free-run timers, the selected 16-bit free-run timers can be activated or cleared simultaneously.
- It simultaneously controls the timer clear bit (SCLR) and the timer enable bit (STOP) of the timer state register (TCCS) for each 16-bit free-run timer that enables the free-run timer simultaneous activation.
- If the timers are not activated or cleared simultaneously, it is possible to activate or clear each 16-bit free-run timer individually by setting the timer enable bit (STOP) and timer clear bit (SCLR) of the timer state register (TCCS).

■ Function of the 16-bit Free-run Timer

- The 16-bit free-run timer consists of 16-bit up/down counter, control register, 16-bit compare clear register (with buffer register), and prescaler.
- You can select one of the nine counter operation clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128, ϕ/256). (ϕ: peripheral clock)
- A compare clear interrupt will be generated when a compare clear register matches the 16-bit free-run timer upon comparison of the two. "0" detection interrupt will be generated while the 16-bit free-run timer is detecting the count value "0".
- The compare clear register comes with selectable buffer registers (Data written to this buffer register will be transferred to the compare clear register). Once data is written to the buffer after the 16-bit free-run timer has stopped, the transfer will be executed immediately. If the timer value "0" is detected while the 16-bit free-run timer is active, data will be transferred from the buffer.
- If there is a reset or if there is a compare match with the software clear or compare clear register, the counter value will be reset to "0000_H".
- This counter output value can be used as a clock count of the output compare, the input capture and the A/D activation compare.

■ Function of the Free-run Timer Selector

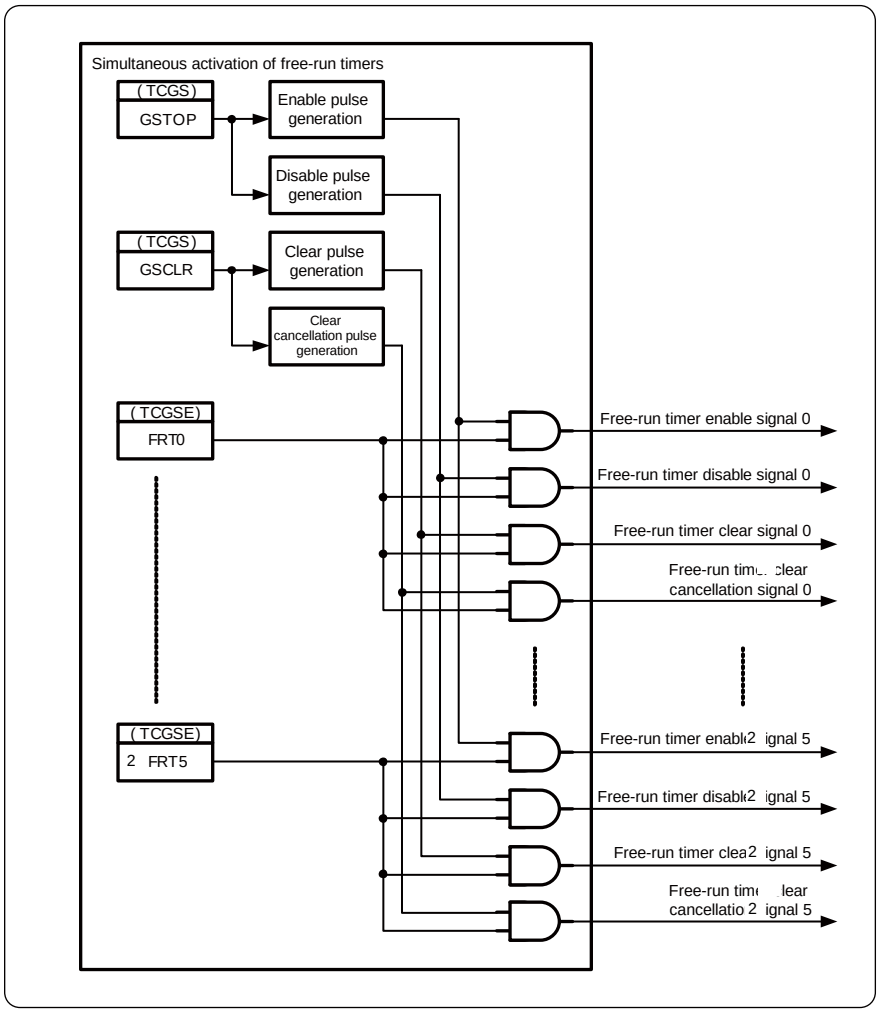
- The free-run timer selection register allows you to select the assignment of the free-run timer from among the 16-bit output capture, the 16-bit input capture, and the A/D activation compare.

3. Configuration

This section is explains the configuration of 16-bit free-run timers.

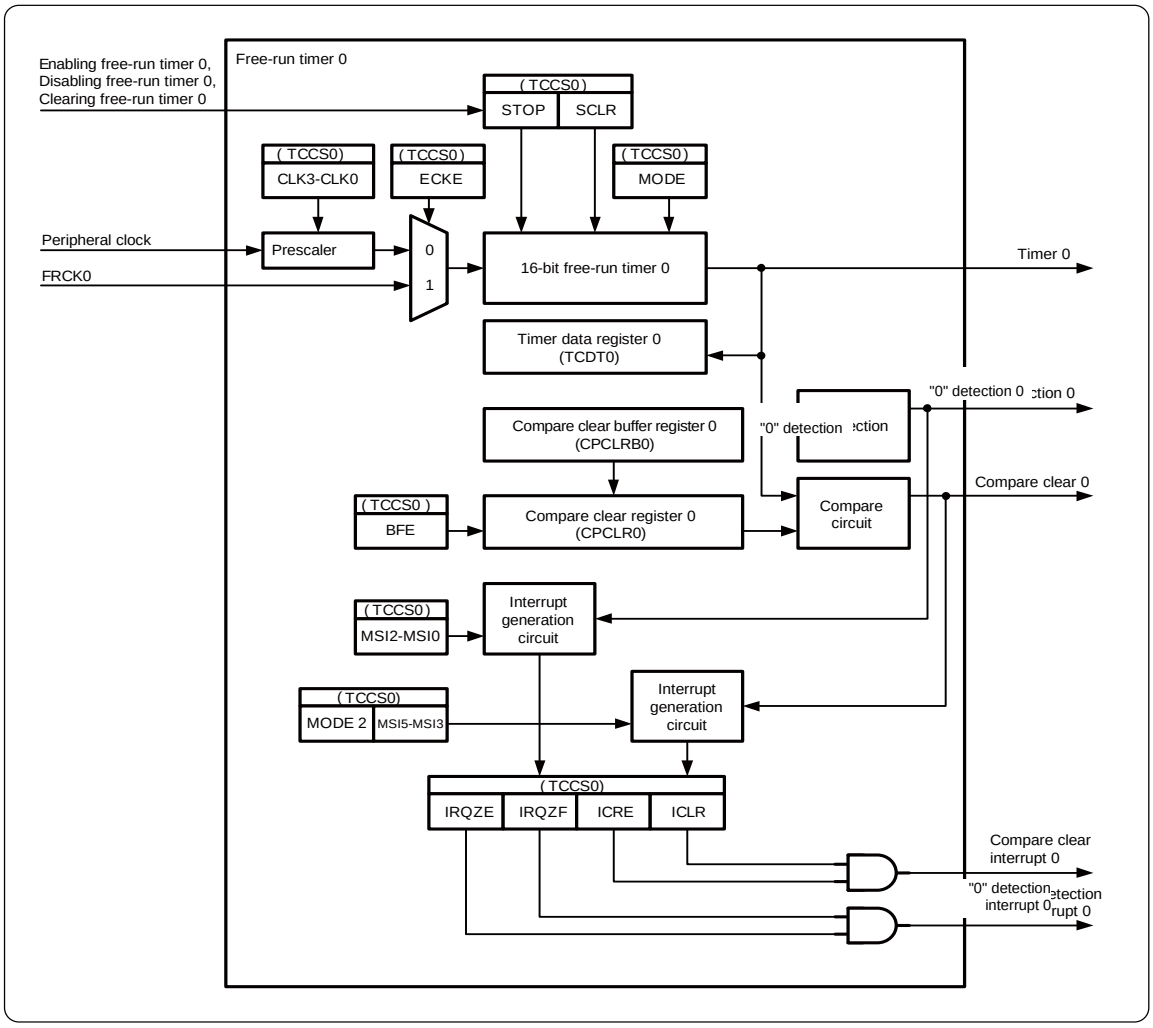
■ Configuration of the 16-bit Free-run Timer Simultaneous Activation

Figure 3-1 Configuration of the Free-run Timer Simultaneous Activation



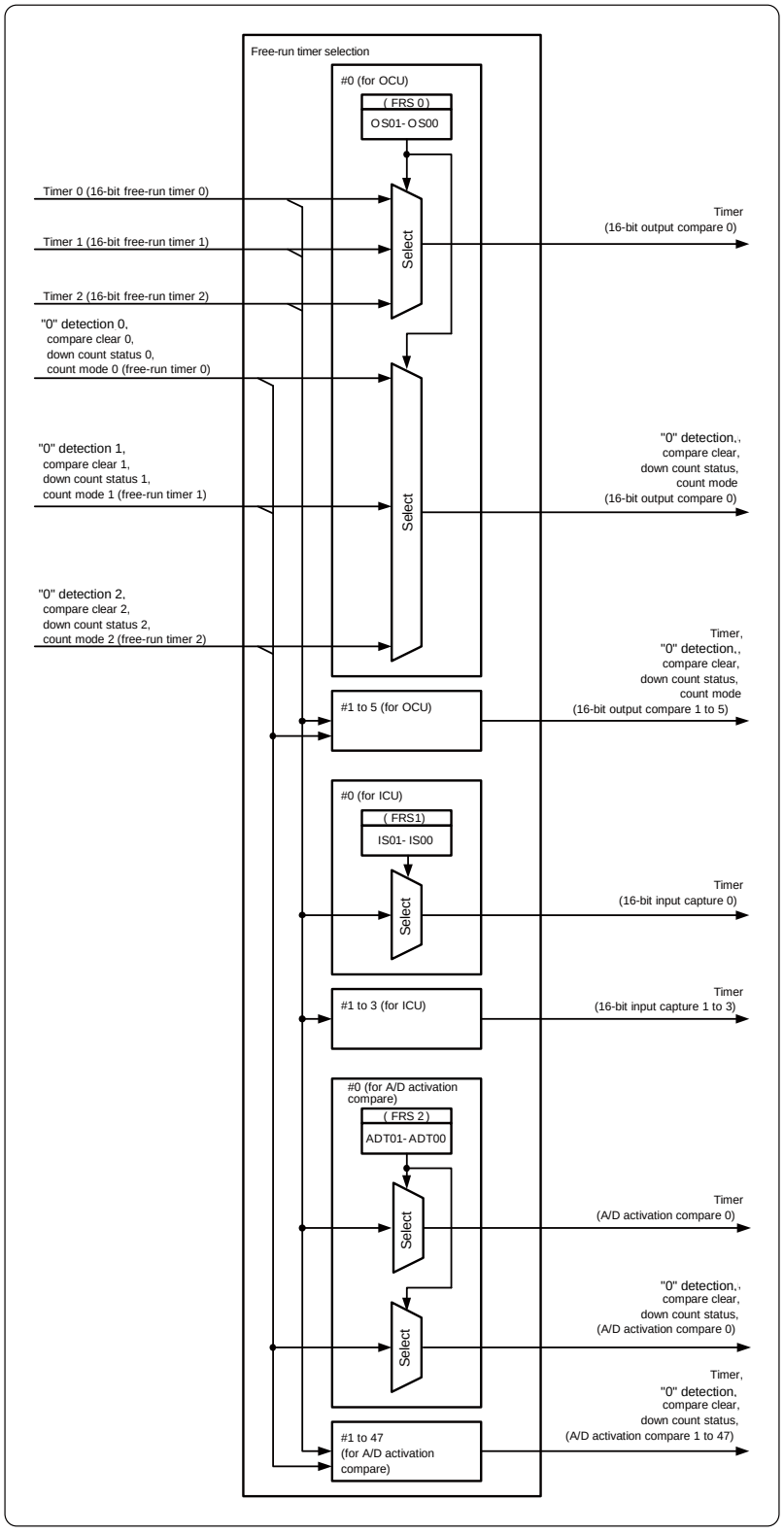
■ Configuration of the 16-bit Free-run Timer

Figure 3-2 Configuration of the 16-bit Free-run Timer (only one channel)



■ Configuration of the Free-run Timer Selector

Figure 3-3 Configuration of the Free-run Timer Selector



4. Registers

This section is explains the registers of 16-bit free-run timers.

■ Table of external pins

Channel	External pins (FRCK)	
	MB91F52xB	MB91F52xD, MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL
0	FRCK0_0	FRCK0_0
1	FRCK1_1	FRCK1_0/FRCK1_1
2	FRCK2_0	FRCK2_0

■ List of registers

Table 4-1 List of Registers for the Free-run Timer Simultaneous Activation

Address	+0	+1	+2	+3
0x1200	Timer synchronous activation register (TCGS)	Reserved	Reserved	Timer synchronous activation enable register (TCGSE)

Table 4-2 List of Registers for the 16-bit Free-run Timer

Address	+0	+1	+2	+3
0x1204	Compare clear buffer register 0 (CPCLRB0) Compare clear register 0 (CPCLR0)		Timer data register (TCDT0)	
0x1208	Timer state control register 0 (TCCS0)			Reserved
0x120C	Compare clear buffer register 1 (CPCLRB1) Compare clear register 1 (CPCLR1)		Timer data register (TCDT1)	
0x 1210	Timer state control register 1 (TCCS1)			Reserved
0x1214	Compare clear buffer register 2 (CPCLRB2) Compare clear register 2 (CPCLR2)		Timer data register (TCDT2)	
0x1218	Timer state control register 2 (TCCS2)			Reserved

Table 4-3 List of Registers for the Free-run Timer Selector

Address	+0	+1	+2	+3
0x1234	Reserved	Free-run timer selection register 0 (FRS0)		
0x1238	Reserved	Reserved	Free-run timer selection register 1 (FRS1)	
0x123C	Free-run timer selection register 2 (FRS2)			
0x1240	Free-run timer selection register 3 (FRS3)			
0x1244	Free-run timer selection register 4 (FRS4)			
0x12D0	Free-run timer selection register 5 (FRS5)			
0x12D4	Free-run timer selection register 6 (FRS6)			
0x12D8	Free-run timer selection register 7 (FRS7)			

4.1. Registers for the Free-run Timer Simultaneous Activation

Registers for the free-run timer simultaneous activation are explained.

The free-run simultaneous activation consists of the timer synchronous activation register and the timer synchronous enable register.

4.1.1. Timer Synchronous Activation Register : (TCGS)

This section explains the bit structure of the timer synchronous activation register.

The timer synchronous activation register (TCGS) is used for enabling simultaneous timer and controlling simultaneous timer clear of the free-run timer. The free-run timer for enabling and clearing the simultaneous timer can be set by the timer synchronous activation enable register (TCGSE).

■ TCGS: Address 1200_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						GSTOP	GSCLR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W

[bit7 to bit2] Reserved

Always write "0" to these bits.

[bit1] GSTOP: Simultaneous timer enable bit

GSTOP	Function	
	Read	Write
0	"0" is always read out.	Enable the counting simultaneously. (Start the counting)
1		Disable the counting simultaneously. (Stop the counting)

- This bit is used to simultaneously start/stop the counting of the free-run timer specified by the timer synchronous activation enable register (TCGSE).
- When this bit is set to "0":
  Starts the counting of the 16-bit free-run timer of the free-run timer specified by the timer synchronous activation enable register (TCGSE). At this time, the STOP bit of the timer state control register (TCCS) of the free-run timer specified by the timer synchronous activation enable register (TCGSE) will be cleared to "0".
- When this bit is set to "1":
  Stops the counting of the 16-bit free-run timer of the free-run timer specified by the timer synchronous activation enable register (TCGSE). At this time, the STOP bit of the timer state control register (TCCS) of the free-run timer specified by the timer synchronous activation enable register (TCGSE) will be set to "1".
- The value read out is always "0".

[bit0] GSCLR: Simultaneous timer clear bit

GSCLR	Function	
	Read	Write
0	"0" is always read out	Counter will not be initialized
1		Counter will be initialized to "0000_H" simultaneously.

- This bit is used to initialize the free-run timer 16-bit free-run timer specified by the timer synchronous activation enable register (TCGSE) to "0000_H".
- When this bit is set to "1":
  Initializes the 16-bit free-run timer of the free-run timer specified by the timer synchronous activation enable register (TCGSE). At this time, the SCLR bit of the timer state control register (TCCS) of the free-run timer specified by the timer synchronous activation enable register (TCGSE) will be set to "1".
- When this bit is set to "0":
  Cancels the instruction for initializing the 16-bit free-run timer of the free-run timer specified by the timer synchronous activation enable register (TCGSE). At this time, the SCLR bit of the timer state control register (TCCS) of the free-run timer specified by the timer synchronous activation enable register (TCGSE) will be cleared to "0".
- The value read out is always "0".

4.1.2. Timer Synchronous Activation Enable Register : TCGSE

This section explains the bit structure of timer synchronous activation enable register.

The timer synchronous activation enable register (TCGSE) sets the free-run timer that enables the simultaneously activation/clear.

■ TCGSE: Address 1203_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					FRT2	FRT1	FRT0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W

[bit7 to bit3] Reserved

Always write "0" to these bits.

[bit2 to bit0] FRT2 to FRT0: Simultaneous activation/clear setting bits

FRT2 to FRT0	Function
0	Do not allow simultaneous activation/clear
1	Allow simultaneous activation/clear

- These bits allow you to set the free-run timer that enables the simultaneous activation/clear.
- When these bits are set to "0":
  The free-run timer will not be activated nor cleared when configuring the timer synchronous activation register (TCGS).
- When these bits are set to "1":
  The free-run timer will be activated or cleared when configuring the timer synchronous activation register (TCGS).

4.2. Registers for the 16-bit Free-run Timer

Registers for the 16-bit free-run timer are explained.

The 16-bit free-run timer consists of the compare clear buffer register, the compare clear register, the timer data register, and the timer state control register.

4.2.1. Compare Clear Buffer Register : CPCLRB/ Compare Clear Register : CPCLR

This section explains the bit structures of the compare clear buffer register and compare clear register.

The compare clear buffer register (CPCLRB) is a 16-bit buffer register contained in the compare clear register (CPCLR).
The CPCLRB and CPCLR registers are located at the same address.

■ CPCLRB0, 1, 2: Address 1204_H, $120C_H$, 1214_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] CL15 to CL00: Compare clear value buffer bits

CL15 to CL00	Function
	Compare clear value buffer

- The compare clear buffer register is a buffer register located at the same address of the compare clear register (CPCLR).
- If the buffer function is disabled (BFE:bit23 of timer state control register (TCCS) is 0) or the free-run timer stops, the value of the compare clear buffer register will be immediately transferred to the compare clear register.
- If the buffer function is enabled, the value will be transferred to the compare clear register when the count value "0" of the 16-bit free-run timer is detected.

Note:

Do not set "0000_H" for the compare clear buffer register.
When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

■ CPCLR0, 1, 2: Address 1204_H, $120C_H$, 1214_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] CL15 to CL00: Compare clear value bits

CL15 to CL00	Function
	Compare clear value

- The compare clear register is used for comparison with the count value of the 16-bit free-run timer.
- In the up-count mode, if this register matches the count value of the 16-bit free-run timer, the 16-bit free-run timer will be reset to "0000_H".
- In the up/down count mode, if this register matches the count value of the 16-bit free-run timer, the 16-bit free-run timer will be converted from up count to down count or it will be converted from down count to up count when "0" is detected.

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

4.2.2. Timer Data Register : TCDT0 to TCDT2

This section explains the bit structure of the timer data register.

The timer data register (TCDT) reads the count value of the 16-bit free-run timer. It is also possible to set the count value of the 16-bit free-run timer.

- **TCDT0: Address 1206_H (Access: Half-word, Word)**
- **TCDT1: Address $120E_H$ (Access: Half-word, Word)**
- **TCDT2: Address 1216_H (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	T15	T14	T13	T12	T11	T10	T09	T08
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	T07	T06	T05	T04	T03	T02	T01	T00
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit15 to bit0] T15 to T00: Count value bits

T15 to T00	Function
	Count value

- The timer data register is used for reading the count value of the 16-bit free-run timer.
- The count value will be cleared to "0000_H" as soon as reset occurs.
- The timer value can be set by writing a value to this register. However, a value needs to be written while the timer is inactive (STOP:bit22 of timer state control register (TCCS) is 1).
- The 16-bit free-run timer will be initialized as soon as any of the following occurs.
  - Reset
  - While the 16-bit free-run timer is active (STOP:bit22 of timer state control register (TCCS) is 0), the clear bit (SCLR:bit20) of the timer state control register (TCCS) is 1
  - The timer count value matches the compare clear register in the up-count mode (MODE: bit21 of timer state control register (TCCS) is 0)

Note:

The 16-bit free-run timer will not be initialized even when the clear bit (SCLR: bit20) of the timer state control register (TCCS) is set to 1 while the 16-bit free-run timer is inactive (STOP: bit22=1 of timer state control register (TCCS)).
When accessing the timer data register, use a half-word or word access instruction.
If a count value is written during the up/down counter mode (MODE:bit21=1 of timer state control register (TCCS)), an unintended counting may be performed.
To write a count value in the up/down counter mode (MODE:bit21=1 of timer state control register (TCCS)), perform the following steps.

1. Stop the 16-bit free-run timer counter. (Writing "1" in STOP:bit21 of timer state control register (TCCS))
2. Set a count value for the timer data register.
3. Perform software clear. (Writing "1" in SCLR:bit20 of timer state control register (TCCS))
4. Start the 16-bit free-run timer counter.

4.2.3. Timer State Control Register : TCCS0 to TCCS2

This section explains the bit structure of timer state control register.

The timer state control register (TCCS) controls the operation of the 16-bit free-run timer.

■ TCCS0: Address 1208_H (Access: Byte, Half-word, Word)

■ TCCS1: Address 1210_H (Access: Byte, Half-word, Word)

■ TCCS2: Address 1218_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R(RM1), W	R/W	R,W	R,W	R,W	R(RM1), W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0
Initial value	0	1	0	0	0	0	0	0
Attribute	R/W	R,W	R/W	R0,W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				MODE2	MSI5	MSI4	MSI3
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1

[bit31] ECKE: Clock selection bit

ECKE	Function
0	Peripheral clock
1	External clock

- This bit is used for selecting the peripheral clock or external clock as a count clock for the 16-bit free-run timer.
- When this bit is set to "0":
  The peripheral clock is selected. To select the count clock frequency, you will also need to select the clock frequency selection bits (CLK3 to CLK0) of the TCCS register.

- When this bit is set to "1":
  The external clock (FRCK) is selected.

Note:

The count clock will be changed as soon as this bit is set. Therefore, change this bit while the output compare and input capture are inactive.

[bit30] IRQZF: "0" detection interrupt flag bit

IRQZF	Function	
	Read	Write
0	No "0" detected	This bit is cleared
1	"0" detected	This bit remains unaffected

- When the count value of the 16-bit free-run timer is set to "0000_H", this bit will be set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- This bit is cleared when the "0" detection interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
This bit will not be set by software clear (write "1" to the SCLR: bit20 of the timer state control register (TCCS)) while the 16-bit free-run timer is active (STOP:bit22 of timer state control register (TCCS) is 0).
In the up/down count mode (MODE: bit21 of the timer state control register (TCCS) is 1), this bit will be set to "1" when an interrupt configured by the interrupt mask selection bits (MSI2 to MSI0: bit28 to bit26 of the timer state control register (TCCS) is other than "000_B") occurs. If no interrupt occurs, this bit will not be set to "1".
In the up count mode (MODE:bit21=0), this bit will be set every time "0" detection occurs regardless of the value of MSI2 to MSI0: bit28 to bit26.
If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit29] IRQZE: "0" detection interrupt request enable bit

IRQZE	Function
0	Interrupt request disabled
1	Interrupt request enabled

When this bit and interrupt flag bit (IRQZF: bit30) are set to "1", an interrupt request for CPU will be generated.

[bit28 to bit26] MSI2 to MSI0: Interrupt mask selection bits

MSI2	MSI1	MSI0	Function
0	0	0	An interrupt will be generated when there is a match for the first time
0	0	1	An interrupt will be generated when there is a match for the second time
0	1	0	An interrupt will be generated when there is a match for the third time
0	1	1	An interrupt will be generated when there is a match for the fourth time
1	0	0	An interrupt will be generated when there is a match for the fifth time
1	0	1	An interrupt will be generated when there is a match for the sixth time
1	1	0	An interrupt will be generated when there is a match for the seventh time
1	1	1	An interrupt will be generated when there is a match for the eighth time

- When MODE2: bit11 of the timer state control register (TCCS) is 0:
  - These bits are used for configuring the mask count of compare clear interrupt in the up count mode (MODE: bit21 of the timer state control register (TCCS) is 0). In the up/down count mode (MODE: bit21 of the timer state control register (TCCS) is 1), they are used to configure the mask count of "0" detection interrupt.
  - When this bit is set to "0", the interrupt factor will not be masked.
- When MODE2: bit11 of the timer state control register (TCCS) is 1:
  - In the up/down count mode (MODE: bit21 of the timer state control register (TCCS) is 1), these bits are used to configure the mask count of "0" detection interrupt.
  - Settings of the up count mode (MODE: bit21 of the timer state control register (TCCS) is 0) are disabled.

Note:

The value read is a mask counter value.
If a read-modify-write instruction is executed, the value read is a mask register value.
The written data will be written to the mask register.
The written value to the mask register while the free-run timer is active (STOP: bit22 of the timer state control register (TCCS) is 0) will be reloaded to the counter only when the mask counter becomes "0".
The written value to the mask register while the free-run timer is inactive (STOP: bit22 of the timer state control register (TCCS) is 1) will be immediately reloaded to the counter.

[bit25] ICLR: Compare clear interrupt flag bit

ICLR	Function	
	Read	Write
0	No compare clear match	This bit is cleared
1	Compare clear match	This bit remains unaffected.

- This bit will be set to "1" when the compare clear value matches the 16-bit free-run timer value.
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- This bit will be cleared when the compare clear match interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
In the up count mode (MODE: bit21 of the timer state control register (TCCS) is 0), this bit will be set to "1" when an interrupt configured by the interrupt mask selection bits occurs.
If no interrupt occurs, this bit will not be set to "1".
In the up/down count mode (MODE: bit21 of the timer state control register (TCCS) is 1), this bit will be set every time a compare clear occurs regardless of the value of the MSI2 to MSI0 bits.
If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit24] ICRE: Compare clear interrupt request enable bit

ICRE	Function
0	Interrupt request disabled
1	Interrupt request enabled

When this bit and compare clear interrupt flag bit (ICLR: bit25) are set to "1", an interrupt request for CPU will be generated.

[bit23] BFE: Compare clear buffer enable bit

BFE	Function
0	Invalidate the compare clear buffer
1	Validate the compare clear buffer

- This bit is used for validating the compare clear buffer register (CPCLRB).
- When this bit is set to "0":
  Compare clear buffer register (CPCLRB) will be invalidated. Thus, you can write to the compare clear register (CPCLR) directly.

- When this bit is set to "1":
  Compare clear buffer register (CPCLRB) will be validated. Data written to and retained in the compare clear buffer register (CPCLRB) will be transferred to the compare clear register once the count value "0" from the 16-bit free-run timer has been detected.

[bit22] STOP: Timer enable bit

STOP	Function
0	Enable counting (Start the counting)
1	Disable counting (Stop the counting)

- This bit is used to start/stop counting of the 16-bit free-run timer.
- When this bit is set to "0":
  Start counting the 16-bit free-run timer.

- When this bit is set to "1":
  Stop counting the 16-bit free-run timer.

- The free-run timer will not be initialized even when the SCLR: bit20 of the timer state control register (TCCS) is set to "1" while the free-run timer is inactive (this bit=1).
- The value to be reflected to this bit is the one specified at the GSTOP bit of the timer synchronous activation register (TCGS) while the FRT bit of the timer synchronous activation enable register (TCGSE) is set to "1".

[bit21] MODE: Timer count mode bit

MODE	Function
0	Up count mode
1	Up/down count mode

- This bit is used to select the count mode of the 16-bit free-run timer.
- When this bit is set to "0":
  The up count mode is selected. The timer continues to count up until the count value matches the compare clear register to be reset to "0000_H". After that, it starts counting up again.

- When this bit is set to "1":
  The up/down count mode is selected. The timer continues to count up until the count value matches the compare clear register. After that, it will be converted to down count. Then, when the count value reaches to "0000_H", it will change to up count once again.

- You can write to this bit regardless of the timer is active or inactive. If the timer is active, the value written to this bit will be transferred to the buffer. Then, when the timer value becomes "0000_H", the count mode changes based on the buffer value.

[bit20] SCLR: Timer clear bit

SCLR	Function	
	Read	Write
0	"0" is always read out.	Counter will not be initialized.
1		Counter will be initialized to "0000_H".

- This bit is used to initialize the 16-bit free-run timer to "0000_H".
- Initialization of the 16-bit free-run timer:
  When this bit is set to "1" while the 16-bit free-run timer is active (STOP: bit22 of the timer state control register (TCCS) is 0), the 16-bit free-run timer will be initialized to "0000_H" in the next count clock. The 16-bit free-run timer will not be initialized when this bit is set to "1" while the 16-bit free-run timer is inactive (STOP: bit22 of the timer state control register (TCCS) is 1).

- The value read out is always "0".
- The value to be reflected to this bit is the one specified at the GSTOP bit of the timer synchronous activation

register (TCGS) while the FRT bit of the timer synchronous activation enable register (TCGSE) is set to 1.

Note:

Writing "1" to this bit will not generate the "0" detection interrupt.
If you write "0" to this bit prior to the next count clock after setting "1", the timer clear will not be executed.

[bit19 to bit16] CLK3 to CLK0 : Clock frequency selection bits

CLK 3	CLK 2	CLK 1	CLK 0	Function					
				Count Clock	ϕ=40MHz	ϕ=20MHz	ϕ=10MHz	ϕ=5MHz	ϕ=2.5MHz
0	0	0	0	ϕ	25ns	50ns	100ns	200ns	400ns
0	0	0	1	ϕ /2	50ns	100ns	200ns	400ns	800ns
0	0	1	0	ϕ /4	100ns	200ns	400ns	800ns	1.6µs
0	0	1	1	ϕ /8	200ns	400ns	800ns	1.6µs	3.2µs
0	1	0	0	ϕ /16	400ns	800ns	1.6µs	3.2µs	6.4µs
0	1	0	1	ϕ /32	800ns	1.6µs	3.2µs	6.4µs	12.8µs
0	1	1	0	ϕ /64	1.6µs	3.2µs	6.4µs	12.8µs	25.6µs
0	1	1	1	ϕ /128	3.2µs	6.4µs	12.8µs	25.6µs	51.2µs
1	0	0	0	ϕ /256	6.4µs	12.8µs	25.6µs	51.2µs	102.4µs
Other settings disabled				-	-	-	-	-	-

These bits are used to select the count clock frequency of the 16-bit free-run timer.

Note:

When setting CLK3 to CLK0 bits, confirm that the free-run timer stops firmly.

[bit15 to bit12] Reserved

Always write "0" to these bits.

[bit11] MODE2 : Interrupt mask mode bit2

MODE2	MODE*	Function
0	0	Value set for MSI5 to MSI3 will be invalid
0	1	Value set for MSI5 to MSI3 will be invalid
1	0	Setting disabled (operation is not guaranteed)
1	1	Value set for MSI5 to MSI3 will be valid

- In the up/down count mode (MODE: bit21 of the timer state control register (TCCS) is 1) of the 16-bit free-run timer, this bit will be used to mask the "0" detection interrupt and compare clear interrupt independently.
- During the MODE:bit21="1" of the timer state control register (TCCS) and if this bit is set to "1", the value configured at MSI5 to MSI3: bit10 to bit8 of this register becomes valid and the compare clear interrupt is masked for the number of times specified. For the mask count of "0" detection interrupt, the value configured at MSI2 to MSI0: bit28 to bit26 of the timer state control register (TCCS) becomes valid.

Note:

During MODE:bit21="0" of the timer state control register (TCCS) and if this bit is set to "1", the operation is not guaranteed.

[bit10 to bit8] MSI5 to MSI3 : Compare clear interrupt mask selection bits

MSI5	MSI4	MSI3	Function
0	0	0	An interrupt occurs when there is a match for the first time
0	0	1	An interrupt occurs when there is a match for the second time
0	1	0	An interrupt occurs when there is a match for the third time
0	1	1	An interrupt occurs when there is a match for the fourth time
1	0	0	An interrupt occurs when there is a match for the fifth time
1	0	1	An interrupt occurs when there is a match for the sixth time
1	1	0	An interrupt occurs when there is a match for the seventh time
1	1	1	An interrupt occurs when there is a match for the eighth time

- These bits, which are used to configure the mask count of compare clear interrupt, are valid only when MODE: bit21 of the timer state control register (TCCS) as well as MODE2: bit11 of this register are 1. Value that can be configured for the mask count of "0" detection interrupt is MSI2 to MSI0: bit28 to bit26 of the timer state control register (TCCS).
- When these bits are set to "000_B", the compare clear interrupt factor will not be masked.

Note:

The value read is a mask counter value.
If a read-modify-write instruction is executed, the value read is a mask register value.
The written data will be written to the mask register.
The written value to the mask register while the free-run timer is active (STOP: bit22 of the timer state control register (TCCS) is 0) will be reloaded to the counter only when the mask counter becomes "0".
The written value to the mask register while the free-run timer is inactive (STOP: bit22 of the timer state control register (TCCS) is 1) will be immediately reloaded to the counter.

[bit7 to bit0] Reserved

Always write "1" to these bits.

4.3. Register for the Free-run Timer Selector

Register for the free-run timer selector are explained.

The free-run timer selector has the free-run timer selection register.

4.3.1. Free-run Timer Selection Register : FRS

This section explains the bit structure of the free-run timer selection register.

The free-run timer selection register (FRS) sets any of 3 channels in the free-run timers for the input capture, output compare, A/D activation compare respectively.

■ FRS0: Address 1234_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		OS51	OS50	Reserved		OS41	OS40
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
	Reserved		OS31	OS30	Reserved		OS21	OS20
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		OS11	OS10	Reserved		OS01	OS00
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

[bit31 to bit24] Reserved

Always write "1" to these bits.

[bit23 ,bit22] Reserved

Always write "0" to these bits.

[bit21, bit20] OS51, OS50: Output compare free-run timer selector selection bits

OS51	OS50	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19, bit18] Reserved

Always write "0" to these bits.

[bit17, bit16] OS41, OS40: Output compare free-run timer selector selection bits

OS41	OS40	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15, bit14] Reserved

Always write "0" to these bits.

[bit13, bit12] OS31, OS30: Output compare free-run timer selector selection bits

OS31	OS30	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11, bit10] Reserved

Always write "0" to these bits.

[bit9, bit8] OS21, OS20: Output compare free-run timer selector selection bits

OS21	OS20	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7, bit6] Reserved

Always write "0" to these bits.

[bit5, bit4] OS11, OS10: Output compare free-run timer selector selection bits

OS11	OS10	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3, bit2] Reserved

Always write "0" to these bits.

[bit1, bit0] OS01, OS00: Output compare free-run timer selector selection bits

OS01	OS00	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the output compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

■ FRS1: Address $123A_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		IS31	IS30	Reserved		IS21	IS20
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		IS11	IS10	Reserved		IS01	IS00
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] Reserved

Always write "0" to these bits.

[bit13, bit12] IS31, IS30: Input capture free-run timer selector selection bits

IS31	IS30	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the input capture.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11, bit10] Reserved

Always write "0" to these bits.

[bit9, bit8] IS21, IS20: Input capture free-run timer selector selection bits

IS21	IS20	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the input capture.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7, bit6] Reserved

Always write "0" to these bits.

[bit5, bit4] IS11, IS10: Input capture free-run timer selector selection bits

IS11	IS10	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the input capture.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3, bit2] Reserved

Always write "0" to these bits.

[bit1, bit0] IS01, IS00: Input capture free-run timer selector selection bits

IS01	IS00	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the input capture.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

■ FRS2: Address $123C_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS71	AS70	Reserved		AS61	AS60
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS51	AS50	Reserved		AS41	AS40
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
	Reserved		AS31	AS30	Reserved		AS21	AS20
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS11	AS10	Reserved		AS01	AS00
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

■ FRS3: Address 1240_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS151	AS150	Reserved		AS141	AS140
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS131	AS130	Reserved		AS121	AS120
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
	Reserved		AS111	AS110	Reserved		AS101	AS100
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS91	AS90	Reserved		AS81	AS80
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

■ FRS4: Address 1244_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS231	AS230	Reserved		AS221	AS220
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS211	AS210	Reserved		AS201	AS200
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
	Reserved		AS191	AS190	Reserved		AS181	AS180
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS171	AS170	Reserved		AS161	AS160
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

■ FRS5: Address $12D0_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS311	AS310	Reserved		AS301	AS300
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS291	AS290	Reserved		AS281	AS280
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
	Reserved		AS271	AS270	Reserved		AS261	AS260
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS251	AS250	Reserved		AS241	AS240
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

■ FRS6: Address $12D4_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS391	AS390	Reserved		AS381	AS380
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS371	AS370	Reserved		AS361	AS360
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
	Reserved		AS351	AS350	Reserved		AS341	AS340
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS331	AS330	Reserved		AS321	AS320
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

■ FRS7: Address 12D8$_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		AS471	AS470	Reserved		AS461	AS460
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		AS451	AS450	Reserved		AS441	AS440
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
	Reserved		AS431	AS430	Reserved		AS421	AS420
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		AS411	AS410	Reserved		AS401	AS400
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

[bit31, bit30] Reserved

Always write "0" to these bits.

[bit29, bit28] AS471, AS470: A/D activation compare free-run timer selector selection bits

AS71/AS151/ AS231/AS311/ AS391/AS471	AS70/AS150/ AS230/AS310/ AS390/AS470	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit27, bit26] Reserved

Always write "0" to these bits.

[bit25, bit24] AS461, AS460: A/D activation compare free-run timer selector selection bits

AS61/AS141/ AS221/AS301/ AS381/AS461	AS60/AS140/ AS220/AS300/ AS380/AS460	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit23, bit22] Reserved

Always write "0" to these bits.

[bit21, bit20] AS451, AS450: A/D activation compare free-run timer selector selection bits

AS51/AS131/ AS211/AS291/ AS371/AS451	AS50/AS130/ AS210/AS290/ AS370/AS450	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19, bit18] Reserved

Always write "0" to these bits.

[bit17, bit16] AS441, AS440: A/D activation compare free-run timer selector selection bits

AS41/AS121/ AS201/AS281/ AS361/AS441	AS40/AS120/ AS200/AS280/ AS360/AS440	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15, bit14] Reserved

Always write "0" to these bits.

[bit13, bit12] AS431, AS430: A/D activation compare free-run timer selector selection bits

AS31/AS111/ AS191/AS271/ AS351/AS431	AS30/AS110/ AS190/AS270/ AS350/AS430	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11, bit10] Reserved

Always write "0" to these bits.

[bit9, bit8] AS421, AS420: A/D activation compare free-run timer selector selection bits

AS21/AS101/ AS181/AS261/ AS341/AS421	AS20/AS100/ AS180/AS260/ AS340/AS420	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7, bit6] Reserved

Always write "0" to these bits.

[bit5, bit4] AS411, AS410: A/D activation compare free-run timer selector selection bits

AS11/AS91/ AS171/AS251/ AS331/AS411	AS10/AS90/ AS170/AS250/ AS330/AS410	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3, bit2] Reserved

Always write "0" to these bits.

[bit1, bit0] AS401, AS400: A/D activation compare free-run timer selector selection bits

AS01/AS81/ AS161/AS241/ AS321/AS401	AS00/AS80/ AS160/AS240/ AS320/AS400	Function
0	0	Free-run timer 0
0	1	Free-run timer 1
1	0	Free-run timer 2
Others		Setting disabled (operation is not guaranteed)

These bits are used to configure the free-run timer assigned to the A/D activation compare.

Note:

Before configuring these bits, make sure to verify that the free-run timer is inactive.

5. Operation

The section explains the operation description of the 16-bit free-run timer.

- **Free-run timer simultaneous activation**
  Of the three 16-bit free-run timers, the selected 16-bit free-run timers will be activated or cleared simultaneously.
- **16-bit free-run timer**
  The 16-bit free-run timer starts counting up from the value configured at the timer data register (TCDT) once the count operation is enabled. The count value will be used as base time of the 16-bit output compare and 16-bit input capture.
- **Free run-timer selector**
  You will be able to select the free run-timer input for the 16-bit output compare and 16-bit input capture respectively.

5.1. Interrupt for the 16-bit Free-run Timer

Interrupt for the 16-bit free-run timer are explained.

Table 5-1 shows the interrupt control bits and interrupt factor of the 16-bit free-run timer.

Table 5-1 Interrupt Control Bits and Interrupt Factor of the 16-bit Free-run Timer

	16-bit free-run timer	
	Compare clear	"0" detection
Interrupt request flag bit	Timer state control register (TCCS), ICLR: bit25	Timer state control register (TCCS), IRQZF: bit30
Interrupt request enable bit	Timer state control register (TCCS), ICRE: bit24	Timer state control register (TCCS), IRQZE: bit29
Interrupt factor	The 16-bit free-run timer value matches the compare clear register (CPCLR).	The 16-bit free-run timer value becomes "0000_H".

When the value of the 16-bit free-run timer matches the compare clear register (CPCLR), ICLR: bit25 of the timer state control register (TCCS) will be set. If interrupt requests are enabled (ICRE: bit24 of TCCS is 1) while this bit is set, an interrupt request is output to the interrupt controller.

When the timer value becomes "0000_H", IRQZF: bit30 of the timer state control register (TCCS) will be set.

If interrupt requests are enabled (IRQZE: bit29 of TCCS is 1) while this bit is set, an interrupt request is output to the interrupt controller.

5.2. Operation of the 16-bit Free-run Timer

Operation of the 16-bit free-run timer are explained.

The 16-bit free-run timer starts counting up from the value configured at the timer data register (TCDT) after reset. The count value will be used as base time of the 16-bit output compare and 16-bit input capture.

5.2.1. Timer Clear

This section explains timer clear.

The count value of the 16-bit free-run timer will be cleared in any of the followings:

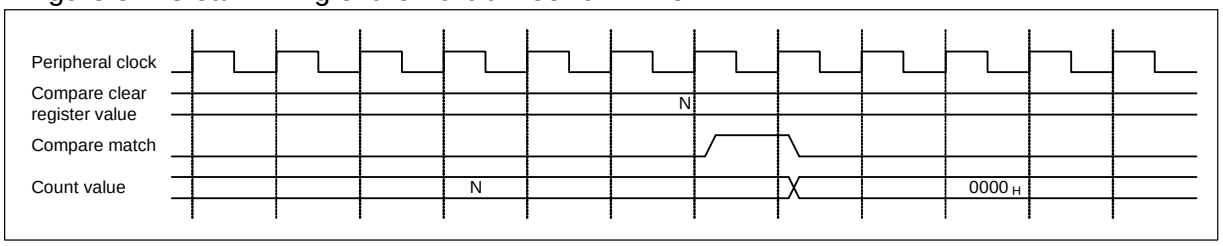
- When there is a match with the compare clear register by the up count mode (MODE:bit21 of TCCS register is 0).
- When "1" is written to SCLR: bit20 of the TCCS register while it is active.
- When "0000_H" is written to the TCDT register while it is inactive.
- When it has been reset.

The counter will be cleared as soon as it has been reset. In the case of a software clear or when there is a match with the compare clear register, the counter will be cleared synchronously with the count timing.

Note:

Even when "1" is written to the SCLR: bit20 of the TCCS register while it is inactive, the count value of the 16-bit free-run timer will not be cleared.
If "0000_H" is written in TCDT register during the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)), an unintended counting may be performed.
See Section "4.2.2 Timer Data Register : TCDT0 to TCDT2" for the setting procedure of TCDT register during the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)).

Figure 5-1 Clear Timing of the 16-bit Free-run Timer



5.2.2. Timer Mode

This section explains timer mode.

For the 16-bit free-run timer, you will be able to select either one of the following modes:

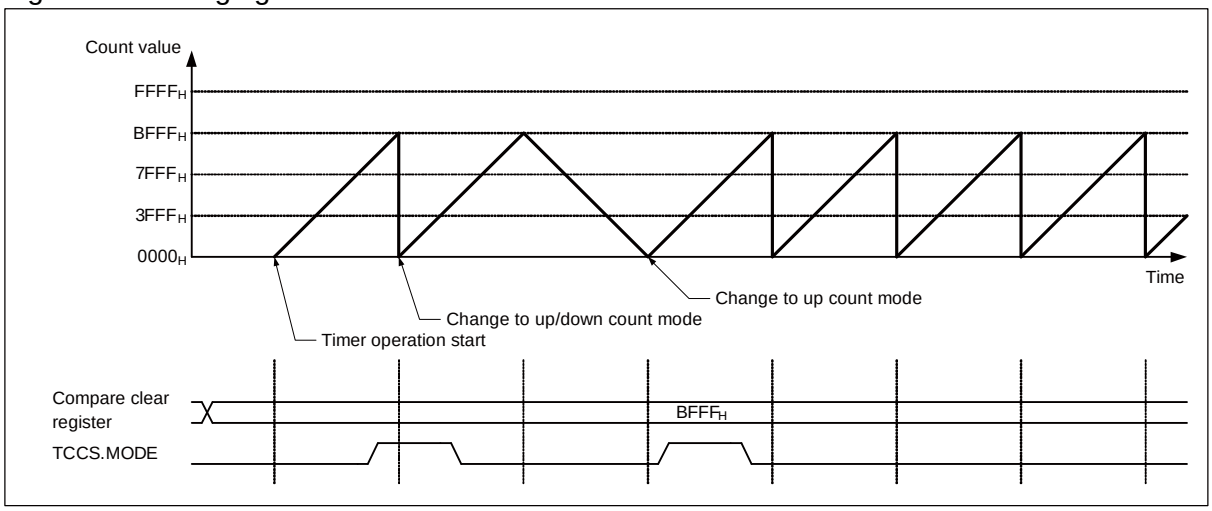
- Up count mode (MODE: bit21 of the TCCS register is 0)
- Up/down count mode (MODE: bit21 of the TCCS register is 1)

In the up count mode, the counter starts counting from the timer data register (TCDT) configured in advance. It continues to count up until the count value matches the value of the compare clear register (CPCLR). The counter will be cleared to "0000_H" and start counting up again.

In the up/down count mode, the counter starts counting from the timer data register (TCDT) configured in advance. It continues to count up until the count value matches the value of the compare clear register (CPCLR). Then, the counter changes counting method from up count to down count. The counter continues to count down until the counter value reaches "0000_H" and starts counting up again.

You will be able to write a value to the mode bit (MODE: bit21 of the TCCS register) whether the timer is active or inactive. If the timer is active, the value written to this bit will be transferred to the buffer. Then, when the timer value becomes "0000_H", the count mode changes.

Figure 5-2 Changing the Timer Mode While the Timer Is Active



5.2.3. Compare Clear Buffer

This section explains compare clear buffer.

The compare clear register (CPCLR) has a buffer function that can be enabled or disabled. When the buffer function is enabled (BFE: bit23 of the TCCS register is 1), data written to the compare clear buffer register (CPCLRB) will be transferred to the CPCLR register once the 16-bit free-run timer value "0" has been detected. When the buffer function is disabled (BFE: bit23 of the TCCS register is 0), you will be able to write data to the CPCLR register directly.

Figure 5-3 Operation in the Up Count Mode when the Compare Clear Buffer is Disabled (BFE: bit23 of the TCCS register is 0)

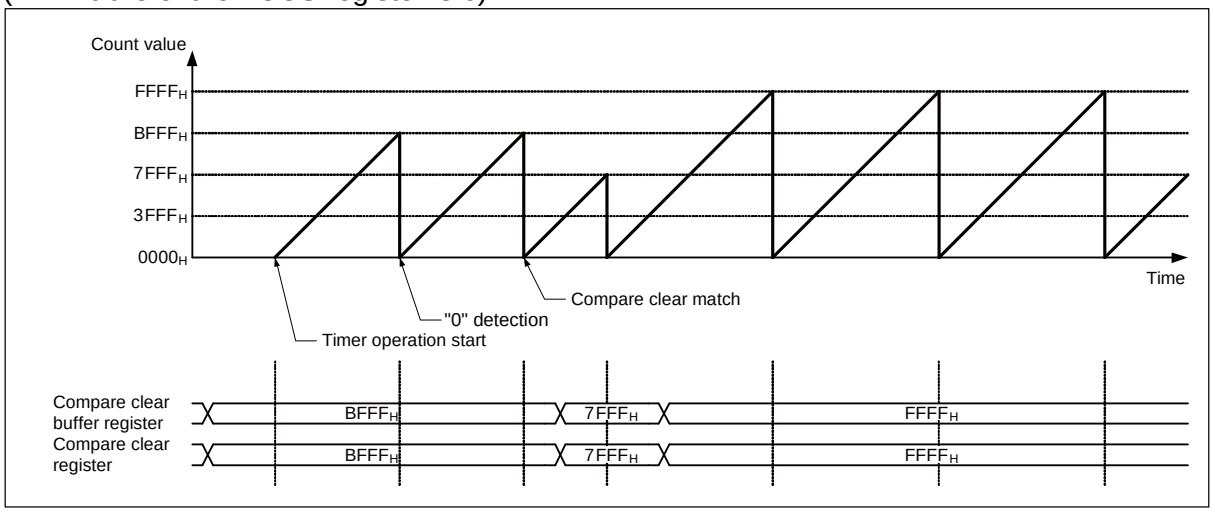


Figure 5-4 Operation in the Up Count Mode when the Compare Clear Buffer is Enabled (BFE: bit23 of the TCCS register is 1)

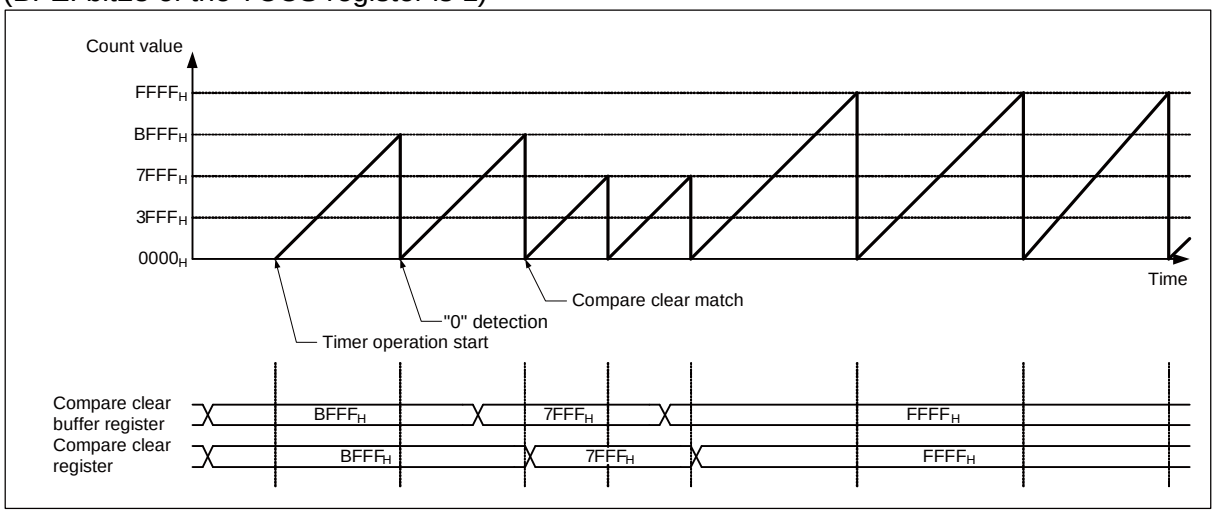
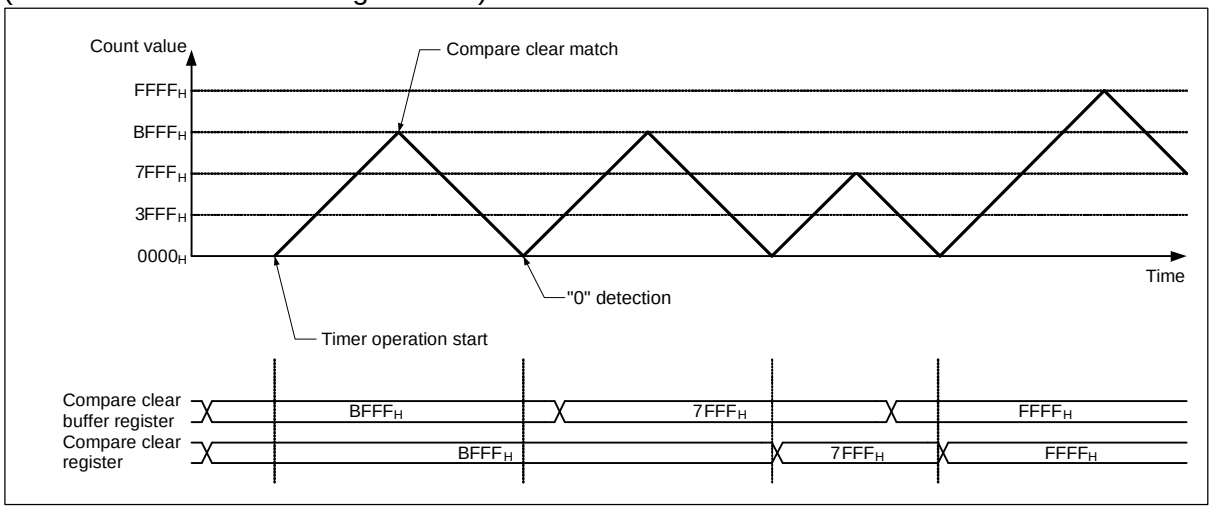


Figure 5-5 Operation in the Up/Down Count Mode when the Compare Clear Buffer is Enabled (BFE: bit23 of the TCCS register is 1)



5.2.4. Timer Interrupt

This section explains timer interrupt.

For the 16-bit free-run timer, you will be able to generate the following two types of interrupt.

- Compare clear interrupt
- "0" detection interrupt

The compare clear interrupt will be generated when the timer value matches the value of the compare clear register.

The "0" detection interrupt will be generated when the timer value becomes "0000_H".

Note:

Software clear (SCLR: bit20 of the TCCS register is 1) does not generate the "0" detection interrupt.

Figure 5-6 Interrupt Generated in the Up Count Mode (MODE: bit21 of the TCCS register is 0)

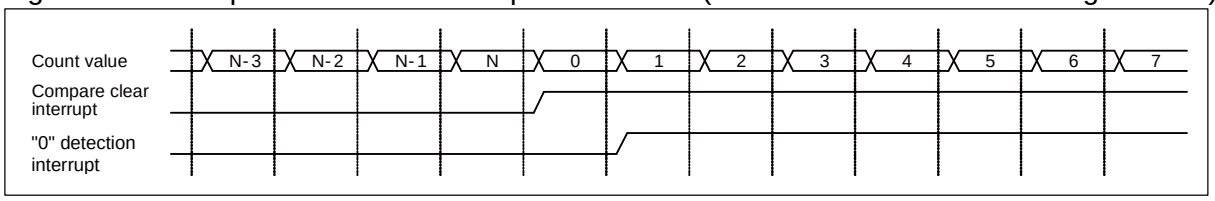
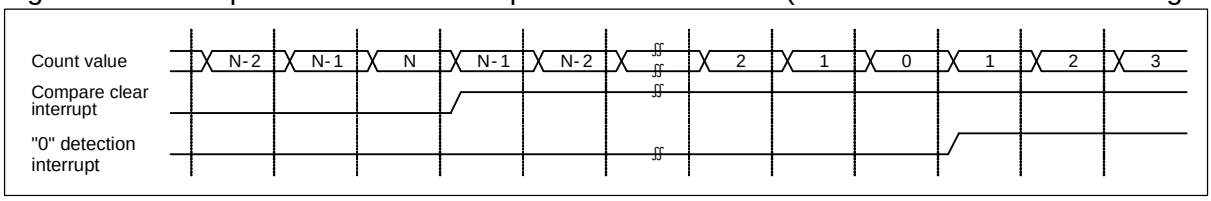


Figure 5-7 Interrupt Generated in the Up/Down Count Mode (MODE: bit21 of the TCCS register is 1)



5.2.5. Interrupt Mask Function

This section explains the interrupt mask function.

You can mask either or both of the "0" detection interrupt and compare match interrupt. The following explains how to mask either one of the interrupts.

- You will be able to mask the interrupt request by setting the MSI2 to MSI0: bit28 to bit26 of the TCCS register. MSI2 to MSI0 bits are 3-bit reload down register that reloads the value once the count value reaches "000_B". You can also load the count value by writing the value to the MSI2 to MSI0 bits directly. Mask count is the value configured at MSI2 to MSI0. When the MSI2 to MSI0 bits become "000_B", the interrupt request will not be masked.
- The interrupt request varies depending on the count mode (MODE: bit21 of the TCCS register). In the up count mode, you will be able to mask the compare clear interrupts only while the "0" detection interrupts will be generated every time "0" is detected. In the up/down count mode, you will be able to mask the "0" detection interrupts only.

The following explains how to mask both types of interrupt requests.

- Only when the free-run timer is in the up/down count mode, you will be able to mask both types of interrupts by setting MODE2 of the TCCS register to 1 and MODE of the TCCS register to 1.

MSI2 to MSI0 bits of the TCCS register are used for masking the "0" detection interrupts and MSI5 to MSI3 bits of the TCCS register are used for masking the compare clear interrupts.

Note:

Software clear (SCLR: bit20 of the TCCS register is 1) does not generate the "0" detection interrupt.

Figure 5-8 Compare Clear Interrupt Masked in the Up Count Mode

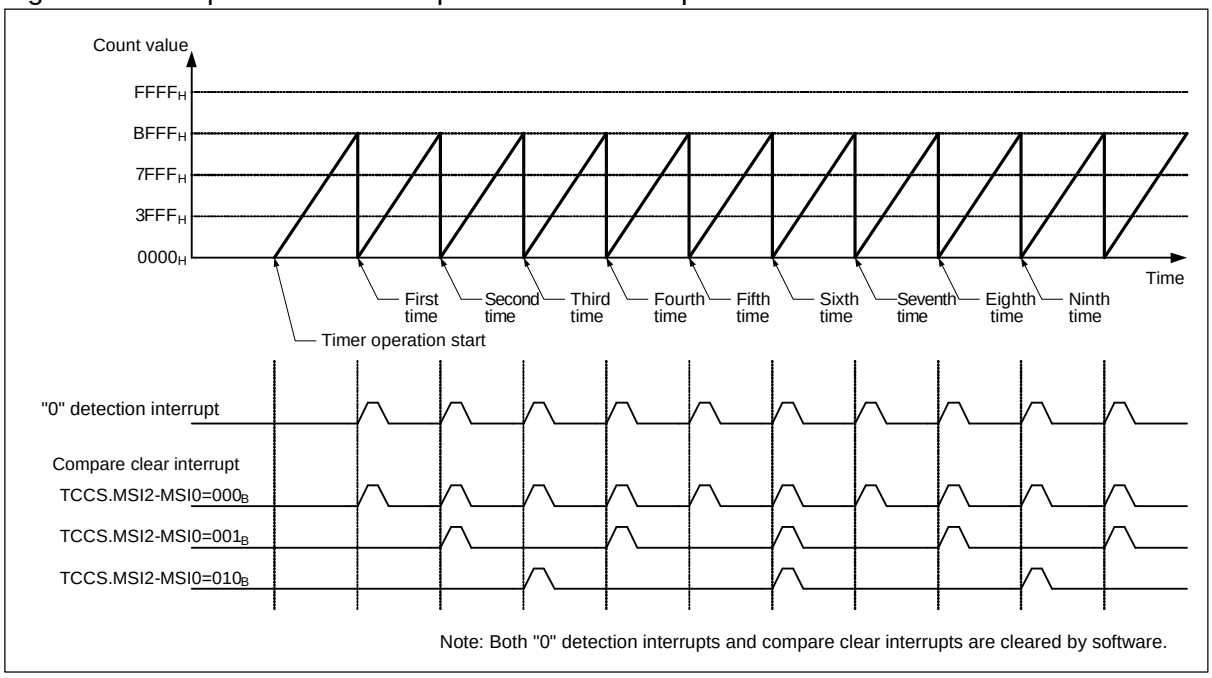


Figure 5-9 "0" Detection Interrupt Masked in the Up/Down Count Mode

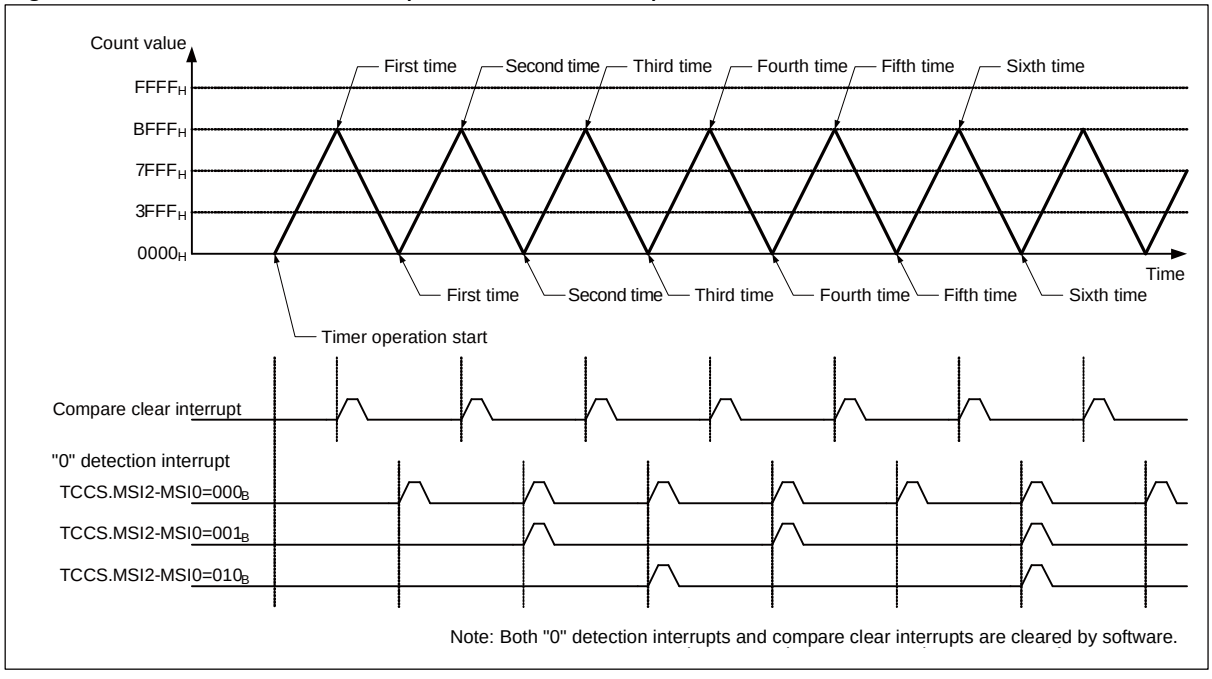
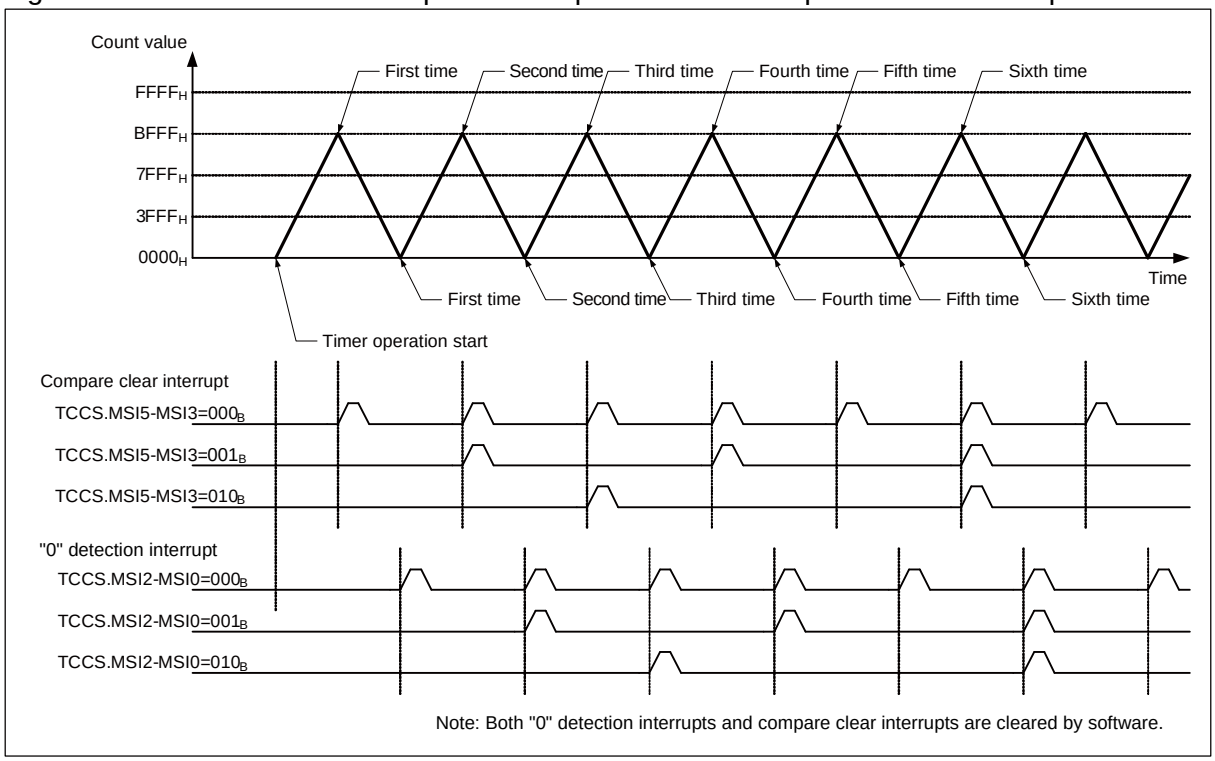


Figure 5-10 "0" Detection Interrupt and Compare Clear Interrupt Masked in the Up/Down Count Mode

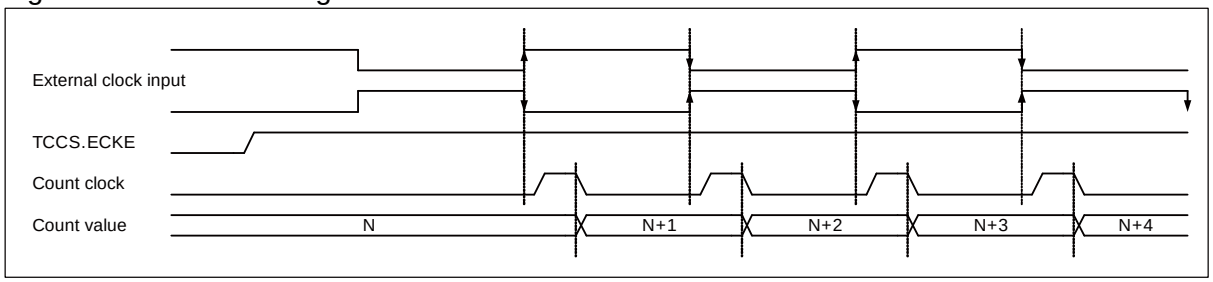


5.2.6. Selected External Count Clock

This section explains the selected external count clock.

The 16-bit free-run timer will be incremented based on the input clock (peripheral clock or external clock). If you select the external clock, the 16-bit free-run timer starts counting up by a rising edge after the external clock mode is selected (ECKE: bit31 of the TCCS register is 1) when the initial value of external input is "1". After that, the timer will count up by both edges. When the initial value of external input is "0", the timer starts counting up by a falling edge. After that, the timer will count up by both edges.

Figure 5-11 Count Timing of the 16-bit Free-run Timer



Note:

Counting of the external clock input uses both edges of the external clock.

5.3. Operation of the Free-run Timer Selector

Operation of the free-run timer selector are explained.

The free-run timer selector is used to configure the free-run timer input for output compare, input capture, and A/D activation compare.
This chip consists of three free-run timers, three (6ch) output compares, and two (4ch) input captures. In addition, the A/D activation compare consists of 48 channels maximum, so you will be able to select any of the following registers shown in the table below.

Table 5-2 Table for Registers

Resources	Register	Remarks
OCU0	FRS0:OS0[1:0]	16-bit output compare
OCU1	FRS0:OS1[1:0]	
OCU2	FRS0:OS2[1:0]	
OCU3	FRS0:OS3[1:0]	
OCU4	FRS0:OS4[1:0]	
OCU5	FRS0:OS5[1:0]	
ICU0	FRS1:IS0[1:0]	16-bit input capture
ICU1	FRS1:IS1[1:0]	
ICU2	FRS1:IS2[1:0]	
ICU3	FRS1:IS3[1:0]	
ADT0	FRS2:AS0[1:0]	A/D activation compare
ADT1	FRS2:AS1[1:0]	
ADT2	FRS2:AS2[1:0]	
ADT3	FRS2:AS3[1:0]	
ADT4	FRS2:AS4[1:0]	
ADT5	FRS2:AS5[1:0]	
ADT6	FRS2:AS6[1:0]	
ADT7	FRS2:AS7[1:0]	
ADT8	FRS3:AS8[1:0]	
ADT9	FRS3:AS9[1:0]	
ADT10	FRS3:AS10[1:0]	
ADT11	FRS3:AS11[1:0]	
ADT12	FRS3:AS12[1:0]	
ADT13	FRS3:AS13[1:0]	
ADT14	FRS3:AS14[1:0]	
ADT15	FRS3:AS15[1:0]	
ADT16	FRS4:AS16[1:0]	
ADT17	FRS4:AS17[1:0]	
ADT18	FRS4:AS18[1:0]	

Resources	Register	Remarks
ADT19	FRS4:AS19[1:0]	A/D activation compare
ADT20	FRS4:AS20[1:0]	
ADT21	FRS4:AS21[1:0]	
ADT22	FRS4:AS22[1:0]	
ADT23	FRS4:AS23[1:0]	
ADT24	FRS5:AS24[1:0]	
ADT25	FRS5:AS25[1:0]	
ADT26	FRS5:AS26[1:0]	
ADT27	FRS5:AS27[1:0]	
ADT28	FRS5:AS28[1:0]	
ADT29	FRS5:AS29[1:0]	
ADT30	FRS5:AS30[1:0]	
ADT31	FRS5:AS31[1:0]	
ADT32	FRS6:AS32[1:0]	
ADT33	FRS6:AS33[1:0]	
ADT34	FRS6:AS34[1:0]	
ADT35	FRS6:AS35[1:0]	
ADT36	FRS6:AS36[1:0]	
ADT37	FRS6:AS37[1:0]	
ADT38	FRS6:AS38[1:0]	
ADT39	FRS6:AS39[1:0]	
ADT40	FRS7:AS40[1:0]	
ADT41	FRS7:AS41[1:0]	
ADT42	FRS7:AS42[1:0]	
ADT43	FRS7:AS43[1:0]	
ADT44	FRS7:AS44[1:0]	
ADT45	FRS7:AS45[1:0]	
ADT46	FRS7:AS46[1:0]	
ADT47	FRS7:AS47[1:0]	

Table 5-3 Table for Setting Values

Setting value	Free-run timer
00_B	FRT0 (initial state)
01_B	FRT1
10_B	FRT2
11_B	Setting disabled (operation is not guaranteed)

Note:

Before configuring the free-run timer selection register, make sure to verify that the free-run timer is inactive.

5.4. Notes on Operating Specifications

Notes on operating specifications are explained.

5.4.1. Notes at Accessing the Buffer Registers

This section explains notes to observe when accessing the buffer registers.

The CPCLR register in the free-run timer has a buffer function. Do not use a read-modify-write instruction when accessing this register.

5.4.2. Notes on Using the 16-bit Free-run Timer

This section explains the notes on using the 16-bit free-run timer.

- **Notes on setting by a program**
  - When you execute reset, the timer value becomes "0000_H", however, the "0" detection interrupt flag will not be configured.
  - Since the timer mode bit (MODE of the TCCS register) has a buffer, the timer mode changed after the 0 detection becomes valid.
  - Software clear (SCLR of the TCCS register is 1) initializes the timer, but it does not generate the 0 detection interrupt.
  - When you start counting while the compare value and count value match, the compare clear flag will not be configured.
  - Set the value other than "0000_H" for the compare value. When setting the value, consider that the following operation will happen.
    - When the timer mode bit (MODE in TCCS register) is in the up count mode, the timer value is updated to "0000_H" and then is fixed to "0000_H". The "0" detection interrupt flag and the compare clear flag continue to be set every count clock.
    - When the timer mode bit (MODE in TCCS register) is in the up down count mode, the timer value repeats the up count operation from "0000_H" to "$FFFF_H$". The "0" detection interrupt flag and the compare clear flag are set when the timer value and "0000_H" match.
- **Notes on interrupts**
  - Always clear the interrupt flag (IRQZF) before setting the interrupt request enable bit (IRQZE) of the timer state control register (TCCS) to "1".
  - Always clear the interrupt flag (ICLR) before setting the interrupt request enable bit (ICRE) of the timer state control register (TCCS) to "1".
- **Notes on accessing the TCCS register**
  - For the read-modify-write instruction, setting value will be read out from the MSI2 to MSI0/MSI5 to MSI3.
  - For the normal reading mode, the counter value will be read out from the MSI2 to MSI0/MSI5 to MSI3.

5.4.3. Notes on Using the Free-run Timer Selector

This section explains notes on using the free-run timer selector.

Be sure to configure selection while the free-run timer is inactive.

Chapter 25: 16-Bit Output Compare

This chapter explains the 16-bit output compare.

1. Overview
2. Features
3. Configuration Diagram
4. Registers
5. Operation

Code : FS18-2v4-91520-2-E

1. Overview

This section explains the overview of the 16-bit output compare.

This product includes six 16-bit output compare channels.

2. Features

This section explains the features of the 16-bit output compare.

■ Functions of 16-bit Output Compare

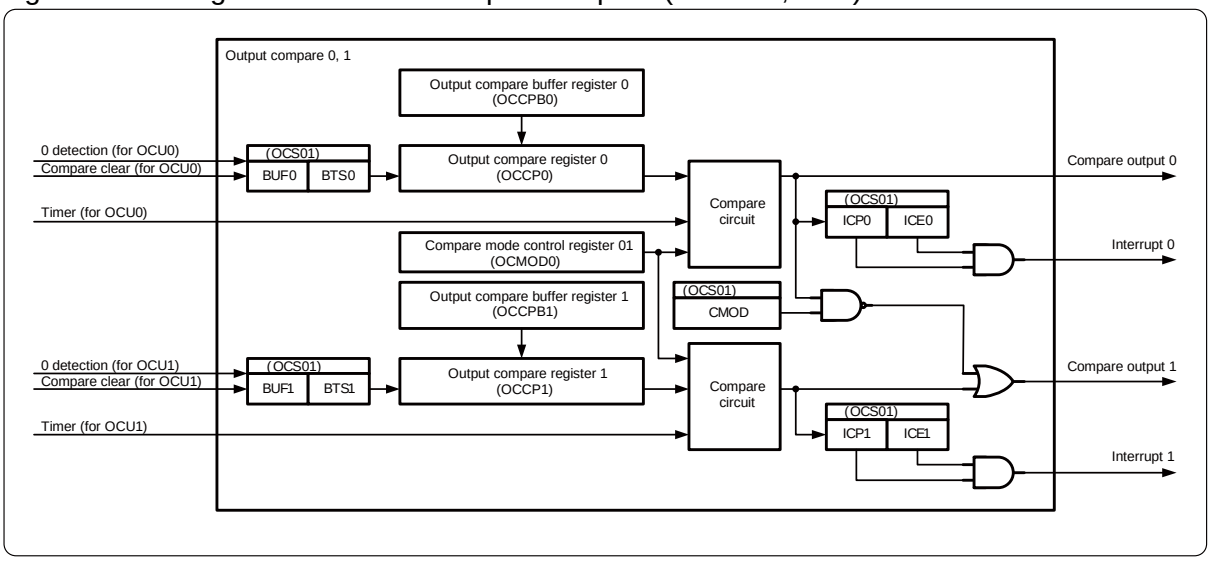
- 16-bit output compare consists of six 16-bit compare registers (with selectable buffer registers), compare output latch, 3 compare control registers, and compare mode control registers. When the 16-bit free-run timer value matches a compare register, an interrupt is generated and the output level is inverted.
- The 6 compare registers can be operated independently. An output pin and an interrupt flag correspond to each of the compare registers.
- Two compare registers can be used as a pair to control output pins. Two compare registers combined as a pair can be used to invert the output pins.
- The initial values for output pins can be set.
- An interrupt can be generated when an output compare register matches the 16-bit free-run timer.
- Any desired free-run timer channel can be set for each compare unit.
- Any of free-run timers 0 through 2 can be selected as the input for output compare 0 through 5 among output compare 6 channels. This can be set at the free-run timer selection register (FRS0). See "Free-run Timer Selection Register: FRS" of "CHAPTER: 16-BIT FREE-RUN TIMER" for details.
- The compare output for an output compare can be output from the waveform generator output pin.
  The PPG timer's GATE signal output can be also controlled. See "CHAPTER: WAVEFORM GENERATOR" for details.

3. Configuration Diagram

This section explains the configuration diagram of the 16-bit output compare.

■ Configuration of 16-bit Output Compare

Figure 3-1 Configuration of 16-bit Output Compare (For ch.0, ch.1)



4. Registers

This section explains the registers of the 16-bit output compare.

■ List of 16-bit Output Compare Registers

Table 4-1 List of 16-bit Output Compare Registers

Address	+0	+1	+2	+3
0x0000124C	Output compare buffer register 0 (OCCPB0), Output compare register 0 (OCCP0)		Output compare buffer register 1 (OCCPB1), Output compare register 1 (OCCP1)	
0x00001250	Compare control register 01 (OCS01)		Reserved	Compare mode control register 01 (OCMOD01)
0x00001254	Output compare buffer register 2 (OCCPB2), Output compare register 2 (OCCP2)		Output compare buffer register 3 (OCCPB3), Output compare register 3 (OCCP3)	

Address	+0	+1	+2	+3
0x00001258	Compare control register 23 (OCS23)		Reserved	Compare mode control register 23 (OCMOD23)
0x0000125C	Output compare buffer register 4 (OCCPB4), Output compare register 4 (OCCP4)		Output compare buffer register 5 (OCCPB5), Output compare register 5 (OCCP5)	
0x00001260	Compare control register 45 (OCS45)		Reserved	Compare mode control register 45 (OCMOD45)

4.1. 16-bit Output Compare Registers

This section explains the registers of the 16-bit output compare.

The 16-bit output compare consists of output compare buffer registers, output compare registers, compare control registers, and compare mode selection registers.

4.1.1. Output Compare Buffer Registers (OCCPB0 to OCCPB5)/Output Compare Registers (OCCP0 to OCCP5)

The bit configuration of the output compare buffer registers / the output compare registers is shown below.

The output compare buffer register (OCCPB) is a 16-bit buffer register for the output compare register (OCCP).
The output compare register (OCCP) is a 16-bit register to be used for comparison with the count value of the 16-bit free-run timer.
Both the OCCPB and OCCP registers are located at the same address.

■ **OCCPB0, 2, 4: Address $124C_H$, 1254_H, $125C_H$ (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] OP15 to OP00: Compare value buffer bits

	Function
OP15 to OP00	Compare value buffer

The output compare buffer register is a buffer register for the output compare register (OCCP). If the buffer function is disabled (BUF0: bit2 = 1 in the compare control register (OCS)) or the free-run timer stops, the value of the output compare buffer register will be immediately transferred to the output compare register. If the buffer function is enabled (BUF0: bit2 = 0 in the compare control register (OCS)), the value will be transferred to the output compare register when a compare clear match or 0 is detected in accordance with the transfer selection bit (BTS0:bit2) in the compare control register (OCS).

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

■ OCCPB1, 3, 5: Address $124E_H$, 1256_H, $125E_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] OP15 to OP00: Compare value buffer bits

	Function
OP15 to OP00	Compare value buffer

The output compare buffer register is a buffer register for the output compare register (OCCP). If the buffer function is disabled (BUF1: bit3 = 1 in the compare control register (OCS)) or the free-run timer stops, the value of the output compare buffer register will be immediately transferred to the output compare register. If the buffer function is enabled (BUF1: bit3 = 0 in the compare control register (OCS)), the value will be transferred to the output compare register when a compare clear match or 0 is detected in accordance with the transfer selection bit (BTS1:bit3) in the compare control register (OCS).

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

■ OCCP0, 2, 4: Address124C$_H$, 1254$_H$, 125C$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

[bit31 to bit16] OP15 to OP00: Compare value bits

OP15 to OP00	Function
	Compare value

- The output compare register is a 16-bit register to be used for comparison with the count value of the 16-bit free-run timer. Before enabling the operation of the 16-bit free-run timer, set a value in the output compare buffer register (OCCPB).
- When the value of the output compare register matches the count value of the 16-bit free-run timer, a compare signal is generated and the output compare interrupt flag bit (IOP0: bit6 in the compare control register (OCS)) is set. When the output level is set (OTD0: bit8 in the compare control register (OCS)), the compare output level for the output compare register (OCCP) can be inverted.
- When all the conditions listed below are met and a value that exceeds the peak value of the 16-bit free-run timer is set to this register, the output compare output is "1" right after the buffer transfer. When this register is set to "0000$_H$", the output compare output is "0" right after the buffer transfer.
  - The free-run timer is in up/down count mode.
  - When the BUF bit in the compare control register (OCS) is "0" (buffer function enabled)
  - When the BTS bit in the compare control register (OCS) is "1" (transfer when there is a compare clear match)
  - When the CMD bit in the compare control register (OCS) is "1"
  - When the MOD bit in the compare mode control register (OCMOD) is "1"

When all the conditions listed above are not met, even if the value of this register matches the peak value of the 16-bit free-run timer in up/down mode, no compare signal is generated.

The outcome is as follows according to the settings of the CMD bit in the compare control register (OCS)

- When the CMOD bit in the compare control register (OCS) is 1
  When this register is set to "FFFF$_H$", the output compare output is "1" regardless of the value of the 16-bit free-run timer and the inversion mode.
  When this register is set to "0000$_H$", the output compare output is "0".
- When the CMOD bit in the compare control register (OCS) is 0
  When this register is set to "FFFF$_H$", the output compare output is "0" regardless of the value of the 16-bit free-run timer and the inversion mode.
  When this register is set to "0000$_H$", the output compare output is "1".

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

■ OCCP1, 3, 5: Address $124E_H$, 1256_H, $125E_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] OP15 to OP00: Compare value bits

OP15 to OP00	Function
	Compare value

- The output compare register is a 16-bit register to be used for comparison with the count value of the 16-bit free-run timer. Before enabling the operation of the 16-bit free-run timer, set a value in the output compare buffer register (OCCPB).
- When the value of the output compare register matches the count value of the 16-bit free-run timer, a compare signal is generated and the output compare interrupt flag bit (IOP1: bit7 in the compare control register (OCS)) is set. When the output level is set (OTD1: bit9 in the compare control register (OCS)), the compare output level for the output compare register (OCCP) can be inverted.
- When all the conditions listed below are met and a value that exceeds the peak value of the 16-bit free-run timer is set to this register, the output compare output is "1" right after the buffer transfer. When this register is set to "0000_H", the output compare output is "0" right after the buffer transfer.
  - The free-run timer is in up/down count mode.
  - When the BUF bit in the compare control register (OCS) is "0" (buffer function enabled)
  - When the BTS bit in the compare control register (OCS) is "1" (transfer when there is a compare clear match)
  - When the CMD bit in the compare control register (OCS) is "1"
  - When the MOD bit in the compare mode control register (OCMOD) is "1"

When all the conditions listed above are not met, even if the value of this register matches the peak value of the 16-bit free-run timer in up/down mode, no compare signal is generated.

The outcome is as follows according to the settings of the CMD bit in the compare control register (OCS).

- When the CMOD bit in the compare control register (OCS) is 1
  When this register is set to "$FFFF_H$", the output compare output is "1" regardless of the value of the 16-bit free-run timer and the inversion mode.
  When this register is set to "0000_H", the output compare output is "0".
- When the CMOD bit in the compare control register (OCS) is 0
  When this register is set to "$FFFF_H$", the output compare output is "0" regardless of the value of the 16-bit free-run timer and the inversion mode.
  When this register is set to "0000_H", the output compare output is "1".

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

4.1.2. Compare Control Register (OCS)

The bit configuration of the compare control register is shown below.

The compare control register (OCS) controls the output level, output enable, output level inversion mode, compare operation enable, compare match interrupt enable, and compare match interrupt flag in OUT0 to OUT5.

■ OCS01: Address 1250_H (Access: Byte, Half-word, Word)

■ OCS23: Address 1258_H (Access: Byte, Half-word, Word)

■ OCS45: Address 1260_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	BTS1	BTS0	CMOD	Reserved		OTD1	OTD0
Initial value	0	1	1	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W0	R/W0	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0
Initial value	0	0	0	0	1	1	0	0
Attribute	R(RM1), W	R(RM1), W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Always write 0 to this bit.

[bit14] BTS1: Buffer transfer selection bit

BTS1	Function
0	Transfer is activated when 0 is detected (ch.1).
1	Transfer is activated when a compare clear match occurs (ch.1).

- This bit is used to select the timing of data transfer from the output compare buffer register (OCCPB1) to the output compare register (OCCP1).
- When this bit is set to "0":
  Data transfer is activated when the count value of "0" is detected on the 16-bit free-run timer.

- When this bit is set to "1":
  Data transfer is activated when a compare clear match occurs on the 16-bit free-run timer.

- For ch.3 and 5, the operation is the same as ch.1.

[bit13] BTS0: Buffer transfer selection bit

BTS0	Function
0	Transfer is activated when 0 is detected (ch.0).
1	Transfer is activated when a compare clear match occurs (ch.0).

- This bit is used to select the timing of data transfer from the output compare buffer register (OCCPB0) to the output compare register (OCCP0).
- When this bit is set to "0":
  Data transfer is activated when the count value of "0" is detected on the 16-bit free-run timer.

- When this bit is set to "1":
  Data transfer is activated when a compare clear match occurs on the 16-bit free-run timer.

- For ch.2 and 4, the operation is the same as ch.0.

[bit12] CMOD: Output level invert mode bit

CMOD	Function	
0	For the compare mode control register: MOD0=0 The compare output 0 is immediately inverted when there is a match with the output compare register (OCCP0). For the compare mode control register: MOD1=0 The compare output 1 is immediately inverted when there is a match with the output compare register (OCCP1).	For the compare mode control register: MOD0=1 or MOD1=1 Set to "1" when a match is detected in up-count mode. Reset to "0" when a match is detected in down-count mode.
1	For the compare mode control register: MOD0=0 The compare output 0 is immediately inverted when there is a match with the output compare register (OCCP0). For the compare mode control register: MOD1=0 The compare output 1 is immediately inverted when there is a match with the output compare register (OCCP0 or OCCP1).	For the compare mode control register: MOD0=1 or MOD1=1 Set to "0" when a match is detected in up-count mode. Reset to "1" when a match is detected in down-count mode.

- This bit is used to switch the compare output level inversion mode immediately when a match is detected.
- The compare output for an output compare can be output from the waveform generator output pin (RTO).
- See "CHAPTER: WAVEFORM GENERATOR" for the output setting method.
- When this bit is set to "0":
  When the compare mode control register: MOD0/MOD1=0

- For the compare mode control register: MOD0=0
  The compare output 0 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCP6).
- For the compare mode control register: MOD1=0
  The compare output 1 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCP7).

For the compare mode control register: MOD0/MOD1=1

- Set to "1" when a match is detected in up-count mode.
- Reset to "0" when a match is detected in down-count mode.

- When this bit is set to "1":
  When the compare mode control register: MOD0/MOD1=0

  - For the compare mode control register: MOD0=0
    The compare output 0 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCP0).

  - For the compare mode control register: MOD1=0
    The compare output 1 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCP0 or OCCP1).
  - When the output compare registers (OCCP0 and OCCP1) have the same value, the operation is the same as when one compare register is used.
    For the compare mode control register: MOD0/MOD1=1

  - Reset to "0" when a match is detected in up-count mode.
  - Set to "1" when a match is detected in down-count mode.
- For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

[bit11, bit10] Reserved

Always write 0 to these bits

[bit9] OTD1: Output level bit

OTD1	Function	
	Read	Write
0	Output value of the compare output 1	The compare output 1 outputs "0".
1		The compare output 1 outputs "1".

- This bit is used to change the compare output 1 level of the output compare.
- The initial value of the compare pin output is "0".
- Be sure to stop the compare operation before writing a value to this bit. The value read from this bit is the output compare value (compare output 1).

Note:

A value can be written to this bit when CST1: bit1=0 in the compare control register (OCS).

[bit8] OTD0: Output level bit

OTD0	Function	
	Read	Write
0	Output value of the compare output 0	The compare output 0 outputs "0".
1		The compare output 0 outputs "1".

- This bit is used to change the compare output 0 level of the output compare.
- The initial value of the compare pin output is "0".
- Be sure to stop the compare operation before writing a value to this bit. The value read from this bit is the output compare value (compare output 0).

Note:

A value can be written to this bit when CST0: bit0=0 in the compare control register (OCS).

[bit7] IOP1: Compare match interrupt flag bit

IOP1	Function	
	Read	Write
0	No compare match interrupt occurs for the output compare register (OCCP1).	This bit is cleared.
1	Compare match interrupt occurs for the output compare register (OCCP1).	This bit remains unaffected.

- This bit is an interrupt flag that indicates whether the value of the output compare register (OCCP1) matched that of the 16-bit free-run timer.
- This bit is set to "1" when the output compare register value matches the 16-bit free-run timer value.
- An output compare interrupt occurs if this bit is set while the compare match interrupt enable bit (IOE1:bit5) is enabled ("1").
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- For ch.3 and 5, the operation is the same as ch.1.

Note:

If a read-modify-write instruction is executed, "1" is always read.

If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit6] IOP0: Compare match interrupt flag bit

IOP0	Function	
	Read	Write
0	No compare match interrupt occurs for the output compare register (OCCP0).	This bit is cleared.
1	Compare match interrupt occurs for the output compare register (OCCP0).	This bit remains unaffected.

- This bit is an interrupt flag that indicates whether the value of the output compare register (OCCP0) matched that of the 16-bit free-run timer.
- This bit is set to "1" when the compare register value matches the 16-bit free-run timer value.
- An output compare interrupt occurs if this bit is set while the compare match interrupt enable bit (IOE0:bit4) is enabled ("1").
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- For ch.2 and 4, the operation is the same as ch.0.

Note:

If a read-modify-write instruction is executed, "1" is always read.

If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit5] IOE1: Compare match interrupt enable bit

IOE1	Function
0	Compare match interrupt is disabled for the output compare register (OCCP1).
1	Compare match interrupt is enabled for the output compare register (OCCP1).

- This bit is used to enable an output compare interrupt for the output compare register (OCCP1).
- An output compare interrupt occurs if the compare match interrupt flag bit (IOP1: bit7) is set while this bit is set to "1".
- For ch.3 and 5, the operation is the same as ch.1.

[bit4] IOE0: Compare match interrupt enable bit

IOE0	Function
0	Compare match interrupt is disabled for the output compare register (OCCP0).
1	Compare match interrupt is enabled for the output compare register (OCCP0).

- This bit is used to enable an output compare interrupt for the output compare register (OCCP0).
- An output compare interrupt occurs if the compare match interrupt flag bit (IOP0: bit6) is set while this bit is set to "1".
- For ch.2 and 4, the operation is the same as ch.0.

[bit3] BUF1: Compare buffer invalidating bit

BUF1	Function
0	Validates the compare buffer of the output compare register (OCCP1).
1	Invalidates the compare buffer of the output compare register (OCCP1).

- This bit is used to invalidate the buffer function of the output compare register (OCCP1).
- When this bit is set to "0": This buffer function is validated.
- For ch.3 and 5, the operation is the same as ch.1.

[bit2] BUF0: Compare buffer invalidating bit

BUF0	Function
0	Validates the compare buffer of the output compare register (OCCP0).
1	Invalidates the compare buffer of the output compare register (OCCP0).

- This bit is used to invalidate the buffer function of the output compare register (OCCP0).
- When this bit is set to "0": This buffer function is validated.
- For ch.2 and 4, the operation is the same as ch.0.

[bit1] CST1: Compare operation enable bit

CST1	Function
0	Disables the compare operation of the output compare register (OCCP1).
1	Enables the compare operation of the output compare register (OCCP1).

- This bit is used to enable the compare operation between the 16-bit free-run timer and the output compare register (OCCP1).
- Before enabling the compare operation, be sure to write a value to the output compare register (OCCP1) and the timer data register of the free-run timer (TCDT[x], where x is a pertinent free-run timer).
- For ch.3 and 5, the operation is the same as ch.1.

[bit0] CST0: Compare operation enable bit

CST0	Function
0	Disables the compare operation of the output compare register (OCCP0).
1	Enables the compare operation of the output compare register (OCCP0).

- This bit is used to enable the compare operation between the 16-bit free-run timer and the output compare register (OCCP0).
- Before enabling the compare operation, be sure to write a value to the output compare register (OCCP0) and the timer data register of the free-run timer (TCDT[x], where x is a pertinent free-run timer).
- For ch.2 and 4, the operation is the same as ch.0.

4.1.3. Compare Mode Control Register (OCMOD)

The bit configuration of the compare Mode control register is shown below.

The compare mode control register (OCMOD) controls the output level upon detection of a compare match by specifying to invert, set, or reset the output level.

■ OCMOD01: Address 1253_H (Access: Byte, Half-word, Word)

■ OCMOD23: Address $125B_H$ (Access: Byte, Half-word, Word)

■ OCMOD45: Address 1263_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						MOD1	MOD0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W

[bit7 to bit2] Reserved

Always write 0 to these bits.

[bit1] MOD1: Compare match mode setting bit

MOD1	Function
0	Inverts the previous output value.
1	Sets the output value to "1" or resets it to "0" according to the setting of the CMOD bit in the compare control register (OCS01).

- This bit specifies the operation to be performed when a compare match is detected in the output compare output 1.
- When this bit is set to "0", the output value is inverted upon a compare match.
- When this bit is set to "1", the output value is set to "1" or reset to "0" upon a compare match. The switch between setting and resetting is performed according to the CMOD bit (common to ch.0 and ch.1) in the compare control register (OCS01).
- For ch.3 and 5, the operation is the same as ch.1.

Note:

Be sure to stop the compare operation before writing a value to this bit.

[bit0] MOD0: Compare match mode setting bit

MOD0	Function
0	Inverts the previous output value.
1	Sets the output value to "1" or resets it to "0" according to the setting of the CMOD bit in the compare control register (OCS01).

- This bit specifies the operation to be performed when a compare match is detected in the output compare output 0.
- When this bit is set to "0", the output value is inverted upon a compare match.

- When this bit is set to "1", the output value is set to "1" or reset to "0" upon a compare match. The switch between setting and resetting is performed according to the CMOD bit (common to ch.0 and ch.1) in the compare control register (OCS01).
- For ch.2 and 4, the operation is the same as ch.0.

Note:

Be sure to stop the compare operation before writing a value to this bit.

5. Operation

This section explains the operations.

5.1 Interrupts for 16-bit Output Compare

5.2 Operation of 16-bit Output Compare

5.3 Notes on Using 16-bit Output Compare

5.1. Interrupts for 16-bit Output Compare

This section explains the interrupts for 16-bit output compare.

Table 5-1 shows the interrupt control bits and interrupt factor of the 16-bit output compare.

Table 5-1 Interrupt Control Bits and Interrupt Factor of 16-bit Output Compare

	16-bit output compare	
	Even-number channel	Odd-number channel
Interrupt request flag bit	Compare control register (OCS) IOP0:bit6	Compare control register (OCS) IOP1:bit7
Interrupt request enable bit	Compare control register (OCS) IOE0:bit4	Compare control register (OCS) IOE1:bit5
Interrupt factor	The 16-bit free-run timer value matches the output compare register 0 (OCCP0).	The 16-bit free-run timer value matches the output compare register 1 (OCCP1).

When the 16-bit free-run timer value matches the output compare register (OCCP), IOP1/IOP0: bit7/bit6 in the compare control register (OCS) are set to "1". If interrupt requests are enabled (IOE1/IOE0: bit5/bit4 = 1 of OCS) in this state, an interrupt request is output to the interrupt controller.

5.2. Operation of 16-bit Output Compare

This section explains the operation of 16-bit output compare.

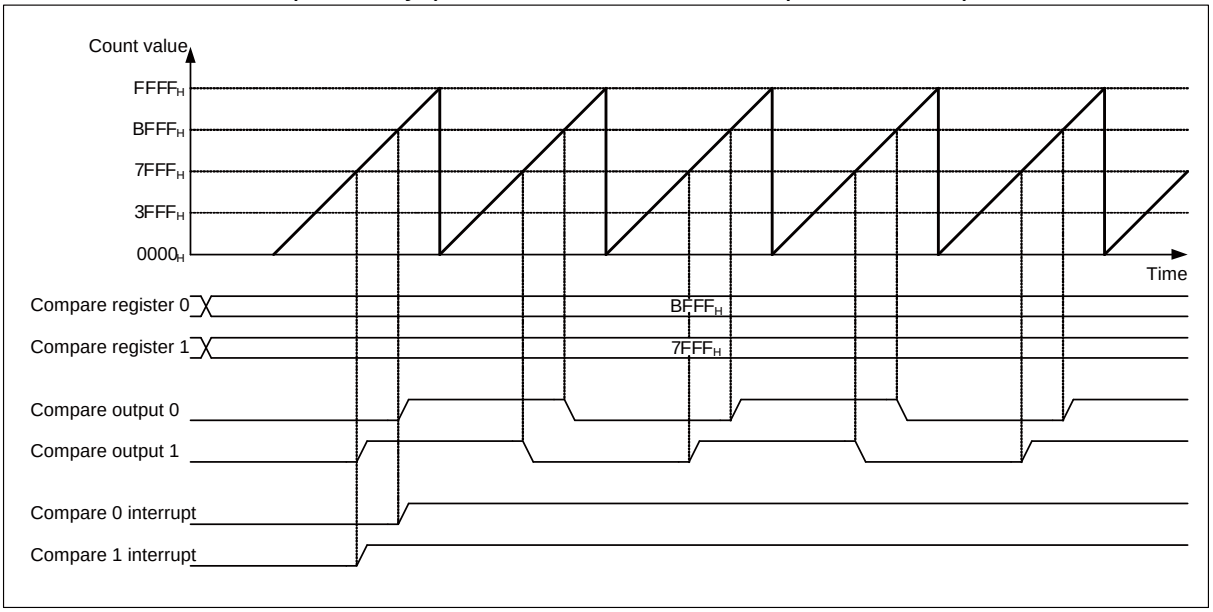
The output compare is used to compare the value set in the specified compare clear register and the value of the 16-bit free-run timer. When a match is detected, the interrupt flag is set and the output level is inverted. If there is a match between the count peak and the compare register value while the free-run timer is in up/down count mode, the match signal is ignored.

5.2.1. Operation of 16-bit Output Compare (Inverted Mode, MOD0= 0 in OCMOD01 Register)

The operation of 16-bit output compare (Inverted mode, MOD0= 0 in OCMOD01 register) is shown.

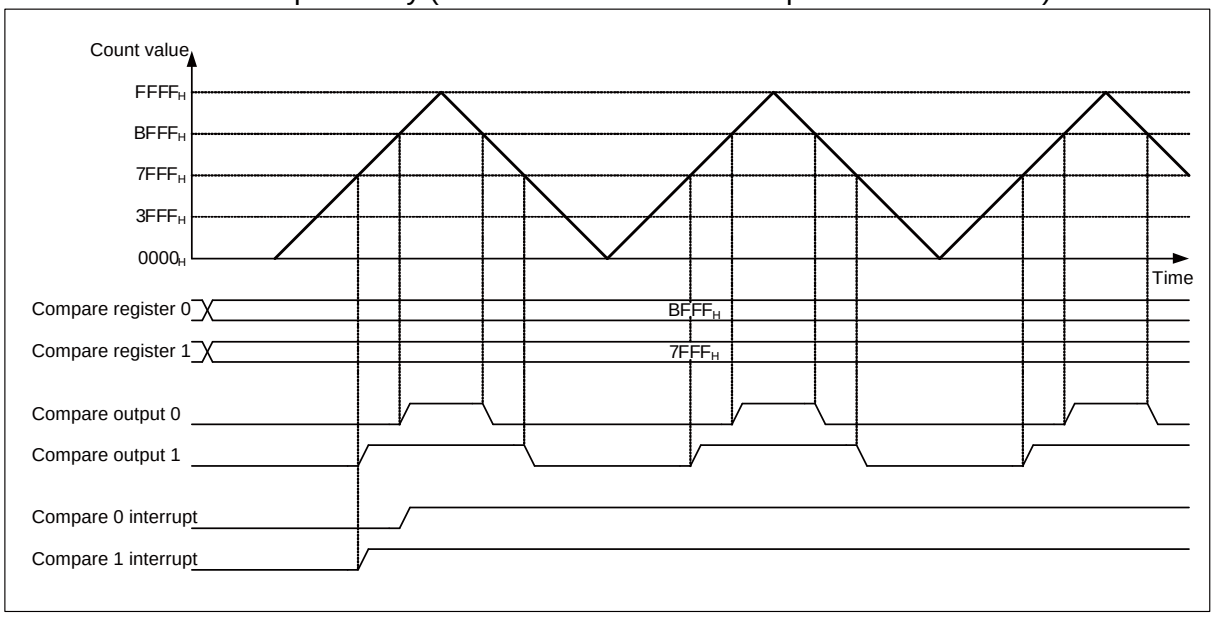
The compare operation can be performed in each of the channels (CMOD: bit12=0 in the compare control register (OCS01)).

Figure 5-1 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used Independently (With the free-run timer in up count mode)



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

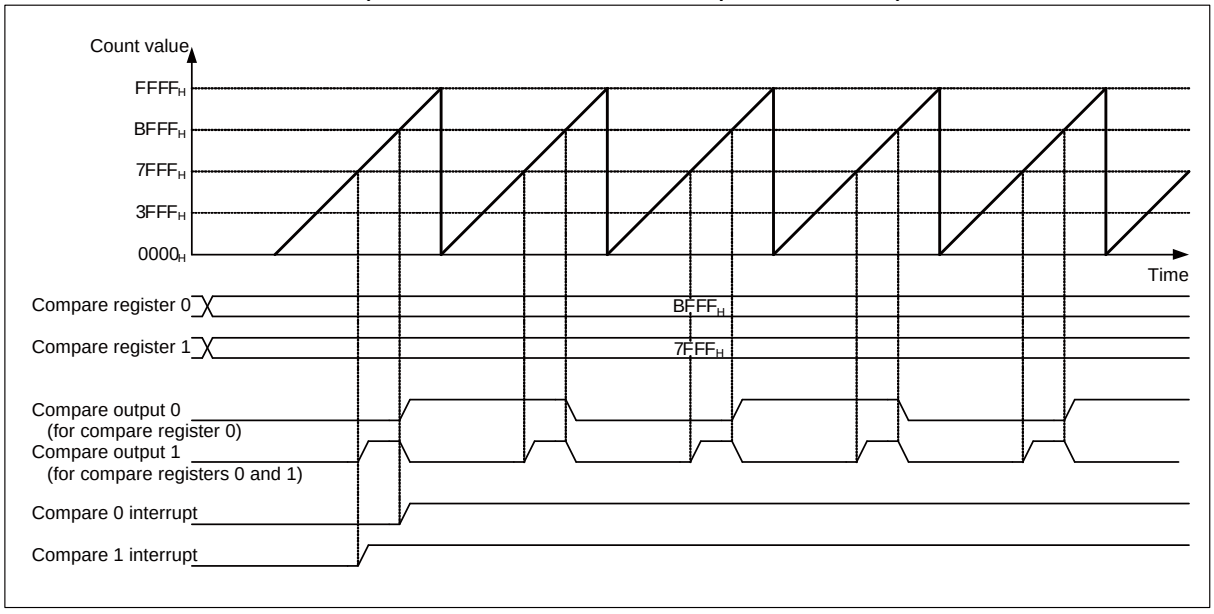
Figure 5-2 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used Independently (With the free-run timer in up/down count mode)



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

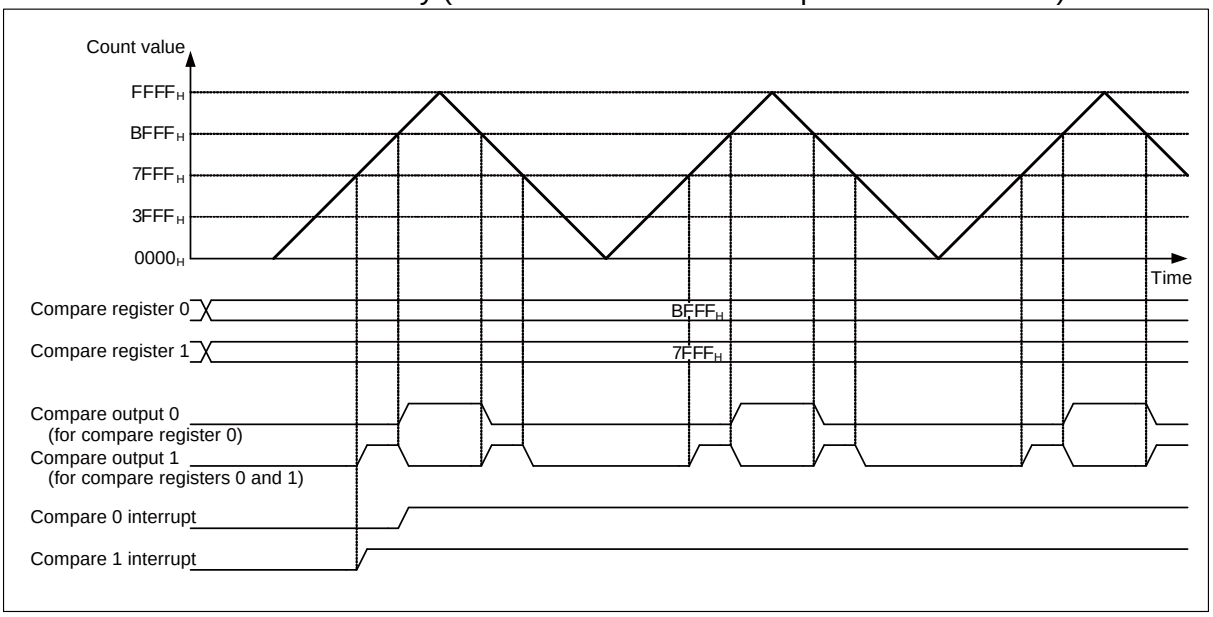
- **The output level can be changed using a pair of compare registers (CMOD: bit12 = 1 in OCS01).**

Figure 5-3 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used as a Pair (With the free-run timer in up count mode)



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

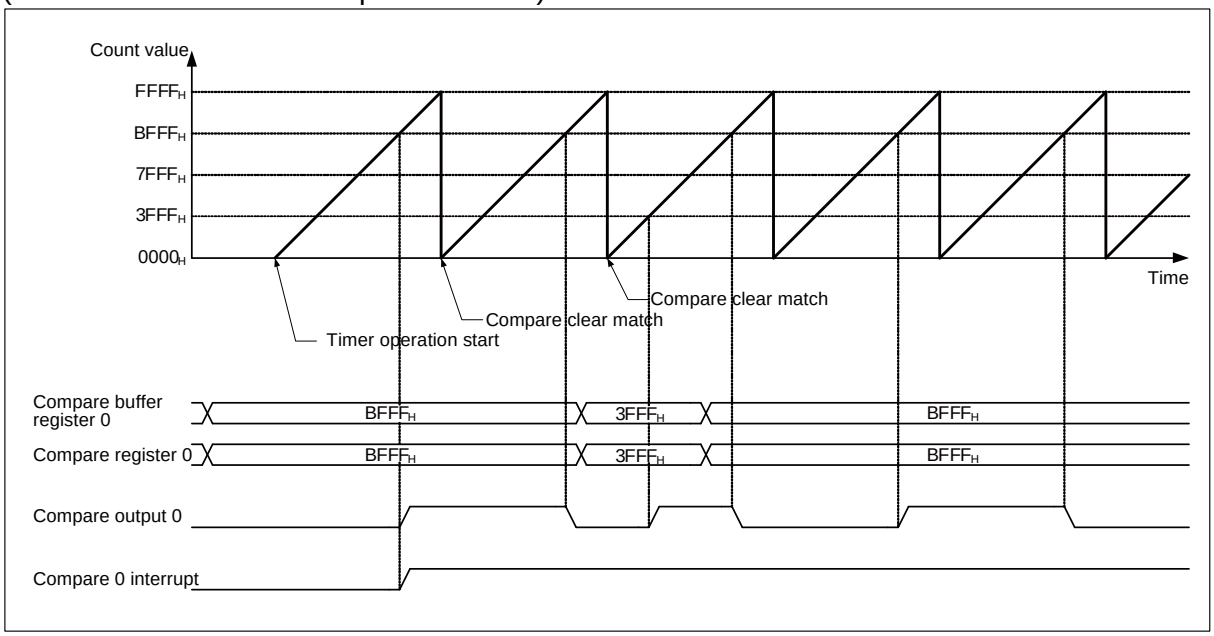
Figure 5-4 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used Simultaneously (With the free-run timer in up/down count mode)



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

● Output level when the compare buffer is invalid

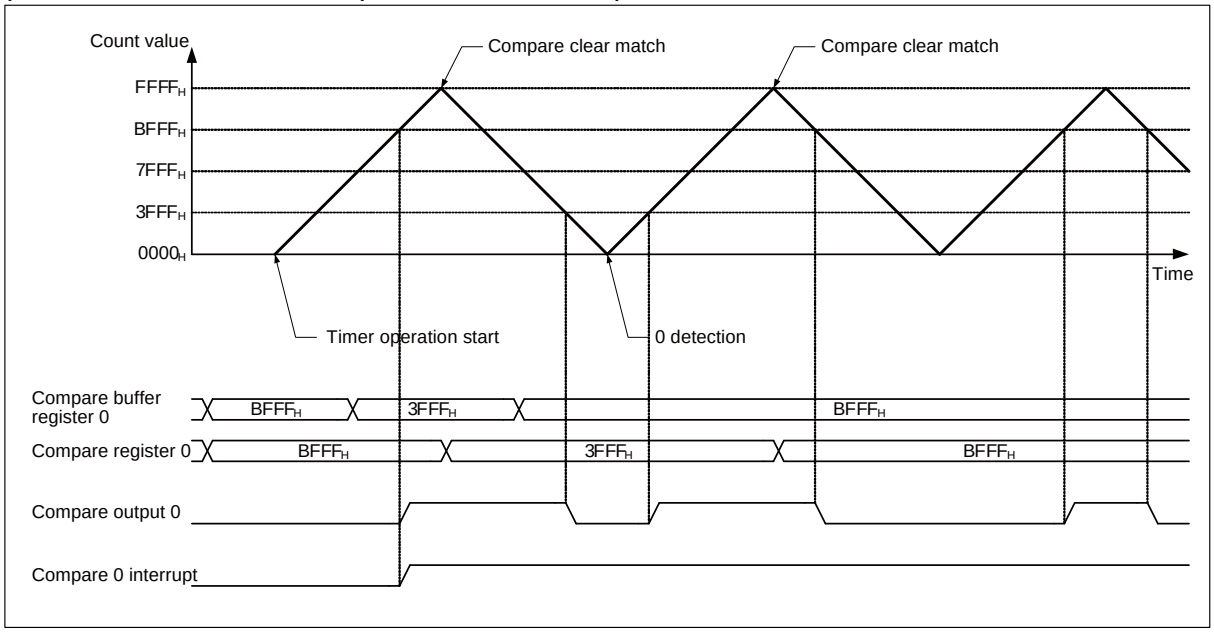
Figure 5-5 Example of Output Waveform When the Compare Buffer Is Invalid
(With the free-run timer in up count mode)



For ch.1, 2, 3, 4 and 5, the operation is the same as ch.0.

● Output level when the compare buffer is selected upon a compare clear match

Figure 5-6 Example of Output Waveform When the Compare Buffer Is Valid
(With the free-run timer in up/down count mode)

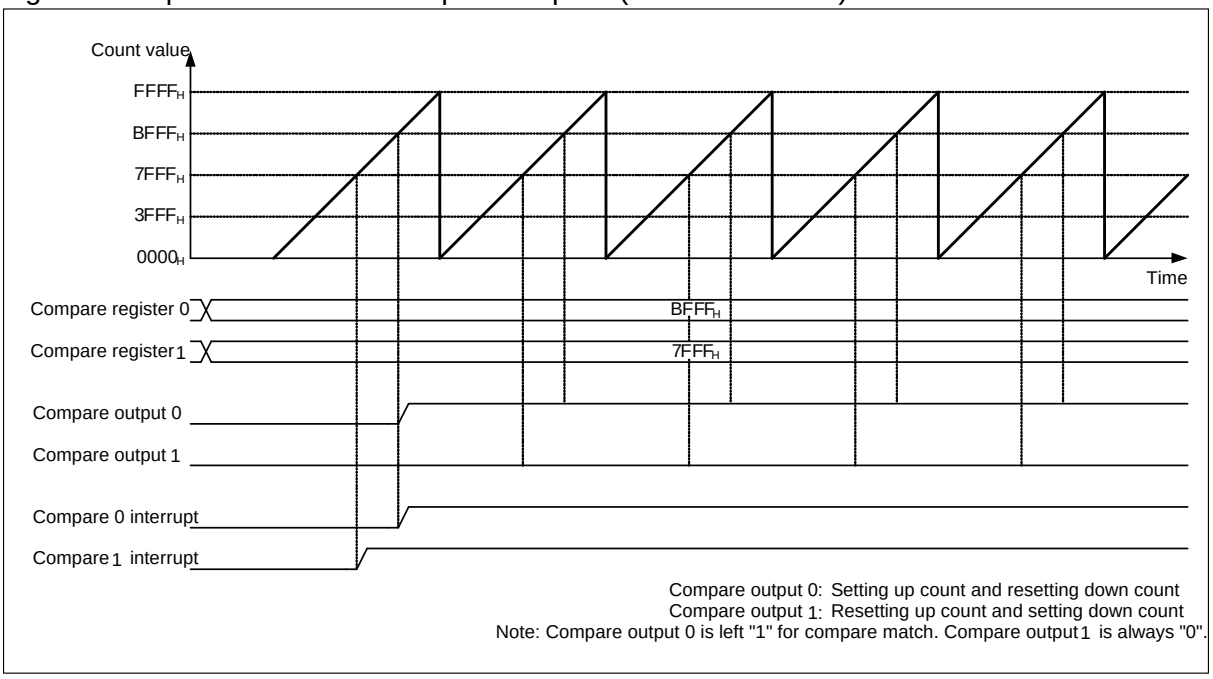


For ch.1, 2, 3, 4 and 5, the operation is the same as ch.0.

5.2.2. Operation of 16-bit Output Compare (Set/Reset Mode, MOD0 = 1 in OCMOD01 Register)

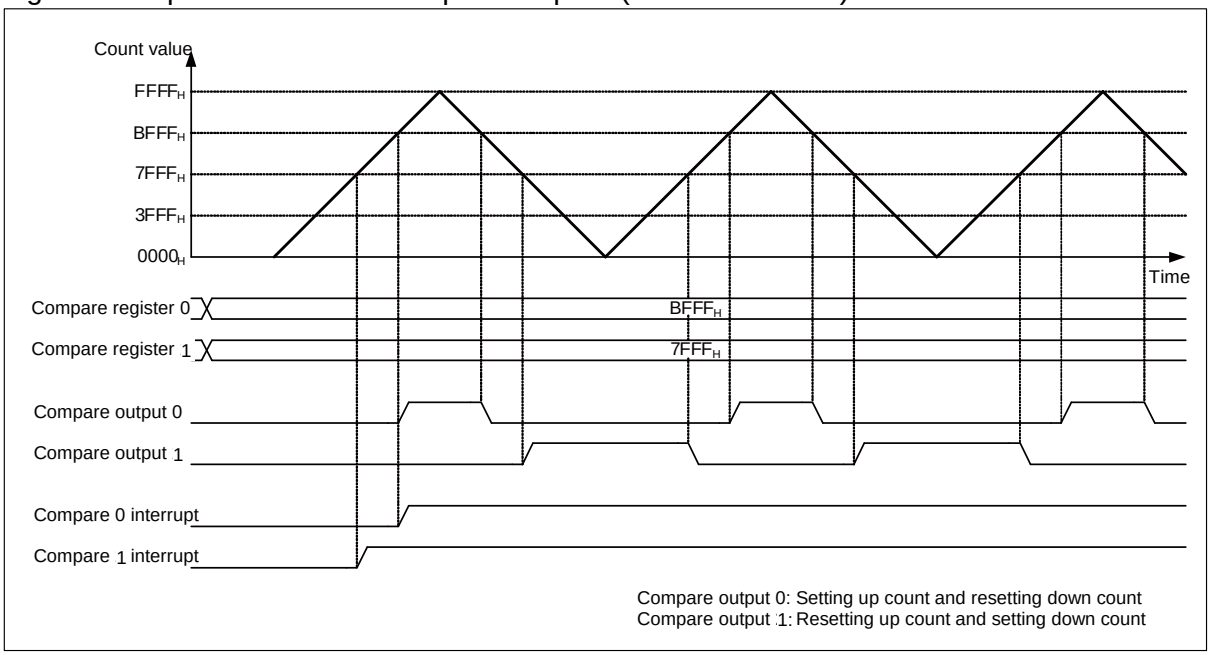
The operation of 16-bit output compare (Set/Reset mode, MOD0 = 1 in OCMOD01 register) is shown.

Figure 5-7 Operation of 16-bit Output Compare (Set/Reset Mode) #1



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

Figure 5-8 Operation of 16-bit Output Compare (Set/Reset Mode) #2



For ch.2, 3 and ch.4, 5, the operation is the same as ch.0, 1.

5.2.3. 16-bit Output Compare Timing

This section explains the 16-bit output compare timing.

When the free-run timer value matches the compare register value, the output compare generates a compare match signal and inverts the output to generate an interrupt. When a compare match occurs, the output is inverted in synchronization with the counter count timing.

Figure 5-9 Compare Register Interrupt Timing

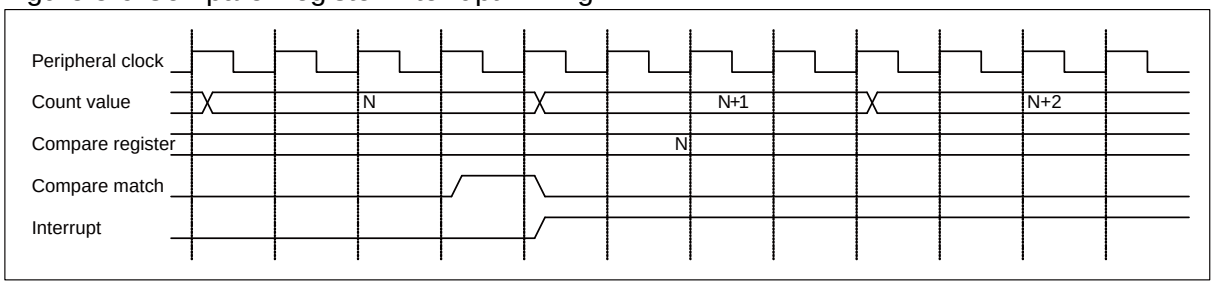
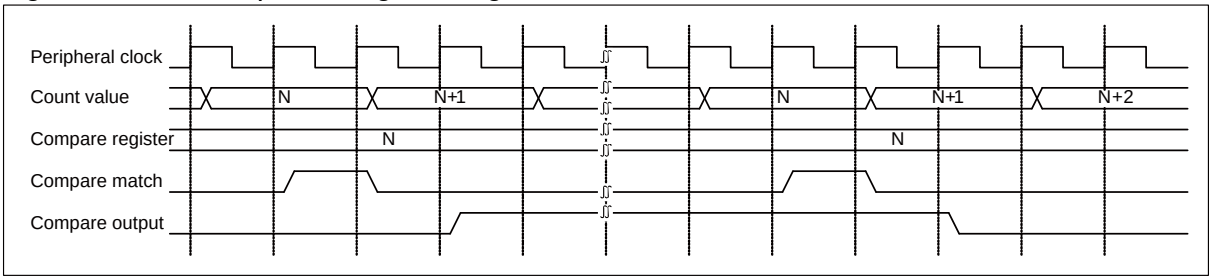


Figure 5-10 Pin Output Change Timing

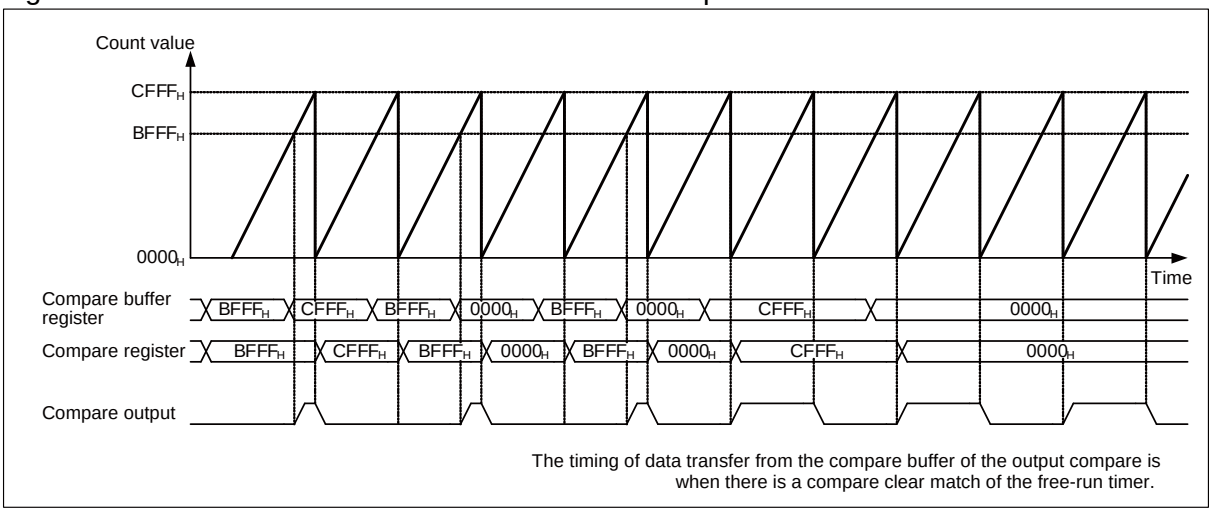


5.2.4. Operation of 16-bit Output Compare and Free-run Timer

This section explains the operation of 16-bit output compare and free-run timer.

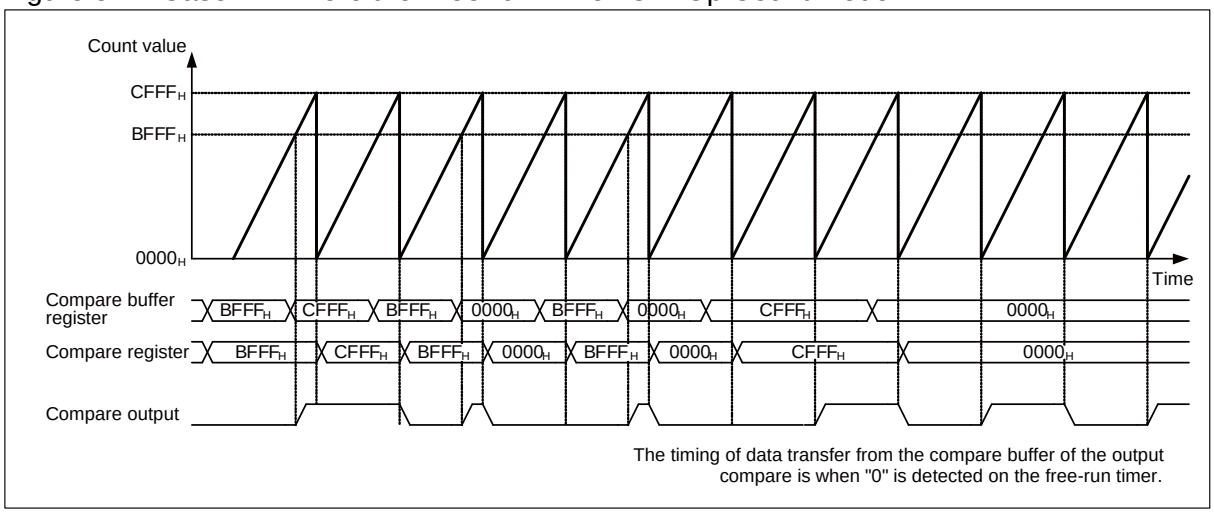
- **Case #1 where the free-run timer is in up count mode**

Figure 5-11 Case #1 Where the Free-run Timer Is in Up Count Mode



Case #2 where the free-run timer is in up count mode

Figure 5-12 Case #2 Where the Free-run Timer Is in Up Count Mode



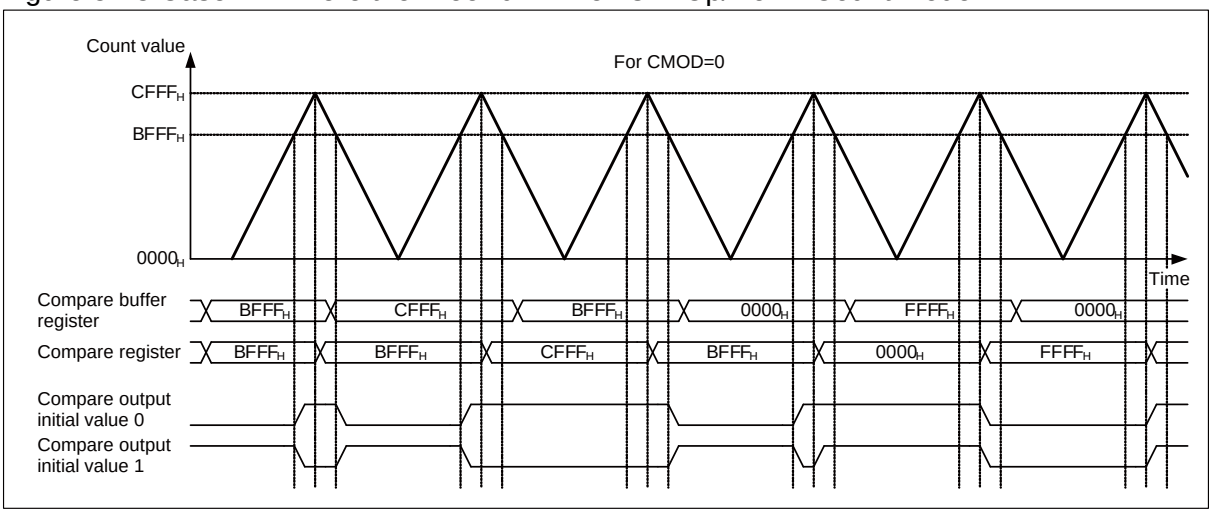
Case #1 where the free-run timer is in up/down count mode

- The timing of data transfer from the compare buffer of the output compare is when there is a compare clear match of the free-run timer.
- When there is an output compare output match, the output is in inverted mode.

Notes:

- When the compare register value is set to "0000_H", the output compare output is set to "1" regardless of the count value of the free-run timer (or reset to "0" when CMOD: bit12 = 1 in OCS).
- When the compare register value is set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer (or set to "1" when CMOD: bit12 = 1 in OCS).
- No comparison is made when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare. Note that a compare match occurs only once at the time of starting the free-run timer when the initial value of the free-run timer is same as the compare clear register value. If, at this time, both the compare clear register value and the compare register value are set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.

Figure 5-13 Case #1 Where the Free-run Timer Is in Up/Down Count Mode

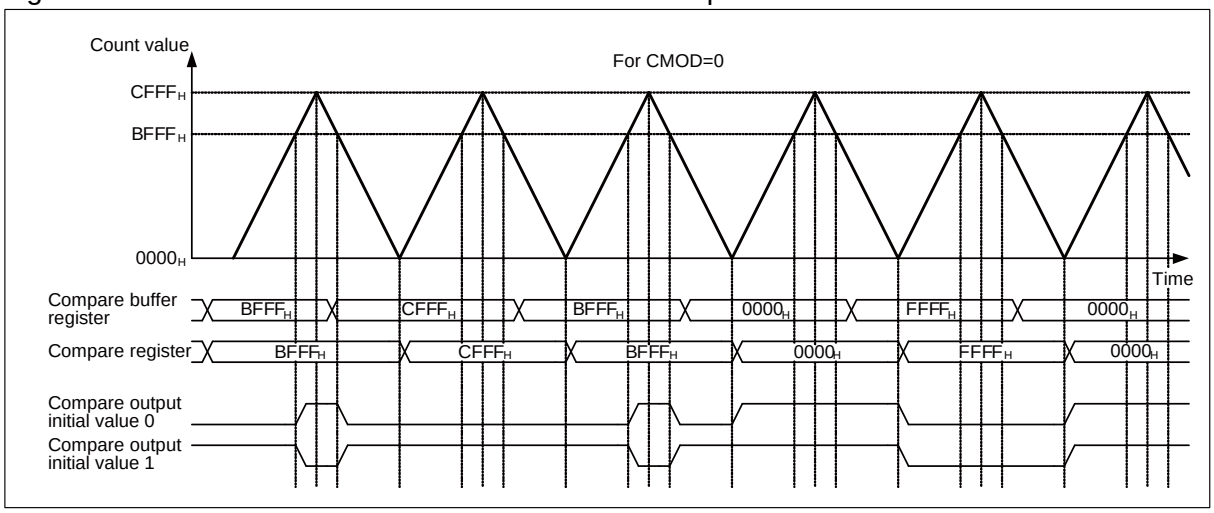


- **Case #2 where the free-run timer is in up/down count mode**
  - The timing of data transfer from the compare buffer of the output compare is when "0" is detected on the free-run timer.
  - When there is an output compare output match, the output is in inverted mode.

Notes:

- When the compare register value is set to "0000_H", the output compare output is set to "1" regardless of the count value of the free-run timer (or reset to "0" when CMOD: bit12 = 1 in OCS).
- When the compare register value is set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer (or set to "1" when CMOD: bit12 = 1 in OCS).
- No comparison is made when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare. Note that a compare match occurs only once at the time of starting the free-run timer when the initial value of the free-run timer is same as the compare clear register value. If, at this time, both the compare clear register value and the compare register value are set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.

Figure 5-14 Case #2 Where the Free-run Timer Is in Up/Down Count Mode



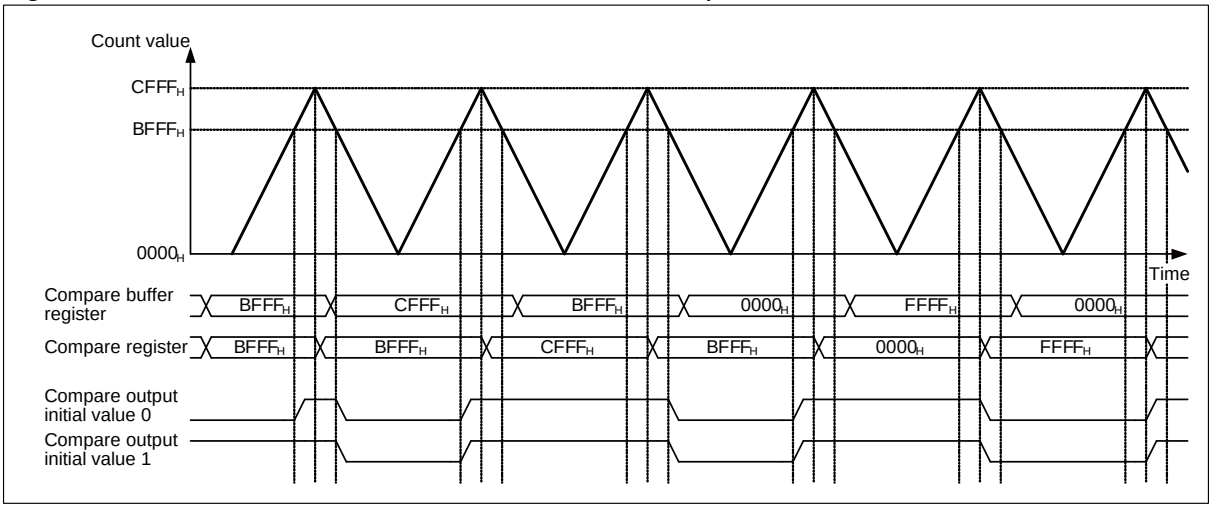
Case #3 where the free-run timer is in up/down count mode

- The timing of data transfer from the compare buffer of the output compare is when there is a compare clear match of the free-run timer.
- The output compare output is set to "1" upon a match in up count mode or reset to "0" upon a match in down count mode (CMOD: bit12=0 in OCS01).
- The ch.2, 3 and ch.4, 5 have the same operation.

Notes:

- When the compare register value is set to "0000_H", the output compare output is set to "1" regardless of the count value of the free-run timer.
- When the compare register value is set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.
- No comparison is made when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare. Note that a compare match occurs only once at the time of starting the free-run timer when the initial value of the free-run timer is same as the compare clear register value. If, at this time, both the compare clear register value and the compare register value are set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.

Figure 5-15 Case #3 Where the Free-run Timer Is in Up/Down Count Mode

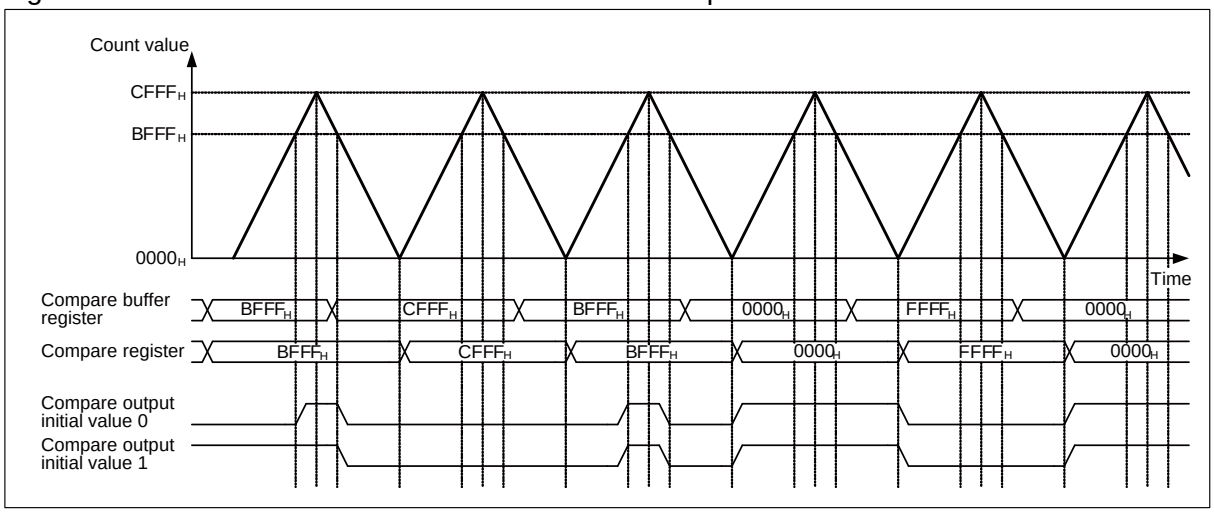


- **Case #4 where the free-run timer is in up/down count mode**
  - The timing of data transfer from the compare buffer of the output compare is when "0" is detected on the free-run timer.
  - The output compare output is set to "1" upon a match in up count mode or reset to "0" upon a match in down count mode (CMOD: bit12=0 in OCS01).
  - The ch.2, 3 and ch.4, 5 have the same operation.

Notes:

- When the compare register value is set to "0000_H", the output compare output is set to "1" regardless of the count value of the free-run timer.
- When the compare register value is set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.
- No comparison is made when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare. Note that a compare match occurs only once at the time of starting the free-run timer when the initial value of the free-run timer is same as the compare clear register value. If, at this time, both the compare clear register value and the compare register value are set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.

Figure 5-16 Case #4 Where the Free-run Timer Is in Up/Down Count Mode



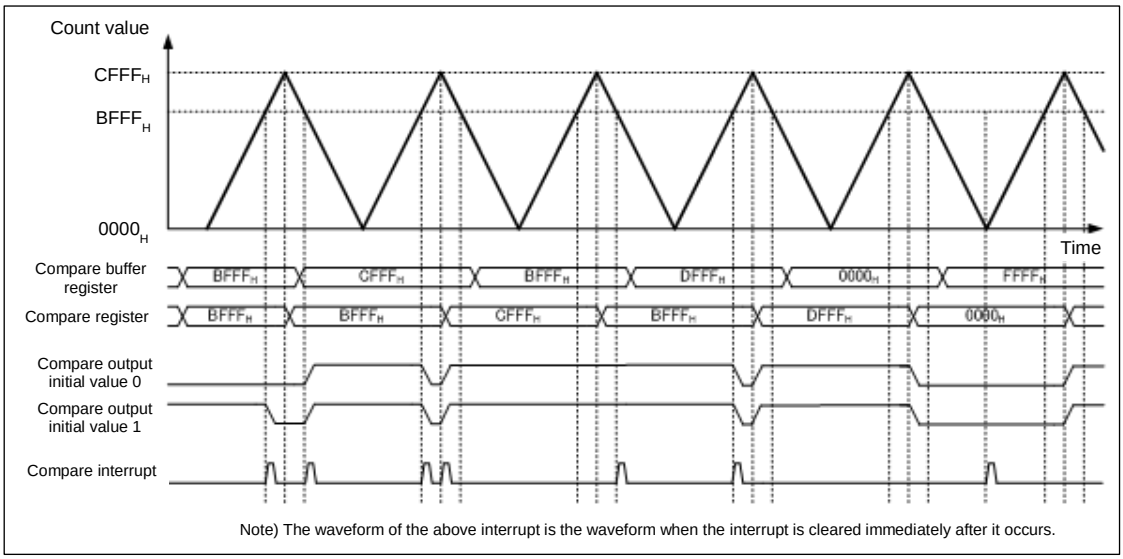
● Case #5 where the free-run timer is in up/down count mode

- The timing of data transfer from the compare buffer of the output compare is when there is a compare clear match of the free-run timer.
- The output compare output is reset to "0" upon a match in up count mode or set to "1" upon a match in down count mode (CMOD: bit12=1 in OCS01).
- The ch.2, 3 and ch.4, 5 have the same operation.

Notes:

- When the compare register value is set to "0000_H", the output compare output is set to "0" when there is a compare clear match of the free-run timer.
- When the compare register value is set to a value greater than or equal to the compare clear register value of the free-run timer, the output compare output is "1" when there is a compare clear match of the free-run timer.
- A comparison will be made and an interrupt flag occurs when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare.

Figure 5-17 Case #5 Where the Free-run Timer Is in Up/Down Count Mode



Note) The waveform of the above interrupt is the waveform when the interrupt is cleared immediately after it occurs.

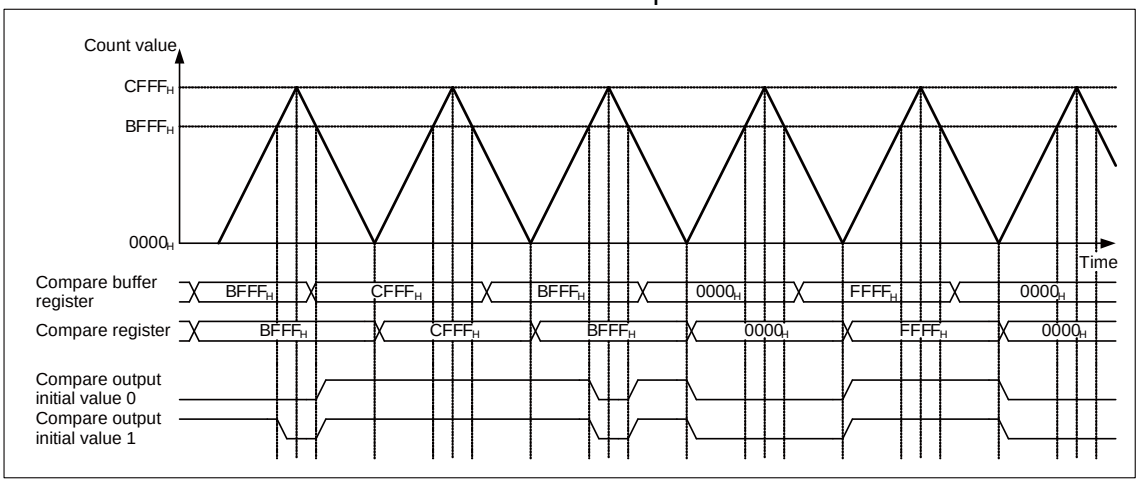
● Case #6 where the free-run timer is in up/down count mode

- The timing of data transfer from the compare buffer of the output compare is when "0" is detected on the free-run timer.
- The output compare output is reset to "0" upon a match in up count mode or set to "1" upon a match in down count mode (CMOD: bit12=1 in OCS01).
- The ch.2, 3 and ch.4, 5 have the same operation.

Notes:

- When the compare register value is set to "0000_H", the output compare output is reset to "0" regardless of the count value of the free-run timer.
- When the compare register value is set to "$FFFF_H$", the output compare output is set to "1" regardless of the count value of the free-run timer.
- No comparison is made when there is a match between the compare clear register value of the free-run timer and the compare register value of the output compare. Note that a compare match occurs only once at the time of starting the free-run timer when the initial value of the free-run timer is same as the compare clear register value. If, at this time, both the compare clear register value and the compare register value are set to "$FFFF_H$", the output compare output is reset to "0" regardless of the count value of the free-run timer.

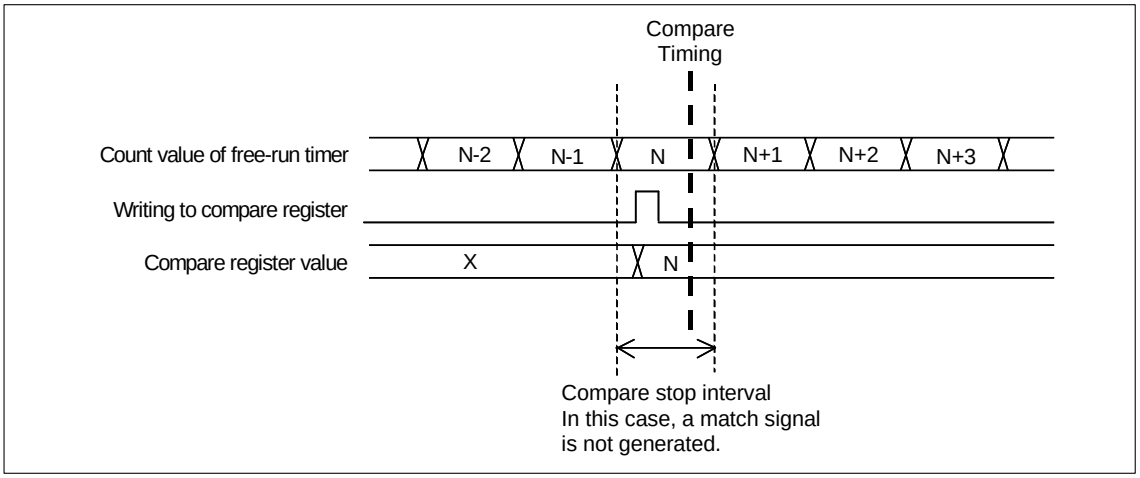
Figure 5-18 Case #6 Where the Free-run Timer Is in Up/Down Count Mode



5.3. Notes on Using 16-bit Output Compare

The notes on using 16-bit output compare is shown.

- Compare stop section in the compare operation
  The compare operation is stopped for one count cycle just after a compare value is written to the compare register as shown below.



- If the settings are CMOD = 1 and OCCP0 = OCCP1, OCCP2 = OCCP3 and OCCP4 = OCCP5, the port is inverted only once when a compare match occurs.
- Be sure to stop the compare operation before specifying the output level of the output compare output.
- Stopping the free-run timer stops the compare operation because the output compare is in synchronization with the free-run timer.
- An interrupt operation occurs independently for each of OCU0 to OCU5 when the compare mode bit CMOD is set

to 1.

- When the free-run timer is used as compare data for the output compare, the free-run timer clock frequency (TCCSL.CLK[3:0]) cannot be set to "0000_B".

Read-modify-write

The interrupt request flag bits (IOP0) and (IOP1) are "1" when read using a read-modify-write instruction.

Chapter 26: 16-Bit Input Capture

This chapter explains the 16-bit input capture.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS23-1v0-91520-2-E

1. Overview

This section explains the overview of the 16-bit input capture.

This product includes four 16-bit input capture channels.

2. Features

This section explains features of the 16-bit input capture.

■ Functions of 16-bit Input Capture

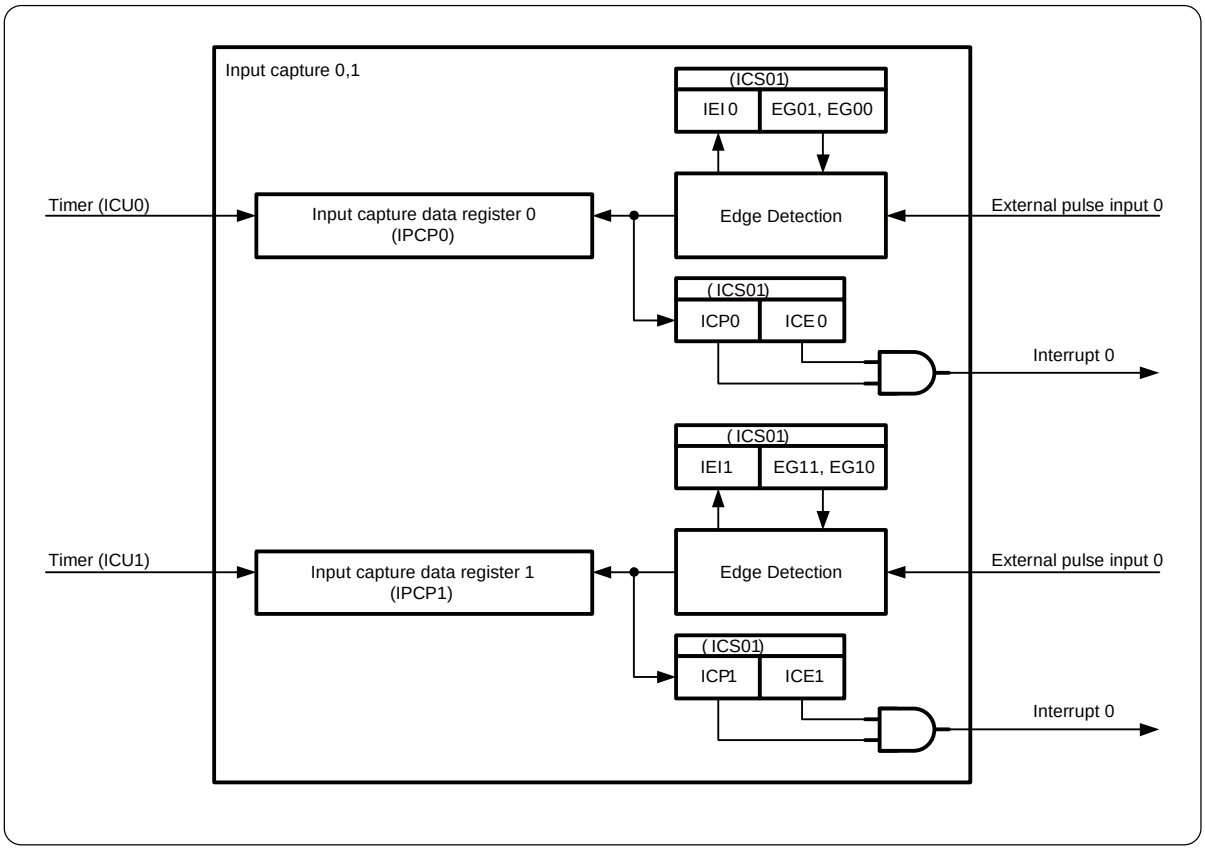
- The 16-bit input capture consists of 4 independent external input pins, capture registers corresponding to this pin, and capture control registers. When an edge of the input signal from the external pin is detected, the value of the 16-bit free-run timer can be stored in the capture register and an interrupt is generated simultaneously.
- 3 types of trigger edge (rising edge, falling edge, and both edges) of the external input signal can be selected and there is a register that indicates whether the trigger edge is rising or falling.
- The 4 input capture channels can be operated independently.
- An interrupt is generated when a valid edge from the external input is detected.
- Any desired free-run timer channel can be set for each compare unit.
- There are 4 input capture channels, input capture 0 through 3, for which any of free-run timers 0 through 2 can be selected as the input. This can be set at the free-run timer selection register (FRS1). See "Free-run Timer Selection Register" of "CHAPTER: 16-BIT FREE-RUN TIMER" for details.

3. Configuration

This section explains the configuration of the 16-bit input capture.

■ Configuration of 16-bit Input Capture

Figure 3-1 Configuration of 16-bit Input Capture (For ch.0, ch.1)



4. Registers

This section explains registers of the 16-bit input capture.

■ Table of external pins

Channel	External pin (ICU input)				
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK, MB91F52xL
0	ICU0_3	ICU0_3	ICU0_2/ ICU0_3	ICU0_1/ ICU0_2/ ICU0_3	ICU0_0/ ICU0_1/ ICU0_2/ ICU0_3
1	ICU1_1/ ICU1_2	ICU1_1/ ICU1_2	ICU1_1/ ICU1_2/ ICU1_3	ICU1_1/ ICU1_2/ ICU1_3	ICU1_0/ ICU1_1/ ICU1_2/ ICU1_3
2	ICU2_2/ ICU2_3	ICU2_2/ ICU2_3	ICU2_1/ ICU2_2/ ICU2_3	ICU2_1/ ICU2_2/ ICU2_3	ICU2_0/ ICU2_1/ ICU2_2/ ICU2_3
3	ICU3_0/ ICU3_1/ ICU3_3	ICU3_0/ ICU3_1/ ICU3_2/ ICU3_3	ICU3_0/ ICU3_1/ ICU3_2/ ICU3_3	ICU3_0/ ICU3_1/ ICU3_2/ ICU3_3	ICU3_0/ ICU3_1/ ICU3_2/ ICU3_3

■ List of 16-bit Input Capture Registers

Table 4-1 List of 16-bit Input Capture Registers

Address	+0	+1	+2	+3
0x0000127C	Input capture data register 0 (IPCP0)		Input capture data register 1 (IPCP1)	
0x00001280	Input capture state control register 01 (ICS01)		Reserved	LIN SYNCH FIELD switching register (LSYNS)
0x00001284	Input capture data register 2 (IPCP2)		Input capture data register 3 (IPCP3)	
0x00001288	Input capture state control register 23 (ICS23)		Reserved	Reserved

4.1. 16-bit Input Capture Registers

This section explains registers of the 16-bit input capture.

The 16-bit input capture consists of input capture data registers and input capture state control registers.

4.1.1. Input Capture Data Register : IPCP0 to IPCP3

This section explains registers of the 16-bit input capture.

An input capture data register (IPCP) retains the count value of the free-run timer at the time of detection of an effective edge of the input waveform.

■ **IPCP0: Address $127C_H$ (Access: Half-word, Word)**

■ **IPCP1: Address $127E_H$ (Access: Half-word, Word)**

■ **IPCP2: Address 1284_H (Access: Half-word, Word)**

■ **IPCP3: Address 1286_H (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15 to bit0] CP15 to CP00: Free-run timer value

CP15 to CP00	Function
	Free-run timer value

- This register is used to store a free-run timer value at the time of detection of an effective edge of the corresponding external pin input waveform.
- The free-run timer value in the above explanation represents the operating state of a free-run timer for which the input capture has been selected.

Note:

When accessing this register, use a half-word or word access instruction. No data can be written to this register.

4.1.2. Input Capture State Control Register : ICS

The bit configuration for the input capture state control register is shown below.

An input capture state control register (ICS) is used to select an edge, enable interrupt request, and control an interrupt request flag. It is also used to indicate an effective edge detected by the input capture.

■ ICS01: Address 1280_H (Access: Byte, Half-word, Word)

■ ICS23: Address 1288_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						IEI1	IEI0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
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

[bit15 to bit10] Reserved

Always write 0 to these bits.

[bit9] IEI1: Effective edge indication bit

IEI1	Function
0	A falling edge is detected.
1	A rising edge is detected.

- This effective edge indication bit for the capture register (IPCP) indicates that a rising or falling edge has been detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is read-only.

Note:

If EG11, EG10: bit3, bit2 of the input capture state control register (ICS) are set to 00_B, the value read from this register is meaningless.

[bit8] IEI0: Effective edge indication bit

IEI0	Function
0	A falling edge is detected.
1	A rising edge is detected.

- This effective edge indication bit for the capture register (IPCP) indicates that a rising or falling edge has been detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is read-only.

Note:

If EG01, EG00: bit1, bit0 of the input capture state control register (ICS) are set to 00_B, the value read from this register is meaningless.

[bit7] ICP1: Interrupt request flag bit

ICP1	Function	
	Read	Write
0	No effective edge is detected.	This bit is cleared.
1	An effective edge is detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the input capture.
- This bit is immediately set to "1" when an effective edge from the external input pin is detected.
- An interrupt is immediately generated when an effective edge is detected while the interrupt request enable bit (ICE1: bit5) is set.
- When this bit is set to "0":This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Notes:

- If a read-modify-write (RMW) instruction is executed, "1" is always read.
- If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit6] ICP0: Interrupt request flag bit

ICP0	Function	
	Read	Write
0	No effective edge is detected.	This bit is cleared.
1	An effective edge is detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the input capture.
- This bit is immediately set to "1" when an effective edge from the external input pin is detected.
- An interrupt is immediately generated when an effective edge is detected while the interrupt request enable bit (ICE0: bit4) is set.
- When this bit is set to "0":This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Notes:

- If a read-modify-write (RMW) instruction is executed, "1" is always read.
- If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit5] ICE1: Interrupt request enable bit

ICE1	Function
0	Interrupt request disabled
1	Interrupt request enabled

- This bit is used to enable an input capture interrupt request for the input capture.
- An input capture interrupt is generated when an interrupt request flag bit (ICP1: bit7) is set while this bit is set to "1".

[bit4] ICE0: Interrupt request enable bit

ICE0	Function
0	Interrupt request disabled
1	Interrupt request enabled

- This bit is used to enable an input capture interrupt request for the input capture.
- An input capture interrupt is generated when an interrupt request flag bit (ICP0: bit6) is set while this bit is set to "1".

[bit3, bit2] EG11, EG10: Edge selection bits

EG11	EG10	Function
0	0	No edge is detected (Stopped).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

- These bits are used to specify an effective edge polarity of the external input for the input capture.
- These bits are also used to enable the operation of input capture.

[bit1, bit0] EG01, EG00: Edge selection bits

EG01	EG00	Function
0	0	No edge is detected (Stopped).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

- These bits are used to specify an effective edge polarity of the external input for the input capture.
- These bits are also used to enable the operation of input capture.

4.1.3. LIN SYNCH FIELD Switching Register : LSYNS

The bit configuration of the LIN SYNCH FIELD switching register is shown below.

The LIN SYNCH FIELD switching register (LSYNS) is used for LIN linkage control.

■ LSYNS: Address 1283_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				LSYN3	LSYN2	LSYN1	LSYN0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

Always write 0 to these bits.

[bit3 to bit0] LSYN3 to LSYN0: Input capture 3 to 0 input selection

LSYN3 to LSYN0	Function
0	External pin input (ICU3 to ICU0)
1	Input Lin Synch Field detection signals from the multi-function serial interface ch.3 to ch.0.

These bits are used to enable Lin Synch Field from the multi-function serial interface ch.3 to ch.0.

5. Operation

This section explains the operation.

5.1 Interrupts for 16-bit Input Capture

5.2 Operation of 16-bit Input Capture

5.3 Notes on Using the 16-bit Input Capture

5.1. Interrupts for 16-bit Input Capture

This section explains the interrupts for 16-bit input capture

Table 5-1shows the interrupt control bits and interrupt factor of the 16-bit input capture.

Table 5-1 Interrupt Control Bits and Interrupt Factor of 16-bit Input Capture

	16-bit input capture	
	Even-number channel	Odd-number channel
Interrupt request flag bit	Input capture state control register (ICS) ICP0: bit6	Input capture state control register (ICS) ICP1: bit7
Interrupt request enable bit	Input capture state control register (ICS) ICE0: bit4	Input capture state control register (ICS) ICE1: bit5
Interrupt factor	An effective edge is detected at the IN pin.	An effective edge is detected at the IN pin.

With 16-bit input capture, when an effective edge is detected at a pin, the input capture state control register (ICS) ICP1/ICP0: bit7/bit6 are set to "1". If interrupt requests are enabled (ICE1/ICE0:bit5 and bit4 of ICS01 is 1) with this state, an interrupt request is output to the interrupt controller.

5.2. Operation of 16-bit Input Capture

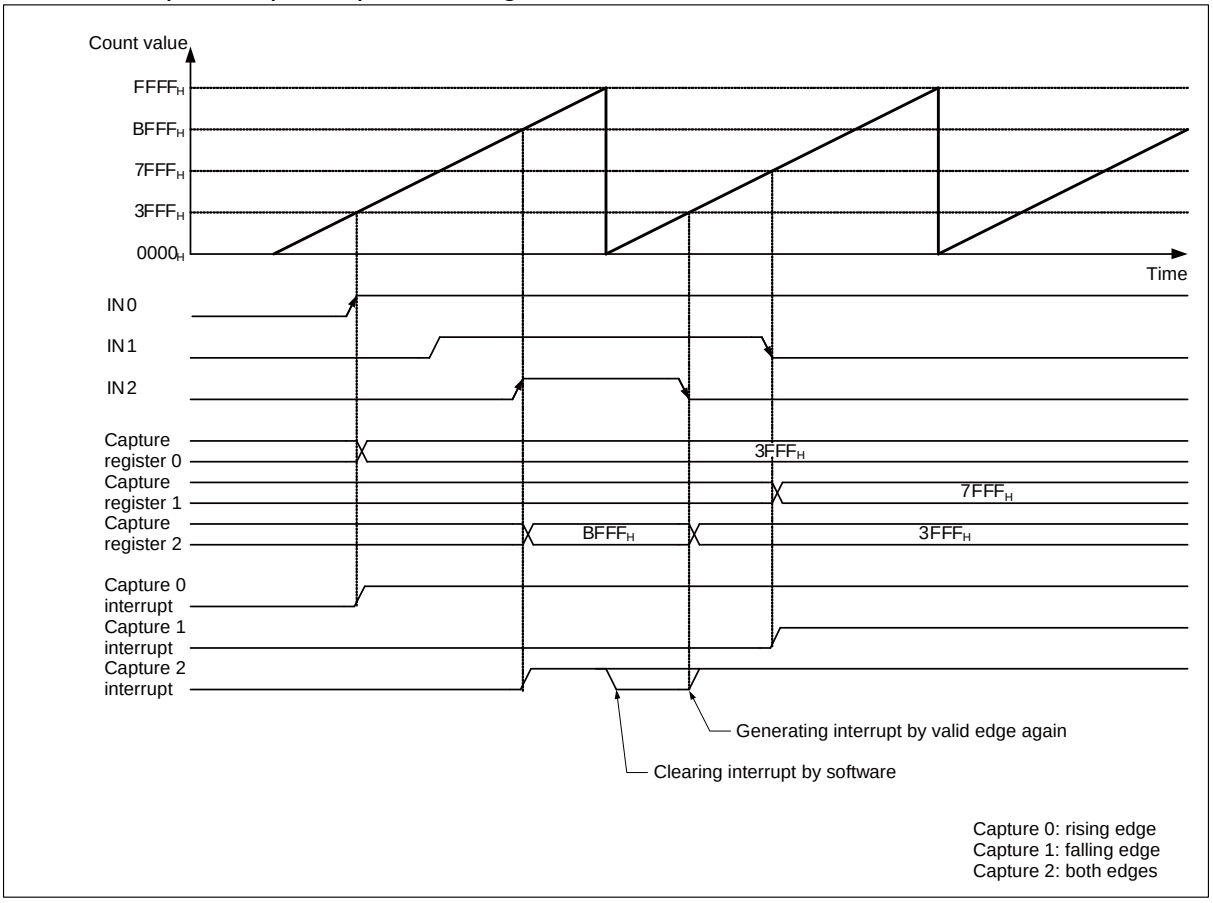
The operation of 16-bit Input capture is shown below.

Input capture is used to detect a specified effective edge. When an effective edge is detected, an interrupt flag is set and the value of the 16-bit free-run timer is loaded to the capture register.

5.2.1. Operation of 16-bit Input Capture

The operation of 16-bit Input capture is shown below.

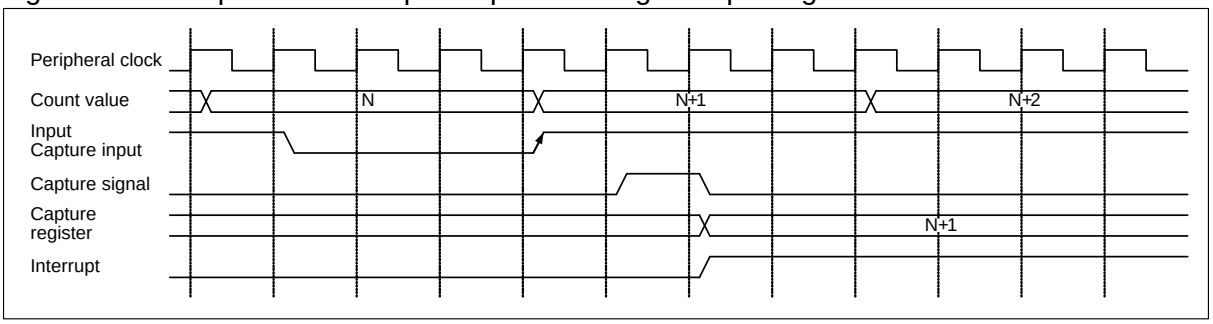
Figure 5-1 Example of Input Capture Timing



5.2.2. 16-bit Input Capture Input Timing

The operation of 16-bit Input capture input timing is shown below.

Figure 5-2 Example of 16-bit Input Capture Timing for Input Signals



5.3. Notes on Using the 16-bit Input Capture

This section explains the notes on using the 16-bit input capture.

If the input capture pin (IN) level is changed during the period from the bit setting of ICP1/ICP0 of the input capture state control register (ICS01) to the processing of an interrupt routine, the ICP1/ICP0 effective edge indication bits (IEI1 and IEI0 of ICS01 register) indicate the latest edge detected.

- For ch.2, 3, the same notes as ch.0, 1 are required.

● Input capture register

Reading from the input capture register must be performed in 16-bit or 32-bit access.

● Read-modify-write

When reading is performed using a read-modify-write instruction, ICP1 and ICP0 of the input capture state control register (ICS01) are read as "1".

- For ch.2, 3, the same notes as ch.0, 1 are required.

● Note on interrupts

Before the input capture state control register (ICS) interrupt request enable bits (ICE1/ICE0) are set to "1", be sure to clear the interrupt flags (ICP1/ICP0).

Chapter 27: Up/Down Counter

This chapter explains the up/down counter.

1. Overview
2. Features
3. Configuration
4. Registers
5. Interrupt
6. Operation and Setting Procedure Examples

Code : FG20-1v0-91520-2-E

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.

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 2-1 lists the up/down counter operating modes.

Table 2-1 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.

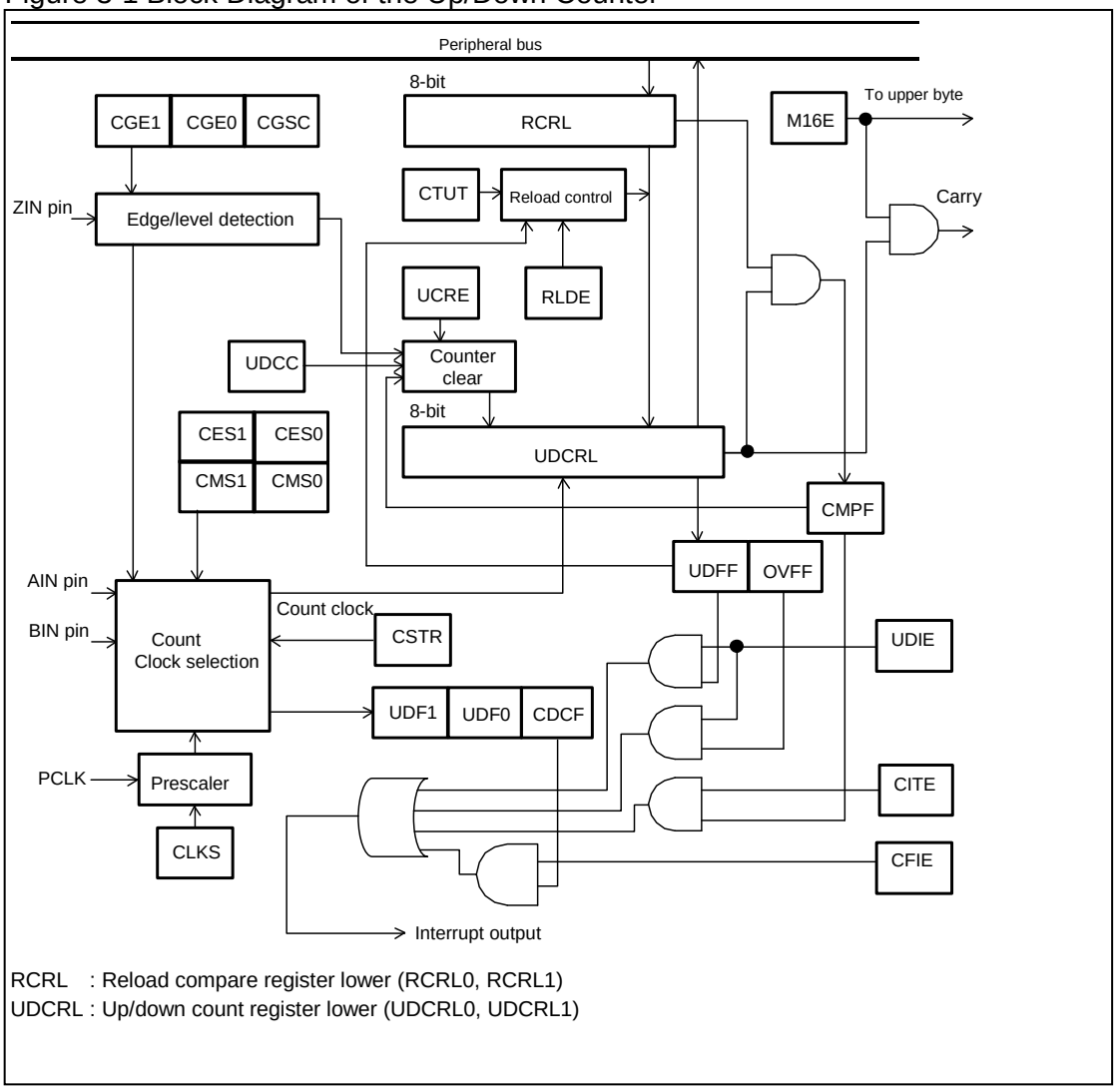
3. Configuration

This section explains the configuration of the up/down counter.

■ Block Diagram of the Up/Down Counter

Figure 3-1 shows the block diagram of the up/down counter using ch.0 as an example.

Figure 3-1 Block Diagram of the Up/Down Counter



- 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 3-1 lists the clocks used for the up/down counter.

Table 3-1 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

4. Registers

This section explains the registers of the up/down counter.

■ Correspondence between Pins and Channels

Table 4-1 shows the correspondence between channels and pins.

Table 4-1 Correspondence between Pins and Channels

Channel	External signal input pins								
	MB91F52xB			MB91F52xD			MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL		
	AIN	BIN	ZIN	AIN	BIN	ZIN	AIN	BIN	ZIN
0	AIN0_0	BIN0_0/ BIN0_1	ZIN0_2	AIN0_0/ AIN0_1	BIN0_0/ BIN0_1	ZIN0_2	AIN0_0/ AIN0_1	BIN0_0/ BIN0_1	ZIN0_0/ ZIN0_1/ ZIN0_2
1	AIN1_0/ AIN1_1	BIN1_0	ZIN1_0/ ZIN1_1	AIN1_0/ AIN1_1	BIN1_0/ BIN1_1	ZIN1_0/ ZIN1_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 4-2 lists the up/down counter register map.

Table 4-2 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

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.

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 0F72$_H$ (Access : Half-word, Word)

■ UDCRH1 : Address 0F82$_H$ (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 $0F73_H$ (Access : Byte, Half-word, Word)

■ UDCRL1 : Address $0F83_H$ (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

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.

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 $0F74_H$ (Access : Byte, Half-word)**

■ **CCR1 : Address $0F84_H$ (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 operation mode has been selected.

[bit11, bit10] CMS1, CMS0 : Operation mode select bits

Select an operation mode of the up/down counter as follows.

- Timer mode
  The timer 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 "0000_H" 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 underflowed 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 "0000_H" 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 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	Setting disabled	Setting disabled

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 $0F77_H$ (Access : Byte)

■ CSR1 : Address $0F87_H$ (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

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 5-1 shows the correspondence between operating modes and interrupt requests.

Table 5-1 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 5-2 shows interrupts that can be used for the up/down counter.

Table 5-2 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 B. 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)".

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.

● Operation mode

One of the following three modes (four types) can be selected as the operation 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 "■ Counting" in "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 "■ Counting" in "6.2 Operation in Up/down Count 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 "6.2 Operation in Up/down Count Mode".
  This function is not available in timer mode.

Table 6-1 shows the setting method for the reload/compare functions.

Table 6-1 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.

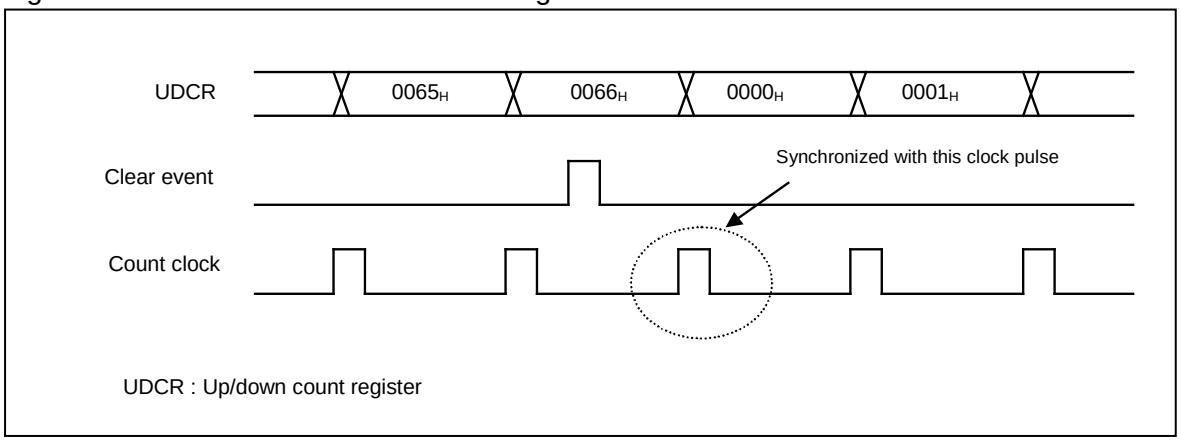
  (The counter is not cleared if an attempt is made to decrement or stop the counter.)
- Clear due to overflow
  Count up/down timing after the counter reaches "$FFFF_H$" (or "FF_H" in 8-bit mode).

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.

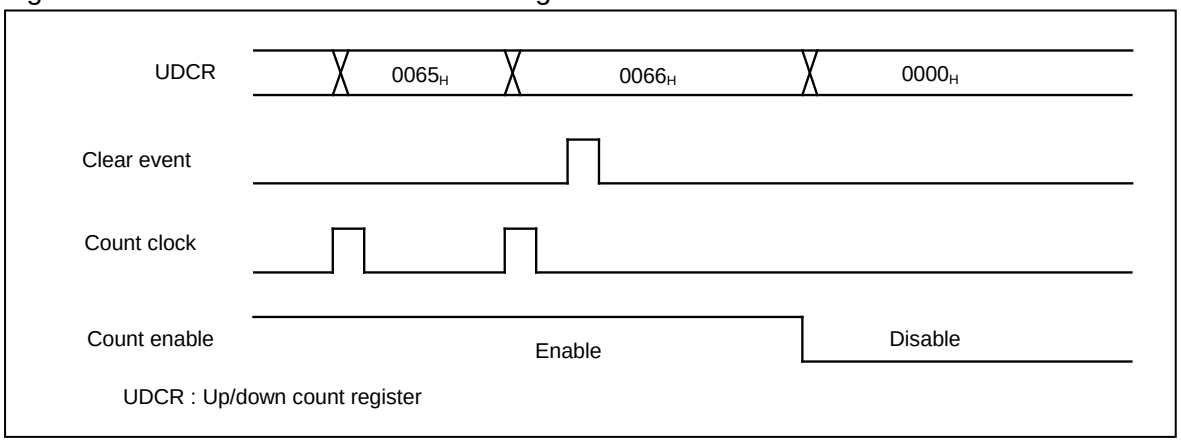
Figure 6-1 shows clear event occurrence timing.

Figure 6-1 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 6-2 shows the clear event occurrence timing.

Figure 6-2 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.

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

1. The reload/compare value is set in the reload compare register (RCR).
2. 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).
3. 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 "4.3 Counter Control Register (CCR0, CCR1)".

Note:

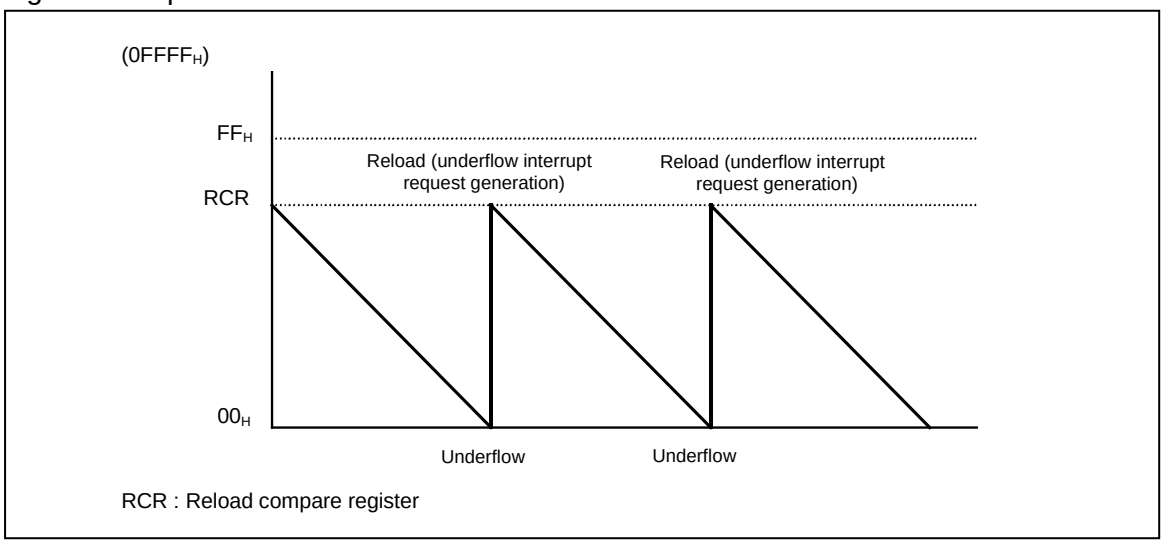
The minimum pulse width required at the ZIN pin is 2T (T is the cycle of the peripheral clock (PCLK)).

● 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 6-3 shows the operation performed when the reload function is in use.

Figure 6-3 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".

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 "4.3 Counter Control Register (CCR0, CCR1)".

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 "6.1 Operation in Timer Mode".

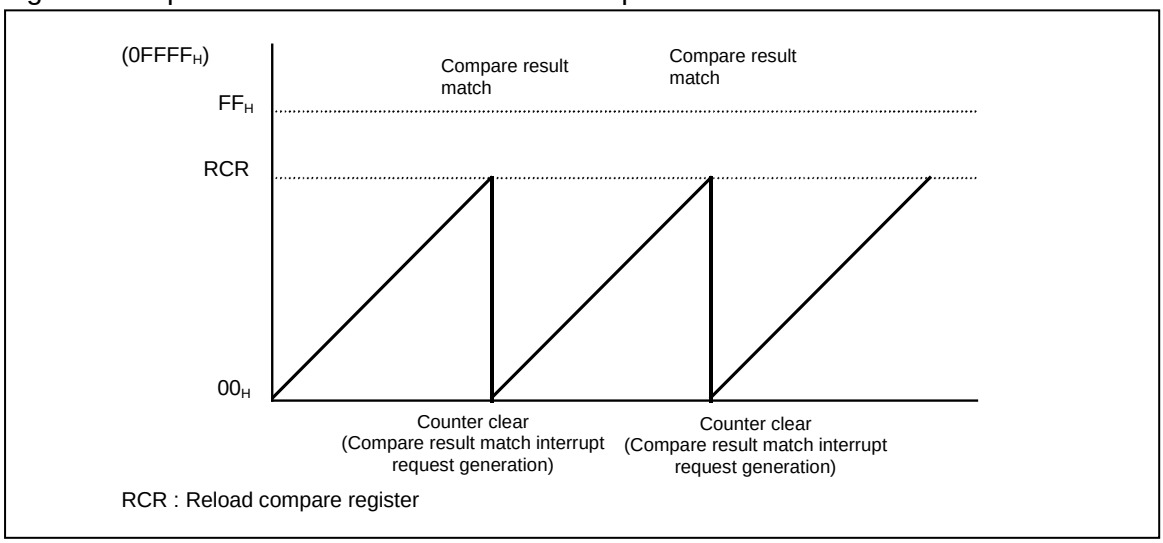
● Operation performed when the compare function is in use

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 6-4 shows the operation performed when the compare function is in use.

Figure 6-4 Operation Performed When the Compare Function is in Use



Note:

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:

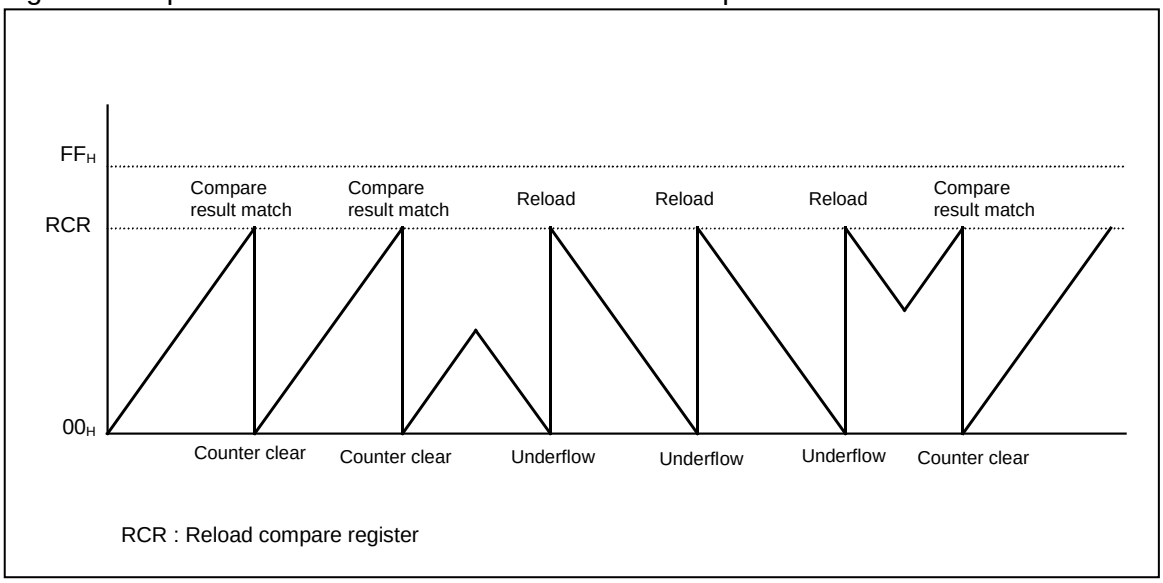
- The next operation is counting down.
- The up/down counter is inactive.

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 6-5 shows operation performed when the reload compare function is in use.

Figure 6-5 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 6-2 lists the counting directions indicated with the UDF1 and UDF0 bits.

Table 6-2 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".

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 6-3 lists the count methods.

Table 6-3 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

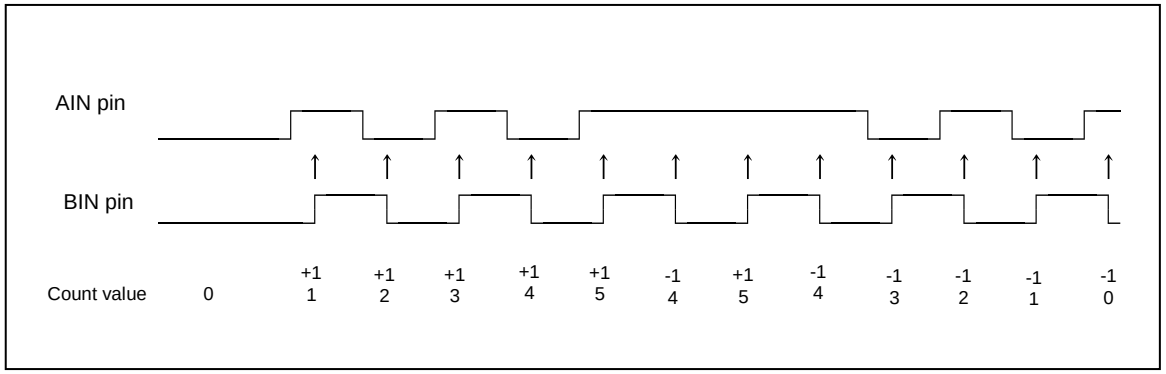
■ Counting

● 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 6-6 shows the operation in the phase difference count mode (multiply-by-two).

Figure 6-6 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 "4.3 Counter Control Register (CCR0, CCR1)".

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 "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 "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 "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 "6.2 Operation in Up/down Count Mode".

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 6-4 lists the count methods.

Table 6-4 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

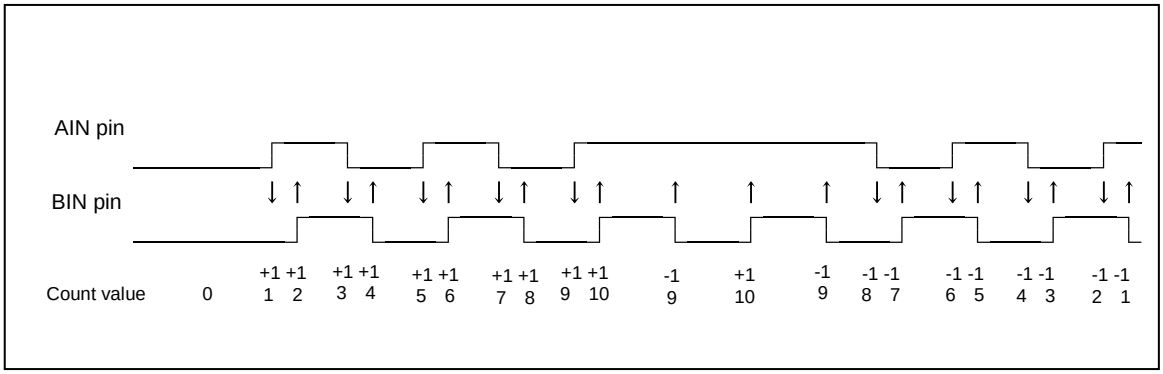
■ Counting

● 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 6-7 shows the operation in the phase difference count mode (multiply-by-four).

Figure 6-7 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 "4.3 Counter Control Register (CCR0, CCR1)".

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 "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 "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 "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 "6.2 Operation in Up/down Count Mode".

Chapter 28: Real-Time Clock(RTC)

This chapter explains the real-time clock (RTC).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Setting
7. Q&A
8. Sample Program
9. Notes

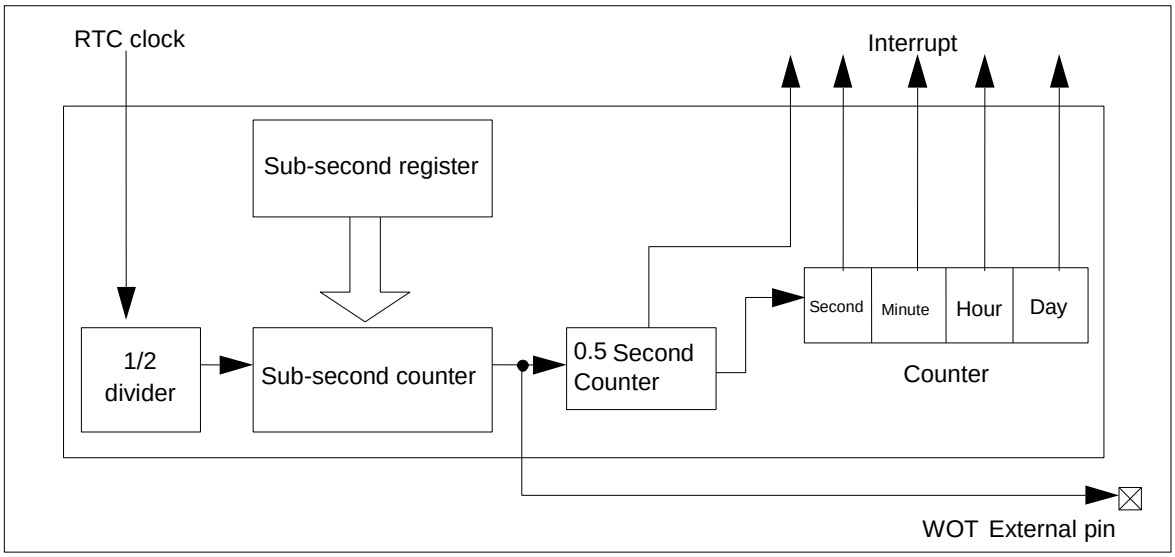
Code : FS57-1v0-91520-2-E

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 1-1 Block Diagram (Overview)



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 set and 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: 0.5second, 1second, 1minute, 1hour, and 1day. In addition, interrupts at any interval (from short interval to long interval) can be generated by changing the sub-second value.

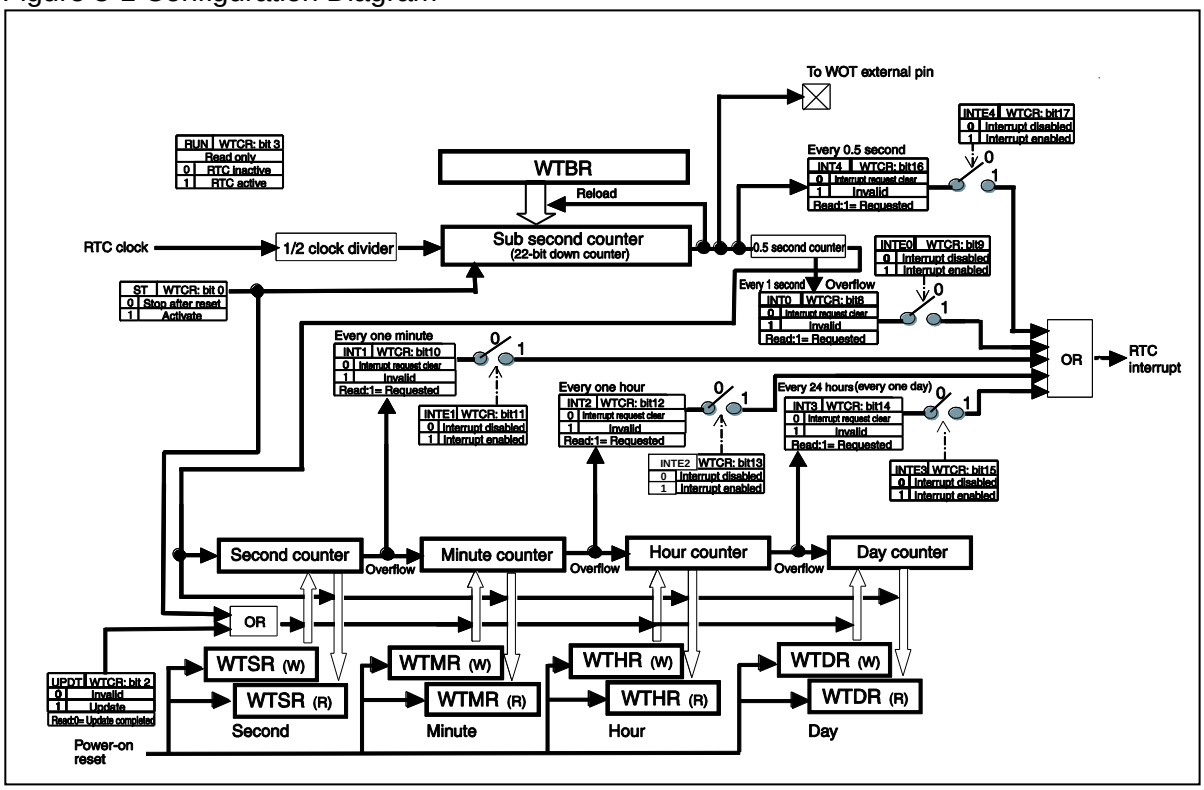
Note:

If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to '31', before CPU state changes to the watch mode with power-shutdown.

3. Configuration

This section explains the configuration of the real-time clock (RTC).

Figure 3-1 Configuration Diagram



4. Registers

This section explains registers of the real-time clock (RTC).

Table 4-1 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)

4.1. RTC Control Register : WTCR

The bit configuration of the RTC control register 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 factors 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 request 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.

Notes:

- 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.

4.2. Sub-second Register : WTBR

The bit configuration of the sub-second register is shown below.

This register contains the reload value of the sub-second counter (22-bit down counter).

■ **WTBRH : Address 0565_H (Access: Byte)**

■ **WTBRM : Address 0566_H (Access: Byte)**

■ **WTBRL : Address 0567_H (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, update the sub-second register 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 4-2 WTBR Setting Example

RTC clock frequency	WTBR Setting value
32kHz	0x001F3F
50kHz	0x0030D3
4MHz	0x0F423F

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 factor.

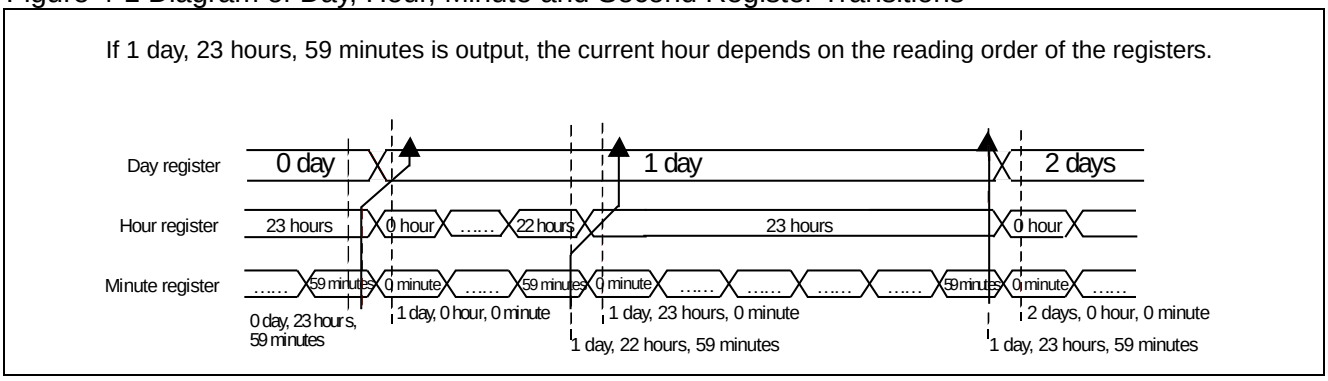
- The Second/ Minute/ Hour/ Day registers contain day and time information. Binary-coded notation is used for second, minute, hour, and day.
- When the register is read out, the counter value will be read out. The written data will be loaded to the counter after the UPDT bit is set to "1".
- As word 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.
  [Ex.] 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 4-1 Diagram of Day, Hour, Minute and Second Register Transitions



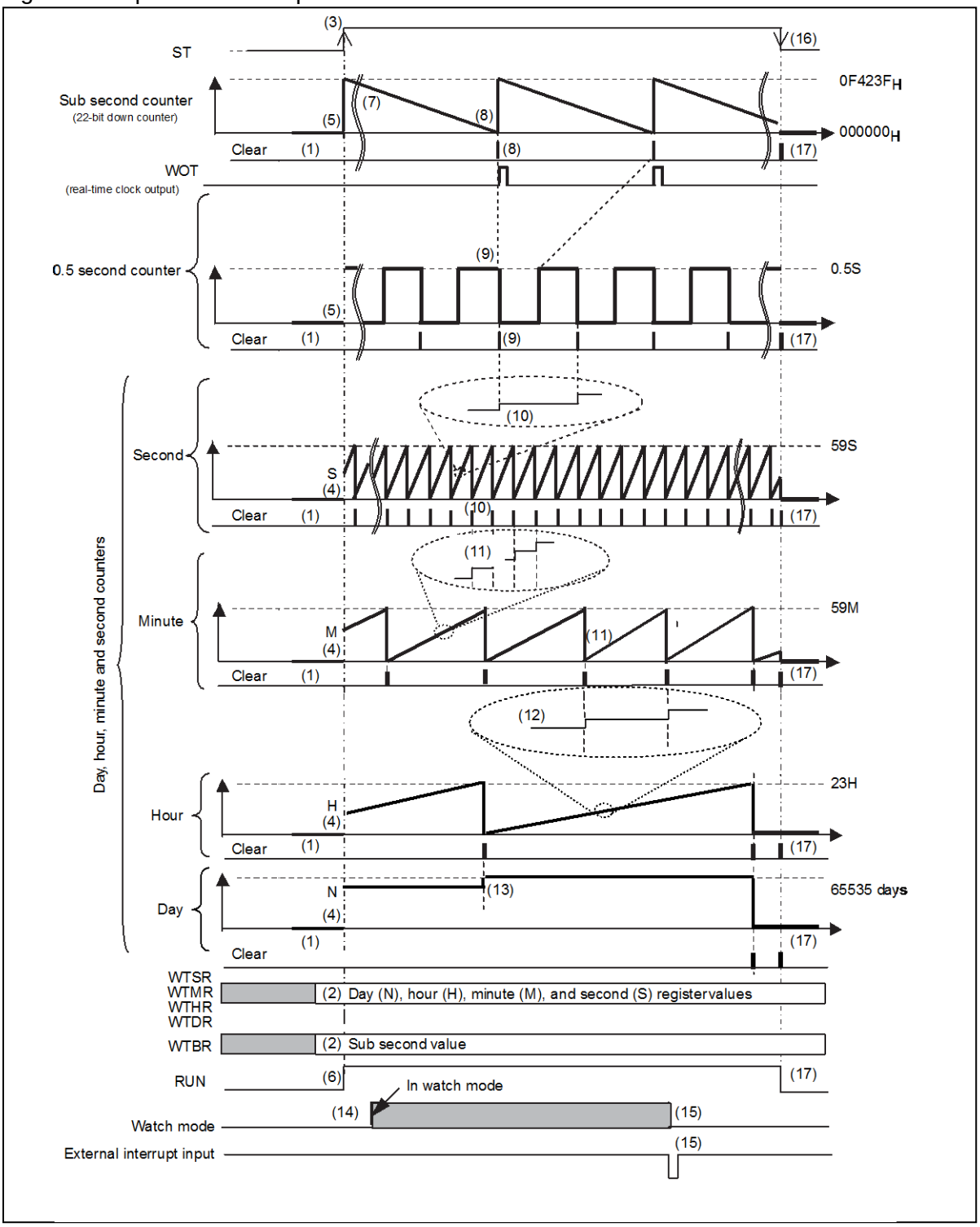
- When the operation clock frequency is obtained by dividing the frequency of the main clock by 2 (while PLL is stopped), the wrong values may be read out from the Hour/Minute/Second registers. This is caused due to synchronization adjustment between reading operations and count operations. Therefore, 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, clear the Day/Hour/Minute/Second registers.

5. Operation

This section explains the operation of the real-time clock (RTC).

This section explains the operations of the real-time clock.

Figure 5-1 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) -Write the values of Day/ Hour/ Minute/ Second to Day/ Hour/ Minute/ Second registers: WTDR, WTHR, WTMR, WTSR by software.
-Write "$0F_H$", "42_H", "$3F_H$" to sub-second registers: WTBRH, WTBRM, WTBRL by software.
-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 "000000_H", 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 (4/2MHz).
(8) When the sub-second counter(22-bit down counter) becomes "000000_H", load the sub-second register value "$0F423F_H$" 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 4MHz, "H" output with a width of 500ns)
(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.

6. Setting

This section explains setting of the real-time clock (RTC).

Table 6-1 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 7.1.
Initialization of the real-time clock	RTC Control Register (WTCR)	See 7.2.
Setting of number of days, time (Day/Hour/Minute/Second)	Day/ Hour/ Minute/ Second registers (WTDR, WTHR, WTMR, WTSR)	See 7.3.
Startup of the real-time clock	RTC Control Register (WTCR)	See 7.4.

Table 6-2 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 7.6.

Table 6-3 Settings Required for Stopping the Real-time Clock

Settings	Setting Registers	Setting procedure
Stop of the real-time clock	RTC Control Register (WTCR)	See 7.7.

Table 6-4 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 7.10.
RTC interrupt setting Interrupt request clear Interrupt request enable	RTC Control Register (WTCR)	See 7.11.

7. Q&A

This section explains Q&A of the real-time clock (RTC).

7.1 How to Set the 0.5 Second Count Interval

7.2 How to Initialize the Real-time Clock

7.3 How to Set/Update Number of Days (Day) and Time (Hour/Minute/Second)

7.4 How to Start/Stop the Count of the Real-time Clock

7.5 How to Confirm That the Real-time Clock Is Running

7.6 How to Know the Number of Days and Time

7.7 How to Stop the Real-time Clock

7.8 How to Calibrate the Real-time Clock

7.9 Interrupt Related Registers

7.10 Interrupt Types and How to Select Them

7.11 How to Enable Interrupts

7.1. How to 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 4-2 WTBR Setting Example to the sub-second register(WTBR) according to the RTC clock frequency.

7.2. How to 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.

7.3. How to 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"

7.4. How to 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"

7.5. How to 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

7.6. How to 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

7.7. How to Stop the Real-time Clock

This section explains how to stop the real-time clock.

See "7.4 How to Start/Stop the Count of the Real-time Clock" .

7.8. How to 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".

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)

Note:

If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.

7.10. Interrupt Types and How to Select Them

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.

7.11. How to 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"

8. Sample Program

This section explains the sample program of the real-time clock(RTC).

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*/
}
```

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, 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, 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.
- If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to '31', before CPU state changes to the watch mode with power-shutdown.
- [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] The internal reset is issued at the return from the standby watch mode (power-shutdown). Therefore, only the reset causes (power-on reset, internal low-voltage reset, and simultaneous assertion of RSTX and NMIX) are recognized. At this time, the register of the RTC is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after wake-up, the user needs to initialize the register of RTC before using it.
- [MB91F52xxxD] The internal reset is issued at the return from the standby watch mode (power-shutdown). Therefore, only the reset causes (power-on reset, internal low-voltage reset, and assertion of RSTX) are recognized. At this time, the register of the RTC is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after wake-up, the user needs to initialize the register of RTC before using it.

Chapter 29: RTC/WDT1 Calibration

This chapter explains the RTC/WDT1 calibration.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : RTCCAL-1v0-91520-2-E

1. Overview

This section gives an overview of the RTC/WDT1 calibration.

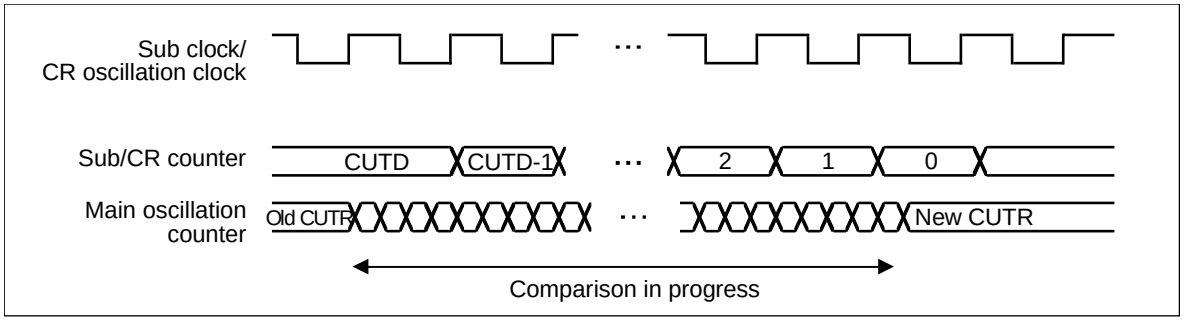
This module calculates the values to calibrate the real-time clock.

2. Features

This section explains features of the RTC/WDT1 calibration.

- **RTC Clock source select register**
  The main clock or the sub clock can be selected. For information on how to select, see "CHAPTER: CLOCK".
- **Real-Time Clock (RTC) Calibration (This function is effective only when the sub clock is used.)**
  Values of the sub second register of the RTC are determined by calculating the sub clock frequency from the main clock frequency on the condition that both the main clock driven counter and the sub clock driven counter operate concurrently (Figure 2-1).
- **WDT1(CR clock) calibration**
  This device has no CR clock calibration function. CR clock errors, however, can be measured by using the register of this module.

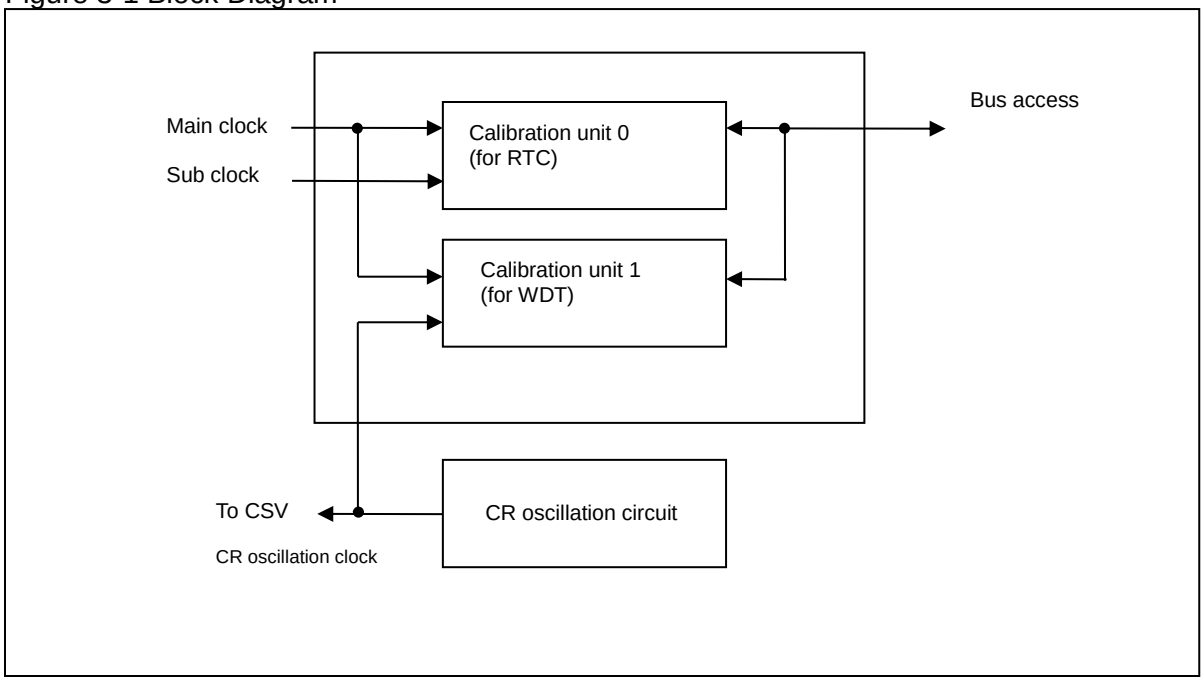
Figure 2-1 Comparison of Counters Driven by Different Clocks



3. Configuration

This section explains configuration of the RTC/WDT1 calibration.

Figure 3-1 Block Diagram



4. Registers

This section explains the registers of the RTC/WDT1 calibration.

Table 4-1 Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x4B8	CUCR0		CUTD0		Calibration unit control register 0 Sub clock timer data register
0x4BC	CUTR0				Main oscillation timer data register 0
0x04C0	Reserved	Reserved	Reserved	Reserved	Reserved
0x4C4	CUCR1		CUTD1		Calibration unit control register 1 CR oscillation timer data register
0x4C8	CUTR1				Main oscillation timer data register 1

4.1. Calibration Unit Control Register 0 : CUCR0 (Calibration Unit Control Register 0)

The bit configuration of the calibration unit control register 0 is shown.

This register configures calibration start and interrupts for RTC calibration unit.

■ CUCR0 : Address $04B8_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved		STRT	Reserved		INT	INTEN
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,WX	R0,WX	R,W	R0,WX	R0,WX	R,W	R/W

[bit15 to bit8] Reserved
The read value of these bits is always "0". Writing to these bits does not influence other functions.

[bit7] Reserved
Be sure to write "0" to this bit.

[bit6, bit5, bit3, bit2] Reserved
The read value of these bits is always "0". Writing to these bits does not influence other functions.

[bit4] STRT (calibration STaRT) : Calibration start
This bit starts counters driven by the main clock and the sub clock. The INT bit will be set after the comparison is completed.

STRT	Function
"0" write	Stops comparison
"1" write	Starts comparison

Setting "0" to this bit stops comparison. While comparing, writing "1" to this bit will not take effect. This bit will be cleared to "0" after the comparison is completed.

[bit1] INT (calibration INTerrupt) : Interrupt
The INT bit will be set to "1" after the comparison is completed. If the INTEN bit is set, an interrupt will occur. This bit is cleared by writing "0".

[bit0] INTEN (calibration INTerrupt ENable) : Interrupt enable
This bit sets whether to generate an interrupt when the INT bit is set.

INTEN	Interrupt
0	Disabled
1	Enabled

4.2. Sub Clock Timer Data Register : CUTD0 (Calibration Unit Timer Data register 0)

The bit configuration of the sub clock timer data register is shown.

This register configures a period of the time during which the sub clock driven counter operates.

■ 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 the number of the sub clock pulses.

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 is shown.

This register indicates the number of the main clock pulses counted within the time interval set by 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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	TDR[23:16]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDR[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

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

[bit31 to bit24] Reserved

The reading value of these bits is always "0". Writing to these bits does not influence other functions.

[bit23 to bit0] TDR[23:0] (Timer Data Register) : Timer data

These bits indicate the number of counts counted in the comparison interval. Read the results after the comparison is completed. The read value during comparison is undefined. Writing has no effect on operation.

4.4. Calibration Unit Control Register 1 : CUCR1 (Calibration Unit Control Register 1)

The bit configuration of the calibration unit control register 1 is shown.

This register configures calibration start and interrupts for the WDT calibration unit.

■ CUCR1 : Address $04C4_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved		STRT	Reserved		INT	INTEN
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,WX	R0,WX	R,W	R0,WX	R0,WX	R,W0	R/W

[bit15 to bit8] Reserved

The read value of these bits is always "0". Writing to these bits does not influence other functions.

[bit7] Reserved

Be sure to write "0" to this bit.

[bit6, bit5, bit3, bit2] Reserved

The read value of these bits is always "0". Writing to these bits does not influence other functions.

[bit4] STRT (calibration STaRT) : Calibration start

This bit starts counters driven by main clock and CR clock. The INT bit will be set after the comparison is completed.

STRT	Function
"0" write	Stops comparison
"1" write	Starts comparison

Setting "0" to this bit stops comparison. While comparing, writing "1" to this bit will not take effect. This bit will be cleared to "0" after the comparison is completed.

[bit1] INT (calibration INTerrupt) : Interrupt

The INT bit will be set to "1" after the comparison is completed. If the INTEN bit is set, an interrupt will occur. This bit is cleared by writing "0".

[bit0] INTEN (calibration INTerrupt Enable) : Interrupt enable

This bit sets whether to generate an interrupt or not when the INT bit is set.

INTEN	Interrupt
0	Disabled
1	Enabled

4.5. CR Clock Timer Data Register : CUTD1 (Calibration Unit Timer Data register 1)

The bit configuration of the CR clock timer data register is shown.

This register sets the time interval during which the CR clock driven counter operates.

■ CUTD1 : Address $04C6_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDD[15:8]							
Initial value	1	1	0	0	0	0	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TDD[7:0]							
Initial value	0	1	0	1	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TDD[15:0] (Timer Data Data field) : Timer data

These bits configure the comparison time interval in the number of the CR clock pulses.

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 is shown.

This register indicates the number of the main clock pulses counted within the time interval set by CUTD1.

■ **CUTR1 : Address $04C8_H$ (Access: Byte, Half-word, Word)**

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	TDR[23:16]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDR[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

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

[bit31 to bit24] Reserved

Always "0" is read. Writing to these bits does not influence other functions.

[bit23 to bit0] TDR[23:0] (Timer Data Register) : Timer data

These bits indicate the number of counts counted in the comparison interval. Read the results after the comparison is completed. The read value during comparison is undefined. Writing has no effect.

5. Operation

This section explains operation.

5.1 Real-Time Clock (RTC) Calibration

5.2 Measurement of Errors in CR Clock

5.3 Note

5.1. Real-Time Clock (RTC) Calibration

This section shows real-time clock (RTC) calibration.

The calibration procedure is as follows:

1. Setting CUTD0
2. Setting CUCR0.INTEN
3. Setting CUCR0.STRT
4. Loop waiting for interrupt
5. Occurrence of interrupt
6. Reading CUTR0
7. Comparison of CUTR0 and CUTD0 can be used to calculate the ratio between the main clock frequency and the sub clock frequency.
8. Setting values of the sub-second register in RTC using the value calculated at (7).

5.2. Measurement of Errors in CR Clock

This section shows measurement of errors in the CR clock.

The procedure for measuring errors in the CR clock is as follows:

1. Setting CUTD1
2. Setting CUCR1.INTEN
3. Setting CUCR1.STRT
4. Loop waiting for interrupt
5. Occurrence of interrupt
6. Reading CUTR1
7. Comparison of CUTR1 and CUTD1 can be used to calculate the ratio between the main clock frequency and the CR clock frequency.

5.3. Note

This section gives a note.

The counter value will become invalid in such a case that transition to standby mode occurs. Write "0" to the STRT bit to stop, and then write "1" again to redo.
TOSC32/OSC100 > 2 × TOSC4 + 3 × TCLKP needs to be satisfied.

TOSC4 : main clock cycle
TOSC32 : sub clock cycle
TOSC100 : oscillation cycle of CR oscillation circuit
TCLKP : peripheral clock oscillation cycle

Chapter 30: Power Consumption Control

This chapter explains the power consumption control.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Usage Example

Code : BZPMU-1v1-91520-2-E

1. Overview

This section gives an overview of the power consumption control.

This device has a variety of low-power consumption modes and can perform the power consumption control according to situations.

2. Features

This section explains features of the power consumption control.

- **Clock control**
  - Clock division
    By changing the division ratio for each running clock, the operating frequency can be lowered accordingly. See "CHAPTER: CLOCK".
- **Sleep mode**
  - CPU sleep mode
    In this mode, the only CPU core stops operating.
  - Bus sleep mode
    In this mode, both the CPU core and on-chip buses stop operating.
- **Standby mode**
  - Watch mode
    In this mode, all operations except some clock oscillations and the timer stop.
  - Stop mode
    In this mode, all clock oscillations and operations stop.
- **Standby mode with power-shutdown**
  - Watch mode with power-shutdown
    In this mode, the device is turned the power off. And all operations except some clock oscillations and the timer stop.
  - Stop mode with power-shutdown
    In this mode, the device is turned the power off and all clock oscillations and operations stop.

Notes:

In case of using the watch mode with power-shutdown, it is necessary to satisfy the below both conditions of (1) and (2).
(1)Interrupt levels that are used as sources for recovering from the watch mode with power-shutdown are ‘31’, before CPU state changes to the watch mode with power-shutdown.
(2) Don’t use NMIX pin as source for recovering from the watch mode with power-shutdown.

3. Configuration

This section shows the configuration of the power consumption control.

Figure 3-1 Block Diagram of Overall Control

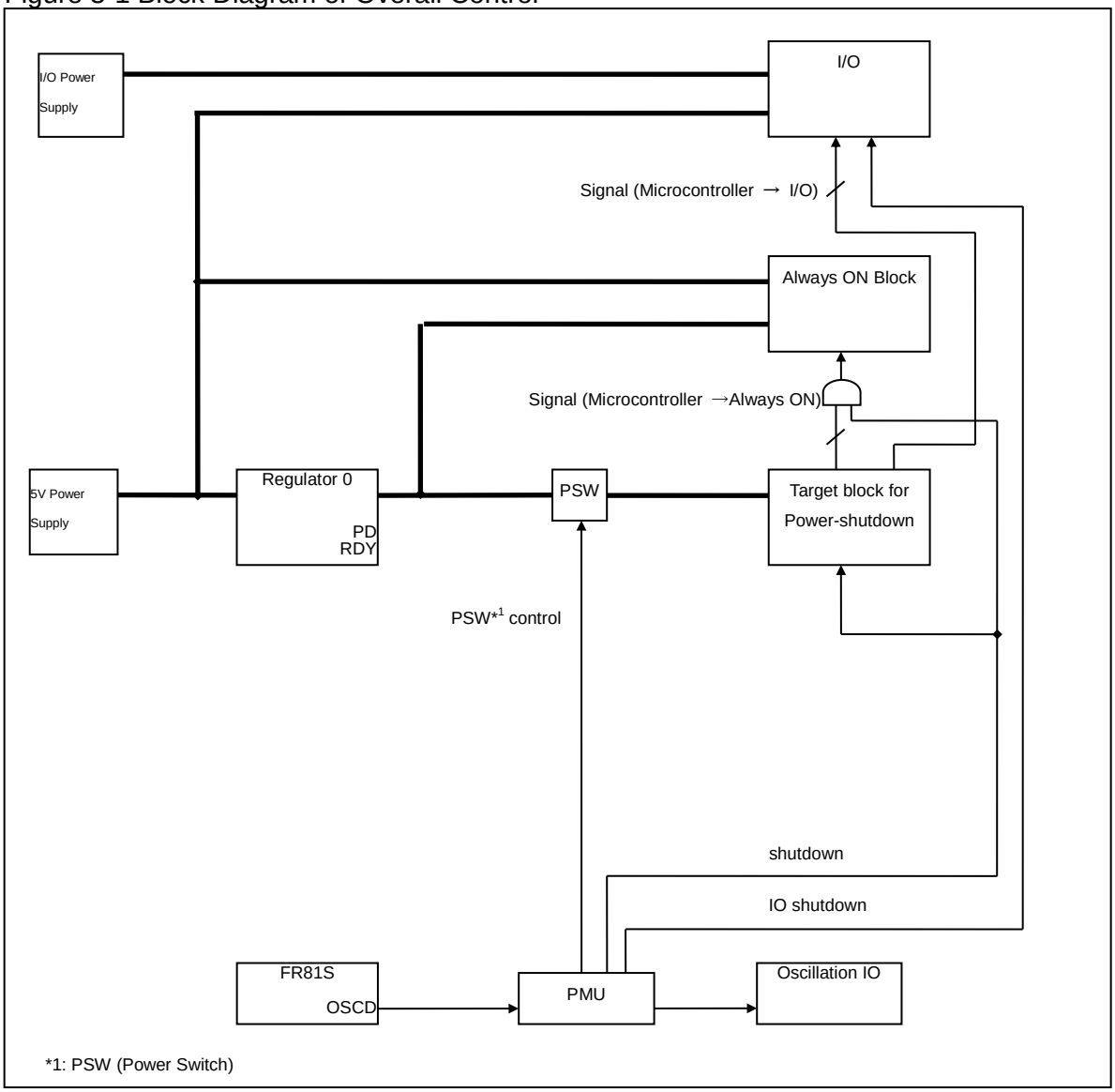
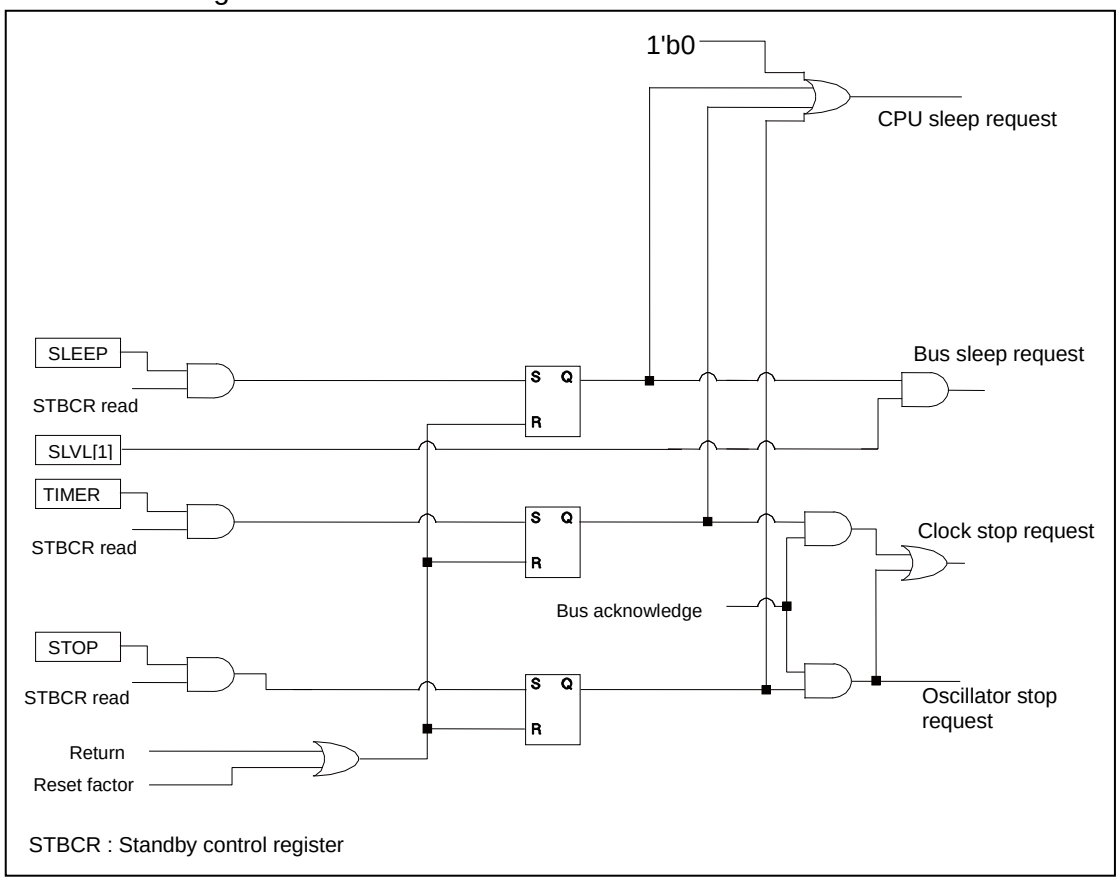


Figure 3-2 Block Diagram of Control



4. Registers

This section shows the registers of the power consumption control.

Table 4-1 Register 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

Note:

The addresses 0x0480 to 0x0481 and 0x0590 are allocated for the register "RESET". (See "CHAPTER: RESET".)

The group of registers (except STBCR) is initialized only in accordance with one or some of the following factors:

1. Power-on reset
2. Internal low-voltage detection
3. [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Simultaneous assertion of RSTX and NMIX external pins
   [MB91F52xxxD] Assertion of RSTX external pin
4. Hardware watchdog timer reset

* Registers are not initialized by reset of the INIT level and RST level. (except for STBCR)

4.1. Standby Control Register: STBCR (STandBy mode Control Register)

The bit configurations of the standby control register are shown below.

This register configures low-power consumption modes.

■ STBCR : Address 0482_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STOP	TIMER	SLEEP	Reserved	Reserved		SLVL[1:0]	
Initial value	0	0	0	0	0	0	1	1
Attribute	R,W	R,W	R,W	R0,W0	R0,W0	R0,W0	R/W	R/W

Note:

Writing to this register by DMA is prohibited.

[bit7] STOP (STOP mode): Stop mode enable

[bit6] TIMER (TIMER mode): Watch mode enable

[bit5] SLEEP (SLEEP mode): Sleep mode enable

Transitions to each standby mode (stop, watch, and sleep) are specified and enabled by these 3 bits. After writing the values shown below to these 3 bits and reading STBCR, the CPU goes into each standby mode.

STOP	TIMER	SLEEP	Enabled transition to each standby mode
0	0	0	No transition (initial value)
0	0	1	Transition to sleep mode by reading STBCR
0	1	X	Transition to watch mode by reading STBCR
1	X	X	Transition to stop mode by reading STBCR

The read value of each bit is as follows regardless of the writing value:

STOP	TIMER	SLEEP	Enabled transition to each standby mode
0	0	0	No transition
0	0	1	Transition to sleep mode
0	1	0	Transition to watch mode
1	0	0	Transition to stop mode

These bits are returned to their initial values by wake up factors arising from each low-power consumption mode.

[bit4] Reserved

The read value is always "0". Be sure to write "0" to this bit.

[bit3, bit2] Reserved

The read value is always "0". Be sure to write "0" to these bits.

[bit1, bit0] SLVL[1:0] (Standby LeVeL) : Standby level setting

These bits control the operations in standby mode and sleep mode as shown below.

Mode	SLVL[1:0]	Operation control
Stop mode	0x	Does not make pins high impedance.
	1x	Makes pins high impedance.
Watch mode	0x	Does not make pins high impedance.
	1x	Makes pins high impedance.
Sleep mode	0x	CPU sleep mode (stop only CPU)
	1x	Bus sleep mode (stop CPU and on-chip bus) *

* : On-chip bus will run only when DMA transfer is in progress.
For information on the pins with high impedance, see "APPENDIX".

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	BRAMSC	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R/W	R0,W0	R0,W0	R0,W0

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset.

[MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, 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". Be sure to write this bit 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	Requesting

This bit is cleared to "0" automatically after writing "1" to this bit and 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 does not affect operation.

[bit3]BRAMSC

BRAMSC	Backup RAM sleep control in standby mode
0	Backup RAM does not sleep in standby mode
1	Backup RAM sleeps in standby mode

[bit2 to bit0] Reserved

The read value is always "0". Be sure to write these bits to "0".

4.3. Power on Timing Control Register : PWRTMCTL (PoWeR on TiMing ConTroL register)

The bit configurations of the Power on Timing control register are shown below.

This register controls timing for power-on.

■ PWRTMCTL : Address 0592_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PTC		
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

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset.

[MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.

[bit7 to bit3] Reserved

The read value is always "0". Be sure to write these bits to "0".

[bit2 to bit0] PTC (Power on Timing Cycle setting)

These bits set the rise time for PSW.

PTC[2:0]	Rise time	Remarks (PMUCLK=32 kHz)
000	5×(1/PMUCLK)	150µs
001	11×(1/PMUCLK)	330µs
010	17×(1/PMUCLK)	510µs
011	29×(1/PMUCLK)	870µs
100	Prohibit	-
101	8×(1/PMUCLK)	240µs
110	14×(1/PMUCLK)	420µs
111	23×(1/PMUCLK)	690µs

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

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset.

[MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, 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.

EIFxx	External interrupt request
0	No request
1	Request

xx -> The number from 15 to 8 is assigned.

These registers are enabled only at shutdown.

These registers are cleared by writing "0". Writing "1" does not affect operation.

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 0595_H (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

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset.

[MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, 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.

EIFxx	External interrupt request
0	No request
1	Request

xx -> The number from 7 to 0 is assigned.

These registers are enabled only at shutdown.

These registers are cleared by writing "0". Writing "1" does not affect operation.

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 0596_H (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

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset.

[MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, 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 "0". Writing "1" does not affect operation.

[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 "0". Writing "1" does not affect operation.

[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 "0". Writing "1" does not affect operation.
The internal reset is issued at the return from the standby mode (power-shutdown) and the main timer interrupt flag is not set.

[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 "0". Writing "1" does not affect operation.
The internal reset is issued at the return from the standby mode (power-shutdown) and the sub timer interrupt flag is not set.

[bit3 to bit0] Reserved

The read value is always "0". Be sure to write these bits to "0".

5. Operation

Operations of the power consumption control are explained.

Features of the power consumption control of the device are explained in the following sections.

5.1 Clock Control

5.2 List of Clocks Supplied in Low-power Consumption Mode

5.3 Sleep Mode

5.4 Standby Mode : Watch Mode

5.5 Standby Mode : Watch Mode with Power-shutdown

5.6 Standby Mode : Stop Mode

5.7 Standby Mode : Stop Mode with Power-shutdown

5.8 Stop State of Microcontroller

5.9 Transition to Illegal Standby Mode

5.10 Restrictions on Power-Shutdown and Normal Standby Control

5.1. Clock Control

This section shows the clock control of the power consumption control.

By adjusting each operating clock of the device, its power consumption and processing capability can be optimized.

5.1.1. Division Setting

This section shows division setting of the clock.

See "CHAPTER: CLOCK".

5.1.2. Stopping of Unused Clocks

This section shows stopping of unused clocks.

The following clock can be independently stopped by settings:

· External bus clock (TCLK): Can be selected to supply/stop in bus sleep mode

For details of the setting, see "CHAPTER: CLOCK".

5.2. List of Clocks Supplied in Low-power Consumption Mode

The list of clocks supplied in low-power consumption mode is shown below.

Table 5-1 List of Clocks Supplied in Low-power Consumption Mode

Clock	Standby		Sleep	
	Stop	Watch	Bus	CPU
CPU clock (CCLK)	○	○	○	×
CAN Prescaler Clock	○	○	*1	×
On-chip bus clock (HCLK)	○	○	○	×
Peripheral clock (PCLK)	○	○	×	×
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 clock/sub clock/PLL clock are stopped by each clock setting register, supply of each clock stops, accordingly.)

*1: When on-chip bus clock (HCLK) is selected as CAN prescaler clock, this clock stops. When PLL

clock is selected, whether CAN prescalar stops or not depends on PLL output. Otherwise, CAN prescaler 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: In order to stop the CR oscillation in standby mode, a setting is needed in advance. See the description of CSVCR.RCE in "CHAPTER: CLOCK SUPERVISOR".

5.3. Sleep Mode

This section describes sleep mode.

Sleep mode is the mode in which CPU and on-chip bus are stopped and only the peripherals run. In sleep mode, there are the following modes according to the difference in the range of functional blocks to be stopped .

- CPU sleep mode : Only CPU is stopped.
- Bus sleep mode : Both CPU and on-chip bus are stopped.

The stop state continues until a wake up request occurs. It is possible to return to programmed operation within a few clock times by generating a wake up request.

Operation of each mode are explained in the following sections

5.3.1. CPU Sleep Mode

This section describes CPU sleep mode.

CPU sleep mode is the mode to stop the CPU operating.

In this mode, the DMA controller and on-chip bus can continue operating, but more power will be consumed than that in bus sleep mode.

5.3.2. Bus Sleep Mode

This section describes bus sleep mode.

Bus sleep mode is the mode to stop CPU and on-chip bus operations. In this mode, the CPU clock (CCLK) and on-chip bus clock (HCLK) will stop.

When accepting a DMA transfer request in bus sleep mode, on-chip bus clock (HCLK) supply resumes temporarily and performs DMA transfers. After the DMA transfer, 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.

5.3.3. Configuration of Sleep Mode

The configuration of sleep mode is described below.

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.

5.3.4. Activation of Sleep Mode

Activation of sleep mode is described below.

To activate sleep mode, follow the steps below.

- Write "001" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read STBCR

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering sleep mode.

[Example] Sample program of sleep mode activation

```
LDI     #value_of_sleep, R0     ; SLEEP bit ="1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of sleep mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

5.3.5. Wake Up from the Sleep Mode

Wake up from the sleep mode is described below.

The sleep mode is terminated under the following conditions:

- Reset
- Generation of interrupt request whose value of corresponding ICR register is value other than "0x1F"
- Generation of NMI request
- Generation of tool break while connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program starts from the instruction next to the instruction which activated the sleep mode.

In the bus sleep mode, the on-chip bus clock (HCLK) is temporarily returned by generating the DMA transfer request and DMA transfer is performed. After the DMA transfer is ended, the on-chip bus clock (HCLK) is stopped again.

5.3.6. Effect of Sleep Mode

Effect of sleep mode is described below.

You can reduce power consumption on the peripheral or external input event wait state drastically by using sleep mode. This mode does not decrease power consumption as much as that of in watch mode or stop mode because the peripheral clock (PCLK) will continue to run. While, a return to the program operation within several clock times is possible by generating a wake up request.

5.4. Standby Mode : Watch Mode

This section describes standby mode: watch mode.

Watch mode is the mode to continue oscillation only for the specific clock and count the clock timer corresponding to that clock. When the sub clock (SBCLK) is selected as the clock source, only the sub clock oscillates and only the sub timer counts.

Notes:

- Enter the device into the standby mode only when main RUN or sub RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "5.9 Transition to Illegal Standby Mode".
- Transition to the standby mode while the Flash memory is being programmed / erased is prohibited.

5.4.1. Configuration of Watch Mode

The configuration of watch mode is described below.

Before activating watch mode, set the state of external pins in watch mode with the bit1:SLVL1 in the STBCR register.

- When setting bit1:SLVL1="0" in the STBCR register, the external pins hold previous state.
- When setting bit1:SLVL1="1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. See "APPENDIX".

5.4.2. Activation of Watch Mode

Activation of watch mode is described below.

To activate watch mode, follow the steps below.

- "0" is written in bit7:SHDE of the PMUCTLR register.
- When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to watch mode.)
- Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering watch mode.

[Example] Sample program of watch mode activation

```
LDI     #value_of_timer, R0     ;TIMER bit ="1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of watch mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

5.4.3. Wake Up from the Watch Mode

Wake up from the watch mode is described below.

The watch mode is terminated under the following conditions:

- Reset

- Generation of interrupt request whose value of corresponding ICR register is value other than "0x1F" (see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)")
- Generation of NMI request
- Generation of tool break while connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program continues to run from the instruction next to the instruction which activated the watch mode.

5.4.4. Effect of Watch Mode

The effect of watch mode is described below.

You can reduce power consumption on the external input event wait state drastically by using watch mode. This mode does not decrease power consumption as much as that of in stop mode because enabled clock oscillation will continue to run. On the other hand, a clock timer can continue to run and a return to the program operation is possible by generating a wake up request in a short time compared with the return from the stop mode.*

* : When continue to run program with activate clocks.

5.5. Standby Mode : Watch Mode with Power-shutdown

This section describes 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 power supply to the microcontroller 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.

Notes:

- Enter the device into the standby mode only when main RUN or sub RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "5.9 Transition to Illegal Standby Mode".
- Transition to the standby mode while the Flash memory is being programmed / erased is prohibited.

5.5.1. Configuration of Watch Mode with Power-shutdown

The configuration of watch mode with power-shutdown is described below.

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".

(2) Interrupt levels that are used as sources for recovering from the watch mode with power-shutdown are 31', before CPU state changes to the watch mode with power-shutdown.

(3) Don't use NMIX pin as source for recovering from the watch mode with power-shutdown.

5.5.2. Activation of Watch Mode with Power-shutdown

Activation of watch mode with power-shutdown is described below.

To activate watch mode with power-shutdown, follow the steps below:

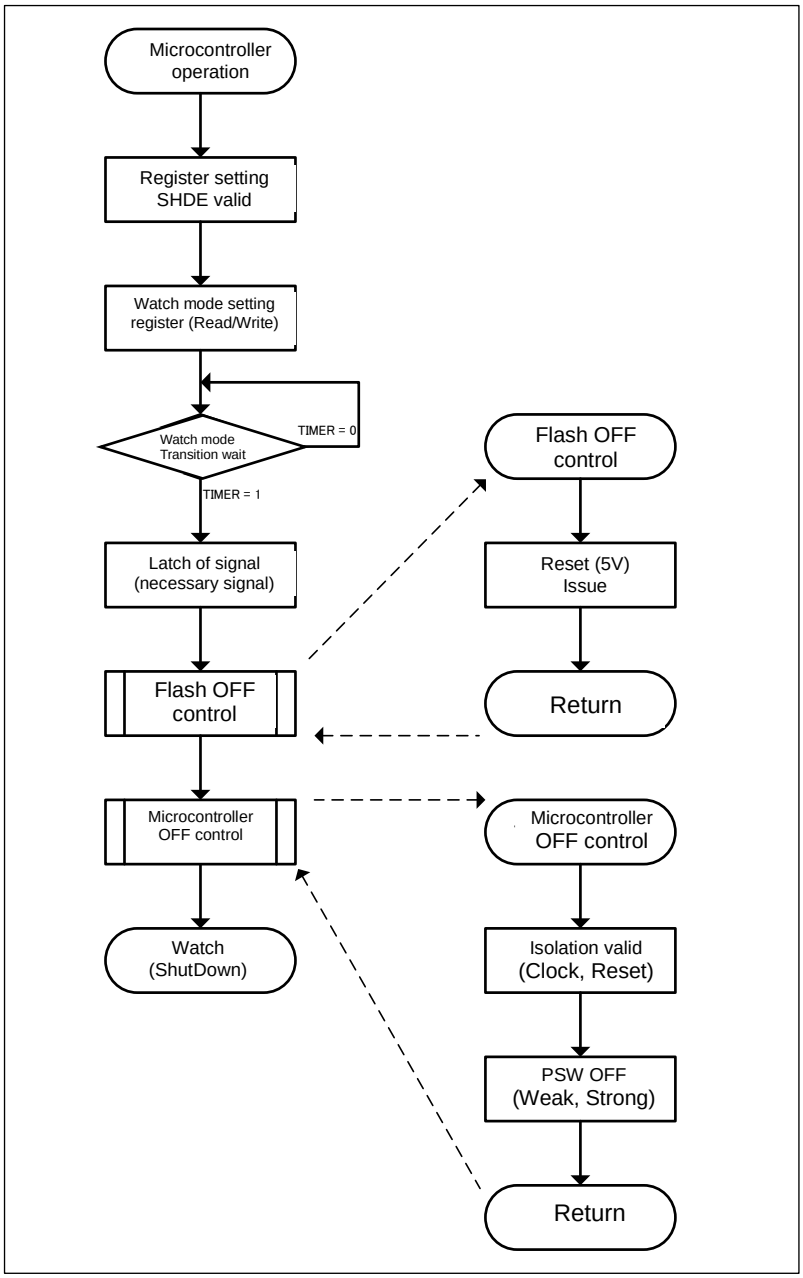
- "1" is written in bit7:SHDE of the PMUCTLR register.
- When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to watch mode with power-shutdown.)
- Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering watch mode 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 5-1 Transition Sequence to Watch Mode with Power-shutdown



5.5.3. Wake Up from the Watch Mode with Power-shutdown

Wake up from the watch mode with power-shutdown is described below.

The watch mode with power-shutdown is terminated under the following conditions:

- Reset
- Generation of external interrupt request
- Generation of RTC interrupt request
- Generation of main/sub timer interrupt request

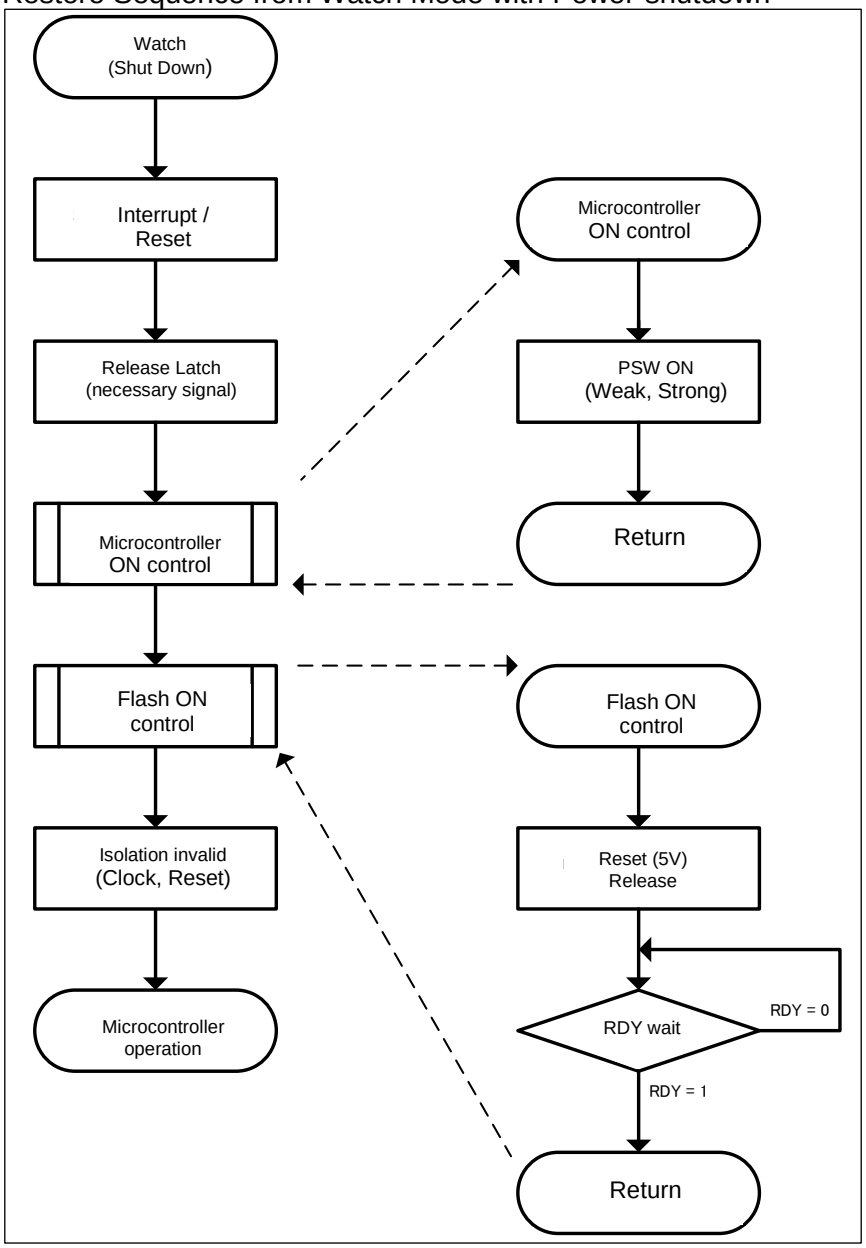
For the wake up caused by an interrupt request, the CPU and the interrupt controller do not necessarily have to be set so as to accept this interrupt request. The CPU always starts operation from the reset state.

The register of RTC and external interrupt input (IOCTMD=1) is not initialized.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Only the reset factors (power-on reset, internal low-voltage reset, and simultaneous assertion of RSTX and NMIX) are accepted during wake-up. At this time, the register of the RTC and external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the RTC/external interrupt input register before using it.

[MB91F52xxxD] Only the reset factors (power-on reset, internal low-voltage reset, and assertion of RSTX) are accepted during wake-up. At this time, the register of the RTC and external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the RTC/external interrupt input register before using it.

Figure 5-2 Restore Sequence from Watch Mode with Power-shutdown



5.5.4. Effect of Watch Mode with Power-shutdown

The effect of watch mode with power-shutdown is described below.

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.

5.6. Standby Mode : Stop Mode

This section describes standby mode: stop mode.

Stop mode is the mode to stop all clock oscillations and minimize power consumption of this device.

Notes:

- Enter the device into the standby mode only when main RUN or sub RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "5.9 Transition to Illegal Standby Mode".
- Transition to the standby mode while the Flash memory is being programmed / erased is prohibited.

5.6.1. Configuration of Stop Mode

The configuration of stop mode is described below.

Before activating stop mode, set the state of external pins in stop mode with the bit1:SLVL1 in the STBCR register.

- When setting bit1:SLVL1="0" in the STBCR register, the external pins hold previous state.
- When setting bit1:SLVL1="1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. See "APPENDIX".

5.6.2. Activation of Stop Mode

Activation of stop mode is described below.

To activate stop mode, follow the steps below.

- "0" is written in bit7:SHDE of the PMUCTLR register.
- When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to stop mode.)
- Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering stop mode.

[Example] Sample program of stop mode activation

```
LDI     #value_of_stop, R0      ; STOP bit ="1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of stop mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

5.6.3. Wake Up from the Stop Mode

Wake up from the stop mode is described below.

The stop mode is terminated under the following conditions:

- Reset
- Generation of interrupt request in which the value of corresponding ICR register is other than "0x1F" (see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)")
- Generation of NMI request
- Generation of tool break while being connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program continues to run from the instruction next to the instruction which activated the stop mode.

5.6.4. Effect of Stop Mode

The effect of stop mode is described below.

You can minimize power consumption on the external input event wait state by using stop mode. While, a return to the program operation after generating a wake up request needs the oscillation stabilization wait time.

5.7. Standby Mode : Stop Mode with Power-shutdown

This section describes 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 the device.

Notes:

- Enter the device into the standby mode only when main RUN or sub RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "5.9 Transition to Illegal Standby Mode".
- Transition to the standby mode while the Flash memory is being programmed / erased is prohibited.

5.7.1. Configuration of Stop Mode with Power-shutdown

The configuration of stop mode with power-shutdown is described below.

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".

5.7.2. Activation of Stop Mode with Power-shutdown

Activation of stop mode with power-shutdown is described below.

To activate stop mode with power-shutdown, follow the steps below:

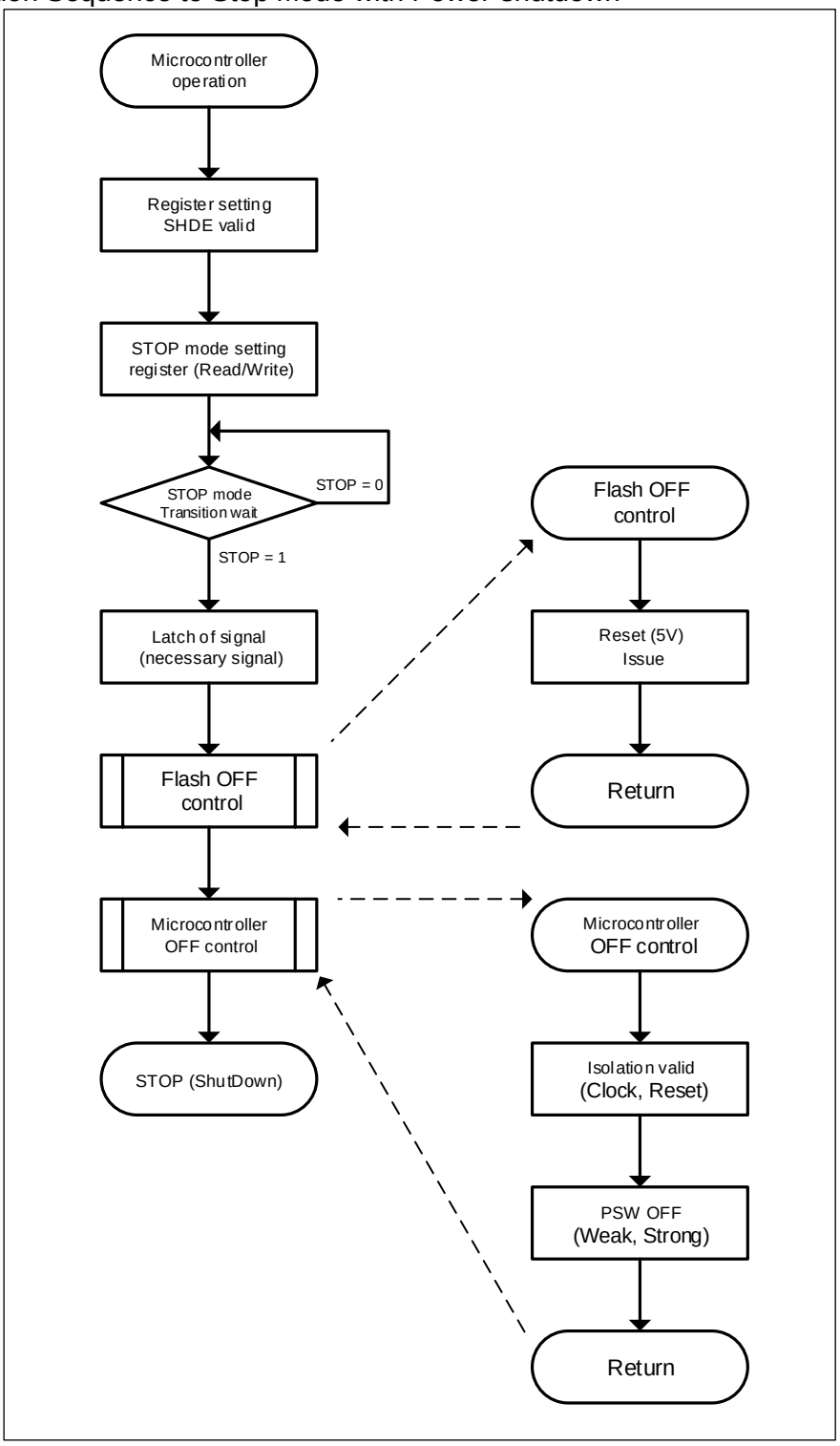
- "1" is written in bit7:SHDE of the PMUCTLR register.
- When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to stop mode with power-shutdown.)
- Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering stop mode 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 5-3 Transition Sequence to Stop Mode with Power-shutdown



5.7.3. Wake Up from the Stop Mode with Power-shutdown

Wake up from stop mode with power-shutdown is described below.

The stop mode with power-shutdown is terminated under the following conditions:

- Reset
- Generation of external interrupt request
- Generation of NMI request

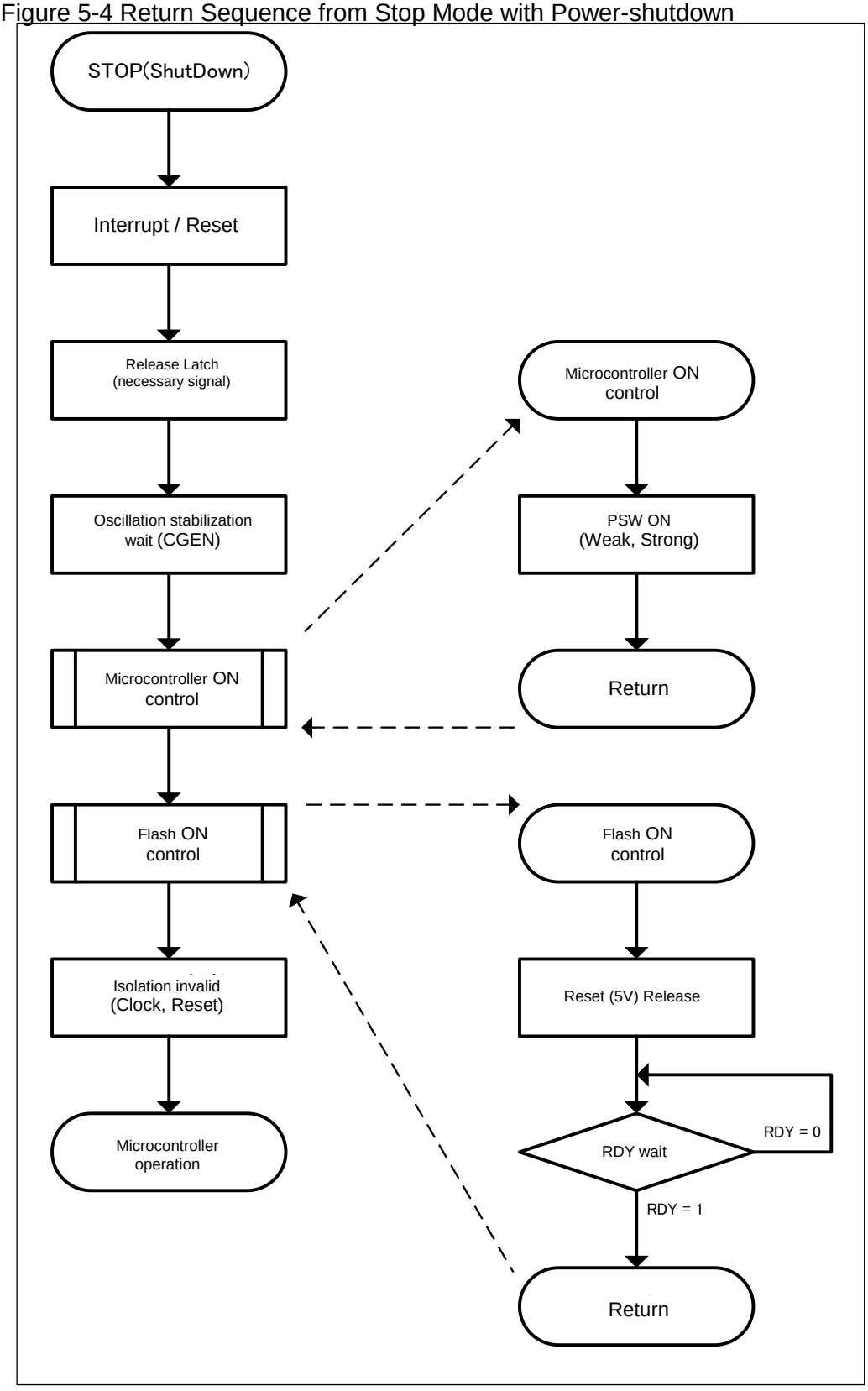
For the wake up caused by an interrupt request, the CPU and the interrupt controller do not necessarily have to be set so as to accept this interrupt request. The CPU always starts operation from the reset state.

The register of the external interrupt input (IOCTMD=1) is not initialized.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Only the reset factors (power-on reset, internal low-voltage reset and simultaneous assertion 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 flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the register before using it.

[MB91F52xxxD] Only the reset factors (power-on reset, internal low-voltage reset, and assertion of RSTX) are accepted during wake-up. At this time, the register of the external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the register before using it.

Figure 5-4 Return Sequence from Stop Mode with Power-shutdown



5.7.4. Effect of Stop Mode with Power-shutdown

The effect of stop mode with power-shutdown is described below.

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.

5.8. Stop State of Microcontroller

The stop state of the microcontroller is described below.

When the transition from the state of the standby mode (watch mode/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 >

1. Connecting OCD
2. Operating PLL

<Standby control not done by microcontroller stop condition>

3. Flash memory power saving control
4. Oscillation stop (At the stop mode or 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 "5.9. Transition to Illegal Standby Mode" for the illegal standby mode transition.

5.9. Transition to Illegal Standby Mode

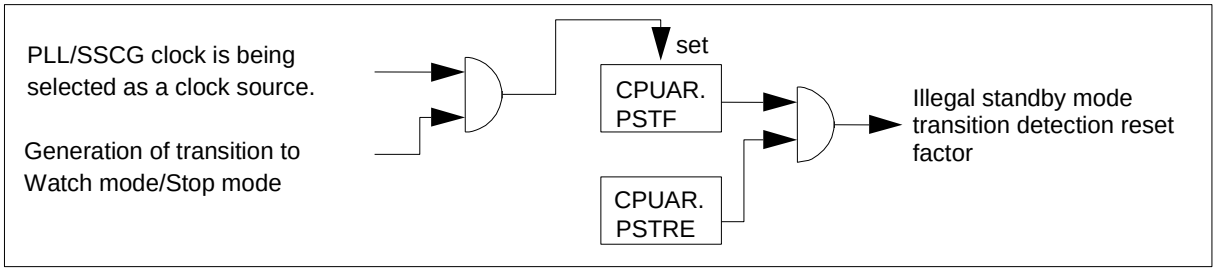
Transition to illegal standby mode is described below.

If the transition from PLL run state to standby mode (watch mode/watch mode with power-shutdown/stop mode/stop mode with power shutdown) is made, standby mode is set and PLL oscillation stabilization is canceled. (Transition to illegal standby mode)

After returning from standby mode, CSELR.CKS[1:0]=00 and CMONR.CKM[1:0]=00 (divide-by-two output of the main clock).

The PSTF flag of the CPUAR register is set concurrently with the transition to standby mode. When the PSTRE bit in the CPUAR register is set, reset occurs by illegal standby mode transition detection reset factor. For the CPUAR register, see " CPU Abnormal Operation Register: CPUAR (CPU Abnormal operation Register) " in "CHAPTER: RESET".

Figure 5-5 Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor



5.10. Restrictions on Power-Shutdown and Normal Standby Control

Restrictions on power-shutdown and normal standby control are described below.

The microcontroller does not perform standby control under the following conditions:

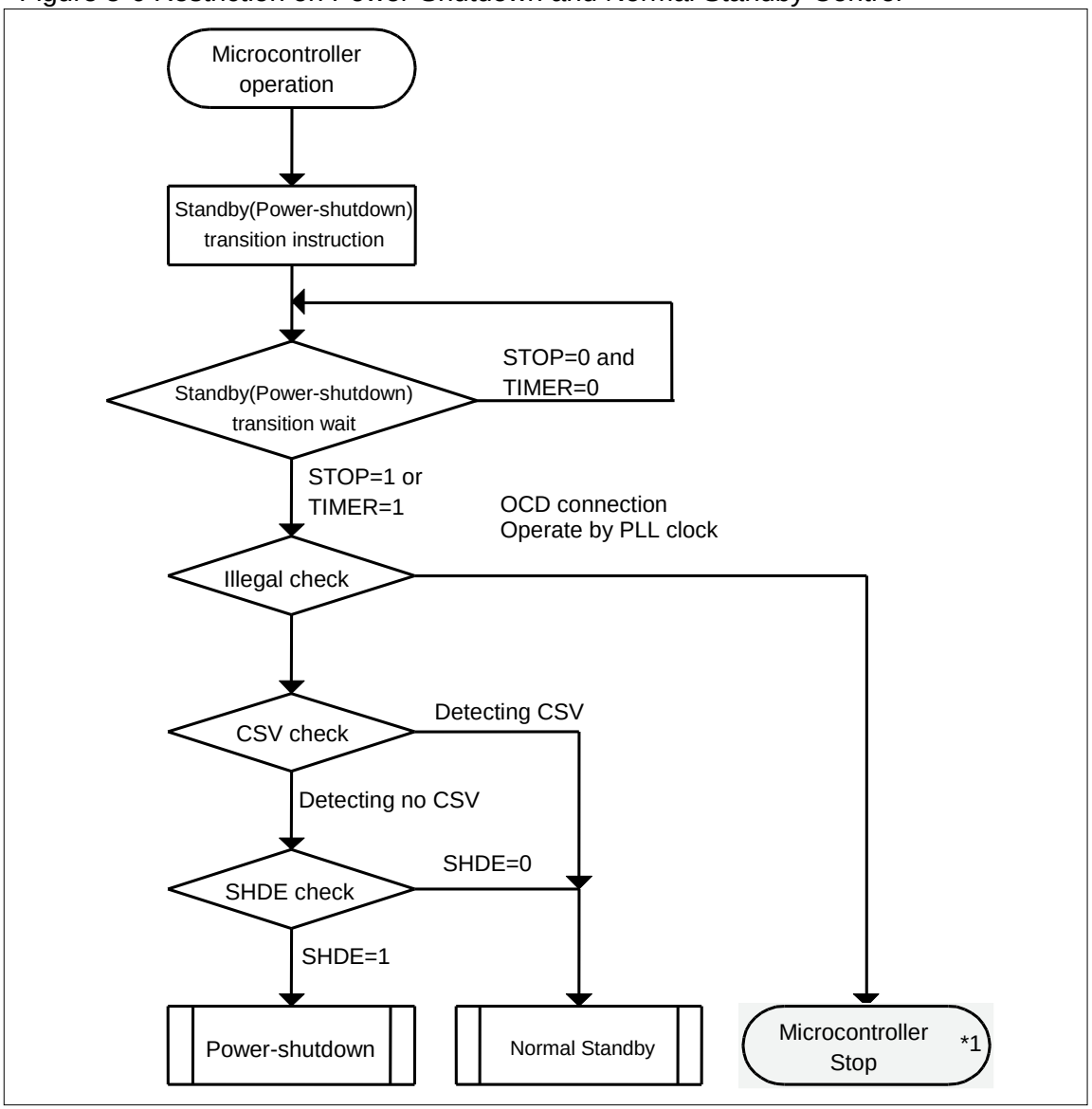
- When the CPU is operating with PLL
- When the OCD is being enabled to operate

The standby control does not operate in the state above, but the CPU is in the standby state. (See 5.8.)

- When missing the clock by CSV function *2, *3.

When standby (power-shutdown) or transition is directed in the state above, it usually becomes standby processing.

Figure 5-6 Restriction on Power-Shutdown and Normal Standby Control



*1: This state is not recognized as power-shutdown and the state that the 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 the CPU operates with the CR clock after main clock stop are detected, the CPU does not perform standby control. However, it is not the limitation case, when sub-clock stop is detected while the CPU run with the main clock.

*3: When standby or power-shutdown transition is directed after the operating clock source is missing, it usually becomes standby processing. Also, note that the CSV function stops if you enable power-shutdown permission in a state in which clock stop is not detected.

Only a part of registers is maintained at returning, because power is not being supplied to almost all blocks inside in standby mode with power-shutdown. Table 5-2 shows the list of registers that are stored at return from standby mode with power-shutdown.

Table 5-2 List of Registers 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	Register	0594_H	
	PMUINTF1	Register	0595_H	
	PMUINTF2	Register	0596_H	
Reset factor register	CPUAR.PMDF	Flag	$051A_H$ bit2	
	CPUAR.PSTF	Flag	$051A_H$ bit1	
	CPUAR.HWDF	Flag	$051A_H$ bit0	
Low-voltage detection register (External low-voltage detection)	LVD5R.LVD5R_F	Flag	0584_H bit0	
	LVD5F.LVD5F_F	Flag	0585_H bit0	
	LVD5F.LVD5F_PD	Register	0585_H bit7	
	LVD5F.LVD5F_OE	Register	0585_H bit3	
Low-voltage detection register (Internal low-voltage detection)	LVD.LVD_F	Flag	0586_H bit0	
	LVD.LVD_PD	Register	0586_H bit7	
	LVD.LVD_OE	Register	0586_H bit3	
CSV register	CSVCR	Register	$056D_H$	
External interrupt register	EIRR0/1	Register	$0550_H/0554_H$	*2
	ENIR0/1	Register	$0551_H/0555_H$	*2
	ELVR0/1	Register	$0552_H/0556_H$	*2
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	Flag	0510_H bit7	*1
	CMONR.SCRDY	Flag	0511_H bit7	*1
	CCRTSELR.CST	Flag	0530_H bit7	*1
	CCRTSELR.CSC	Flag	0530_H bit0	*1

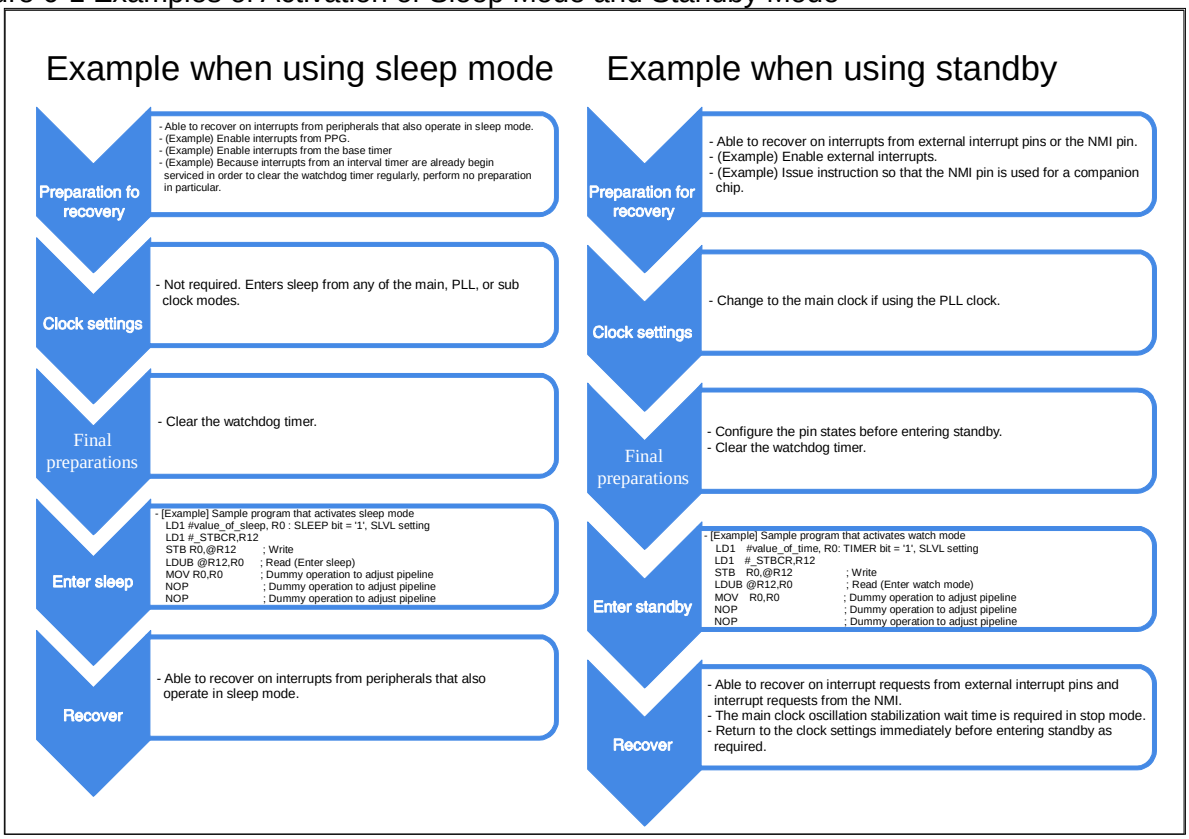
*1: These registers are initialized at return from stop mode with power-shutdown.
*2: These registers are initialized at PMUCTLR.IOCTMD=0.

6. Usage Example

Power consumption control usage examples are shown below.

These are examples of activation of sleep mode and standby mode.

Figure 6-1 Examples of Activation of Sleep Mode and Standby Mode



Chapter 31: Low-Voltage Detection (Internal Low-Voltage Detection)

This chapter explains the low-voltage detection (internal low-voltage detection).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

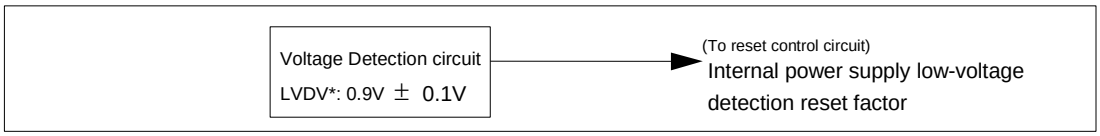
Code : BZLVDR_LVDI-2v2-91520-2-E

1. Overview

This section gives an overview of the low-voltage detection (internal low-voltage detection).

The internal low-voltage detection is the function that monitors an internal power supply voltage and detects a fall of the power supply voltage below the low-voltage detection voltage level. When the internal low-voltage below the detection voltage level is detected, a detection flag is set and the device goes to the reset state by the low-voltage detection reset.

Figure 1-1 Block Diagram (Overview)



*: The detection voltage of the internal low voltage detection is 0.9V±0.1V. This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage.

2. Features

This section explains features of the low-voltage detection (internal low-voltage detection).

The internal low-voltage detection circuit

Function: Generates a reset signal to initialize settings if a voltage LVDV ± 0.1V or less is detected. (LVDV : 0.9 V)

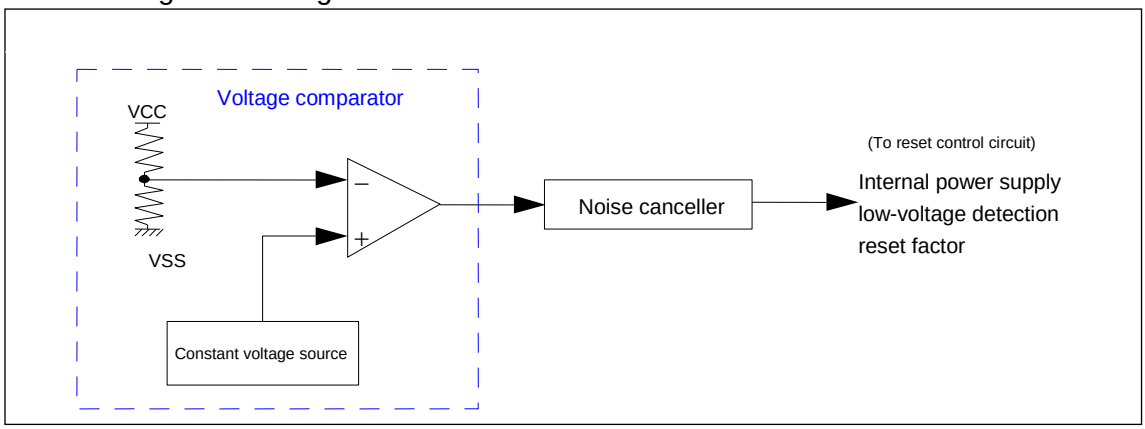
- Number of units: 1
- Operation: Continues to operate in sleep mode, stop mode, and watch mode.

Voltage comparator: Compares the internal power supply voltage to the detection voltage level, and changes output from "H" to "L" if a low-voltage is detected.
After the power is turned on the voltage comparator operates constantly.

3. Configuration

This section shows the configuration of the low-voltage detection (internal low-voltage detection).

Figure 3-1 Configuration Diagram



4. Registers

This section shows the registers of the low-voltage detection (internal low-voltage detection).

Table 4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0584	LVD5R	LVD5F	LVD	Reserved	Internal low-voltage detection register

4.1. Internal Low-Voltage Detection Register : LVD (Low-Voltage Detect internal power fall register)

The bit configuration of the internal low-voltage detection register is shown.

This register has the internal low-voltage detection flag (LVD_F) and the control bit.

■ LVD : Address 0586$_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LVD_PD	LVD_SEL[2:0]			LVD_OE	Reserved		LVD_F
Initial value	0	1	0	0	0	0	0	0
Attribute	R/W	R/W1	R/W0	R/W0	R/W	R0,WX	R0,WX	R(RM1), W

[bit7] LVD_PD (Low-Voltage Detect fall Power Down)

This bit sets whether a fall of the internal power supply voltage in the microcontroller should be detected or not.

LVD_PD	Setting for detection of internal power supply voltage fall power down in the microcontroller
0	Disabled (Detection is executed.)
1	Enabled (Detection is stopped.)

(Note): This bit is initialized by only power-on reset.

Note:

Set detection enable (OE = 0) after 100 μs, if this bit sets the status of power-down enable to disable (operation start). If set it before 100 μs, some detection flag setting will be occur.

[bit6 to bit4] LVD_SEL[2:0] (Low-Voltage Detect power fall SELect)

These bits select the detection level of a fall of the internal power supply voltage.

LVD_SEL[2:0]	Setting for detection level of fall of internal power supply voltage	Guaranteed MCU operation voltage range
100 *	0.9V ± 0.1V	No
Other than those above	Setting is prohibited	-

(Note): These bits can be rewritten only when LVD_OE="1".

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage.

[bit3] LVD_OE (Low-Voltage Detect power fall Output Enable)

This bit is the output enable signal for internal voltage fall detection.

LVD_OE	Internal voltage fall detection output enable setting
0	Enable
1	Disable

(Note): This bit is initialized by only power-on reset.

[bit2, bit1] Reserved

[bit0] LVD_F (Low-Voltage Detect power fall Flag)

This bit indicates an internal power supply voltage fall detection flag.

LVD_F	Internal power supply fall detection flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a fall in the internal power supply voltage is detected, the LVD_F bit is set to "1".

It will be initialized only when the external reset is input.

5. Operation

This section explains operations of the low-voltage detection (internal low-voltage detection).

5.1 Internal Low-voltage Detection

5.1. Internal Low-voltage Detection

The internal low-voltage detection is explained.

The internal low-voltage detection is the function that monitors an internal power supply voltage and detects it falling below the detection voltage level. When the internal low-voltage below the detection level is detected, a detection flag is set and a reset signal to initialize setting is generated.

If the internal voltage falls below the detection voltage level, it takes the oscillation stabilization wait time after the internal voltage is recovered. For details, see "CHAPTER: RESET".

Oscillation stabilization wait time	2^{15} × Main clock cycle

6. Notes

This section provides notes on the low-voltage detection (internal low-voltage detection).

● Operation of internal low-voltage detection

If the internal power supply voltage falls and the internal low-voltage detection flag in the microcontroller is set (LVD:LVD_F="1"), internal reset is generated by the function of low-voltage detection reset. Thus, writing and reading of the internal low-voltage detection register (LVD) in the microcontroller is not allowed.

The internal low-voltage detection circuit can operate even though the device is in its sleep mode, stop mode, and watch mode, consuming a certain amount of current.

The internal low-voltage detection circuit can be set to operate/stop by a user.

● Initial value of internal low-voltage detection flag (LVD:LVD_F)

The internal low-voltage detection flag is cleared by external reset or by writing "0" to the LVD_F bit of the internal low-voltage detection register (LVD).

● Oscillation stabilization wait time

If the internal voltage falls below the detection voltage level, it takes the oscillation stabilization wait time after the internal voltage recovers. For details, see "CHAPTER: RESET".

● Hysteresis of detection/reset release voltage

Since the detection voltage and reset release voltage exhibit hysteresis of 0.1V, the reset release voltage becomes the set detection voltage + 0.1V. For example, when LVD: 0.9V ± 0.1V is set, the reset release voltage becomes 1.0V ± 0.1V.

Chapter 32: Low-Voltage Detection (External Low-Voltage Detection)

This chapter explains the low-voltage detection (external low-voltage detection).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

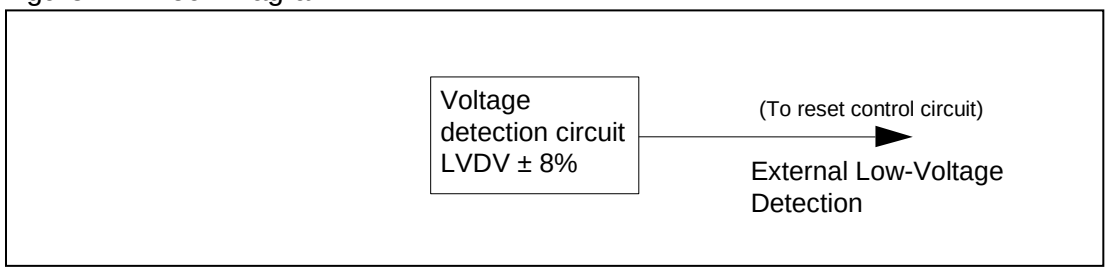
Code : BZLVDR_LVDE-2v2-91520-2-E

1. Overview

This section gives an overview of the low-voltage detection (external low-voltage detection).

The external low-voltage detection is the function that monitors external voltage and detects a fall of the power supply voltage below the low-voltage detection voltage level.

Figure 1-1 Block Diagram



(Note): Rising LVDV: 2.3V
Falling LVDV: 2.8 to 4.3V (11 steps) variable*

*: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage (2.7V).

2. Features

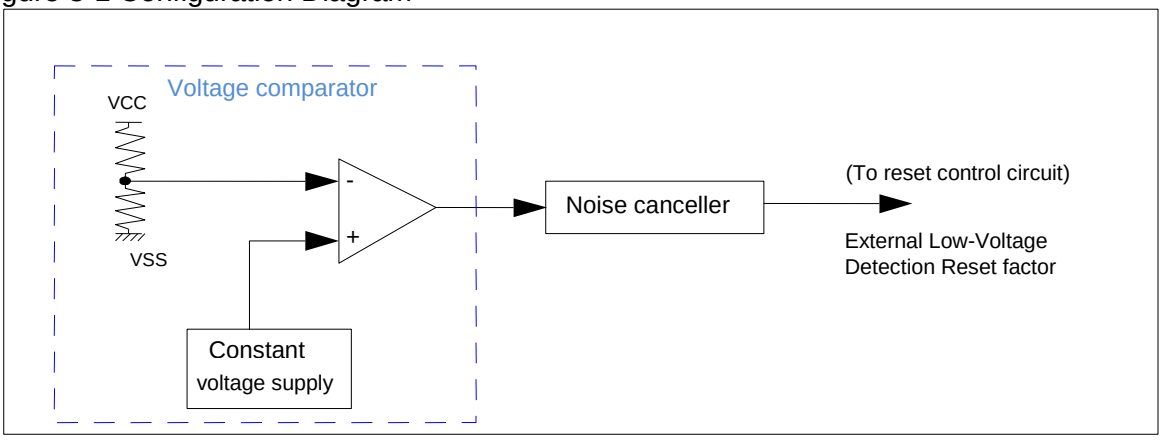
This section explains features of the low-voltage detection (external low-voltage detection).

- Function : Generates the reset signal to initialize settings if the voltage LVDV ±8% or less is detected.
  (Rising: LVDV: 2.3V (fixed), falling LVDV: 2.8 to 4.3V (variable))
- Number of units : One
- Operation : Switches operation/stop by user's settings.
  During writes to the internal RAM, the low-voltage reset occurs after the write has finished.
- Voltage comparator :Compares the detection voltage and the power supply voltage, outputting "L" if low-voltage is detected.
- Either to apply a reset or to generate an interrupt, when a low-voltage is detected, can be selected.

3. Configuration

This section explains the configuration of the low-voltage detection (external low-voltage detection).

Figure 3-1 Configuration Diagram



4. Registers

This section explains the registers of the low-voltage detection (external low-voltage detection).

Table 4-1 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

4.1. External Low-Voltage Detection Rise Detection Register : LVD5R (Low-Voltage Detect external 5v Rise register)

The bit configuration of the external low-voltage detection rise detection register (LVD5R) is shown.

This register is the external power supply voltage rise detection flag.

■ LVD5R : Address 0584_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							LVD5R_F
Initial value	0	0	0	0	0	0	0	1
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1), W

[bit7 to bit1] Reserved

[bit0] LVD5R_F (Low-Voltage Detect external 5v Rise Flag): External voltage rise detection flag
This bit is an external voltage rise detection flag.

LVD5R_F	External power supply rise detection flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a power-on reset is detected, the LVD5R_F bit is set to "1".
The bit will be cleared when external reset is input.

4.2. External Low-Voltage Detection Fall Detection Register : LVD5F (Low-Voltage Detect external 5v Fall register)

The bit configuration of the external low-voltage detection fall detection register (LVD5F) is shown.

This register is used in order to clear the low-voltage detection reset flag and set the low-voltage detection circuit.

■ LVD5F : Address 0585_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LVD5F_PD	LVD5F_SEL[3:1]			LVD5F_OE	LVD5F_SEL[0]	LVD5F_RI	LVD5F_F
Initial value	1/0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R(RM1),W

Note:
The initial value of bit7 is different by device.

[bit7] LVD5F_PD (Low-Voltage Detect external 5v Fall Power Down): External power supply fall power down setting

This bit is used in order to set whether to detect a fall in external voltage or not.

LVD5F_PD	External power supply fall power down setting
0	Invalid (Performs detection) (Initial value "ON". Device option.)
1	Valid (Stops detection) (Initial value "OFF". Device option.)

(Note): This bit is initialized by only power-on reset.

Note:

When setting this bit from power down enable to disable (operation start), set to detection enable (LVD5F_OE=0) 100µs after setting LVD5F_OE=1. If set it before 100 µs, some detection flag setting will be occur.

The initial state of external low-voltage detection is different by device. Therefore, the initial value of this bit is different by device. For details of device, see "3. Product Line-up" in "CHAPTER: OVERVIEW".

[bit6 to bit4, bit2] LVD5F_SEL (Low-Voltage Detect external 5v Fall SELect): External fall detection voltage setting

These bits are the selection signal for a detection level of external voltage fall detection.

LVD5F_SEL[3:0]	External power supply fall detection voltage setting	Guaranteed MCU operation voltage range
0000 *	2.80V ±8%	No
0001	3.00V ±8%	Yes
0010	3.20V ±8%	
0011	3.60V ±8%	
0100	3.70V ±8%	
0101	3.80V ±8%	
0110	3.90V ±8%	
0111	4.00V ±8%	
1000	4.10V ±8%	
1001	4.20V ±8%	
1010	4.30V ±8%	
others	Setting prohibited	-

(Note): LVD5F_SEL[3:0] bits can be rewritten only when LVD5F_OE = "1".

*: This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level (2.8V±8% = 2.576V ~ 3.024V) is below the minimum guaranteed MCU operation voltage (2.7V).

[bit3] LVD5F_OE (Low-Voltage Detect external 5v Fall Output Enable): External power-supply fall detection output enable setting

This bit is the output enable signal for external voltage fall detection.

LVD5F_OE	External power supply fall detection output enable setting
0	Enable
1	Stop

(Note): This bit is initialized by only power-on reset.

[bit1] LVD5F_RI (Low-Voltage Detect external 5v Fall Reset Interrupt select):

This bit selects either low-voltage detection reset or interrupt.

LVD5F_RI	Low-voltage detection reset / Interrupt selection setting
0	Reset
1	Interrupt

[bit0] LVD5F_F (Low-Voltage Detect external 5v Fall Flag): External fall detection flag

This bit is an external voltage fall detection flag.

LVD5F_F	External voltage fall detection flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a fall in external voltage is detected, the LVD5F_F bit is set to "1".

This bit is cleared when an external reset is input.

5. Operation

This section explains operation of the low-voltage detection (external low-voltage detection).

The external low-voltage detection monitors the external voltage and generates an initialization reset or interrupt if the external voltage drops below the configured value.

Those values of this register cannot be guaranteed if a low-voltage is detected and a settings initialization reset occurs. After the low-voltage reset is released, the reset sequence will be executed without the oscillation stabilization wait time, and then the program is restarted from the address specified by the reset vector.

6. Notes

This section provides notes on the low-voltage detection (external low-voltage detection).

Notes on using the low-voltage detection reset circuit

● Operation by program

- The low-voltage detection reset circuit operates in accordance with settings, except for the external low-voltage detection rise detection 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 the stop mode is cleared.

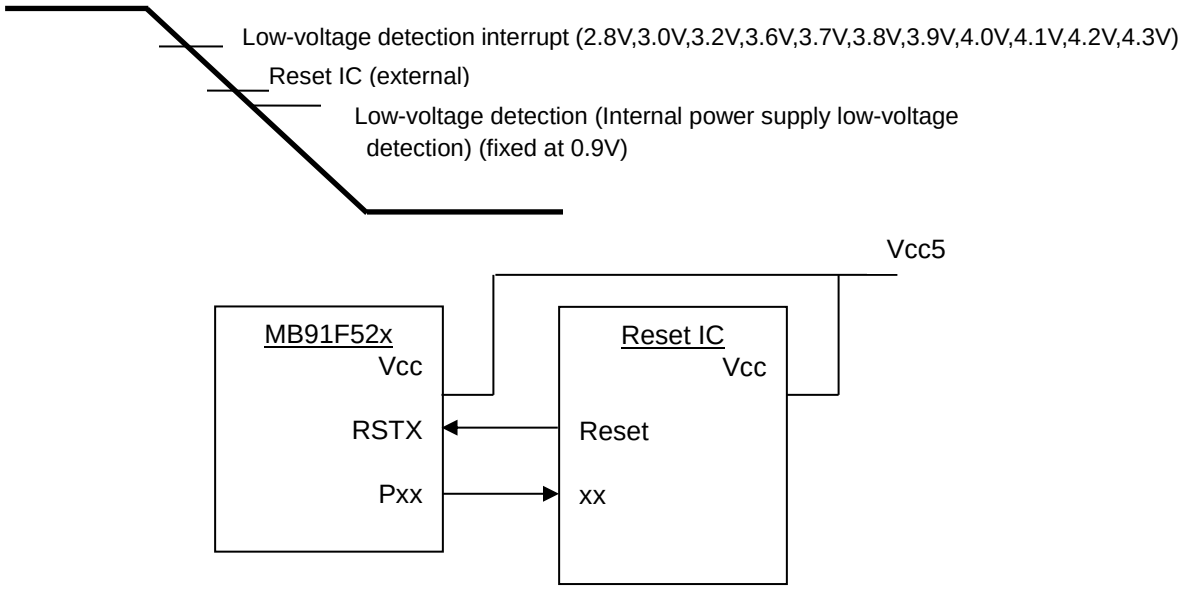
● Hysteresis of detection/reset release voltage

- Since the detection voltage and reset voltage exhibit hysteresis of 0.1V, the reset release voltage becomes the set detection voltage + 0.1V.

  For fall detection power supply voltage, the set detection voltage indicates the detection voltage.
  For example, when 4.1V±8% is set, the release voltage becomes 4.2V±8%.
  For rise detection power supply voltage, the set detection voltage indicates the reset release voltage.
  For example, when 2.5V ± 8% is set, the detection voltage becomes 2.4V ± 8%.

● Be sure to connect an external reset IC if an interrupt is generated when low-voltage is detected.

In addition, be sure to set voltage of the reset request signal 2.7V or more at which operation of the CPU is assured.



Chapter 33: Wild Register

This chapter explains the wild register.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Usage Example

Code : FR81S10_WR-1v1-91520-2-E

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.

2. Features

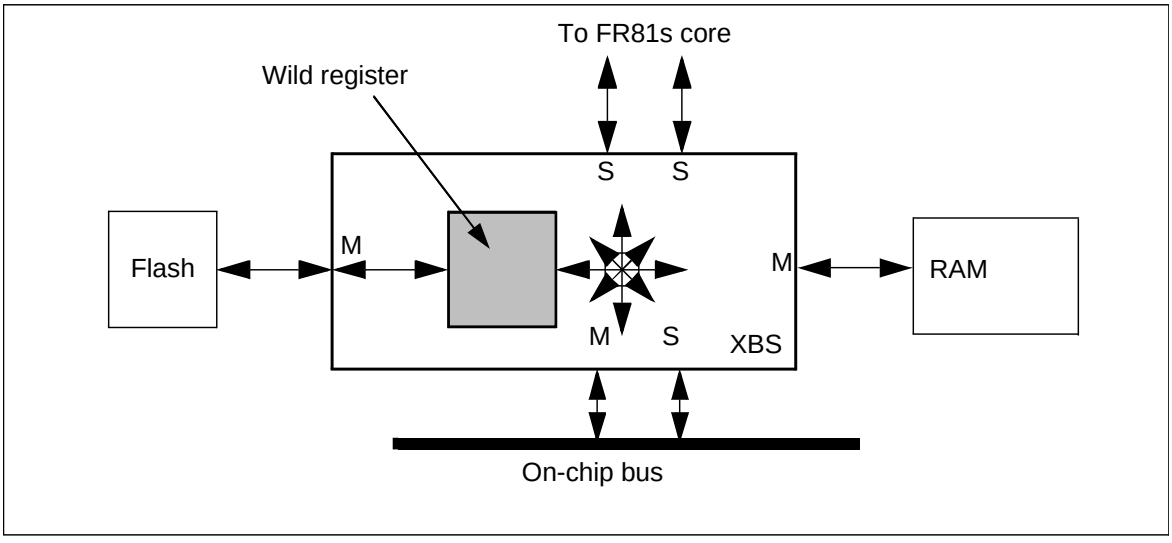
This section explains features of the wild register.

- 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

3. Configuration

This section explains the configuration of the wild register.

Figure 3-1 Configuration Diagram



Note:

When the access wait to the Flash memory is set to one cycle, this function cannot be used.

4. Registers

This section explains registers of the wild register.

Table 4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0858	Reserved		WREN		Wild register data enabled register
0x0880	WRAR00				Wild register address register 00
0x0884	WRDR00				Wild register data register 00
0x0888	WRAR01				Wild register address register 01
0x088C	WRDR01				Wild register data register 01
0x0890	WRAR02				Wild register address register 02
0x0894	WRDR02				Wild register data register 02
0x0898	WRAR03				Wild register address register 03
0x089C	WRDR03				Wild register data register 03
0x08A0	WRAR04				Wild register address register 04
0x08A4	WRDR04				Wild register data register 04
0x08A8	WRAR05				Wild register address register 05
0x08AC	WRDR05				Wild register data register 05
0x08B0	WRAR06				Wild register address register 06
0x08B4	WRDR06				Wild register data register 06
0x08B8	WRAR07				Wild register address register 07
0x08BC	WRDR07				Wild register data register 07
0x08C0	WRAR08				Wild register address register 08
0x08C4	WRDR08				Wild register data register 08
0x08C8	WRAR09				Wild register address register 09
0x08CC	WRDR09				Wild register data register 09
0x08D0	WRAR10				Wild register address register 10

Address	Registers				Register function
	+0	+1	+2	+3	
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

4.1. Wild Register Data Enable Register : WREN (Wild Register ENable register)

The bit configuration of the wild register data enable register is shown.

This register sets whether the wild register function is enabled or disabled on each channel.

■ WREN : Address 085A$_H$ (Access: Half-word)

	bit15	bit14	•	•	•	bit2	bit1	bit0
	WREN[15:0]							
Initial value	0	0	•	•	•	0	0	0
Attribute	R/W	R/W	•	•	•	R/W	R/W	R/W

[bit15 to bit0] WREN[15:0] (Wild Register ENable) : Enable bits

These bits set whether the wild register function is enabled or disabled on each channel.

WRENn (n = 0 to 15)	Function
0	Disables the wild register function of channel n
1	Enables the wild register function of channel n

4.2. Wild Register Address Register 00 to 15 : WRAR00 to 15 (Wild Register Address Register 00 to 15)

The bit configuration of wild register address register 00 to 15 is shown.

These registers set the address to be amended by the wild register function. The read value is undefined when the wild register operation is enabled.

Always set these registers in units of 32 bits.

■ WRAR : Address 0880_H to $08F8_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		WRAR[21:16]					
Initial value	0	0	X	X	X	X	X	X
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	WRAR[15:8]							
Initial value	X	X	X	X	X	X	X	X
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WRAR[7:2]						Reserved	
Initial value	X	X	X	X	X	X	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R0,WX

[bit21 to bit2] WRAR[21:2] (Wild Register Address Register) : Address register

These bits set the address to patch. The target address is (WRAR & 0x003FFFFC).

The read value is undefined when the wild register operation is enabled.

4.3. Wild Register Data Register 00 to 15 : WRDR00 to 15 (Wild Register Data Register 00 to 15)

The bit configuration of wild register data register 00 to 15 is shown.

These registers set the replacement data. When the contents of the memory at the addresses specified by the wild register address registers (WRAR00 to WRAR15) are read, the value set in these registers is returned instead of the actual contents of the memory.

The read value of these registers is undefined while the wild register function is operating.

Always set these registers in units of 32 bits.

■ WRDR : Address 0884_H to $08FC_H$ (Access: Word)

	bit31	bit30	•	•	•	bit2	bit1	bit0
	WRDR[31:0]							
Initial value	X	X	•	•	•	X	X	X
Attribute	R/W	R/W	•	•	•	R/W	R/W	R/W

[bit31 to bit0] WRDR[31:0] (Wild Register Data Register) : Data register

These bits set the replacement value.

The read value of these registers is undefined while the wild register function is operating.

5. Operation

This section explains the operation of the wild register.

This function is used to patch the Flash area. Because the enable register is initialized by reset, this register needs to be set on each reset when being used.

Addresses need to be set so that they do not overlap each other. When addresses overlap, the read value is undefined.

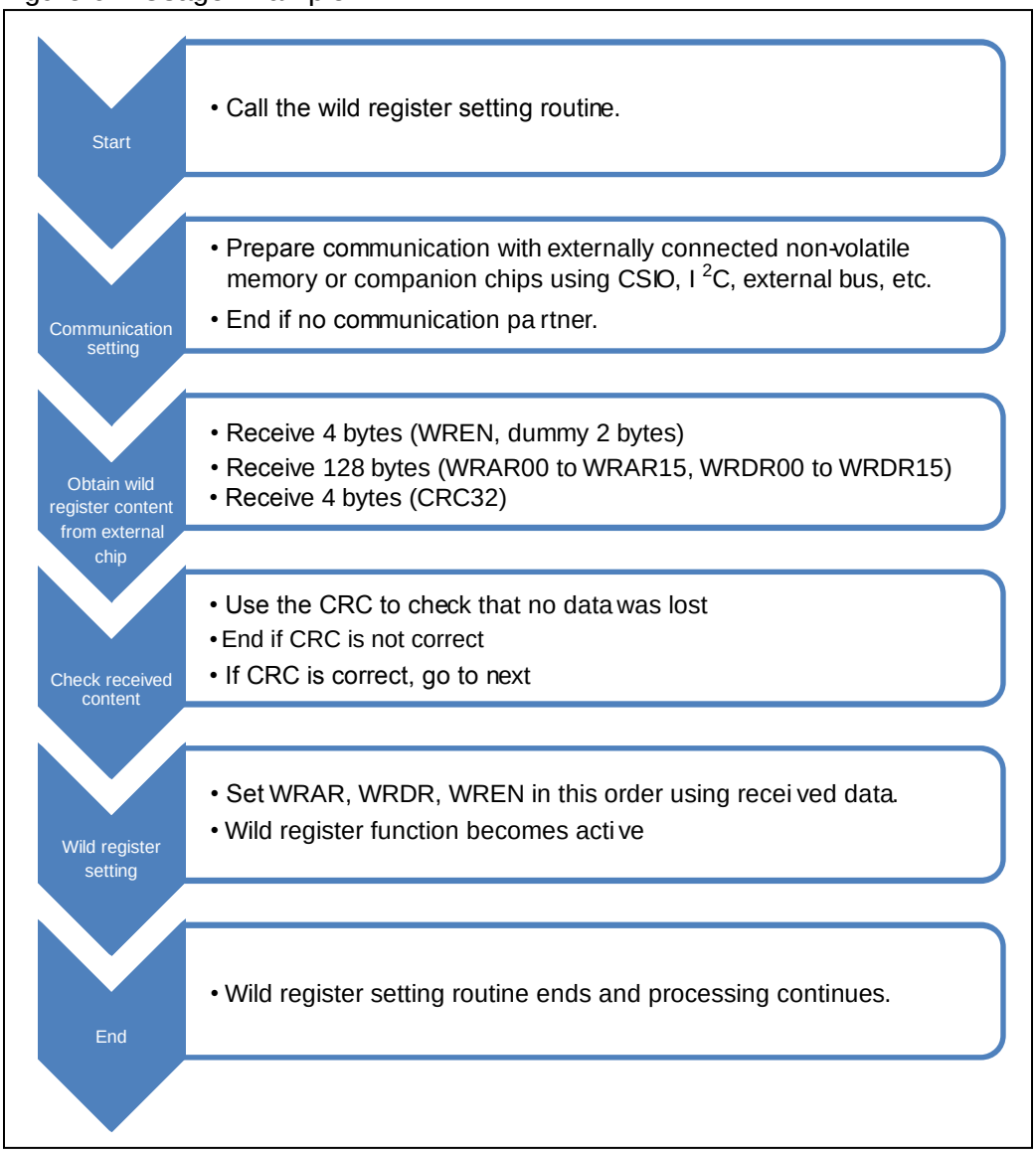
The data's byte line is the big endian.
The target area to replace is the Flash area only.

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 6-1 Usage Example



Chapter 34: Clock Supervisor

This chapter explains the clock supervisor.

1. Overview
2. Configuration
3. Register
4. Operation

Code : FJ58-1v1-91520-2-E

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 supervisor for the sub clock is independent of the supervisor for the main. The clock supervisor can be enabled, and disabled separately.

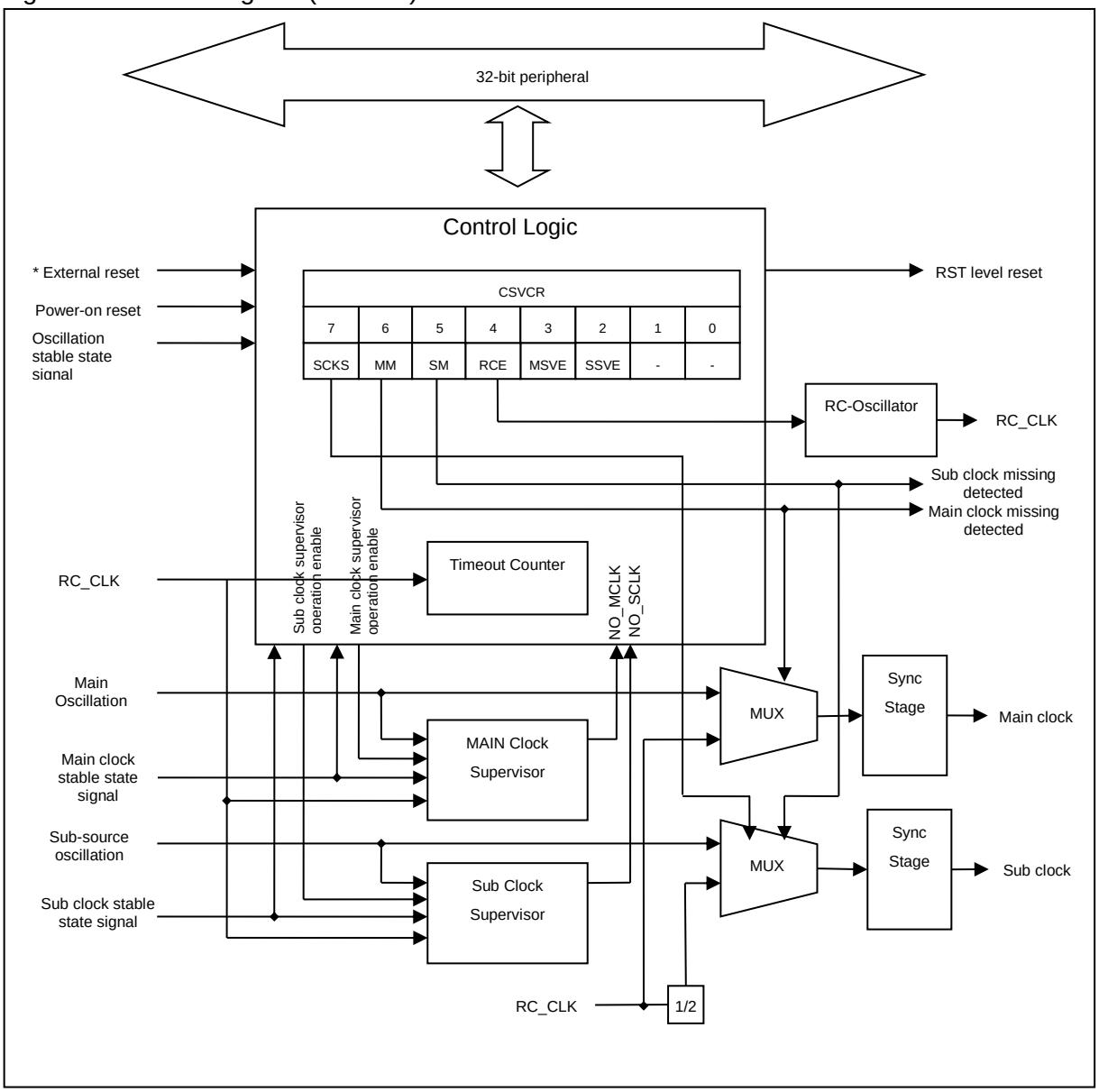
2. Configuration

This section shows the configuration of the clock supervisor.

The blocks that configure the clock supervisor are shown below.

- Clock supervisor
- Timeout counter
- Control logic
- CR oscillator

Figure 2-1 Block Diagram (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.

3. Register

This section explains a register of the clock supervisor.

Table 3-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x056C	Reserved	CSVCR	Reserved	Reserved	Clock supervisor control register

3.1. Clock Supervisor Control Register : CSVCR(Clock SuperVisor Control Register)

The bit configuration of the clock supervisor control register (CSVCR) is explained.

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
	SCKS	MM	SM	RCE	MSVE	SSVE	Reserved	Reserved
Initial value	0	0	0	1	1/0	1	0	0
Attribute	R/W	R,W	R,W	R/W	R/W	R/W	R0/W0	R0/W0

Note: The initial value of bit3 is different by device.

[bit7] SCKS (Sub Clock mode Select) : Selecting sub clock mode

Be sure to set this bit to "1" when sub clock mode is used with the single clock product.
The sub clock mode, same as the dual clock product, originates from divide-by-two output of the CR.
While the sub clock is being selected as a source clock (CSELR:CKS=11), writing "0" is ignored.
For dual clock product, this bit cannot be used. Be sure to write "0" to this bit.
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit will be cleared to "0" on power-on, external reset, or simultaneous assertion with NMIX. Other kind of reset does not affect this bit.
[MB91F52xxxD] This bit will be cleared to "0" on power-on, or external reset. Other kind of reset does not affect this bit.

SCKS	Description
0	Sub clock mode with CR clock as a source clock is disabled. (Initial value)
1	Sub clock mode with CR clock as a source clock is enabled.

[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 do not affect this bit.

MM	Read	Write
0	Main oscillation clock stop undetected	When the main clock is restored oscillating, this bit can be cleared
1	Main oscillation clock stop detected	No effect

Note:

Do not enable the PLL/SSCG oscillation operation when this bit is "1".

[bit5] 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.
This bit will be invalid when the single clock product is set to operate in the sub clock mode (SCKS=1, CSELR:SCEN=1).

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 (RC-oscillator Enable) : CR oscillator Enable

The oscillation of the CR oscillator is permitted at the standby mode when this bit is set to "1". Setting 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 does not affect this bit.
This bit will be invalid when the single clock product is set to operate in the sub clock mode (SCKS=1, CSELR:SCEN=1).

RCE	Description
0	CR oscillation disabled at STBY mode
1	CR oscillation enabled at STBY mode (Initial value)

[bit3] MSVE (Main clock SuperVisor Enable) : Main clock supervisor enable

When this bit is set to "1", the main clock supervisor is enabled.
This bit is initialized to "1" only when the power is turned on.
Other types of resets do not affect this bit.

MSVE	Description
0	Main clock supervisor disabled (Initial value "OFF". Device option.)
1	Main clock supervisor enabled (Initial value "ON". Device option.)

(Note) The initial state of main clock supervisor is different by device. Therefore, the initial value of this bit is different by device. For details of device, see "3. Product Line-up" in "CHAPTER: OVERVIEW".

[bit2] SSVE (Sub clock SuperVisor Enable) : Sub clock supervisor enable

When this bit is set to "1", the sub clock supervisor is enabled.
This bit is initialized to "1" only when the power is turned on.
Other types of resets do not affect this bit.
This bit will be invalid when the single clock product is set to operate in the sub clock mode (SCKS=1, CSELR:SCEN=1).

SSVE	Description
0	Sub clock supervisor disabled
1	Sub clock supervisor enabled (Initial value)

[bit1] Reserved

"0" should be written to this bit.

[bit0] Reserved

"0" should be written to this bit.

4. Operation

This section explains the operation of the clock supervisor.

After the clock replaces the CR oscillator, it is reset at once when the main clock stops while CPU is operating with the main clock. When the period of 20µs to 80µs and the clock is not input, it is judged that it stops. Because the bit indicating that the main clock has stopped remains in the register, it is possible to judge that a problem has occurred with the software.

After the clock replaces the CR oscillator, it is reset at once when sub clock stops while CPU is operating with sub clock. When the period of 206µs to 640µs and the clock is not input, it is judged that it stops. Because the bit indicating that the sub clock has stopped remains in the register, it is possible to judge that a problem has occurred with the software.

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. As for the stop of the sub clock, you can recognize by reading the register.

The main clock supervisor stops automatically when the main clock is intentionally stopped. When the sub clock is intentionally stopped, 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 oscillation operation if the main clock operates as a replacement for the CR oscillator after detecting the main clock stop.

The following explains the operational mode of the clock supervisor.

4.1. Initial State

This section explains the initial state.

At 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 released, the oscillation is automatically restarted.

■ Main Clock Supervisor

For devices whose initial value is ON, the main clock supervisor is enabled after the main oscillation stabilization wait time has elapsed.

For devices whose initial value is OFF, it is disabled in the initial state. It can be enabled when the clock supervisor is enabled again.

When the main clock supervisor is enabled, if the main clock stops, the main clock is replaced by the CR oscillation clock.

Moreover, the MM bit of the CSVCR register is set to "1" and an RST level reset is generated.

(Note) For details of the ON/OFF initial settings for devices, refer to "3. Product Line-up" in chapter "OVERVIEW".
Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.
In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

■ Sub Clock Supervisor

After the timeout period measured by internal CR oscillator passes, it is enabled.

When the sub-clock supervisor is enabled, the behavior when a sub-clock stops changes according to whether MCU is operating with the main clock or with a sub-clock.

- For the main clock mode
  When a sub-clock stops while operating in the main clock mode, the sub-clock is replaced by the CR oscillation clock divided by 2. Afterwards, although the SM bit of the CSVCR register is set to "1", operation continues in the main clock mode with no reset generated.

  Under such conditions, if a change to the sub-clock mode takes place, the clock changes to the sub-clock mode that operates with the CR oscillation clock.

- For the sub clock mode
  When a sub clock stops while operating in the sub-clock mode, CR oscillation clock divided by 2 replaces a sub-clock. Afterwards, the SM bit of the CSVCR register is set to "1", and reset of the RST level is generated.

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".

When there is a problem with the main clock or the sub-clock, the stop of the CR oscillator is prohibited. It can be confirmed whether or not the problem exists in the clock by the MM bit 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".

4.3. Re-enabling the Clock Supervisor

This section explains re-enabling the clock supervisor.

■ Main Clock Supervisor

To re-enable the main clock supervisor function, set the MSVE bit of the CSVCR register to "1".

When the CR oscillator is stopped, enabling the main clock supervisor function is prohibited.

(Note) Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.
In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

■ Sub Clock Supervisor

To permit the sub clock supervisor function again, the SSVE bit of the CSVCR register is set to "1". When the CR oscillator is stopped, enabling the sub-clock supervisor function is prohibited.

4.4. Sub Clock Mode

This section explains the sub clock mode of the clock supervisor.

When the device changes to the sub-clock mode with the main clock supervisor function enabled, the main clock supervisor function stops automatically.

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.

(Note) Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.
In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main clock supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

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.

(Note): Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.
In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

Note:

When the main clock supervisor function is disabled, if stop mode is entered, the supervisor remains disabled even

after recovering from 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.

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 RTC is connected to the main clock with the main clock supervisor function enabled, the system switches to the CR oscillation clock and a reset is issued when the main clock stops.

The watch mode is made clear, and RTC is initialized.

When the RTC is connected to the main clock with the main clock supervisor function prohibited, even if the main clock stops it is not detected, with the result that RTC simply stops.

■ Sub Clock Supervisor

The sub clock supervisor function is not influenced from the transition to the watch mode.

When the RTC is connected to the sub-clock with the sub-clock supervisor function enabled, the system switches to the CR oscillation clock and a reset is not issued when the sub-clock stops.

When the RTC is connected to the sub-clock with the sub-clock supervisor function prohibited, even if the sub-clock stops it is not detected, with the result that RTC simply stops.

4.7. Checking the Reset Factor Using the Clock Supervisor

Checking the reset factor using the clock supervisor is shown.

The method for checking whether or not the clock supervisor detected a clock problem and generated a reset is shown below.

First, read the RSTRR register (see "4.1 Reset Source Register: RSTRR (ReSeT Result Register)" in "CHAPTER: RESET") to check the reset factor.

[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] 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 assertion of RSTX and NMIX external pins was generated. [MB91F52xxxD] 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, or clock supervisor reset was generated.

Please read the CSVCR register in this case, and confirm the MM bit. Also, read the RSTRR register (see "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.

Table 4-1 Reset Factor

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 "4.1 Reset Source Register: RSTRR (ReSeT Result Register)" in "CHAPTER: RESET").

4.8. Return from CR Clock

Return from the CR clock is shown.

■ Main Clock Supervisor

The main clock stops when the CPU detects that the MM bit has been set after recovering from a reset, and it is possible to determine that the system has switched to the CR oscillation clock. At this time, it is possible to return to the main clock by writing "0" in the MM bit if it can be confirmed that the main clock is restored.

When the main clock is not restored, writing "0" in the MM bit does not have any influence. The MM bit keeps maintaining "1".

The MM bit is cleared if the main clock is under operation when "0" is written in the MM bit, and the clock returns to the main clock via a synchronous stage.

It can perform polling on the MM bit until the main clock is restored.

```
ldi #_csvcr,r1
clear_CSV_loop:
bandh #0b1001,@r1 ;; Clear MM+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 CPU detects that the SM bit has been set and it is possible to determine that the system has switched to the CR oscillation clock. At this time, it is possible to return to the 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 if a sub clock is under operation 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.

4.9. Sub Clock Mode Enabled by Setting SCKS Bit

Sub clock mode enabled by setting the SCKS bit is shown.

If the SCKS bit of the single clock product is set to "1", the device can be used in sub clock mode which originates from divide-by-two output (50kHz) of the CR clock.

For details of procedures to select sub clock mode, see "CHAPTER: CLOCK".

Chapter 35: Regulator Control

This chapter explains the regulator control.

1. Overview
2. Features
3. Configuration
4. Register
5. Operation

Code : BZLVDR_REGU-2v1-91520-2-E

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 two regulator modes.

- Main mode (except when shut down)
- Standby mode (at STOP (shutdown) mode and Watch (shutdown) mode)

2. Features

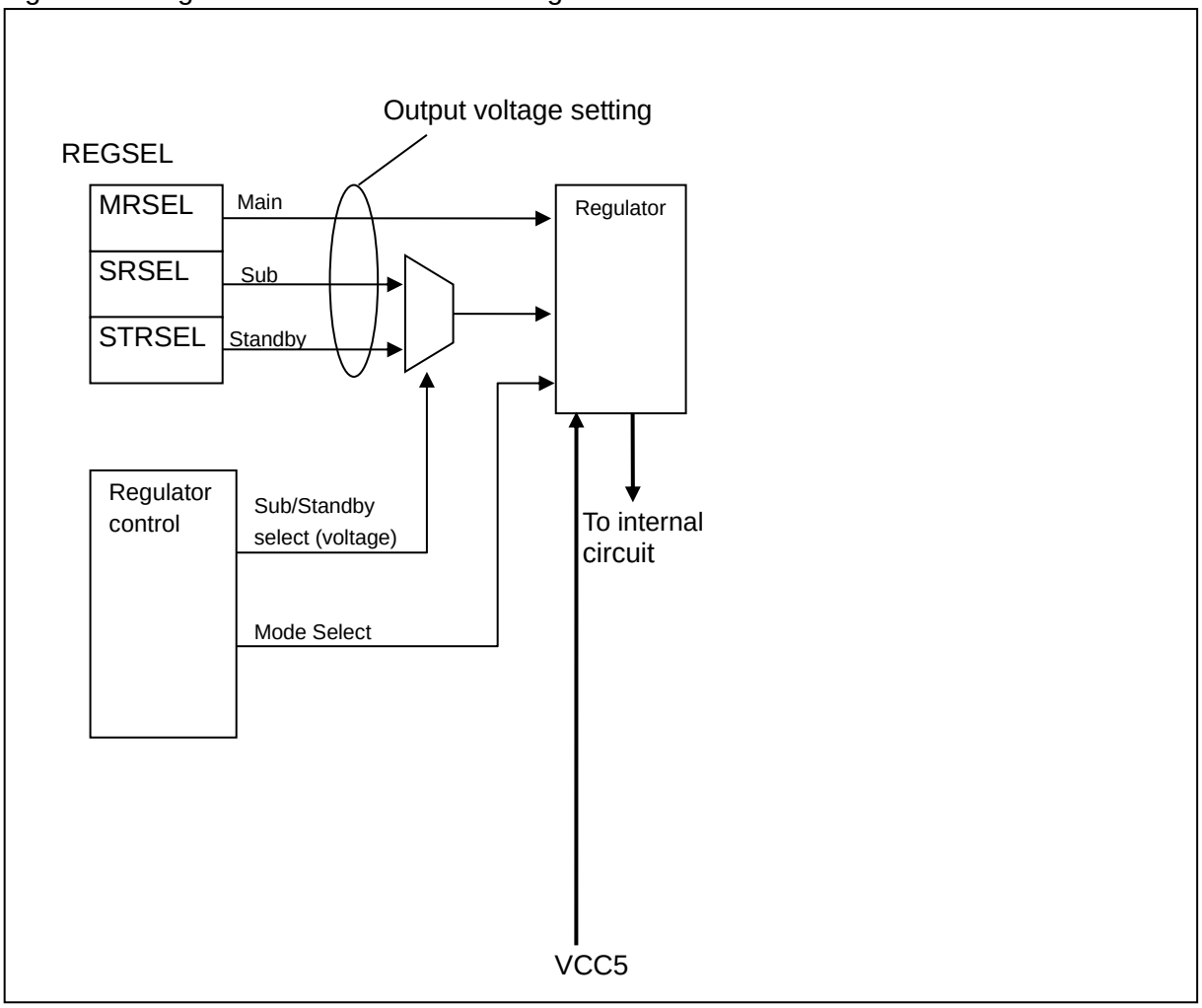
This section explains features of the regulator control.

The regulator mode is automatically changed according to the device state transition.

3. Configuration

This section explains the configuration of the regulator control.

Figure 3-1 Regulator Control Overview Diagram



*: The difference between the sub mode and the standby mode is only the output voltage settings.

4. Register

This section explains a register of the regulator control.

Table 4-1 Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0580	REGSEL	Reserved	Reserved	Reserved	Regulator Output Voltage Selection Register

4.1. Regulator Output Voltage Select Register : REGSEL (REGulator output voltage SELect register)

The bit configuration of the regulator output voltage selection register is shown below.

This register selects the output voltage level of each regulator mode (main/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 (regulator mode : main mode).

MRSEL[1:0]	Main regulator output voltage
00	Reserved
01	1.2±0.1V
10	Reserved
11	Reserved

[bit5, bit4] SRSEL[1:0] (Sub Regulator voltage SELect)

These bits set the output voltage level of sub regulator (regulator mode : sub mode).

SRSEL[1:0]	Sub regulator output voltage
00	Reserved
01	Reserved
10	1.2±0.1V
11	Reserved

[bit3 to bit1] STRSEL[2:0] (STandby Regulator voltage SELect)

These bits set the output voltage level of standby regulator (regulator mode : standby mode).

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 1.2V as the set value (STRSEL[2:0]=110).

[bit0] Reserved

5. Operation

This section explains the operation of the regulator control.

Before entering standby mode, set STRSEL[2:0] to "110". Note that this value is not set immediately after a reset.

Chapter 36: External Bus Interface

This chapter explains the external bus interface.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FR81S10_EBI-1v1-91520-2-E

1. Overview

This section explains the overview of the external bus interface.

This chapter explains each of the functions of the external bus interface.

External bus interface is available with the following models:

MB91F52xK (144 pin), MB91F52xL (176 pin)

2. Features

This section explains the features of the external bus interface.

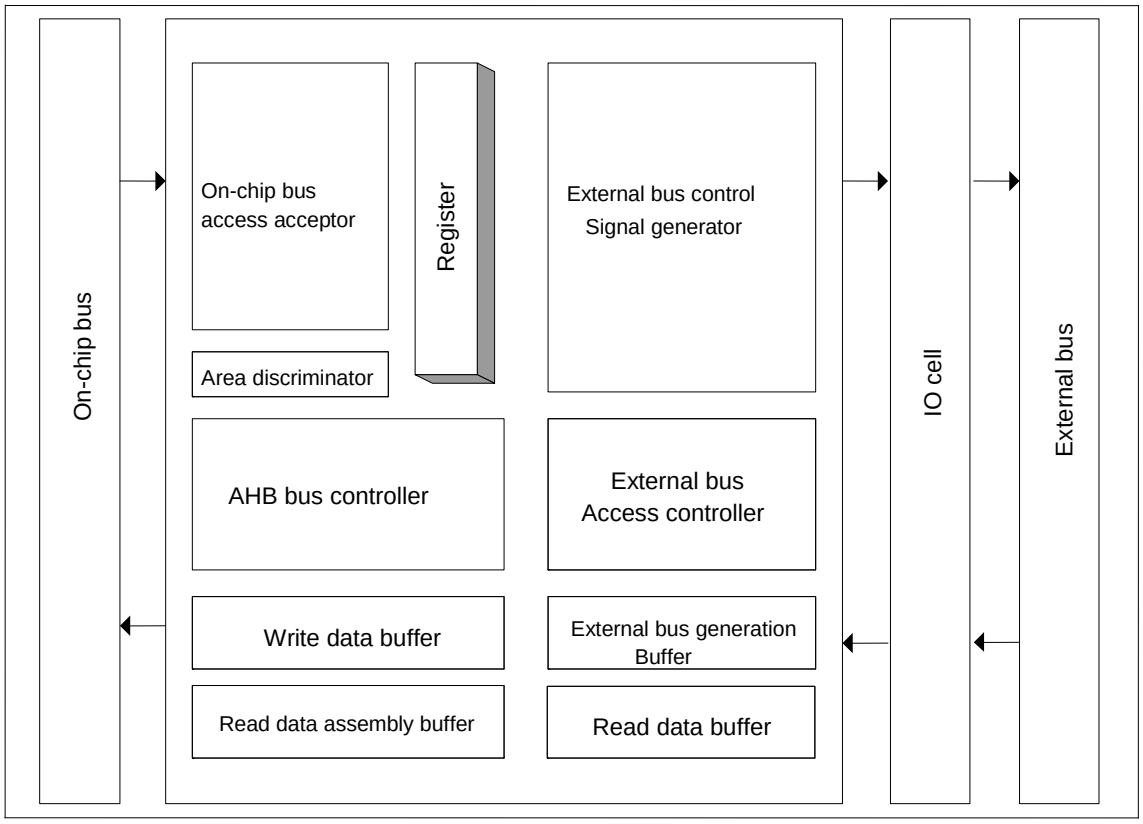
- Address up to 22 bits long (4 MByte space) can be output. (The address space can be extended to 8 MByte by treating the lowermost bit as fixed and extending the upper bit by 1 bit, depending on setting the ACR0 to ACR3:ADTY bit.)
- Supports Address/data split bus
  - Able to connect to asynchronous memory
- Supports Address/data multiplexed 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 KByte to 4 MByte
- 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
  - Address/data split bus
  - Address/data multiplexed 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
    - Insert idle cycle between read access and write access
  - 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
  - Address/data multiplexed bus
  - Address output cycle count

3. Configuration

This section shows the configuration of the external bus interface.

Figure 3-1 Block Diagram of External Bus Interface



4. Registers

This section explains the registers of the external bus interface.

Table 4-1 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.)

4.1. CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3)

The bit configuration of the CS area setting registers is 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 "5.10 CS Setting Flow" for the setting procedure for these registers.

- **ASR0 : Address 0600_H (Access : Word)**
- **ASR1 : Address 0604_H (Access : Word)**
- **ASR2 : Address 0608_H (Access : Word)**
- **ASR3 : Address $060C_H$ (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 $0000_0000_0000_0000_0000_0000_1111_0001_B$
Other than ASR0 $XXXX_XXXX_XXXX_XXXX_0000_0000_XXXX_0XX0_B$

*2 : "R0,W0" for 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 initial values for other than ASR0 are undefined. Set the start address in 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 1Mbyte 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	64Kbyte	SADR[31:16]
0001	128Kbyte	SADR[31:17]
0010	256Kbyte	SADR[31:18]
0011	512Kbyte	SADR[31:19]
0100	1Mbyte	SADR[31:20]
0101	2Mbyte	SADR[31:21]
0110	4Mbyte	SADR[31:22]
0111	8Mbyte	SADR[31:23]
1000	16Mbyte	SADR[31:24]
1001	32Mbyte	SADR[31:25]
1010	64Mbyte	SADR[31:26]
1011	128Mbyte	SADR[31:27]
1100	256Mbyte	SADR[31:28]
1101	512Mbyte	SADR[31:29]
1110	1Gbyte	SADR[31:30]
1111	2Gbyte (Initial value)	SADR[31]

[bit3] Reserved

Always write "0" to this bit.

[bit2] WREN (WRite ENable) : 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 initial values for other than ASR are 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) : 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 values 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 initial values for other than ASR are "0".

4.2. CS Bus Setting Registers: ACR0 to ACR3 (Area Configuration Register 0-3)

The bit configuration of the CS bus setting registers is shown below.

These registers set the bus of the CS area. Each CS area has a single ACR register. See "5.10 CS Setting Flow" 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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
Attribute	R/W	R/W	R0,W0	R0,W0	R/W	R/W	RX,W0	RX,W0

* [Initial value] ACR0 $0000_0000_0000_0000_0000_0000_0100_0000_B$
Other than ACR0 $0000_0000_0000_0000_0000_0000_XX00_XX0X_B$

[bit31 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 "01". The initial values for other than ACR0 are undefined.

[bit5, bit4] Reserved

Always write "0" to these bits.

[bit3] ADTY (ADdress output TYpe) : Address Type

This bit sets the address output type.

ADTY	Description
0	Normal output
1	During 16-bit addressing, addresses are shifted by 1 bit and output. See "5.6 Address Information" for details.

The initial value for ACR0 is "00", and initial values for other than ASR are undefined.

[bit2] BSTY (BuS TYpe) : Bus Type

This bit sets the bus type.

BSTY	Description
0	Address/data split bus
1	Address/data multiplexed bus

The initial value for ACR0 is "00", and initial values for other than ASR are undefined.

[bit1, bit0] Reserved

Always write "0" to these bits.

4.3. CS Wait Registers : AWR0 to AWR3 (Area Wait Register 0-3)

The bit configuration of the CS wait registers is 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 "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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
Attribute	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	R0,W0

* [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) : 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) : 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) : 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)

RIDL[1:0]	Read Access Idle Cycle
01	1 cycle
10	2 cycles
11	3 cycles

[bit17, bit16] WRCV[1:0] (Write ReCoVery cycle) : Write Recovery Cycle

This bit sets the write recovery cycle 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 signal 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) : 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 multiplexed bus is configured (ACR:BSTY=1), set the AWR parameters to satisfy the following conditions.

$ACS + CSRD \geq 1$ and $ACS + CSWR \geq 1$

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) : 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) : 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 multiplexed 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) : 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. The minimum ASX output extension cycle is 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) : 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.

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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	*	*	*	*
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	*	*	*	*	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W0	R/W0	R/W0	R/W0

* [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.

5. Operation

This section explains the operation of the external bus interface.

5.1 External Pin Table

5.2 External Bus Signal Protocol

5.3 Address Alignment

5.4 Split Access

5.5 Data Alignment

5.6 Address Information

5.7 Idle Cycle Insertion Function

5.8 External Bus Output Signal Timing Settings

5.9 RDY Pin Access Cycle Extension Function

5.10 CS Setting Flow

5.11 Example of Connecting to Asynchronous Memory

5.12 Example of Connection to Little Endian Device

5.1. External Pin Table

This section shows the external pin table.

Table 5-1 shows the external pins for the external bus interface of this series.

Table 5-1 External Pin Table

External pin of this series	Pin number of this series						Description
	64pin	80pin	100pin	120pin	144pin	176pin	
SYSCLK	-	-	-	-	35	43	System clock output
ASX	-	-	-	-	5	7	Address strobe output
CS0X	-	-	-	-	6	8	Chip selected output
CS1X	-	-	-	-	7	9	
CS2X	-	-	-	-	38	46	
CS3X	-	-	-	-	39	49	
RDX	-	-	-	-	8	10	Read strobe output
WR0X	-	-	-	-	9	11	Write strobe output
WR1X	-	-	-	-	10	12	
RDY	-	-	-	-	41	51	Bus ready input
D16	-	-	-	-	131	159	Data input/output and address output (during address multiplexing)
D17	-	-	-	-	132	160	
D18	-	-	-	-	133	161	
D19	-	-	-	-	134	162	
D20	-	-	-	-	135	163	
D21	-	-	-	-	136	165	
D22	-	-	-	-	137	167	
D23	-	-	-	-	138	168	
D24	-	-	-	-	139	170	
D25	-	-	-	-	140	171	
D26	-	-	-	-	141	172	
D27	-	-	-	-	142	174	
D28	-	-	-	-	143	175	
D29	-	-	-	-	2	2	
D30	-	-	-	-	3	3	
D31	-	-	-	-	4	5	

External pin of this series	Pin number of this series						Description
	64pin	80pin	100pin	120pin	144pin	176pin	
A00	-	-	-	-	11	14	Address output
A01	-	-	-	-	12	15	
A02	-	-	-	-	13	17	
A03	-	-	-	-	14	18	
A04	-	-	-	-	15	19	
A05	-	-	-	-	16	20	
A06	-	-	-	-	17	21	
A07	-	-	-	-	20	24	
A08	-	-	-	-	21	25	
A09	-	-	-	-	22	26	
A10	-	-	-	-	23	29	
A11	-	-	-	-	24	30	
A12	-	-	-	-	25	31	
A13	-	-	-	-	26	32	
A14	-	-	-	-	27	33	
A15	-	-	-	-	28	34	
A16	-	-	-	-	29	35	
A17	-	-	-	-	30	37	
A18	-	-	-	-	31	39	
A19	-	-	-	-	32	40	
A20	-	-	-	-	33	41	
A21	-	-	-	-	34	42	

5.2. External Bus Signal Protocol

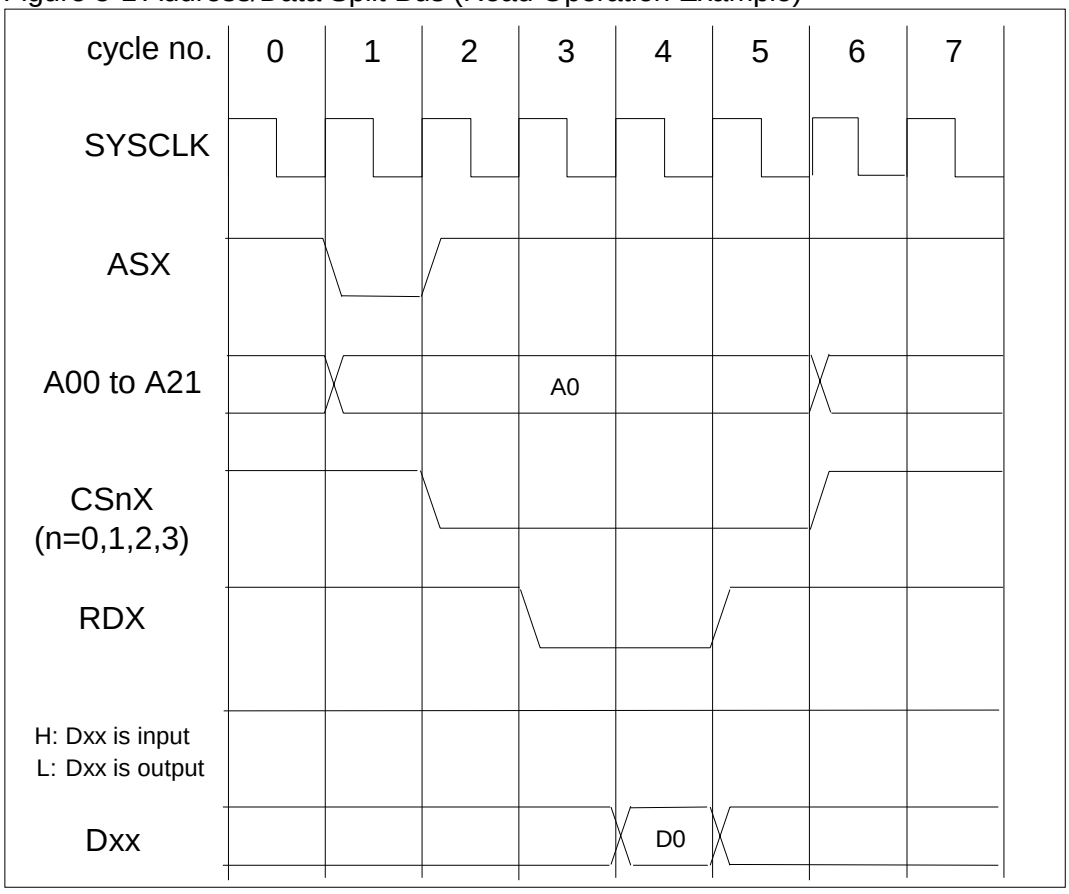
This section shows the external bus signal protocol.

5.2.1. Address/Data Split Bus Read Protocol

This section shows the read protocol for address/data split bus.

This section shows the protocol for read access using an address/data split bus.

Figure 5-1 Address/Data Split Bus (Read Operation Example)



- **Operation 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 started, "L" begins to be output to CSnX (n= 0 to 3), and continues until the access finishes. 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 from when the output of CSnX="L" started, "L" is output to RDX. External bus devices are required to return read data to D16 to D31 within the strobe period indicated by RDX="L".

cycle4 : After the configured count has finished from when the output of RDX="L" started, RDX output is returned to "H". 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

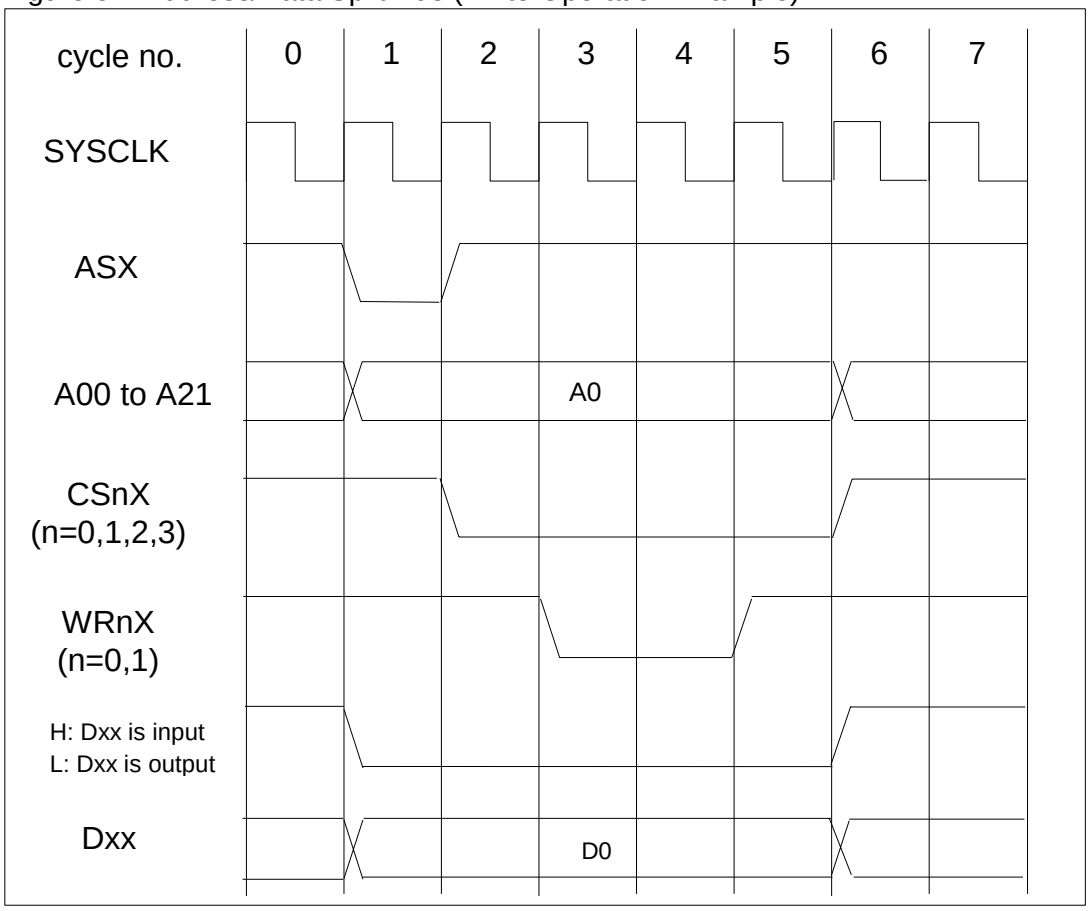
The external bus device is required to return valid data in D16 to D31 within the period where RDX="L". This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

5.2.2. Address/Data split bus write protocol

This section shows the write protocol for address/data split bus.

This section shows the protocol for write access using an address/data split bus.

Figure 5-2 Address/Data Split Bus (Write Operation Example)



- **Operation 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

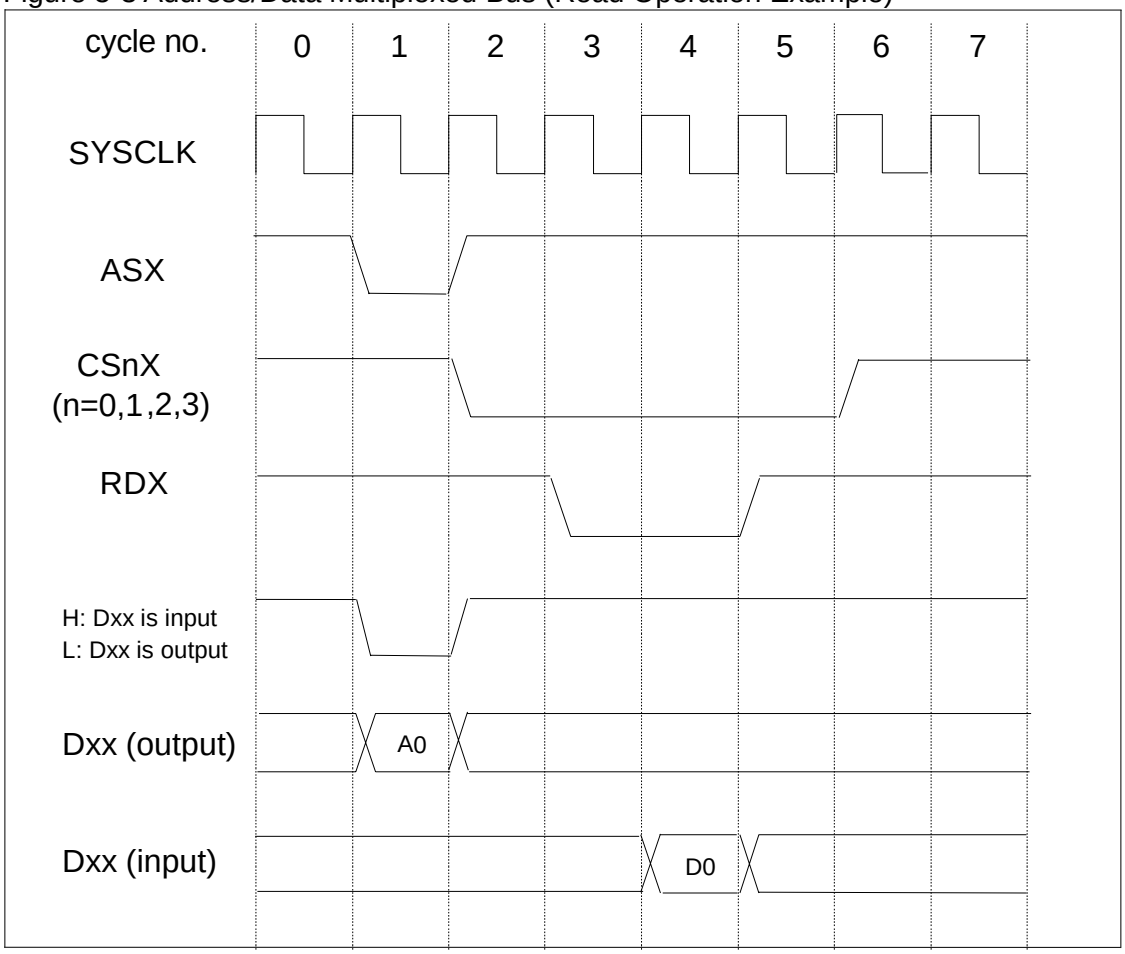
Write data is output from when the access begins. The write data output continues until the access finishes. External bus devices are required to fetch the data of D16 to D31 within the period where WRnX="L".

5.2.3. Address/data multiplexed bus read protocol

This section shows the read protocol for address/data multiplexed bus.

This section shows the protocol for read access using an address/data multiplexed bus.

Figure 5-3 Address/Data Multiplexed Bus (Read Operation Example)



- **Operation 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

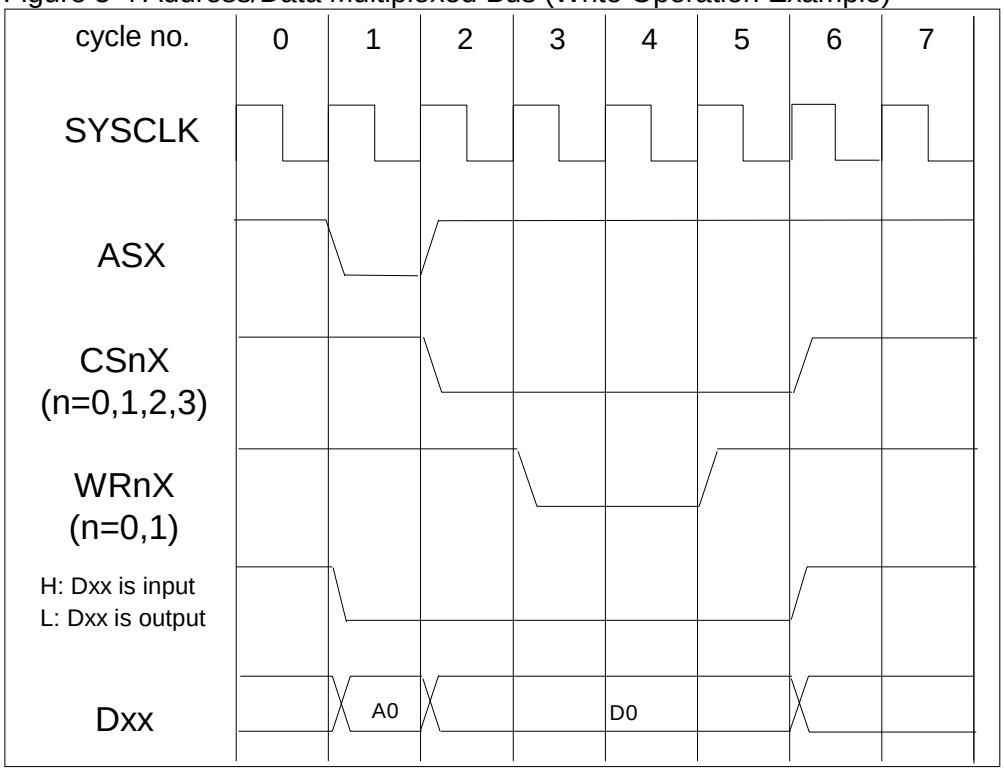
Address information is output from when the access begins. After the configured count has finished, this enters the input state and accepts the read data from the external bus device. This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

5.2.4. Address/Data multiplexed bus write protocol

This section shows the write protocol for address/data multiplexed bus.

This section shows the protocol for write access using an address/data multiplexed bus.

Figure 5-4 Address/Data Multiplexed Bus (Write Operation Example)



Operation 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.

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".

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.

Table 5-2 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

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.

Table 5-3 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	○	○

Table 5-4 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	-	○	-

Access		Split access	Output pins				
Size	Address lowermost 2 bits		A01, A00	D31 to D24	D23 to D16	WR0X	WR1X
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	-	○	-

Table 5-5 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	○	○

Table 5-6 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	-	○	-

5.6. Address Information

This section shows the address information.

5.6.1. Address information and output pins

This section shows the address information and output pins.

- **Address/data split bus**
  22-bit address information is output to A00 to A21.

- **Address/data multiplexed 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.

5.6.2. Address type

This section shows the 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.

Table 5-7 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			

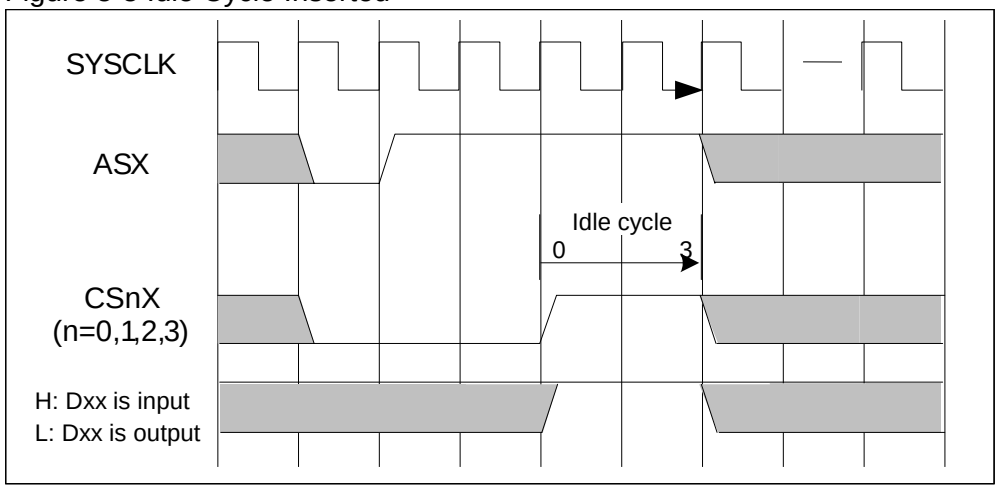
ACR register		Bus width [bit]	A21 to A00	Output pins D31 to D16 during address output cycle	
ADTY	BSTY			D31 to D24	D23 to D16
0	1	8	Address[21:0]	Address[7:0]	-
		16	Address[21:0]	Address[15:8]	Address[7:0]
1	0	8	Address[21:0]	-	-
		16	Address[22:1]		
1	1	8	Address[21:0]	Address[7:0]	-
		16	Address[22:1]	Address[16:9]	Address[8:1]

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 5-5 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].

5.8. External Bus Output Signal Timing Settings

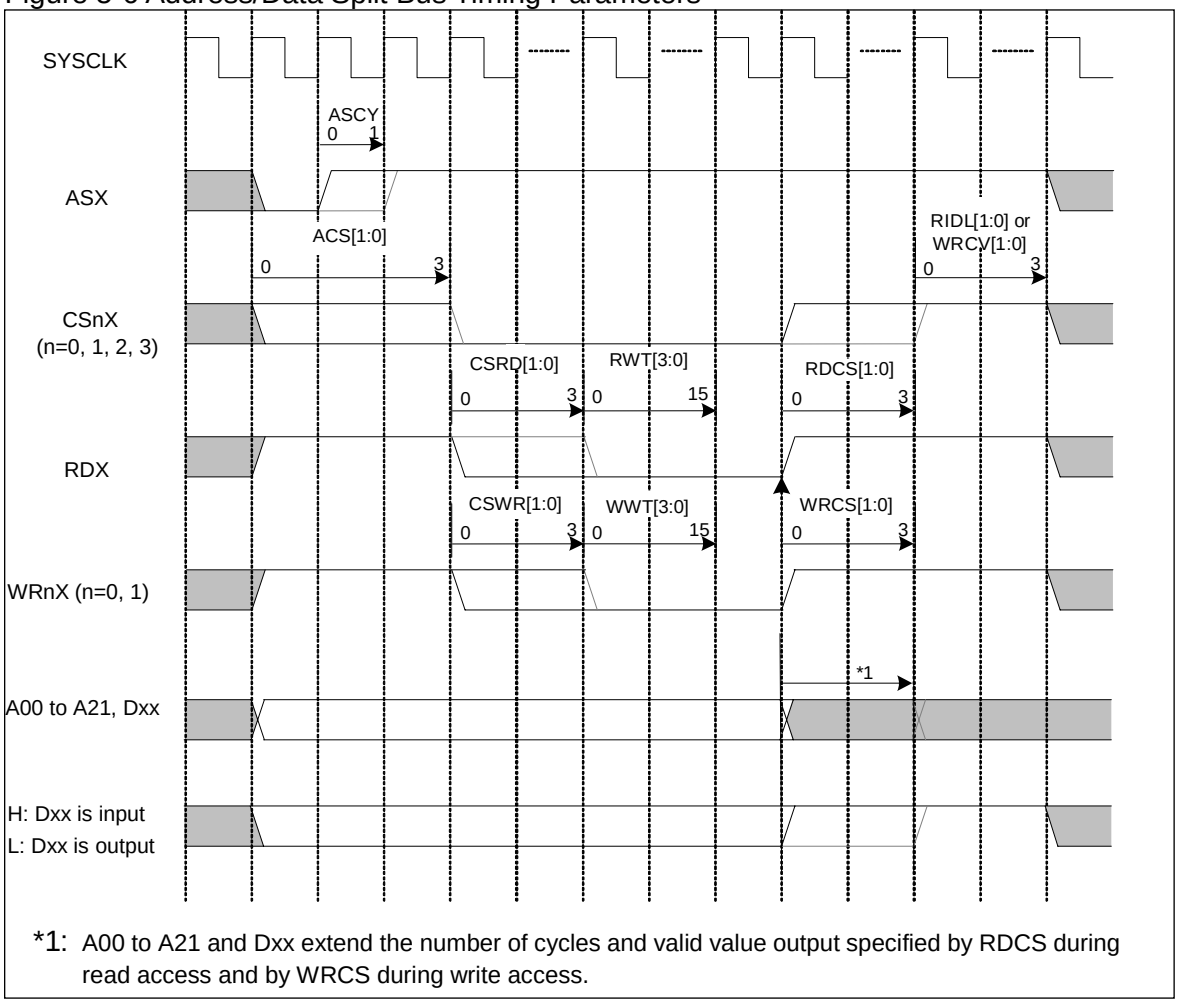
This section shows the external bus output signal timing settings.

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 5-6 Address/Data Split Bus Timing Parameters



*1: A00 to A21 and Dxx extend the number of cycles and valid value output specified by RDCS during read access and by WRCS during write access.

Table 5-8 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, after CSRD count from CSnX "L" output start, "L" output to RDX starts.

Parameter name	Function name	Description
RWT[3:0] (Read access auto WaiT)	Read access auto wait	During read access, after (RWT+1) count from RDX "L" output start, RDX output is returned to "H".
RDCS[1:0] (RDX to CSnX hold cycle)	RDX → CSnX hold cycle	During read access, after RDCS count from the cycle when RDX output is returned to "H", CSnX output is returned to "H".
CSWR[1:0] (CSnX to WRnX setup cycle)	CSnX → WRnX setup cycle	During write access, after CSWR count from CSnX "L" output start, "L" output to WRnX starts.
WWT[3:0] (Write access auto WaiT)	Write access auto wait	During write access, after (WWT+1) count, WRnX(n=0,1) output is returned to "H".
WRCS[1:0] (WRnX to CSnX hold cycle)	WRnX → CSnX hold cycle	During write access, after WRCS count from the cycle when WRnX output is returned to "H", CSnX output is returned 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

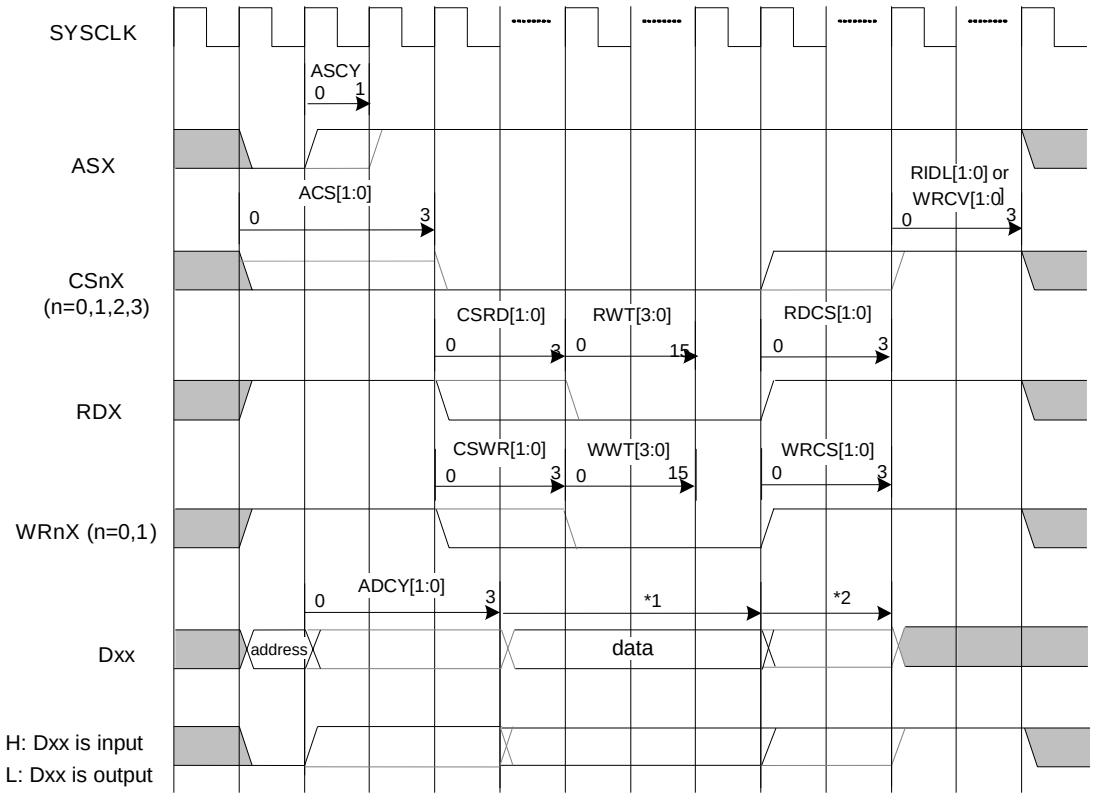
The following conditions need to be met in order to correctly establish the protocol.

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 5-7 Address/Data Multiplexed Bus Timing Parameters



*1: During write access, Dxx outputs write data immediately after the address output cycle ends.
*2: Dxx extends the number of cycles and valid value output specified by WRCS during write access.

Table 5-9 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.

Parameter name	Function name	Description
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 does not affect the start conditions of other counters. As a result, there are some 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

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.

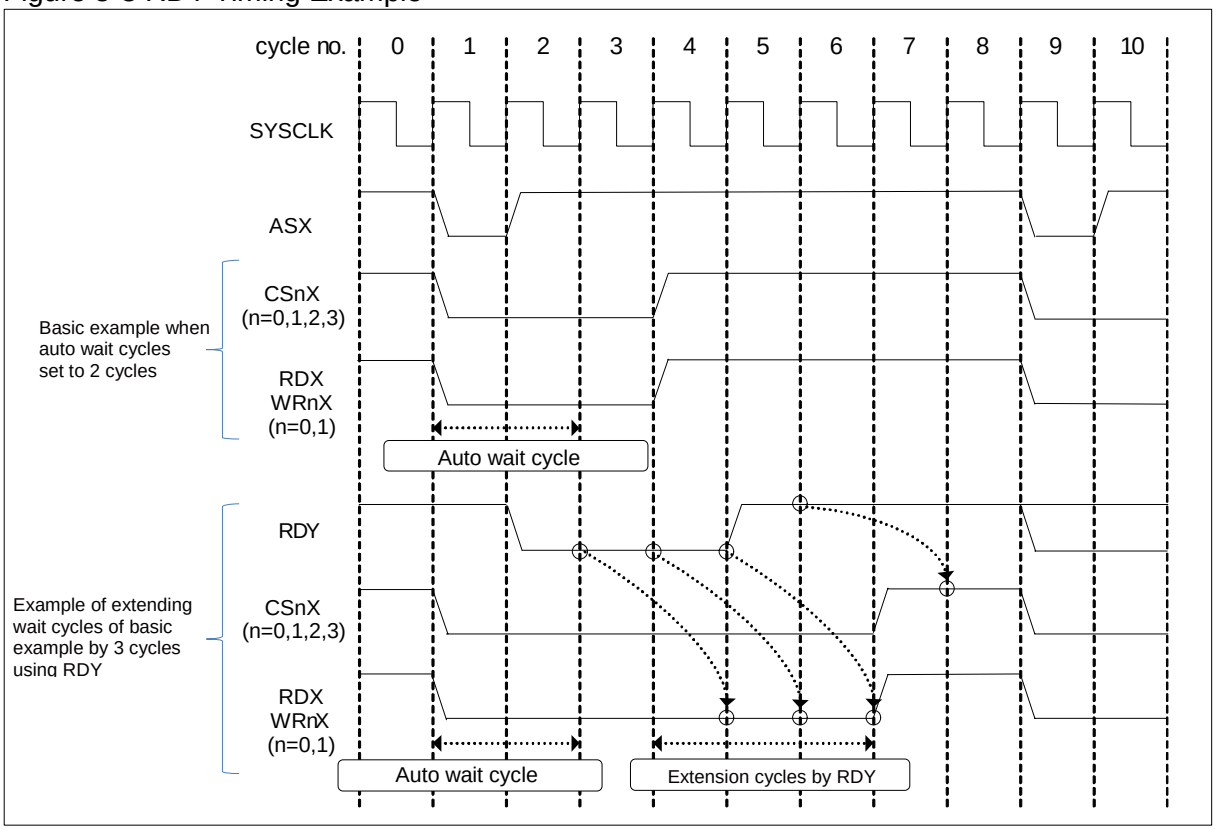
After the auto wait cycle has finished, the read and write strobe cycles are extended while "0" is input to RDY. If "1" is input to RDY after this, the read or write strobe cycle finishes in the next cycle.

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 5-8 RDY Timing Example



5.10. CS Setting Flow

This section explains the CS setting flow.

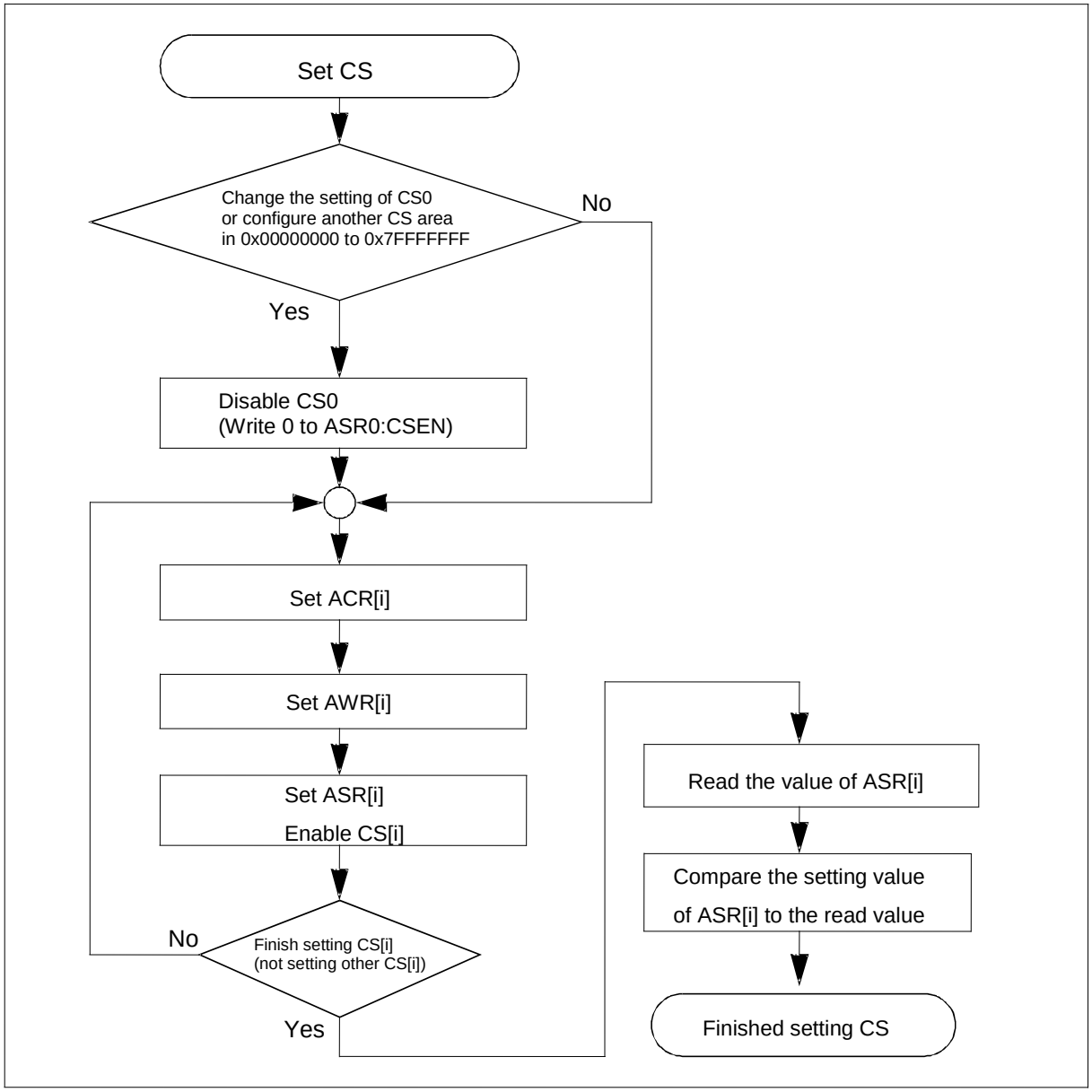
This section explains CS setting method.

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, 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 if the CS0 area is to be changed.
- 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 5-9 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.

Write the above setting values 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. Write the setting values to AWR as words.

Figure 5-10 Parameters that can be Configured in AWR

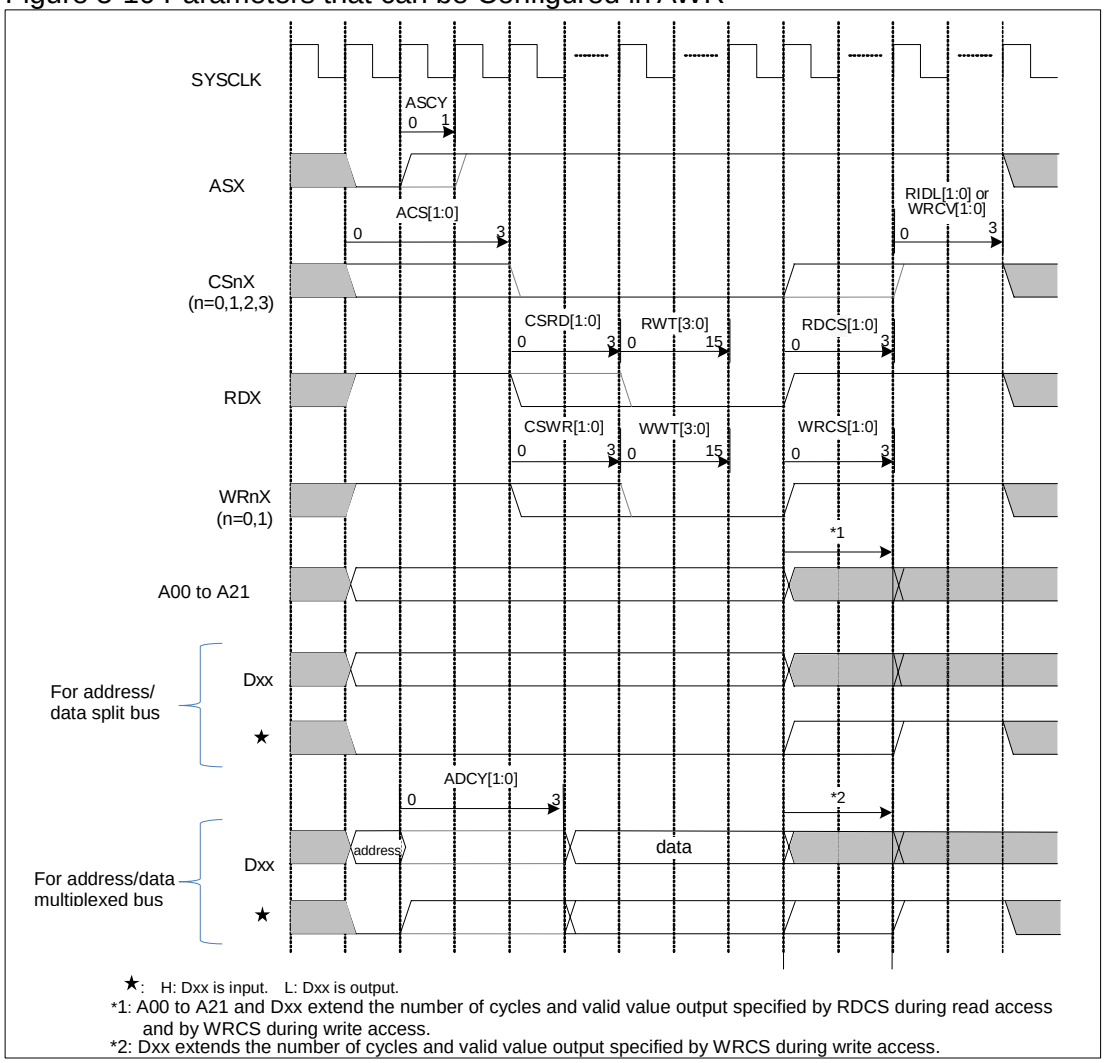


Table 5-10 List of Parameter

Parameter name	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.

Write the above setting values 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 "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 "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 setting 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 1MByte:

  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

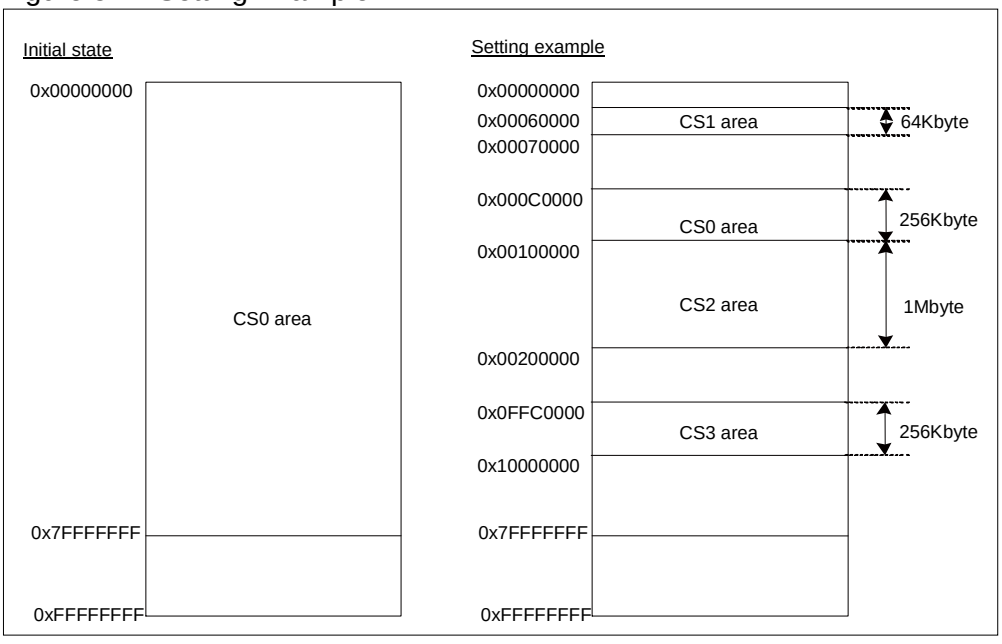
  → 0x00100000 to 0x001FFFFF becomes the CS2 area.

- **CS3 settings**

  ASR3:ASZ[3:0]=0010
  ASR3:SADR[31:16]=0x0FFC

  → 0x0FFC0000 to 0x0FFFFFFF becomes the CS3 area.

Figure 5-11 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 5-12 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

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 endian
- 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     #_ACR1, r0          // #_ACR1 is the ACR1 address value
ldi     #0x40, r1           // Set ACR1 to 0x40
st      r1, @r0
_set_AWR1
ldi     #_AWR1, r0          // #_AWR1 is the AWR1 address value
ldi     #0x034b5a00 r1      // Set AWR1 to 0x034b5a00
st      r1, @r0
_set_ASR1
ldi     #_ASR1, r0          // #_ASR1 is the ASR1 address value
ldi     #0x00400005 r1      // Set ASR0 to 0x00400005
st      r1, @r0
ld      @r0, r2
cmp     r1, r2              // Check the setting value of ASR1
```

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 5-13 Example 1 of Connection to SRAM (8-bit SRAM×2)

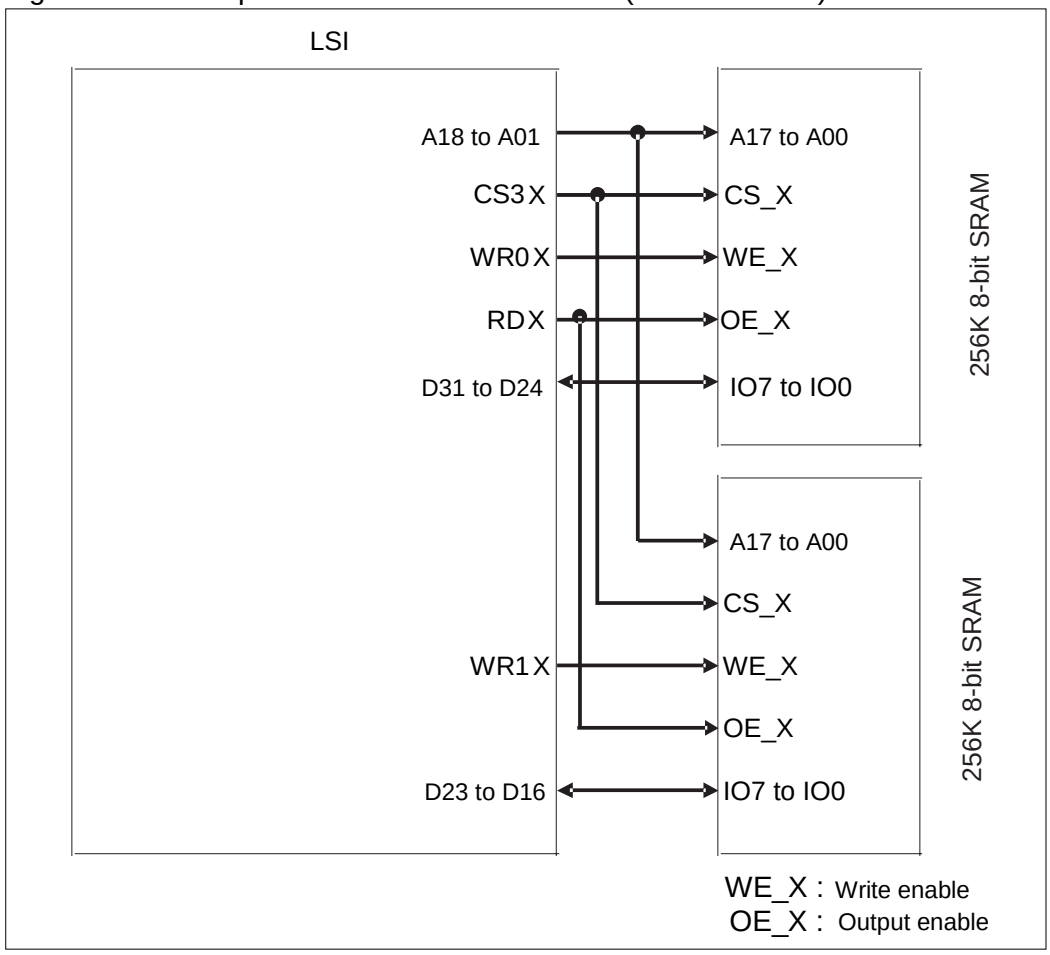
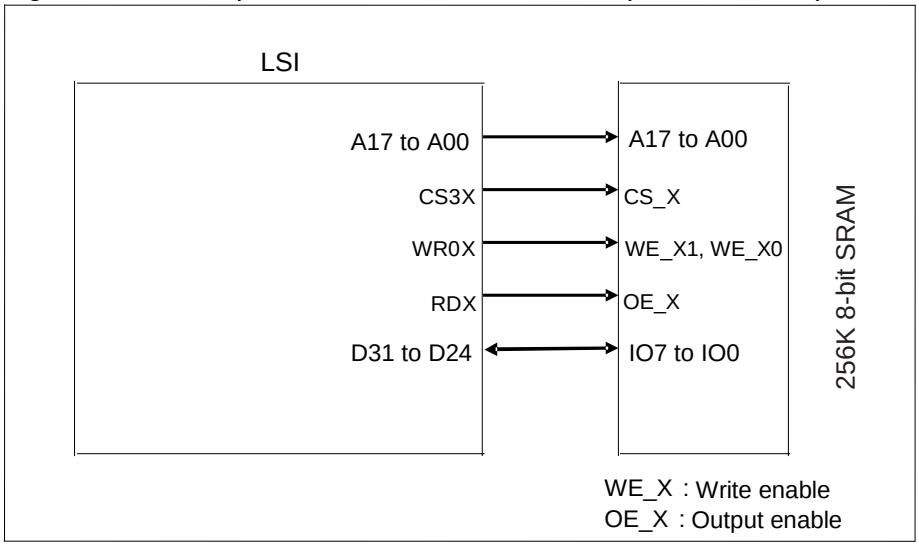


Figure 5-14 Example 2 of Connection to SRAM (8-bit SRAM×1)



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 5-15 16-bit Bus Width

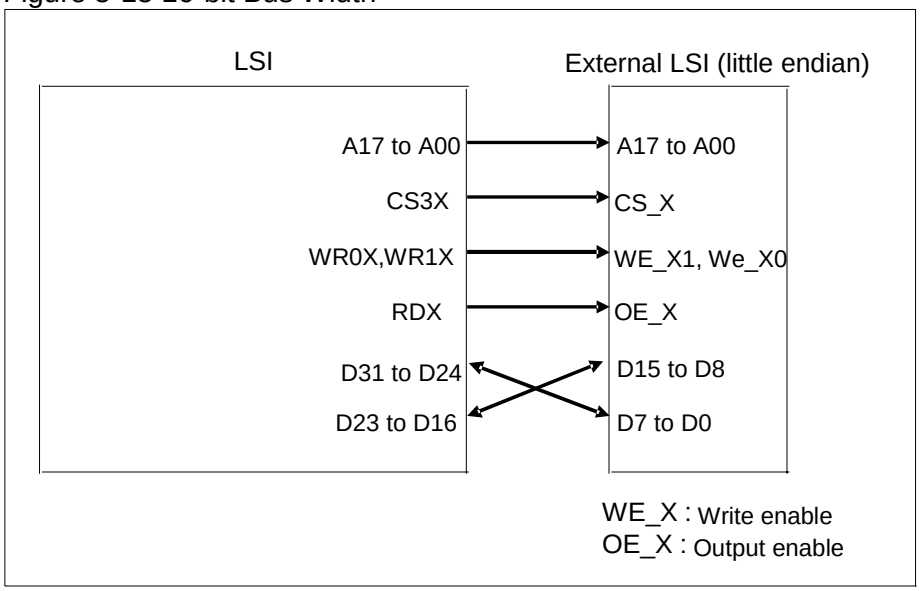
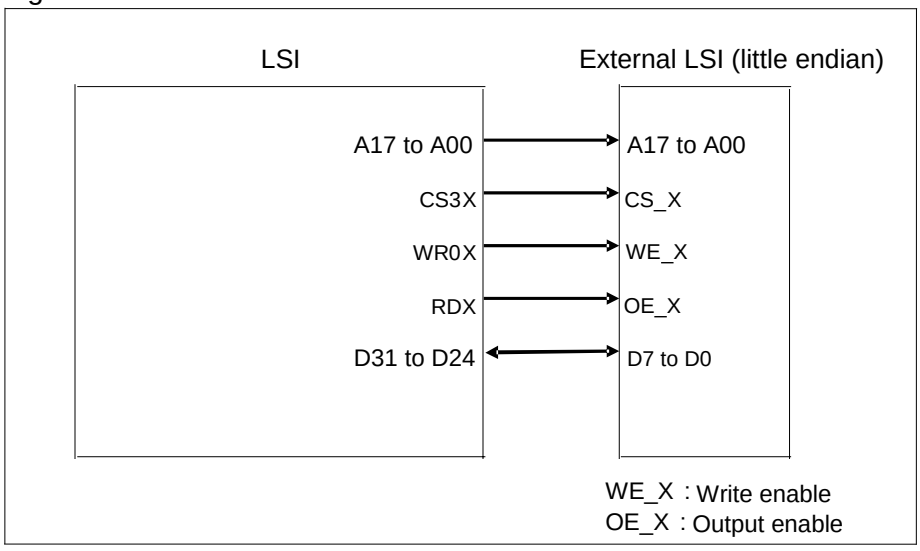


Figure 5-16 8-bit Bus Width



Chapter 37: Bus Performance Counters

This chapter explains the bus performance counters.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FR81S10_BPC-1v1-91520-2-E

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.

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

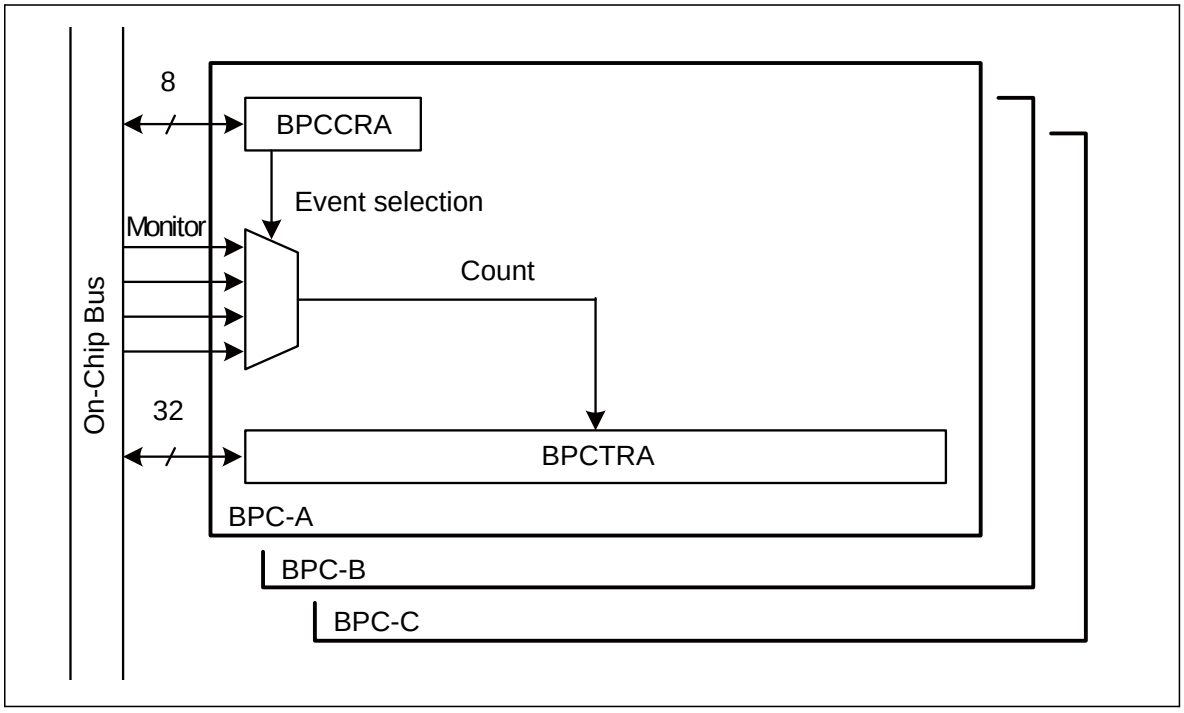
One of the following operations can be selected for counting in each channel

- Specific bus master (CPU, DMAC, other, or all)
- Specific target (ICH, MCH, other, or all)

3. Configuration

This section explains the configuration of the bus performance counters.

Figure 3-1 Block Diagram



4. Registers

This section explains the registers of the bus performance counters.

Table 4-1 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

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

FUNC[1:0]	Event
10	Number of write accesses
11	Number of wait cycles

[bit5 to bit2] MST[3:0] (bus MaSTer select) : Bus master selection

These bits select the bus master for the events which are measured by BPC.

MST[3:0]	Bus master
0000	All bus masters (initial value)
0001	CPU (XBS)
0010	DMAC
0011	Reserved
0100	Reserved
Except for the above	Reserved

[bit1, bit0] SLV[1:0] (SLaVe select) : Slave selection

These bits select the slave for the events which are measured by BPC.

SLV	Slave
00	All slaves (initial value)
01	MCH (registers, external bus)
10	ICH (peripherals)
11	Slaves other than MCH/ICH

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

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

4.4. BPC-A Count Register : BPCTRA (Bus Performance CounTer Register A)

The bit configuration of the BPC-A count register is shown below.

This register is a 32-bit length count register that counts the events configured by BPCCRA.

■ BPCTRA : Address 0714_H (Access: Word)

	bit31	bit30	•	•	•	bit3	bit2	bit1	bit0
	BPCTRA[31:0]								
Initial value	0	0	•	•	•	0	0	0	0
Attribute	R/W	R/W	•	•	•	R/W	R/W	R/W	R/W

[bit31 to bit0] BPCTRA[31:0] (Bus Performance CounTer Register A) : BPC-A count

If bit7, bit6: FUNC of the BPCCRA are set to a value other than "00", the count of the target events begins. This register is readable and writable, and can only be accessed using 32-bit access. Because the counter is not initialized when the count is started, set the initial value when starting a new count. Furthermore, because there is no overflow control, if the counter overflows it returns to zero and continues counting.

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

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

5. Operation

This section explains the operations.

5.1 Setting

5.2 Starting and Stopping

5.3 Operation

5.4 Measurement and Result Processing

5.1. Setting

This section explains the setting.

Before starting each of the BPC channels, write "0x00000000" to BPCTRA, BPCTRB, and BPCTRC, and initialize each counter. Initialize each counter in the same way when changing the measurement target. Because the counter value is undefined after reset, always write the counter value before enabling operation.

When starting each BPC channel, configure the measurement target of each counter using BPCCRA, BPCCRB, and BPCCRC.

The events monitored by the settings of the bus performance counter A (B, C) control register (BPCCRA (B, C)) are as follows. Operation is not guaranteed for any combination that does not exist in the following table. Moreover, it does not count in emulator mode.

Table 5-1 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
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

FUNC[1:0]	MST[3:0]	SLV[1:0]	Target event
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

5.2. Starting and Stopping

This section explains the starting and stopping.

The target event count is started by setting the FUNC[1:0] bits of the bus performance counter A control register (BPCCRA) to a value other than "00". However, at this time the count starts from the current value without initializing the bus performance counter A register (BPCTRA). The operation of the bus performance counter stops when BPCCRA:FUNC[1:0] is set to "00".

5.3. Operation

This section explains the operation.

Once operation has been enabled by setting the control register, each of the measurement target operations continues to be counted while the on-chip bus is operating. However, the count is paused in the circumstances shown below.

- While in emulator mode

The count operation when each of the low-power consumption modes is set is as follows.

- CPU sleep mode
  Each measurement target operation is counted.
- Bus sleep mode
  Only counted during DMA transfers that operate the on-chip bus. During other periods, counting is not performed because the measurement target operations do not occur.

- Standby mode (watch mode / stop mode)
  Counting is not performed because the measurement target operations do not occur.

The control register is initialized when a reset occurs. Counting is not performed immediately after a reset occurs.

5.4. Measurement and Result Processing

This section explains the measurement and result processing.

The use of BPC is anticipated for when ICE is connected or when using a monitor debugger. The configuring of measurements and reading of results are performed in debug mode while the user program execution is halted.

Examples of measurements are as follows.

- Measure between two points in a user program
- Measure a reference time base

These are explained below.

- Measuring between two points in a user program
  During this measurement, the measurement starting point and measurement ending point in the user program are configured as follows.

  - Measurement starting point: Starting point of the user program execution
  - Measurement ending point: Breakpoint in the user program

The measurement sequence is as follows.

1. Configure the measurement and initialize the counter in debug mode
2. Start executing the user program from the measurement starting point
3. Break on the measurement ending point and stop executing the user program
4. Switch to debug mode and read the measurement results

- 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 DMAC access vs. CPU access, specific bus master access that occupies the total access, etc.
2. Occurred event proportion
   Ex. Proportion of write access vs. read access, proportion of total cycles made up of wait cycles, etc.
3. Target accessed proportion
   Ex. Proportion of MCH vs. ICH, proportion of total accesses made up of accesses to a specific target, etc.
4. Proportion of specific accesses from a specific bus master to a specific target
   Ex. Proportion of total access made up of read accesses from CPU to MCH, etc.
5. Proportion of wait cycles occurring in specific target
6. Ex. Proportion of total cycles made up of wait cycles during MCH access
7. Analyze operation of each bus between two specific points in a program
8. Ex. Proportion of total cycles between two specific points in the program consisting of read, write, wait cycles, etc.
9. 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.

Chapter 38: CRC

This chapter explains the CRC.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS15-2v1-91520-2-E

1. Overview

This section explains the overview of the CRC.

This module calculates CRC values.

CRC (Cyclic Redundancy Check) is a kind of error detection methods. CRC codes are remainders left when input data strings, regarded as high-degree polynomials, are divided by predefined generator polynomials. Normally, a CRC code is attached at the end of a data string, and received data is regarded as correct if the data leaves no remainder when divided by the same generator polynomial.

2. Features

This section explains features of the CRC.

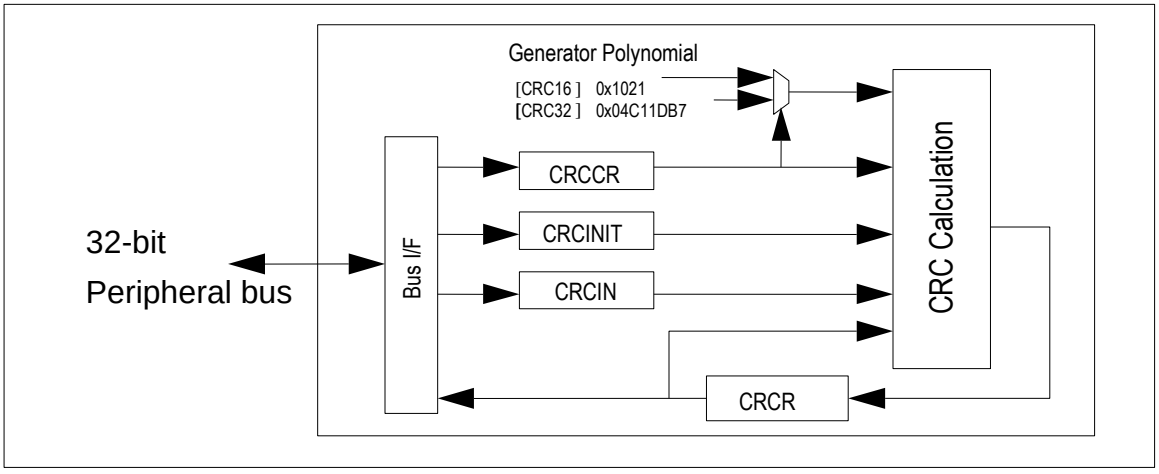
This module calculates CCITT CRC16 and IEEE-802.3 CRC32. This module cannot calculate CRC values based on other generator polynomials because the generator polynomials of this module are fixed for the values of CCITT CRC16 and IEEE-802.3 CRC32.

- CCITT CRC16 generator polynomials : 0x1021
- IEEE-802.3 CRC32 generator polynomials : 0x04C11DB7

3. Configuration

This section explains the configuration of the CRC.

Figure 3-1 Block Diagram



4. Registers

This section explains registers of the CRC.

Table 4-1 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

4.1. CRC Control Register : CRCCR (CRC Control Register)

The bit configuration of the CRC control register is shown below.

This register controls the CRC calculation.

■ CRCCR : Address 1133_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W

[bit7] Reserved
This bit must always be written to "0".

[bit6] FXOR (Final XOR) : Final XOR Control bit
CRC results are output as the XOR value and XOR. The XOR values are 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 setting.

[bit2] LTLEND (LitTtLe-ENDian) : Byte order setting bit

This bit sets byte orders. Specifies byte orders in a writing width. When this bit is "0", big endian is applied, and when this bit is "1", little endian is applied.

[bit1] CRC32 (CRC32) : CRC mode selecting bit

This bit selects a mode for CRC16 and CRC32. When CRC32=1 is true, the arithmetic operation mode of CRC32 is applied.

[bit0] INIT (INITialize) : Initialization bit

Initialization bit. When "1" is written to this bit, software performs the initialization. This bit does not have a value and "0" is always returned at readout. In initialization, hardware loads the value of the initial value register to the CRC register. Initialization needs to be performed once at the beginning of the CRC calculation.

4.2. CRC Initial Value Register : CRCINIT (CRC Initial value register)

The bit configuration of the CRC initial value register is shown below.

This register sets the initial value for the CRC calculation.

■ CRCINIT : Address 1134_H (Access: Byte, Half-word, Word)

	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 (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.

4.3. CRC Input Data Register : CRCIN (CRC INput data register)

The bit configuration of the CRC input data register is shown below.

This register sets the input data for the CRC calculation.

■ CRCIN : Address 1138_H (Access: Byte, Half-word, Word)

	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 (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.

4.4. CRC Register : CRCR (CRC Register)

The bit configuration of the CRC register is shown below.

This register outputs the result for the CRC calculation.

■ CRCR : Address $113C_H$ (Access: Byte, Half-word, Word)

	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 (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.

5. Operation

This section explains the CRC.

5.1 CRC Definition

5.2 Reset Operation

5.3 Initialization

5.4 Byte and Bit Orders

5.5 CRC Calculation Sequence

5.6 Examples

5.1. CRC Definition

The CRC definition is shown below.

- **CCITT CRC16 Standard**

Generator polynomials	0x1021	(CRCCR.CRC32=0)
Initial value	0xFFFF	
Final XOR value	0x0000	(CRCCR.FXOR=0)
Bit order	MSB First	(CRCCR.LSBFST=0)
Output bit order	MSB First	(CRCCR.CRCLSF=0)

(Any byte order can be set for input and output)

- **IEEE-802.3 CRC32 Ethernet Standard**

Generator polynomials	0x04C11DB7	(CRCCR.CRC32=1)
Initial value	0xFFFF_FFFF	
Final XOR value	0xFFFF_FFFF	(CRCCR.FXOR=1)
Bit order	LSB First	(CRCCR.LSBFST=1)
Output bit order	LSB First	(CRCCR.CRCLSF=1)

(Any byte order can be set for input and output)

5.2. Reset Operation

The reset operation of the CRC is shown below.

To reset, set 0xFFFF_FFFF to the initial value register (CRCINIT) and CRC register (CRCR). Others are cleared to "0".

5.3. Initialization

The initialization of the CRC is shown below.

In initialization by CRCCR.INIT, the value of the initial value register is loaded to the CRC register (CRCR).

5.4. Byte and Bit Orders

The byte and bit orders of the CRC is shown below.

This section explains the byte and bit orders using examples. Inputs the following 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	11010101	10000000
(First)			(Last)

Notes:

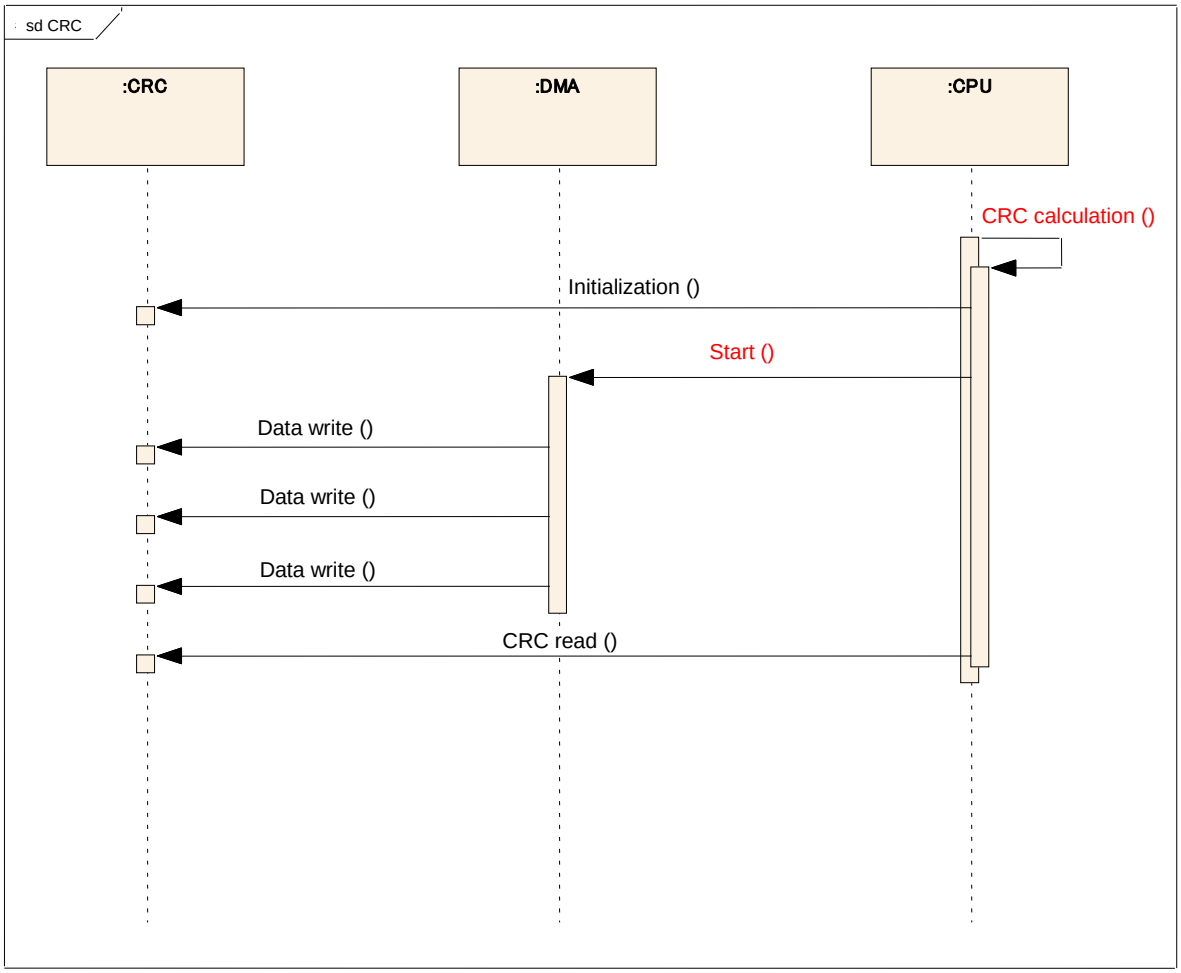
- When CRCCR.CRCLTE=1 is true, the byte order for the CRC result is changed in 32-bit width both for CRC16 and CRC32.
- Note that output position for CRC16 is bit31 to bit16.

5.5. CRC Calculation Sequence

The CRC calculation sequence is shown below.

The sequence for the CRC calculation is shown below. In the following explanation, the initial value register (CRCINIT) setting, CRC16/32 selection (CRCCR.CRC32), byte order and bit order settings (CRCCR.LTLEND, CRCCR.LSBFST) have been done. (When the initial value of ALL "H" is acceptable, the setting for the initial value register (CRCINIT) can be omitted.)

Figure 5-1 CRC Calculation Sequence



- To initialize, write "1" to the initialization bit (CRCCR.INIT). The value of the initial value register will be loaded to the CRC register (CRCR).
- Input data is written to the Input Data register (CRCIN). The writing operation starts the CRC calculation. Input data can be written continuously. In addition, there can be different bit widths of writing in a sequence.
- The CRC code is obtained with the readout of the CRC register (CRCR).

5.6. Examples

The examples is shown below.

5.6.1 Example 1 CRC16, Fixed Byte Input

5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

5.6.3 Example 3 CRC32, Byte Order, Big-endian

5.6.4 Example 4 CRC32, Byte Order, Little-endian

5.6.1. Example 1 CRC16, Fixed Byte Input

Example 1 CRC16 and fixed byte input are shown below.

Figure 5-2 Example 1

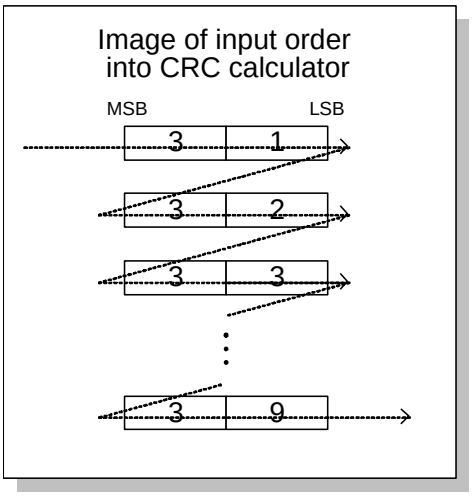
```
//****************************************
// CRC16 (CRC ITU-T)
// polynomial: 0x1021
// initial value: 0xFFFF
// CRCCR.CRC32:  0 // CRC16
// CRCCR.LTLEND: 0 // big endian
// CRCCR.LSBFST: 0 // MSB First
// CRCCR.CRCLTE: 0 // CRC big endian
// CRCCR.CRCLSF: 0 // CRC MSB First
// CRCCR.FXOR:   0 // CRC Final XOR off
//****************************************

//
// Example 1-1 (Byte-unit writing)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789"
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 1-2 (CRC check)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);
B_WRITE (CRCIN, 0x29); // <-- CRC
B_WRITE (CRCIN, 0xB1); // <-- CRC

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

(The following is assumed)

B_WRITE -- Byte writing
H_WRITE -- Half-word writing
W_WRITE -- Word writing

B_READ -- Byte reading
H_READ -- Half-word reading
W_READ -- Word reading

CRCCR -- Control register address
CRCINIT -- Initial value register address
CRCIN -- Input data register address
CRCR -- Current CRC register address



- Bytes and half words can be written into any position. In this example, data is written into +0 position continuously.

- When CRC16 is used, the CRC result is output in bit15 to bit0 for big-endian byte order and thus the address for H_READ8 (Half-word reading) is +2 in the example.

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 5-3 Example 2

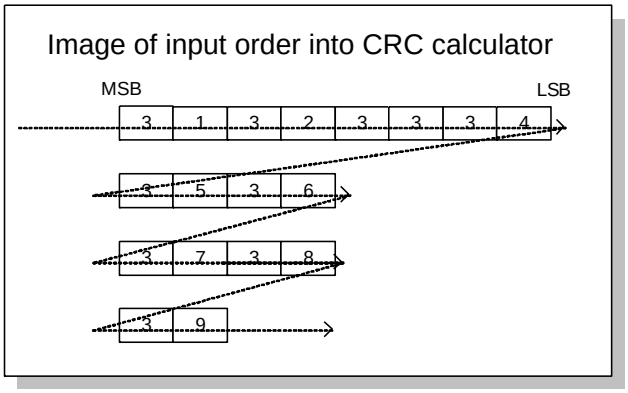
```
//************************************
// CRC16 (CRC ITU-T)
// polynomial:    0x1021
// initial value: 0xFFFF
// CRCCR.CRC32    0 // CRC16
// CRCCR.LTLEND:  0 // big endian
// CRCCR.LSBFST:  0 // MSB First
// CRCCR.CRCLTE:  0 // CRC big endian
// CRCCR.CRCLSF:  0 // CRC MSB First
// CRCCR.FXOR:    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.

5.6.3. Example 3 CRC32, Byte Order, Big-endian

Example 3 CRC32, the byte order and big-endian are shown below.

Figure 5-4 Example 3

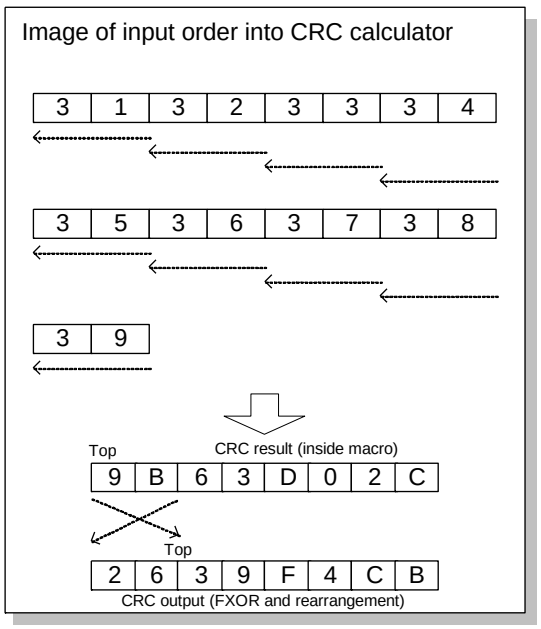
```
//**********************************
// CRC32 (IEEE-802.3)
// polynomial: 0x04C11DB7
// initial value: 0xFFFF_FFFF
// CRCCR.CRC32 1 // CRC32
// CRCCR.LTLEND: 0 // big endian
// CRCCR.LSBFST: 1 // LSB First
// CRCCR.CRCLTE: 0 // CRC big endian
// CRCCR.CRCLSF: 1 // CRC LSB First
// CRCCR.FXOR: 1 // 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.

5.6.4. Example 4 CRC32, Byte Order, Little-endian

Example 4 CRC32, the byte order and Little-endian are shown below.

Figure 5-5 Example 4

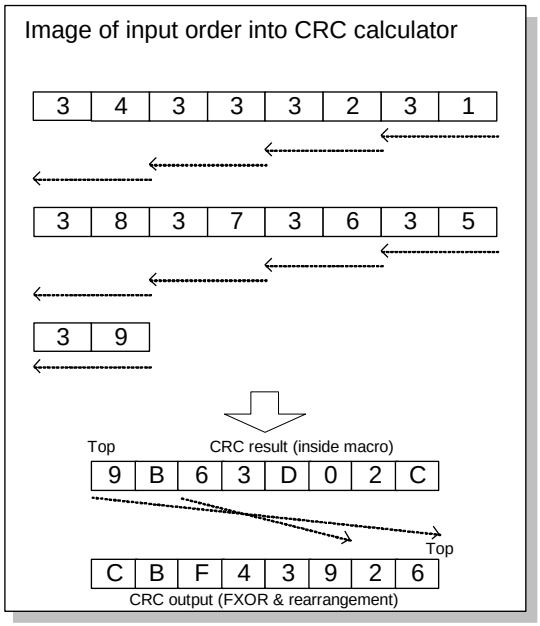
```
//*************************************
// CRC32 (IEEE-802.3)
// polynomial: 0x04C11DB7
// initial value: 0xFFFF_FFFF
// CRCCR.CRC32 1 // CRC32
// CRCCR.LTLEND: 1 // little endian
// CRCCR.LSBFST: 1 // LSB First
// CRCCR.CRCLTE: 1 // CRC little endian
// CRCCR.CRCLSF: 1 // CRC LSB First
// CRCCR.FXOR: 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.

Chapter 39: RAMECC

This chapter explains the RAMECC function.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS28_ECC-3v1-91520-2-E

1. Overview

This section gives an overview of the RAMECC.

The function of the RAMECC is to correct a single-bit error and to detect a double-bit error for those data read from or written to RAM.

2. Features

This section explains features of the RAMECC.

■ Target RAM

- XBS RAM
  - MB91F522: 48K bytes
  - MB91F523: 48K bytes
  - MB91F524: 64K bytes
  - MB91F525: 96K bytes
  - MB91F526: 128K bytes

- Backup-RAM
  - 8K bytes

■ RAMECC Function

Errors up to double bits are detected for those data read from or written to RAM.
Moreover, if a single-bit error is detected it will be corrected.

■ Interrupt Function

A double bit error is detected and RAM double bit error interrupt signal is generated.

■ Test Mode

A false error occurs for the software debugging.

3. Configuration

This section shows the configuration of the RAMECC.

Figure 3-1 Block Diagram of XBS RAM ECC Function (Configuration)

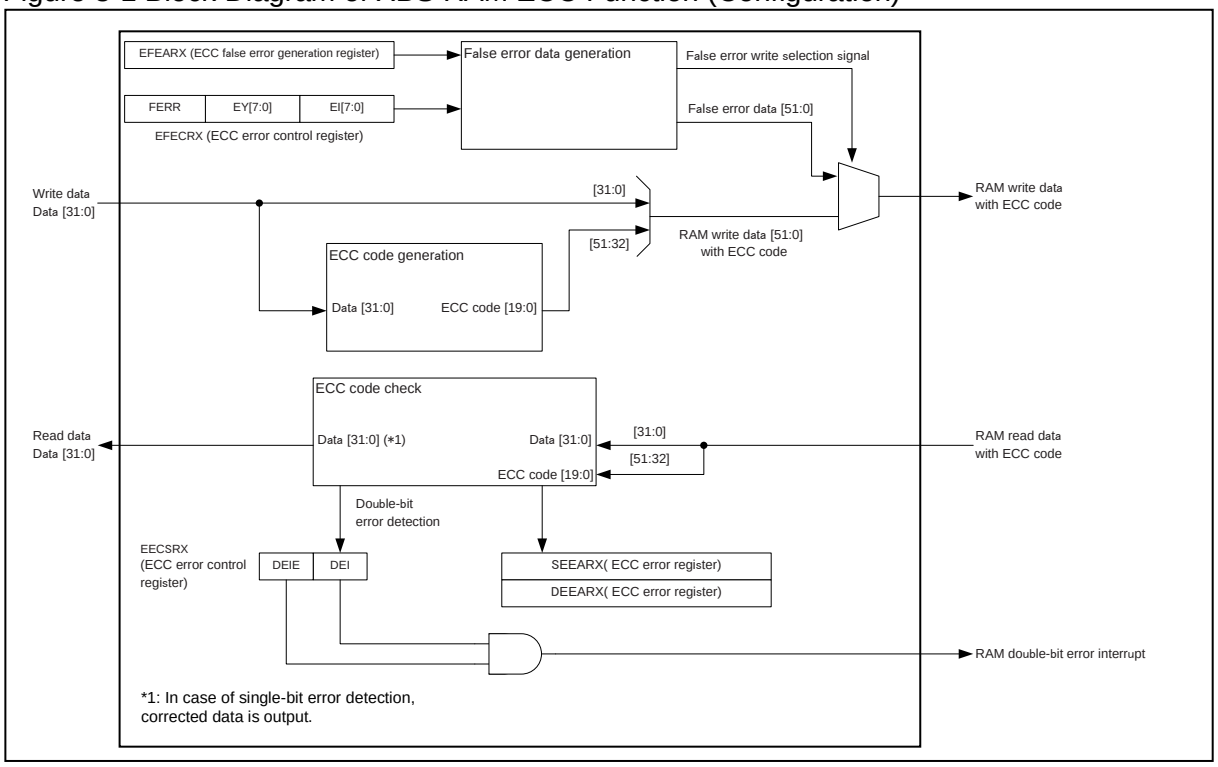
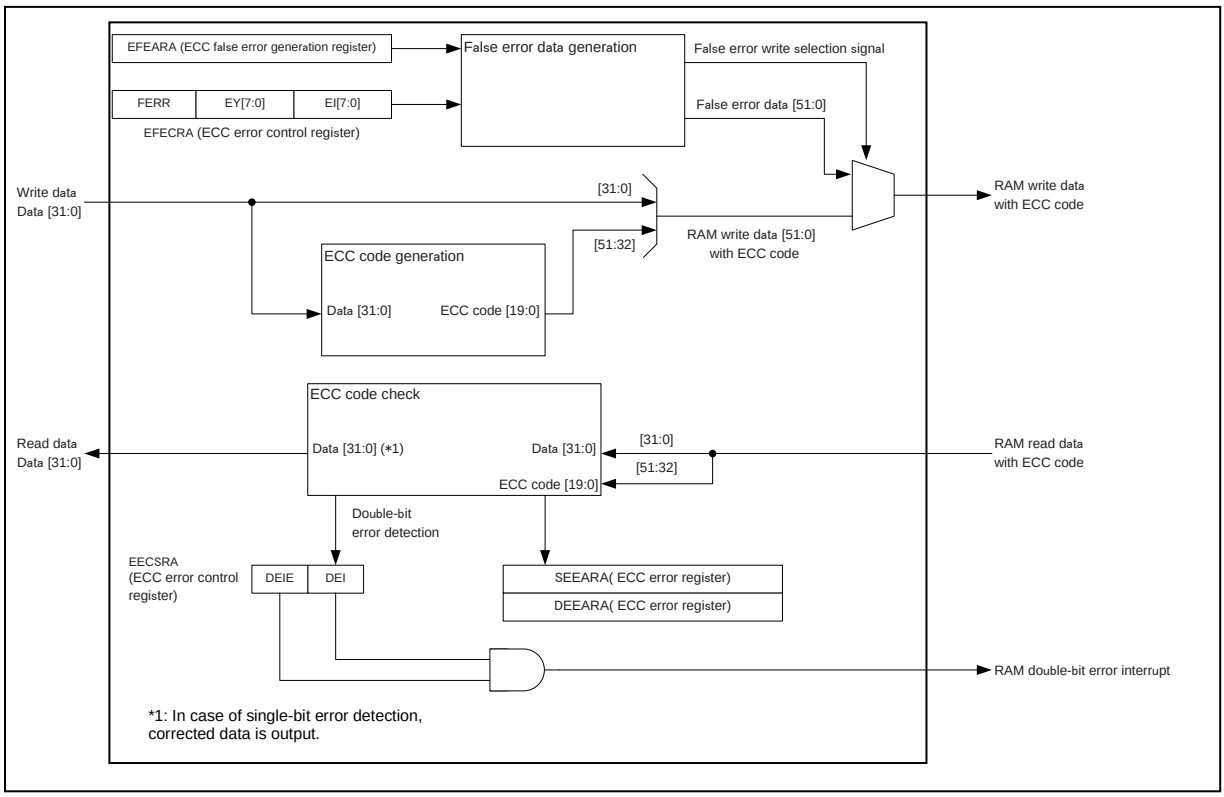


Figure 3-2 Block Diagram of Backup-RAM ECC Function (Configuration)



4. Registers

This section explains the registers of the RAMECC.

Table 4-1 Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x2400	SEEARX		DEEARX		Single-bit ECC error address register XBS RAM Double-bit ECC error address register XBS RAM
0x2404	EECSRX	Reserved	EFEARX		ECC error control register XBS RAM ECC false error generation address register XBS RAM
0x2408	Reserved	EFECRX			ECC false error generation control register XBS RAM
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 generation address register BACKUP-RAM
0x3008	Reserved	EFECRA			ECC false error generation control register BACKUP-RAM

4.1. Single-bit ECC Error Address Register XBS RAM : SEEARX

The bit configuration of single-bit ECC error address register XBS RAM is shown.

When the single-bit error correction is performed during the ECC check of XBS RAM, this register maintains the address at which it occurred.

■ SEEARX : Address 2400_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] D14 to D0 : Single-bit error occurrence address bits

When the single-bit error correction is performed during the ECC check, these bits maintain the address at which it occurred.

If the event above is further detected when a value has already been set to these bits, the original value is maintained without overwriting these bits.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the offset address mentioned above, and then adding the base address of XBS RAM.

(Absolute address) = (0001_0000_H) + (Offset indicated by SEEARX + 2'b00)

4.2. Double-bit ECC Error Address Register XBS RAM : DEEARX

The bit configuration of double-bit ECC error address register XBS RAM is shown.

When the double-bit error detection is performed during the ECC check of XBS RAM, this register maintains the address at which it occurred.

■ DEEARX : Address 2402_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] Reserved.

Always write "0" to this bit.

[bit14 to bit0] D14 to D0 : Double-bit error occurrence address bits

When the double-bit error detection is performed during the ECC check, these bits maintain the address at which it occurred.

If the event above is further detected when a value has already been set to these bits, the original value is maintained without overwriting these bits.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the offset address mentioned above, and then adding the base address of XBS RAM.

(Absolute address) = (0001_0000_H) + (Offset indicated by DEEARX + 2'b00)

4.3. ECC Error Control Register XBS RAM : EECSRX

The bit configuration of ECC error control register XBS RAM is shown.

During the ECC check of XBS RAM, this register maintains the status that indicates whether or not the single-bit error correction or the double-bit error detection has been performed and specifies whether or not to enable interrupts by the double-bit error detection.

■ EECSRX : Address 2404_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				DEIE	DEI	Reserved	SEI
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W0	R(RM1), W0

[bit7 to bit4] Reserved

Always write "0" to these bits.

[bit3] DEIE : Double-bit error factor interrupt enable bit

DEIE	Function
0	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] Reserved

Always write "0" to these bits.

[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.

4.4. ECC False Error Generation Address Register XBS RAM : EFEARX

The bit configuration of the ECC false error generation address register XBS RAM is shown.

The ECC false error generation address register specifies the address where a false error of XBS RAM is generated.

■ EFEARX : Address 2406_H (Access : Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] D14 to D0 : False error generation address setting bits

These bits set the address where false ECC error of XBS RAM is caused.

When EFECRX.FERR=1, write access to this address is generated, and an ECC error is generated by intentionally including an error in the data to be written, according to the settings of EFECRX.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the base address, and then adding the base address of XBS RAM.

(Absolute address) = (0001_0000_H) + (Offset set by EFEARX + 2'b00)

4.5. ECC False Error Generation Control Register XBS RAM : EFECRX

The bit configuration of the ECC false error generation control register XBS RAM is shown.

The ECC false error generation control register (EFECRX) specifies each false error by its byte position and its bit position where the false error is generated.

■ EFECRX : Address 2409_H (Access : Byte, Half-word, Word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							FERR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EY7	EY6	EY5	EY4	EY3	EY2	EY1	EY0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit23 to bit17] Reserved

Always write "0" to these bits.

[bit16] FERR : False error generation enable bit

FERR	Function
0	False ECC error generation disable
1	False ECC error generation enable

This bit enables a false ECC error for XBS RAM.

"0": Prohibits a false ECC error. (Normal operation)
Also, writing "0" from software is ignored.

"1": Enables a false ECC error.

When this bit is set to "1", the following operation sequence is automatically performed.

1. Start writing data including an intentional error in the address specified by EFEARX following EY7 to EY0, EI7 to EI0.
2. Read the same address and detect ECC error.
3. Clear this bit to "0".

[bit15 to bit8] EY7 to EY0 : False error generation byte setting bits

EY7 to EY0	Target byte on RAM
EY0	RAM data[7:0]
EY1	RAM data[15:8]
EY2	RAM data[23:16]
EY3	RAM data[31:24]
EY4	RAM data[36:32]
EY5	RAM data[41:37]
EY6	RAM data[46:42]
EY7	RAM data[51:47]

These bits specify the byte position of the target that causes false ECC error for XBS RAM.
For example, when EY2 is filled with "1" and other false error generation byte setting bits are filled with "0", the target byte where a false error is generated is RAM data[23:16] only. In other bytes on the RAM, no false error is generated.
In addition to the foregoing, when both EY2 and EY3 are filled with "1" and others are filled with "0", the target byte where a false error is generated is RAM data[31:16].

[bit7 to bit0] EI7 to EI0 : False error generation bit setting bits

EI7 to EI0	Target bit on byte
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
EI7	[7]

These bits specify the bit position of the target that causes false ECC error for XBS RAM.
For example, when both EY2 and EI4 are filled with "1", and others are filled with "0", the target bit where a false error is generated is RAM data[20]. As a result, a single bit error can be corrected.
In addition to the foregoing, when EY2, EI4, and EI7 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[23] and RAM data[20]. As a result, a double bit error can be detected.
Moreover, when EY2, EY3, and EI4 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[28] and RAM data[20]. As a result, a single bit error can be corrected in each byte.

Note:

2408_H is reserved bit. The operations in half-word and word-access are as below.
The read value is always "0".
Always write "0".

4.6. Single-bit ECC Error Address Register BACKUP-RAM : SEEARA

The bit configuration of the single-bit ECC error address register BACKUP-RAM is shown.

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
Always write "0" to these bits.

[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 are 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) = (0000_4000_H) + (Offset indicated by SEEARA + 2'b00)

4.7. Double-bit ECC Error Address Register BACKUP-RAM : DEEARA

The bit configuration of the double-bit ECC error address register BACKUP-RAM is shown.

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

Always write "0" to these bits.

[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) = (0000_4000_H) + (Offset indicated by DEEARA + 2'b00)

4.8. ECC Error Control Register BACKUP-RAM : EECSRA

The bit configuration of the ECC error control register BACKUP-RAM is shown.

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 the double-bit error detection.

■ EECSRA : Address 3004_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				DEIE	DEI	Reserved	SEI
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W0	R(RM1), W0

[bit7 to bit4] Reserved

Always write "0" to these bits

[bit3] DEIE : Double-bit error factor interrupt enable bit

DEIE	Function
0	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] Reserved

Always write "0" to these bits.

[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.

4.9. ECC False Error Generation Address Register BACKUP-RAM : EFEARA

The bit configuration of the ECC false error generation address register BACKUP-RAM is shown.

The ECC false error generation address register specifies the address where a false error of Backup-RAM 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" to these bits.

[bit10 to bit0] D10 to D0 : False error generation address setting bits

These bits set the address where false ECC error of Backup-RAM is caused.

When EFECRA.FERR=1, write access to this address is generated, and an ECC error is generated by intentionally including an error in the data to be written, according to the settings of EFECRA.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the base address, and then adding the base address of Backup-RAM.

(Absolute address) = (0000_4000_H) + (Offset set by EFEARA + 2'b00)

4.10. ECC False Error Generation Control Register BACKUP-RAM : EFECRA

The bit configuration of the ECC false error generation control register BACKUP-RAM is shown.

The ECC false 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" to these bits.

[bit16] FERR : False error generation enable bit

FERR	Function
0	False ECC error generation disable
1	False ECC error generation enable

This bit enables a false ECC error for Backup-RAM.

"0": Prohibits a false ECC error. (normal operation)
Also, writing "0" from software is ignored.

"1": Enables a false ECC error.

When this bit is set to "1", the following operation sequence is automatically performed.

1. Start writing data including an intentional error in the address specified by EFEARA following EY7 to EY0, EI7 to EI0.
2. Read the same address and detect ECC error.
3. Clear this bit to "0".

[bit15 to bit8] EY7 to EY0 : False error generation byte setting bits

EY7 to EY0	Target byte on RAM
EY0	RAM data[7:0]
EY1	RAM data[15:8]
EY2	RAM data[23:16]
EY3	RAM data[31:24]
EY4	RAM data[36:32]
EY5	RAM data[41:37]
EY6	RAM data[46:42]
EY7	RAM data[51:47]

These bits specify the byte position of the target that causes false ECC error for Backup-RAM.
For example, when EY2 is filled with "1" and other false error generation byte setting bits are filled with "0", the target byte where a false error is generated is RAM data[23:16] only. In other bytes on the RAM, no false error is generated.
In addition to the foregoing, when both EY2 and EY3 are filled with "1" and others are filled with "0", the target byte where a false error is generated is RAM data[31:16].

[bit7 to bit0] EI7 to EI0 : False error generation bit setting bits

EI7 to EI0	Target bit on byte
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
EI7	[7]

These bits specify bit position of the target that causes false ECC error for Backup-RAM.
For example, when both EY2 and EI4 are filled with "1", and others are filled with "0", the target bit where a false error is generated is RAM data[20]. As a result, a single bit error can be corrected.
In addition to the foregoing, when EY2, EI4, and EI7 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[23] and RAM data[20]. As a result, a double bit error can be detected.
Moreover, when EY2, EY3, and EI4 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[28] and RAM data[20]. As a result, a single bit error can be corrected in each byte.

Note:

3008_H is reserved bit. The operations in half-word and word-access are as below.
The read value is always "0".
Always write "0".

5. Operation

This section explains operations.

5.1 RAMECC Function

5.2 Interrupt-related Register

5.3 Test Mode

5.4 Note

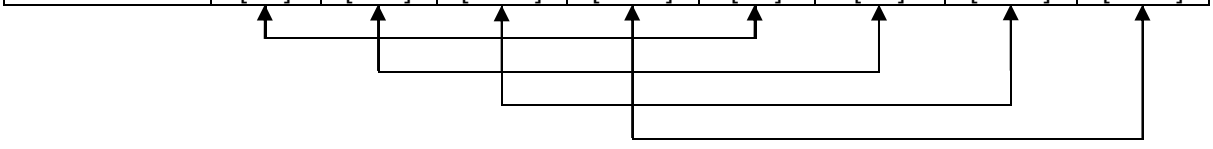
5.1. RAMECC Function

The RAMECC function is explained.

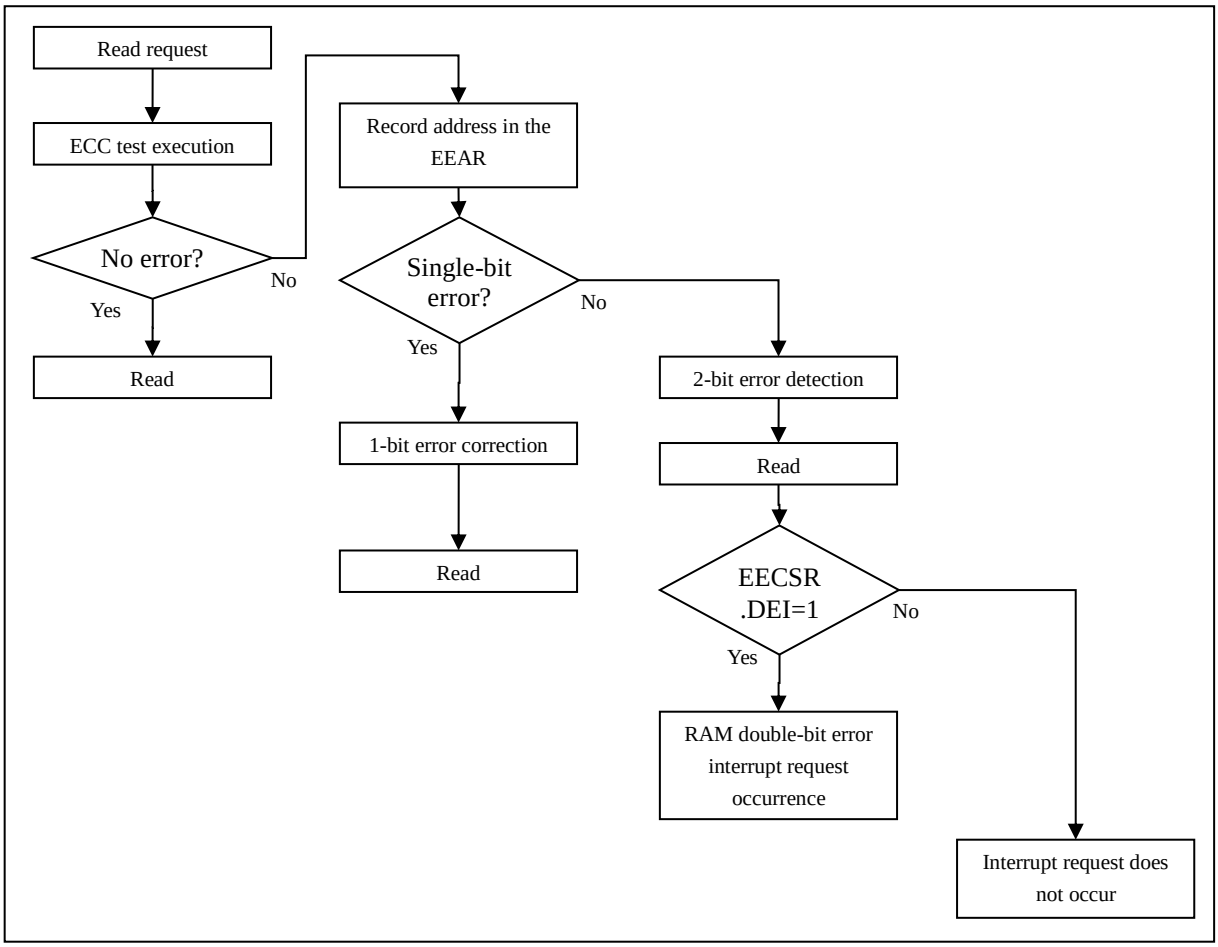
The RAMECC always functions (It, however, stops at RAM diagnosis). When an error is detected, the address where the error occurred is hold in the register EEAR. If another error is detected while the previous address is being held in the register EEAR, the EEAR register will not be overwritten. Thus the previous address is maintained.

ECC code matrix records redundant five bits as ECC code by byte units.

RAM_DATA [51:0]	[7:0]	[15:8]	[23:16]	[31:24]	[36:32]	[41:37]	[46:42]	[51:47]
DATA[31:0]	DATA [7:0]	DATA [15:8]	DATA [23:16]	DATA [31:24]	ECC [4:0]	ECC [9:5]	ECC [14:10]	ECC [19:15]



Flow chart of the operation is shown below.



5.2. Interrupt-related Register

This section explains the interrupt-related register.

Write "1" in the DEIE bit according to the usage in order to generate the interrupt, and set the RAMECC interrupt vector.

Interrupt factor	Interrupt vector	Interrupt level
DEI (RAM double-bit error interrupt)	#15($000FFFC0_H$)	15(F_H) Fixed

See "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)" for details of the interrupt level and the interrupt vector.
Since the interrupt request flag (DEI) is not automatically cleared, clear the flag forcibly with software before returning from the interrupt processing. (Write "0" into the DEI bit).

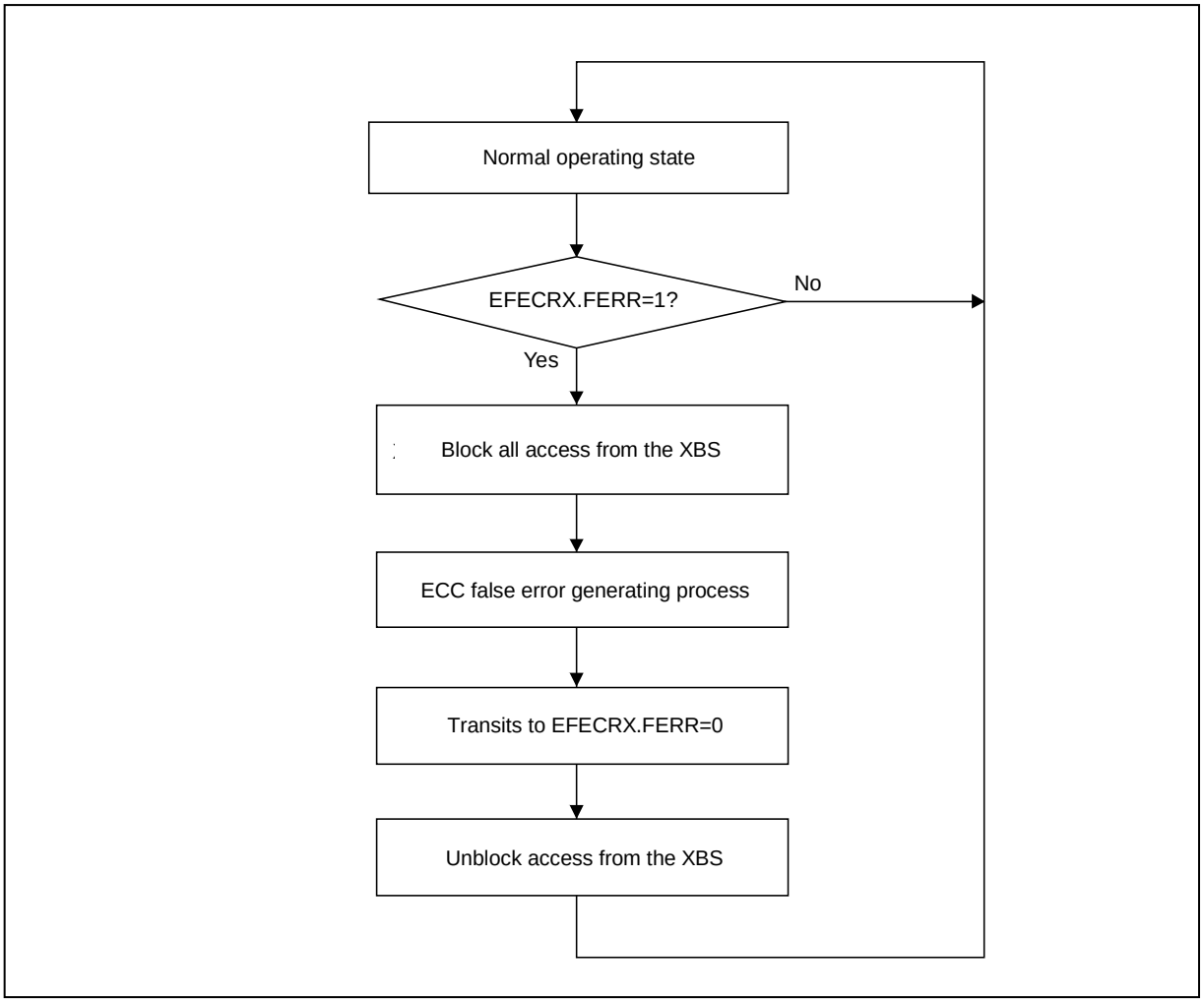
5.3. Test Mode

This section explains the test mode.

In this mode an ECC false error is generated in order to debug software.
The ECC false error of XBS RAM, as well as Backup-RAM, is generated in accordance with the following procedures:

1. Specify the address where a false error is generated to the ESS false error generation address register (EFEARX).
2. Set bytes and bits in ECC false error generation control register XBS RAM (EFECRX).
   (a) Specify the byte position where a false error is generated to EFECRX.EY[7:0].
   (b) Specify the bit position where a false error is generated to EFECRX.EI[7:0].
3. Write "1" to the FERR bit of the ECC false error generation control register XBS RAM (EFECRX).

Those data including errors intentionally are written to the address specified with EFEARX, where 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.
The operation after "1" is written to the FERR bit will be performed automatically.



5.4. Note

This section explains note.

A single bit error might be detected at the fault detection in three bits or more, and the correction operation not intended start.

If XBS RAM is accessed while the ECC false error (a pseudo ECC error) of XBS RAM is generated, the access is blocked. Therefore, access XBS RAM after completing the process of the ECC false error (a pseudo ECC error). (as well as Backup RAM)

Chapter 40: Multi-Function Serial Interface

This chapter explains the multi-function serial interface.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation of UART
6. Operation of CSIO
7. Operation of LIN Interface (v2.1)
8. Operation of I^2C

Code : FIP002-2v4-91520-2-E

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 Interface (v2.1) (LIN Communication Control Interface (v2.1)) and I^2C serial communication function.

2. Features

This section explains features of the multi-function serial interface.

This product is equipped with 12-channel multi-function serial interface communication module. To use this device, you will select UART, CSIO, LIN Interface (v2.1), or I^2C using the serial mode register (SMR).

Moreover, ch.6, ch.8, ch.9, and ch.11 are 5V tolerant input.

The numbers of available channels are shown below.

MB91F52xB (64pin) : 8

MB91F52xD (80pin) : 9

MB91F52xF (100pin) : 12

MB91F52xJ (120pin) : 12

MB91F52xK (144pin) : 12

MB91F52xL (176pin) : 12

For available external signals, see Table 4-1, Table 4-2, Table 4-3, Table 4-4, and Table 4-5.

Note:

I^2C supports only relocation 0 of ch.3 to ch.8, ch.10, and ch.11.

■ UART

UART (asynchronous serial interface) is the general-purpose serial data communication interface designed to communicate with external devices asynchronously (start-stop synchronization). It supports bidirectional communication function (normal mode), master/slave type communication function (multi-processor mode: both master and slave are supported). It is also equipped with FIFO for transmission/reception.

Item	Function
Data	· Full-duplex double buffering (when FIFO is unused) · Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling to the bus clock for three times and determine the reception value by the majority of the sampling value.
Transfer format	Asynchronous
Baud rate	· Dedicated baud rate generator (comprising 15-bit reload counter) · External clock input can be adjusted by the reload counter
Data length	· 5 to 9 bits (normal mode), 7, 8 bits (multi-processor mode)
Signaling system	NRZ (Non Return to Zero), Inverted NRZ
Start bit detection	· Synchronize with the start bit falling edge (NRZ system) · Synchronize with the start bit rising edge (inverted NRZ system)
Reception error detection	· Framing error · Overrun error · Parity error*
Timer feature	· Employs 16-bit serial timer · Dividing ratio of operating clock is selectable (1/1 to 1/256)
Interrupt request	· Reception interrupt (Reception completed, framing error, overrun error, parity error*) · Transmission interrupt (transmission data empty, transmission bus idle) · Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) · Both transmission and reception employ DMA function · Status interrupt (Serial timer interrupt)
Master/slave mode communication function (multi-processor mode)	1 (Master)-to-n (slave) communication is supported (both master and slave systems are supported)
FIFO option	· Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) · Transmission FIFO and reception FIFO can be selected · Transmission data can be retransmitted · Reception FIFO interrupt timing can be modified by software · FIFO reset is supported independently
DMA transfer support	Transmission: Supported Reception: Supported Status: Not supported

*: Parity error is for the normal mode only.

■ CSIO

CSIO (Clock Synchronous Serial Interface) is a general-purpose serial data communication interface for synchronous communication with external devices (SPI supported). It is also equipped with the FIFO for transmission/reception (64 bytes each).

Item	Function
Data buffer	· Full-duplex double buffering (when FIFO is unused) · Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Transfer format	· Clock synchronous (without start bit/stop bit) · Master/slave function · SPI supported (both master/slave mode supported)
Baud rate	· Dedicated baud rate generator provided (comprising 15-bit reload counter, master mode) · An external clock can be entered. (Slave operation)
Data length	Can be changed to 5 to 16, 20, 24, 32 bits
Reception error detection	Overrun error
Interrupt request	· Reception interrupt (reception completed, overrun error) · Transmission interrupt (transmission data empty, transmission bus idle, chip error interrupt) · Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) · Both transmission and reception employ DMA function * · Status interrupt (Serial timer interrupt)
Serial chip select	· Ch.0: Serial chip select function invalid · Ch.1,2,3,8,9,10,11: 1-ch control (single control) · Ch.4,5,6,7: 4-ch control (single control, round control) · Variable setup/hold/deselect times can be set · Active level can be selected for each channel
Synchronous transmission feature	· Synchronizes serial timer and is capable of automatic data transmission periodically
Timer feature	· Employs 16-bit serial timer · Dividing ratio of operating clock is selectable (1/1 to 1/256)
Synchronous mode	Master or slave function
Pin access	Serial data output pin can be set to "1"
FIFO option	· Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) · Transmission FIFO and reception FIFO can be selected · Transmission data can be retransmitted · Reception FIFO interrupt timing can be modified by software · FIFO reset is supported independently
DMA transfer support	Transmission: Supported Reception: Supported Status: Not supported

*: When the access size of transmit/receive data is 16 bit and FIFO is not used, continuous transfer cannot be performed.
To perform continuous transfer, set the access width of transmit/receive data to 32-bit access (SSR:AWC=1) or use FIFO.

■ LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))

● Manual Mode

LIN interface (v2.1) (LIN Communication Control Interface (v2.1)) provides functions to support LIN bus. It is also equipped with the FIFO for transmission/reception (64 bytes each).

Item	Function
Data buffer	· Full-duplex double buffering (when FIFO is unused) · Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling for three times by the bus clock and determine the reception value by the majority of the sampling value.
Transfer mode	· Asynchronous
Baud rate	· Dedicated baud rate generator provided (comprising of 15-bit reload counter) · External clock input can be adjusted by the reload counter · Automatic baud rate adjustment with Sync Field reception
Data length	8 bits
Signaling system	NRZ (Non Return to Zero)
Start Bit Detection	Synchronize with the start bit falling edge
Reception error detection	· Framing error · Overrun error
Interrupt request	· Reception interrupt (Reception completed, framing error, overrun error) · Transmission interrupt (transmission data empty, transmission bus idle) · Status interrupt (LIN Break field detection, serial timer interrupt) · Interrupt request for ICU (LIN synch field detected: LSYN) · Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) · Both transmission and reception employ DMA function
Timer feature	· Employs 16-bit serial timer · Dividing ratio of operating clock is selectable (1/1 to 1/256)
LIN bus option	· LIN protocol revision 2.1 is supported. · Master device operation · Slave device operation · LIN Break field generation (can be changed to 13 to 16 bits) · LIN Break Delimiter generation (can be changed to 1 to 4 bits) · LIN Break field detection · Detection of start/stop edges for LIN synch field connected to input capture by input capture (See "CHAPTER: INPUT CAPTURE".)
FIFO option	· Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) · Transmission FIFO and reception FIFO can be selected · Transmission data can be retransmitted · Reception FIFO interrupt timing can be modified by software · FIFO reset is supported independently
DMA transfer support	· Transmission: Supported · Reception: Supported · Status: Not supported

Assist Mode

LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) provides functions to support LIN bus. Automatic transmission/automatic detection of the header in the LIN communication is possible. It is also equipped with the FIFO for transmission/reception (64 bytes each).

Note: This supports master mode only. This cannot be used in slave mode.

Item	Function
Data buffer	· Full-duplex double buffering (when FIFO is unused) · Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling for three times by the bus clock and determine the reception value by the majority of the sampling value.
Transfer mode	Asynchronous
Baud rate	· Dedicated baud rate generator provided (comprising of 15-bit reload counter) · External clock input can be adjusted by the reload counter · Automatic baud rate adjustment with Sync Field reception
Data length	9 bits
Signaling system	NRZ (Non Return to Zero)
Start Bit Detection	Synchronize with the start bit falling edge
Reception error detection	<The error is detected by the self-check of the transmission side> · LIN bus error <The error is detected by the self-check of the transmission side and by the reception side > · Framing error · Overrun error · LIN ID parity error · LIN checksum error <The error is detected by the reception side of the automatic baud rate adjustment prohibition> · LIN Sync Data error
Interrupt request	· Transmission interrupt (1) Data transmission interrupt (Transmission data empty, transmission bus idle) (2) Transmission FIFO interrupt (When transmission FIFO is the interrupt threshold or less, or transmission FIFO is empty). · Reception interrupt (1) Data reception interrupt (Reception completed) (2) Reception FIFO interrupt (When reception FIFO is the interrupt threshold or more). (3) Various error interrupts (LIN bus error, LIN ID parity error, LIN Sync Data error, framing error, overrun error, and LIN checksum error) · Status interrupt (1) Automatic header completion interrupt (2) Sync Field detection interrupt (3) Checksum arithmetic operation completion interrupt · Both transmission and reception employ DMA function
Timer feature	· Employs 16-bit serial timer · Dividing ratio of operating clock is selectable (1/1 to 1/256)

Item	Function
LIN bus option	· LIN protocol revision 2.1 supported · Automatic transmission/reception of master/slave device headers (1) LIN Break Field generation (can be changed to 13 to 20-bit lengths) (2) LIN Break Delimiter generation (can be changed to 1 to 4-bit lengths) (3) Sync Field automatic generation and automatic check of data value (0x55) (4) ID Field parity value automatic generation and check (5) Check sum automatic generation/check (standard/extension supported) · Detection of start/stop edges for LIN sync Field connected to input capture by input capture (See "CHAPTER : INPUT CAPTURE")
FIFO option	· Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) · Transmission FIFO and reception FIFO can be selected · Transmission data can be retransmitted · Reception FIFO interrupt timing can be modified by software · FIFO reset is supported independently
LIN communication test function	· Serial communication test function · Pseudo trouble generation function (LIN bus error, LIN ID parity error, LIN checksum error, LIN Sync Data error, and framing error)

■ I^2C

I^2C interface (I^2C Communication Control Interface) supports I^2C bus, and runs as a master/slave device on the I^2C bus. It is also equipped with the FIFO for transmission/reception (64 bytes each).

Item	Function
Data buffer	· Full duplex-double buffer (FIFO is unused) · Transmission/reception FIFO (16 bytes each) (when FIFO is used)
Serial input	Eliminate noise up to two clocks using bus clock for serial clock or serial data input.
Transfer mode	Synchronization
Baud rate	· Dedicated baud rate generator provided (comprising 15-bit reload counter) · External clock input is adjustable with the reload counter.
Data length	8-bit
Signaling system	NRZ (Non Return to Zero)
Interrupt request	· Reception interrupt · Transmission interrupt · Status interrupt (INT interrupt, serial timer interrupt) · Interrupt request for ICU · Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) · Both transmission and reception employ DMA function

Item	Function
I^2C	· 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
Timer feature	· Employs 16-bit serial timer · Dividing ratio of operating clock is selectable (1/1 to 1/256)
FIFO option	· Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) · Transmission FIFO and reception FIFO can be selected · Transmission data can be retransmitted · Reception FIFO interrupt timing can be modified by software · FIFO reset is supported independently
DMA transfer support	· Transmission: Supported · Reception: Not supported · Status: Not supported

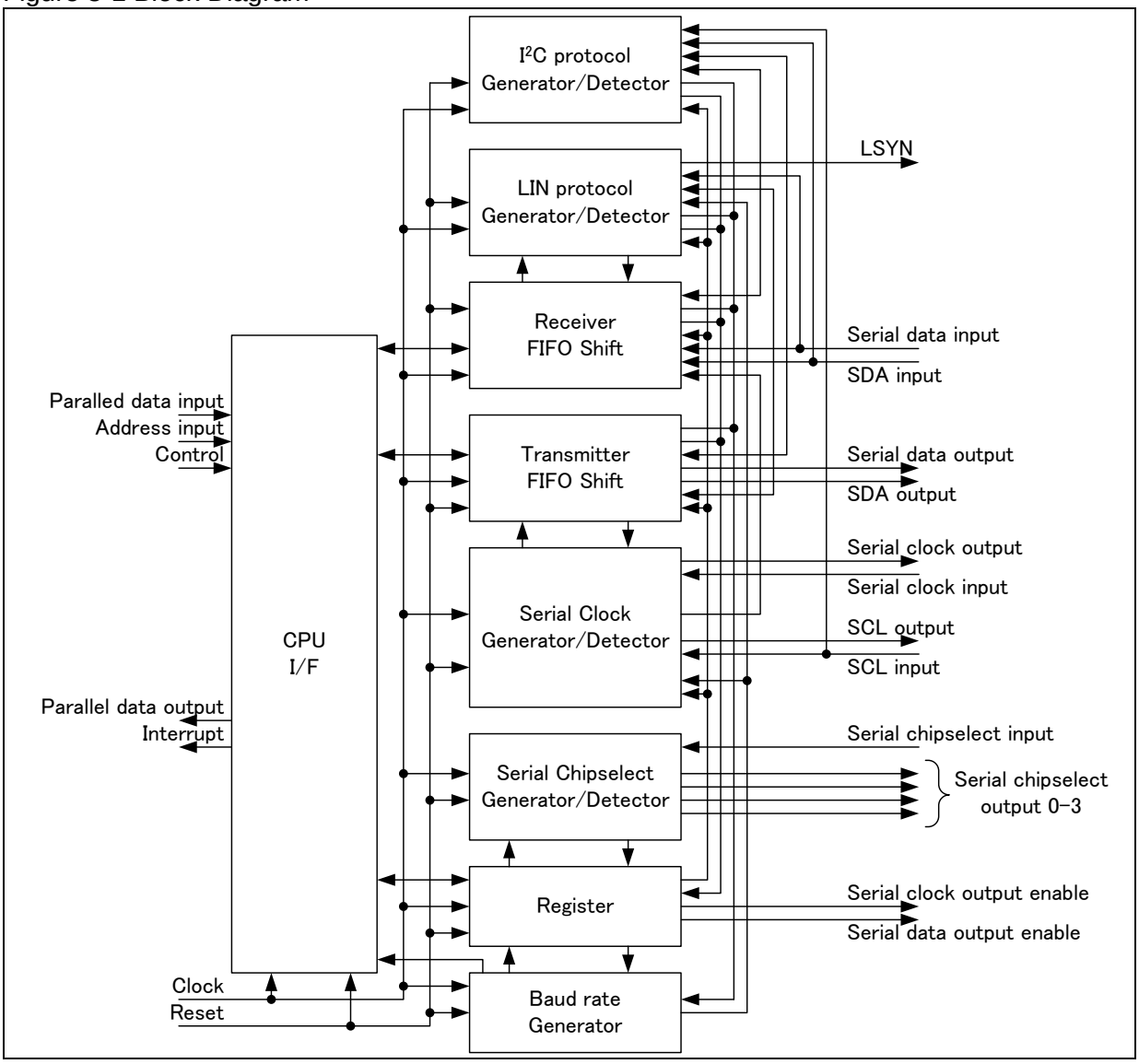
Note:

I^2C supports only ch.3 to ch.8, ch.10, and ch.11.

3. Configuration

This section explains configuration of the multi-function serial interface.

Figure 3-1 Block Diagram



4. Registers

This section explains registers of the multi-function serial interface.

■ Table of Base Addresses (Base_addr) and External Pins

Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)

Channels	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_1	SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	none	none	none	none	none	none	none
2	0x17A0	none	none	none	none	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0	none	none	none
4	0x17F0	SCK4_2	SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	none	SCS42_0	SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	none
6	0x1840	SCK6_0	SOT6_0	SIN6_0	none	none	none	SCS63_0
7	0x1868	none	none	none	none	none	none	none
8	0x1890	none	none	none	none	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	none	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	none	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5, ch.6, and ch.11 can be used as I^2C (standard mode support).)

Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	none	none	none	none	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0/	none	none	none
4	0x17F0	SCK4_1/ SCK4_2	SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0	SCS42_0/ SCS42_1	SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	none
6	0x1840	SCK6_0	SOT6_0	SIN6_0	none	SCS61_0	none	SCS63_0
7	0x1868	none	none	none	none	none	none	none
8	0x1890	none	none	none	none	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	none	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5, ch.6, and ch.11 can be used as I^2C (standard mode support).)

Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_1	SOT1_0	SIN1_0	SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0/	none	none	none
4	0x17F0	SCK4_1/ SCK4_2	SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5 to ch.8, and ch.11 can be used as I^2C (standard mode support).)

Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_0/ SCK1_1	SOT1_0	SIN1_0	SCS1_0/ SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_0/ SCK3_1/ SCK3_2	SOT3_0/ SOT3_1/ SOT3_2	SIN3_0/ SIN3_1	SCS3_0/ SCS3_1	none	none	none
4	0x17F0	SCK4_0/ SCK4_1/ SCK4_2	SOT4_0/ SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.3 and ch.4 can be used as I^2C (fast mode/standard mode support). Ch.5 to ch.8, and ch.11 can be used as I^2C (standard mode support).)

Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_0/ SCK1_1	SOT1_0	SIN1_0	SCS1_0/ SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_0/ SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_0/ SCK3_1/ SCK3_2	SOT3_0/ SOT3_1/ SOT3_2	SIN3_0/ SIN3_1	SCS3_0/ SCS3_1	none	none	none
4	0x17F0	SCK4_0/ SCK4_1/ SCK4_2	SOT4_0/ SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	SCS50_0	SCS51_0	SCS52_0	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_0/ SCK10_1	SOT10_0/ SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.3 and ch.4 can be used as I^2C (fast mode/standard mode support). Ch.5 to ch.8, ch.10, and ch.11 can be used as I^2C (standard mode support).)

■ Registers Map

Table 4-6 Registers Map

Address	Registers				Registers function
	+0	+1	+2	+3	
Base+ 00_H	[UART] SCRn [CSIO] SCRn [LIN] SCRn [I^2C] IBCRn	[Common] SMRn	[UART] SSRn [CSIO] SSRn [LIN] SSRn [I^2C] SSRn	[UART] ESCRn [CSIO] ESCRn [LIN] ESCRn [I^2C] IBSRn	[UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [I^2C] I^2C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [I^2C] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register [I^2C] I^2C Bus status register
Base+ 04_H	[UART] Reserved [CSIO] RDR1n/TDR1n [LIN] Reserved [I^2C] Reserved		[UART] RDR0n/TDR0n [CSIO] RDR0n/TDR0n [LIN] RDR0n/TDR0n [I^2C] RDR0n/TDR0n		[CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register [I^2C] Transmission/receive data register
Base+ 08_H	[UART] SACSRn [CSIO] SACSRn [LIN] SACSRn [I^2C] SACSRn		[UART] STMRn [CSIO] STMRn [LIN] STMRn [I^2C] STMRn		[UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [I^2C] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register [I^2C] Serial timer register
Base+ $0C_H$	[UART] STMCRn [CSIO] STMCRn [LIN] STMCRn [I^2C] STMCRn		[UART] Reserved [CSIO] SCSCRn [LIN] SFURn [I^2C] Reserved		[UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [I^2C] Serial timer compare register [CSIO] Serial chip select control register [LIN] Sync field upper limit register
Base+ 10_H	[UART] Reserved [CSIO] SCSTR3n [LIN] LAMSRn [I^2C] Reserved	[UART] Reserved [CSIO] SCSTR2n [LIN] LAMCRn [I^2C] Reserved	[UART] Reserved [CSIO] SCSTR1n [LIN] SFLR1n [I^2C] Reserved	[UART] Reserved [CSIO] SCSTR0n [LIN] SFLR0n [I^2C] Reserved	[CSIO] Serial chip select timing register [LIN] LIN assist mode status register [LIN] LIN assist mode control register [LIN] Sync field lower limit register

Address	Registers				Registers function
	+0	+1	+2	+3	
Base+ 14_H	Reserved	[UART] Reserved [CSIO] SCSFR2n [LIN] Reserved [I^2C] Reserved	[UART] Reserved [CSIO] SCSFR1n [LIN] Reserved [I^2C] Reserved	[UART] Reserved [CSIO] SCSFR0n [LIN] Reserved [I^2C] Reserved	[CSIO] Serial chip select format register
Base+ 18_H	[UART] Reserved [CSIO] TBYTE3n [LIN] LAMESRn [I^2C] Reserved	[UART] Reserved [CSIO] TBYTE2n [LIN] LAMERTn [I^2C] Reserved	[UART] Reserved [CSIO] TBYTE1n [LIN] LAMIERn [I^2C] Reserved	[UART] Reserved [CSIO] TBYTE0n [LIN] LAMTIDn/ LAMRIDn [I^2C] Reserved	[CSIO] Transfer byte register [LIN] LIN assist mode error status register [LIN] LIN assist mode trouble examination register [LIN] LIN assist mode interrupt enable register [LIN] LIN assist mode transmission/reception ID register
Base+ $1C_H$	[UART] BGRn [CSIO] BGRn [LIN] BGRn [I^2C] BGRn		[UART] Reserved [CSIO] Reserved [LIN] Reserved [I^2C] ISMKn	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I^2C] ISBAn	[UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register [I^2C] Baud rate generator register [I^2C] 7-bit slave address mask register [I^2C] 7-bit slave address register
Base+ 20_H	[Common] FCR1n	[Common] FCR0n	[Common] FBYTEn		[Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register
Base+ 24_H	[Common] FTICRn		Reserved		[Common] Transmission FIFO interrupt control register (FTICR)

4.1. Common Registers

Common registers are shown.

4.1.1. Serial Mode Register: SMR

This section explains the bit structure of the serial mode register.

This register selects the serial communication method (UART or I^2C). Bits 3 to 0 change their function according to the method selected (UART, CSIO, or I^2C).

■ SMRn (n= 0 to 11): Address Base addr + 01_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
MD[2:0]			Reserved	SBL/ SCINV/ RIE	BDS/TIE	SCKE/ (Reserved)	SOE/ (Reserved)	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W0	R/W	R/W (R/W0)	R/W (R/W0)	R/W (R/W0)	Attribute

[bit7 to bit5] MD[2:0] (MoDe): Operation mode

These bits set the communication method.

"000_B": Operating mode 0 (asynchronous normal mode) is set.
"001_B": Operating mode 1 (asynchronous multi-processor mode) is set.
"010_B": Operating mode 2 (CSIO mode) is set.
"011_B": Operating mode 3 (LIN communication mode) is set.
"100_B": Operating mode 4 (I^2C mode) is set.

Notes:

- Settings other than those listed above are prohibited.
- Configure each register after setting the operation mode.
- [UART][CSIO][LIN] Before changing the operation mode, execute programmable clear (SCR:UPCL=1).
- [I^2C] Before changing the operation mode, disable I^2C (ISMK:EN=0).

[bit4] Reserved

Always set this bit to "0".

[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]

This bit configures the bit length of stop bit (frame end mark for transmission data):

When SBL="0" and ESCR.ESBL="0" are set: Stop bit is set to 1 bit.
When SBL="1" and ESCR.ESBL="0" are set: Stop bit is set to 2 bits.
When SBL="0" and ESCR.ESBL="1" are set: Stop bit is set to 3 bits.
When SBL="1" and ESCR.ESBL="1" are set: Stop bit is set to 4 bits.

Notes:

- When receiving, only the first bit of the stop bits will always be detected.
- This bit should be set when transmission is disabled (SCR:TXE=0).

[CSIO]
This bit inverts the serial clock format. When chip select is used in master mode (SCR: MS=0), this bit is used for serial chip select pin 0 communication.

When this bit is set to "0":
- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a rising edge of the serial clock in the normal transfer while it is sampled at a falling edge of the serial clock in the SPI transfer.

When this bit is set to "1":
- Serial clock output mark level is set to "L".
- Transmission data is output in synchronization with a rising edge of the serial clock in the normal transfer while it is output in synchronization with a falling edge of the serial clock in the SPI transfer.
- Reception data is sampled at a falling edge of the serial clock in the normal transfer while it is sampled at a rising edge of the serial clock in the SPI transfer.

Notes:

- Set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).
- Set this bit when serial clock output is disabled (SCKE=0).
- After the SCINV bit is set, set reception enable (SCR: RXE=1).
- This bit is used in one of cases below.
  - When chip select pin is disabled (SCSCR:CSEN3-0="0000"b)
  - While in slave mode (SCR:MS=1)
  - When data format of chip select is disabled (ESCR:CSFE=0)
  - When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

[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:

Set this bit to "0" when receiving data with the use of INT bit of I^2C bus control register (IBCR) while DMA mode is disabled (SSR:DMA=0).

[bit2] BDS/TIE (Bit Direction Select/Transmit Interrupt Enable): Transfer direction selection bit/ transmission interrupt enable bit

[LIN]

LIN must always write "0" to this bit.

[UART][CSIO]

This bit selects whether to transfer the transfer serial data from the least significant bit (LSB-first, BDS=0) or from the most significant bit (MSB-first, BDS=1).

When chip select is used with the master mode (SCR:MS=0), this bit is used for communication with serial chip select pin 0.

Notes:

- Set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).
- [CSIO] This bit is used in one of cases below.
  - When chip select pin is disabled (SCSCR:CSEN3-0="0000"b)
  - While in slave mode (SCR:MS=1)
  - When data format of chip select is disabled (ESCR:CSFE=0)
  - When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

[I^2C]

- 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:

Set this bit to "0" when transmitting data with the use of INT bit of I^2C bus control register (IBCR) while DMA mode is disabled (SSR:DMA=0).

[bit1] SCKE (Serial ClocK Enable): Serial clock output enable bit

[CSIO]

This bit controls the I/O ports of a serial clock.

When this bit is set to "0": The SCK pin functions as a serial clock input pin.

When this bit is set to "1": It becomes a serial clock output pin allowing clock output while transmitting.

Notes:

- When you use a SCK pin as a serial clock input (SCKE=0), set a general purpose input/output port to an input port.
- After SCINV bit is set, set serial clock output enable (SCKE=1).
- When you use a SCK pin as a serial clock output, set the SCK pin as a peripheral output pin (set with EPFR). See "CHAPTER: I/O PORTS" for how to make setups.

[UART][LIN][I^2C]

This bit is a reserved bit. Always set this bit to "0".

[bit0] SOE (Serial Output Enable): Serial data output enable bit

[UART][CSIO][LIN]

This bit enables/disables output of serial data.

When this bit is set to "0": Serial data output stops.

When this bit is set to "1": The SOT pin functions as a serial data output pin (SOT).

Note:

Set a SOT pin as a peripheral output pin (set with EPFR). See "CHAPTER: I/O PORTS" for how to make setups.

[I^2C]

This bit is a reserved bit. Always set this bit to "0".

4.1.2. FIFO Control Register 1: FCR1

This section explains the bit structure of the FIFO control register 1.

FIFO control register (FCR1) is used for the test settings of FIFO, selection of transmission/reception FIFO, settings of transmission FIFO interrupt enable, and control of interrupt flag.

■ FCR1n(n=1 to 11): Address Base addr + 20_H(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	
0	0	0	0	0	1	0	0	Initial value
R/W0	R/W0	R/W0	R/W	R/W	R(RM1), W	R/W	R/W	Attribute

[bit7 to bit5] Reserved

Always set these bits to "0".

[bit4] FLSTE (Flag for data LoST detection Enable): Retransmission data lost detection enable bit

This bit enables FIFO retransmission data lost flag (FLST) detection.

When this bit is set to "0": FLST bit detection disabled

When this bit is set to "1": FLST bit detection enabled

Note:

When this bit is set to "1", set this bit to "1" after setting "1" to the FSET bit.

[bit3] FRIIE (Flag for Receive FIFO IdlE detection Enable): Reception FIFO idle detection enable bit

This bit configures whether or not to detect the reception idle state for 8-bit time or longer while the reception FIFO contains valid data. When reception interrupts are enabled (SCR:RIE=1), a reception interrupt will be generated once it detects reception idle state.

When this bit is set to "0": Reception idle state detection disabled

When this bit is set to "1": Reception idle state detection enabled

Note:

When the reception FIFO is used, set this bit to "1".

[bit2] FDRQ (transmit FIFO Data ReQuest): Transmission FIFO data request bit

It is a data request bit for transmission FIFO. When this bit is set to "1", it indicates that transmission data is being requested. When transmission FIFO interrupts are enabled (FTIE=1) at this time, a FIFO transmission interrupt request will be output.

FDRQ set condition

- When the transmission FIFO interrupt control is not used.
  - FBYTE (for transmission) = 0 (transmission FIFO is empty)
  - The transmission FIFO reset.
- When the transmission FIFO interrupt control is used.
  - FTICR setting value ≥ FTICR reading value (The storage data count of the transmission FIFO is the interrupt trigger level or less.)
  - The transmission FIFO reset.

FDRQ reset condition

- Writing "0" to this bit.
- If the transmission FIFO becomes full.

Notes:

- When transmission FIFO is enabled, writing "0" to this bit is valid.
- When FBYTE (for transmission) is "0", writing "0" to this bit is prohibited.
- When this bit is "0", the change in the FSEL bit is prohibited.
- When "1" is set to this bit, it does not affect the operation.
- If a read-modify-write instruction is executed, "1" will be read.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- [LIN] When it is setting value or less, writing "0" to this bit is prohibited.

[bit1] FTIE (Flag for Transmit Interrupt Enable): Transmission FIFO interrupt enable bit

It is an interrupt enable bit for transmission FIFO. If you set this bit to "1", an interrupt will be generated when the FDRQ bit is "1".

[bit0] FSEL (FIFO SELect): FIFO selection bit

This bit is used to select transmission/reception FIFO.

When this bit is set to "0", FIFO1 is assigned as the transmission FIFO, and FIFO2, the reception FIFO.
When this bit is set to "1", FIFO2 is assigned as the transmission FIFO, and FIFO1, the reception FIFO.

Notes:

- This bit will not be cleared by FIFO reset (FCL2, FCL1=1).
- When you change this bit, disable the FIFO operation (FCR0:FE2, FE1=0) first.
- If FDRQ="0", changing this bit is prohibited.

4.1.3. FIFO Control Register 0: FCR0

This section explains the bit structure of FIFO control register 0.

FIFO control register 0 (FCR0) is used to enable/disable FIFO operation, reset FIFO, save read pointer, and configure retransmission.

■ FCR0n(n=0 to 11): Address Base addr + 21_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1	
0	0	0	0	0	0	0	0	Initial value
R/W0	R,WX	R/W	R0,W	R0,W	R0,W	R/W	R/W	Attribute

[bit7] Reserved

This bit must always be written to "0".

[bit6] FLST (FIFO data LoST): FIFO retransmission data lost flag bit

This bit indicates that the retransmission data of transmission FIFO has been lost.

FLST set condition

- When you write (overwrite) FIFO while the FLSTE bit of FIFO control register 1 (FCR1) is "1" and the read pointers saved by the FSET bit matches the write pointer of transmission FIFO

FLST reset condition

- FIFO reset (writing "1" to FCL)
- Writing "1" to the FSET bit

If this bit is set to"1", it will overwrite the data indicated by the read pointer saved by the FSET bit. As a result, you will not be able to configure the retransmission by the FLD bit even when an error occurs. To execute a retransmission while this bit is set to "1", reset FIFO and write data to FIFO once again.

[bit5] FLD (FIFO pointer reLoaD bit) FIFO pointer reload bit

This bit reloads the data saved by the FSET bit at transmission FIFO to the read pointer. This bit is used for a retransmission in case that a communication error occurs. Once the retransmission setting has completed, this bit will be cleared to "0".

Notes:

- Do not write any other than FIFO reset while this bit is set to "1" since a reload to the read pointer is in progress.
- During the FIFO enable state or while a transmission is in progress, writing "1" to this bit is prohibited.
- [UART] [CSIO] [LIN] This bit must be set to "1" after SCR:TIE bit and SCR:TBIE bit are set to "0" and set SCR:TIE bit and SCR:TBIE bit to "1" after transmission FIFO is enabled.
- [I^2C] This bit must be set to "1" after SMR:TIE bit is set to "0" and set SMR:TIE bit to "1" after transmission FIFO is enabled.

[bit4] FSET (FIFO pointer SET): FIFO pointer save bit

This bit is used to save read pointer of transmission FIFO. If you save read pointer prior to communication, you will be able to retransmit while the FLST bit is "0" in case that a communication error occurs.

If this bit is set to "1": Save the current read pointer value.

If this bit is set to "0": No effect.

Note:

Set this bit to "1" when the transmission byte count (FBYTE) is 0.

[bit3] FCL2 (FIFO Clear 2): FIFO2 reset bit

This bit resets FIFO2.

When this bit is set to "1", it initializes the internal state of FIFO2.

Only the FCR0:FLST bit will be initialized while other bits of FCR1/0 register are retained.

Notes:

- Execute FIFO2 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- Valid data count of FBYTE2 register will be "0".

[bit2] FCL1 (FIFO Clear 1): FIFO1 reset bit

This bit resets FIFO1.

When this bit is set to "1", it initializes the internal state of FIFO1.

Only the FCR0:FLST bit will be initialized while other bits of FCR1/0 register are retained.

Notes:

- Execute FIFO1 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- Valid data count of FBYTE1 register will be "0".

[bit1] FE2 (FIFO Enable 2): FIFO2 operation enable bit

This bit enables/disables operation of FIFO2.

- To use FIFO2, set this bit to "1".
- When FIFO2 is selected as reception FIFO by the FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, you will not be able to set this bit to "1".
- When FIFO2 is used as transmission FIFO, if the transmission buffer is empty (SSR:TDRE="1"), or when FIFO2 is used as reception FIFO, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO2 disabled, the state of FIFO2 will be retained.
- [UART] [CSIO] When any data is present in FIFO2 and it is enabled for transmission (SCR:TXE=1) after FIFO2 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
- [CSIO] When you use FIFO2 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "0" when reception buffer is empty (SSR:RDRF="0") and no valid data is present in the reception FIFO (FBYTE2=0).
- [CSIO] When you use FIFO2 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "1" when reception buffer is empty (SSR:RDRF="0").
- [LIN] When FIFO2 is set for transmission FIFO, this bit is set to "1" while any data is present in FIFO2 and LIN interface (v2.1) transmission is enabled (TXE=1), transmission will immediately be started. In this case, after TIE bit and TBIE bit are set to "0", set "1" to this bit, then TIE bit and TBIE bit.

.

Notes:

[I^2C]

- Changes to enable or disable must be made while IBSR:BB bit is "0" or IBCR:INT bit is "1".
- When selected as a reception FIFO to detect reservation address and operate as a slave transmission, set this bit to "0" with reservation address detection interrupt and set IBCR:ACKE="0".
- When SSR:RDRF bit is "1" while used as a reception FIFO and this bit is changed from "1" to "0", the reception FIFO will not be disabled until SSR:RDRF bit becomes "0".
- When any data is present in FIFO2 while used as a transmission FIFO and this bit is changed from "0" to "1", set this bit to "1" after SMR:TIE bit is set to "0" and set SMR:TIE bit to "1".

[bit0] FE1 (FIFO Enable 1) FIFO1 operation enable bit

This bit enables/disables operation of FIFO1.

- To use FIFO1, set this bit to "1".
- When FIFO2 is selected as reception FIFO by the FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, you will not be able to set this bit to "1".
- When FIFO2 is used as transmission FIFO, if the transmission buffer is empty (SSR:TDRE="1"), or when FIFO2 is used as reception FIFO, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO1 disabled, the state of FIFO1 will be retained.
- [UART] [CSIO] When any data is present in FIFO1 and it is enabled for transmission (SCR:TXE=1) after FIFO1 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
- [CSIO] When you use FIFO1 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "0" when reception buffer is empty (SSR:RDRF="0") and no valid data is present in the reception FIFO (FBYTE2=0).
- [CSIO] When you use as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "1" when reception buffer is empty (SSR:RDRF="0").
- [LIN] When FIFO1 is set for transmission FIFO, this bit is set to "1" while any data is present in FIFO1 and LIN interface (v2.1) transmission is enabled (TXE=1), transmission will immediately be started. In this case, after TIE bit and TBIE bit are set to "0", set "1" to this bit, then TIE bit and TBIE bit.

Notes:

[I^2C]

- Changes to enable or disable must be made while IBSR:BB bit is "0" or IBCR:INT bit is "1".
- When selected as a reception FIFO to detect reservation address and operate as a slave transmission, set this bit to "0" with reservation address detection interrupt and set IBCR:ACKE="0".
- When SSR:RDRF bit is "1" while used as a reception FIFO and this bit is changed from "1" to "0", the reception FIFO will not be disabled until SSR:RDRF bit becomes "0".
- When any data is present in FIFO1 while used as a transmission FIFO and this bit is changed from "0" to"1", set this bit to "1" after SMR:TIE bit is set "0" and set SMR:TIE bit to "1".

4.1.4. FIFO BYTE Register: FBYTE

This section explains the bit structure of the FIFO byte register.

This register has different functions for reading and writing.
For reading, FIFO byte register (FBYTE) shows valid data count of FIFO.
For writing, you will be able to configure whether to generate a reception interrupt when the reception FIFO receives the specified data count.

■ FBYTEn(n=0 to 11): Address Base addr + 22_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
FBYTE2[7:0]								
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
FBYTE1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[bit15 to bit8] FBYTE2 (FIFO BYTE 2): FIFO2 data count display bits

[bit7 to bit0] FBYTE1 (FIFO BYTE 1): FIFO1 data count display bits

The FBYTE register indicates valid data count written to or received at FIFO. The table below shows the details of FCR1:FSEL bit settings.

FSEL	FIFO selection	Data count display
0	FIFO2: Reception FIFO, FIFO1:Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value of FBYTE transfer count is 08_H.
- Set the data count at which you want to generate a reception interrupt flag with FBYTE for reception FIFO. If the specified transfer count and data count display of FBYTE register match, the interrupt flag (SSR:RDRF) will be set to "1".
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1"
  - Data count contained in the reception FIFO does not reach the transfer count
- If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.
- [CSIO] To receive data in the master operation mode (master reception), clear the SCR:TIE bit and SCR:TBIE bit to "0", set the reception data count at the FBYTE register of transmission FIFO, and write "0" to the FCR1:FDRQ bit. Then, serial clocks for the volume of data configured will be output when the SCR:TXE bit is "1", which allows you to receive the data volume you have configured. To set the SCR:TIE bit and the SCR:TBIE bit to "1", set them to 1 after FCR1:FDRQ changes to "1".
- [CSIO] When transmission data is written to TDR once, transmission FIFO's FBYTE will be incremented by +1. When SSR:AWC=0 and the data length is 20, 24, 32, a transmission data writing to TDR must be separated to 2 times. The transmission FIFO's FBYTE will be incremented by +2.

- [CSIO] When reception data is read from RDR once, reception FIFO's FBYTE will be decremented by 1. When SSR:AWC=0 and the data length is 20, 24, 32, a reception data read from RDR must be separated to 2 times. The reception FIFO's FBYTE will be decremented by 2.
- [I^2C] To receive data in the master operation mode (master reception), clear the SMR:TIE bit to "0", set the reception data count at FBYTE register of transmission FIFO, and write "0" to the FCR1:FDRQ bit. SCL clocks for the data volume configured will be output. Then, the IBCR:INT bit will be set to "1". To set the SMR:TIE bit to "1", set it to 1 after FCR1:FDRQ changes to "1".

Notes:

- [UART] [LIN] Set FBYTE register of the transmission FIFO to "8'h00".
- [UART] [LIN] Disable reception before making any change.
- [CSIO] [I^2C] Other than the case of receiving data in the master operation mode, set FBYTE register of the transmission FIFO to "8'h00".
- [CSIO] When you configure the transmission data count for data reception in the master operation mode, make sure that the transmission FIFO is empty and the SCR:TIE and SSR:TBIE bits are "0".
- [CSIO] When you disable reception (SCR:RXE=0) while data is being received in the master operation mode, you will need to disable the transmission FIFO before disabling the transmission/reception FIFO.
- [CSIO] Make any change to reception FIFO's FBYTE after disabled reception.
- [LIN] After setting FIFO select bit (FCR1:FSEL), set FIFO byte register (FBYTE).
- [LIN] FIFO select bit (FCR1:FSEL) and FIFO byte register (FBYTE) cannot be set at the same time.
- [LIN] For FIFO data count display during transmission, a value with 1 subtracted from the number of writing for transmission data is displayed as a valid data count. This is because, if transmission data is written while data which has not been transmitted yet to the TDR register exists, the data is stored in transmission FIFO. After the data in the TDR register is transmitted, the data which has not been transmitted in transmission FIFO is transmitted to the TDR register.
- [LIN] For FIFO data count display during reception, the number of data which is received by reception FIFO and which has not been read yet is displayed. The data which has been received by the RDR register is not included.
- [I^2C] 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.
- [I^2C] 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 SMR:TIE bit is "0".
- [I^2C] Change this in any of the following conditions.
  - When I^2C interface is disabled (ISMK:EN=0)
  - When IBCR:INT=1 during master mode reception under SSR:DMA=0
  - When SSR:TBI=1 during master mode reception under SSR:DMA=1
- [I^2C] To receive data in the master mode operation (master mode reception), do not write dummy data to the transmission data register (TDR) when setting the SMR:TIE bit to "0" and setting the receive data count for the FBYTE register of transmission FIFO.
- [Common] Data configured at FBYTE of the reception FIFO should be "1" or greater.
- [Common] This register cannot use the read-modify-write instruction.
- [Common] Settings that go over the FIFO capacity are prohibited.

4.1.5. Transmission FIFO Interrupt Control Register: FTICR

This section explains the bit structure of the transmission FIFO interrupt control register.

Transmission FIFO interrupt control register (FTICR) configures the interrupt by the transmission effective data count of the FIFO.

■ FTICRn(n=0 to 11): Address Base addr + 24_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
FTICR2[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
FTICR1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8] FTICR2: FIFO2 data count display bits

[bit7 to bit0] FTICR1: FIFO1 data count display bits

The FTICR register sets the interrupt trigger level by the effective data count of the transmission FIFO (residual quantity). The table below shows the details of FCR1:FSEL bit settings.

FSEL	Transmission FIFO selection	Transmission FIFO interrupt control register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial values of the effective data count that generates the interrupt of the FTICR register are 0x00.
- FTICR register sets the data count of transmission interrupt generation to FTICR of transmission FIFO. When the display of set number of data count and effective data count of transmission FIFO (FBYTE) matches or becomes small, interrupt flag (FDRQ) is set to "1".
- The effective data count to transmission FIFO is displayed.

FTICR2, FTICR1: FIFIO2 data count display bits, FIFO1 data count display bits

Write	The effective data count that generates the interrupt is set.
Read	The effective data count is read.

Notes:

- The setting that exceeds the capacity of FIFO is prohibited.
- The set value cannot be read.
- This register cannot use the read-modify-write instruction.

4.2. Registers for UART

Registers for UART are shown.

4.2.1. Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission and reception, disable/enable transmission/reception interrupts, disable/enable transmission bus idle interrupts, and reset UART.

■ SCRn(n=0 to 11): Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	Reserved	Reserved	RIE	TIE	TBIE	RXE	TXE	
0	-	-	0	0	0	0	0	Initial value
R0,W	RX,WX	RX,WX	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of UART. When this bit is set to "1": · Directly reset UART (software reset). In this case, the register settings will be maintained. Note that any active transmission or reception will be cut off immediately. · Baud rate generator restarts by reloading the setting value of the BGR1/0 register. · All the transmission and reception interrupt factors (SSR:PE,FRE,ORE,RDRF,TDRE,TBI,TINT) are initialized(0000110_B). When this bit is set to "0": No effect. A read always results in "0". Notes: · Execute a programmable clear after disabling interrupts. · When using FIFO, disable FIFO (FCR0:FE2,FE1=0) before you execute a programmable clear. · Even if programmable clear is executed (SSR:UPCL=1), the value of serial timer register (STMR) is not cleared.
bit6, bit5	Reserved bits	Read: The value is undefined. Write: No effect on operation.
bit4	RIE: Reception interrupt enable bit	· This bit enables or disables the output of reception interrupt request to the CPU. · When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bit (SSR:PE, ORE, FRE) is set to "1", a reception interrupt request will be output.

Bit name		Function
bit3	TIE: Transmission interrupt enable bit	· This bit enables or disables the output of transmission interrupt request to the CPU. · When the TIE bit and the SSR:TDRE bit are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	· This bit enables or disables the output of transmission bus idle interrupt request to the CPU. · When the TBIE bit and TBI bit are set to "1", a transmission bus idle interrupt request will be output.
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.

4.2.2. Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) checks the status of transmission/reception and the reception error flag, and clears the reception error flag.

■ SSRn(n=0 to 11): Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved	PE	FRE	ORE	RDRF	TDRE	TBI	
0	-	0	0	0	0	1	1	Initial value
R0,W	RX,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears the PE, FRE, ORE flags of the serial status register (SSR). · To clear an error flag, write "1" to this bit. · Writing "0" does not affect anything. A read always results in "0".
bit6	Reserved bit	Read: The value is indefinite. Write: No effect on operation.
bit5	PE: Parity error flag bit (Functions only in the operation mode 0)	"0" Read: No parity error "1" Read: There is a parity error · If a parity error occurs while a reception is in progress (ESCR:PEN=1), this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). · When the PE bit and the SCR:RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.
bit4	FRE: Framing error flag bit	"0" Read: No framing error "1" Read: There is a framing error · If a framing error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). · When the FRE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.

Bit name		Function
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error. "1" Read: There is an overrun error. · If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). · When the ORE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: The receive data register (RDR) contains data. · The flag indicates the state of the receive data register (RDR). · When received data is loaded in the RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". · When the RDRF bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. · While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. · In the case where all the conditions below are met while using reception FIFO, when reception idle continues for more than 8 baud rate clocks, RDRF will be set to "1". · Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" · The reception FIFO contains data without receiving the specified number of data sets If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. · While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register is empty. · The flag indicates the state of the transmit data register (TDR). · When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. · When the TDRE bit and the SCR:TIE bit are set to "1", a transmission interrupt request will be output. · When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". · For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmission is in progress "1" Read: No transmission is in progress · This bit indicates that UART has no transmission in progress. · When transmission data has been written to the transmit data register (TDR), this bit will become "0". · When the transmit data register is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". · When you set "1" to the UPCL bit of the serial control register (SCR), the TBI bit will be set to "1". · When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output.

4.2.3. Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended communication control register (ESCR) configures the data length of transmission/reception, enables/disables the parity bit, selects a parity bit, inverts the serial data format, and selects the length of stop bit.

■ ESCRn(n=0 to 11): Address Base addr + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	ESBL	INV	PEN	P	L[2:0]			
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	Reserved bit	Always set this bit to "0".
bit6	ESBL: Extended stop bit length select bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SBL="0" and ESCR:ESBL="0" are set: Stop bit is set to 1 bit. When SBL="1" and ESCR:ESBL="0" are set: Stop bit is set to 2 bits. When SBL="0" and ESCR:ESBL="1" are set: Stop bit is set to 3 bits. When SBL="1" and ESCR:ESBL="1" are set: Stop bit is set to 4 bits. Notes: · When receiving, only the first bit of the stop bits will always be detected. · This bit should be set when transmission is disabled (TXE=0).
bit5	INV: Inverted serial data format bit	This bit selects the serial data format to be either NRZ format or inverted NRZ format. · When this bit is set to "0": NRZ format is set. · When this bit is set to "1": Inverted NRZ format is set.

Bit name		Function
bit4	PEN: Parity enable bit (Functions only in the operation mode 0)	This bit configures whether to enable addition (transmission) and detection (reception) of the parity bit. · When this bit is set to "0", no parity bit will be added. · When this bit is set to "1", a parity bit will be added. Note: In operation mode 1, this bit will be fixed to "0"internally.
bit3	P: Parity selection bit (Functions only in the operation mode 0)	When parity is enabled (ESCR:PEN=1), this bit selects odd parity"1" or even parity "0". · When this bit is set to "0": Selects even parity · When this bit is set to "1": Selects odd parity
bit2, bit1, bit0	L2, L1, L0: Data length select bits	These bits specify the data length of transmission/reception data. · "000_B": Data length will be set to 8 bits. · "001_B": Data length will be set to 5 bits. · "010_B": Data length will be set to 6 bits. · "011_B": Data length will be set to 7 bits. · "100_B": Data length will be set to 9 bits. Notes: · Settings other than those shown above are prohibited. · In operation mode 1, set the data length to 7 or 8 bits. The other settings are prohibited.

4.2.4. Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When you use the address to read, it functions as the receive data register, and when you use the address to write, it functions as the transmit data register. When FIFO is enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

■ Read

■ RDR0n(n=0 to 11): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved							D8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W	Attribute
7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is a 9-bit data buffer register for serial data reception.

· Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).

- Depending on the data length, "0" is inserted in the upper bit as shown below.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X is the reception data bit)

- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When reception interrupts are enabled (SSR:RIE=1), a reception interrupt request will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the receive data register (RDR) has been read out.
- In case a reception error occurs (SSR: PE, ORE or FRE is "1"), data in the receive data register (RDR) will become invalid.
- In operation mode 1 (multi-processor mode), the operation will be 7-bit or 8-bit long. The AD bit received will be stored at the D8 bit.
- For the 9-bit long transfer and in operation mode 1, RDR will be read in 16-bit access mode.

Notes:

- When using reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to SSR:RDRF.
- When using reception FIFO, if the reception FIFO becomes empty, SSR:RDRF will be cleared to "0".
- If a reception error occurs (SSR: PE, ORE, or FRE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, reception data will not be stored in the reception FIFO.

■ Write

■ TDR0n(n=0 to 11): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-							D8	
-	-	-	-	-	-	-	1	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the 9-bit data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output

from the serial data output pin (SOT pin).

- Depending on the data length, data will be invalidated from the upper bit as shown below.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- You will be able to write transmission data when the transmission data empty flag (SSR:TDRE) is set to "1". If the transmission interrupt is enabled, a transmission interrupt will occur. Writing transmission data should be performed by the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".
- You will not be able to write transmission data when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.
- In operation mode 1 (multi-processor mode), the operation will be 7-bit or 8-bit long. The AD bit will be transmitted by writing at the D8 bit.
- For the 9-bit long transfer and in operation mode 1, write a value to the TDR in 16-bit access mode.

Notes:

- Transmission data register is write-only register and receive data register is read-only register. The value written is different from the read value since the transmission/reception registers are located at the same address. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operation cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR:TDRE) when using the transmission FIFO, see "5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

4.2.5. Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

Serial Aid Control Status Register (SACSR) configures serial test operation control, timer interrupt enable/disable, synchronous transmission enable/disable, operating clock division rate of serial timer, and serial timer enable/disable.

■ SACSRn(n=0 to 11): Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	Reserved						TINT	
0	0	0	0	0	0	0	0	Initial value
R/W	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	Reserved		TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	RX,W0	RX,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] STST: Serial test bit

This bit selects to enable or disable serial test mode.

When serial test mode is enabled, SOT and SIN will be connected in the multi-function serial interface so that the data output from SOT can be received from SIN without any modification.

When serial test mode is enabled, SOT pin will be fixed to "H" and data input to SIN pin will be ignored.

STST	Serial test bit
0	Disables serial test mode
1	Enables serial test mode

Notes:

This bit can be changed only when transmission and reception is disabled (SCR:TXE=0, SCR:RXE=0).

[bit14 to bit9] Reserved

Always set these bits to "0".

[bit8] TINT: Timer Interrupt Flag

When Serial Timer Register (STMR) matches Serial Timer Comparison Register (STMCR), the Serial Timer Register (STMR) becomes "0" and this bit will be set to "1".

When this bit is "1" and timer interrupt enable bit (TINTE) is "1", status interrupt request will be output.

When this bit is set to "0", it will be reset to "0".

Writing "1" to this bit is invalid.

TINT	Description
0	No timer interrupt request
1	Timer interrupt request

Notes:

- Performing software reset (SCR:UPCL="1") will reset this bit to "0".
- A read with a read-modify-write instruction will read "1".

[bit7] TINTE: Timer Interrupt Enable Bit

This bit enables/disables timer interrupt to the CPU.

When this bit is "1" and timer interrupt flag (TINT) is "1", status interrupt request will be output.

TINTE	Description
0	Disables interrupt from serial timer
1	Enables interrupt from serial timer

[bit6, bit5] Reserved

Always set these bits to "0".

[bit4 to bit1] TDIV3-0: Timer Operating Clock Division Bits

These bits set division rate of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer Operating Clock						
				Division rate	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	ϕ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	ϕ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

ϕ: Bus clock

Notes:

- These bits can be changed only when serial timer enable bit (TMRE) is "0".
- Any setup other than the above is prohibited.

[bit0] TMRE: Serial timer enable Bit

This bit enables/disables serial timer operations.

TMRE	Serial timer enable bit
0	Stops serial timer. While stopped, values in the Serial Timer Register (STMR) are retained.
1	When this bit is changed from "0" to "1", the value of the Serial Timer Register (STMR) will be initialized to "0" and serial timer will be started.

4.2.6. Serial TiMer Register: STMR

This section explains the bit structure of the serial timer register.

Serial Timer Register (STMR) indicates the timer values of the serial timer.

■ STMRn(n=0 to 11) : Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TM15-0: Timer Data Bits

These bits indicate the timer values of the serial timer.

While the timer is running, the timer values of the serial timer will be increased by "1" every timer operating clock .(SACSR: Set with TDIV3-0).

Note:

These bits will be initialized to "0" when the timer started running.

4.2.7. Serial Timer Comparison Register: STMCR

This section explains the bit structure of the serial timer comparison register.

Serial Timer Comparison Register (STMCR) sets the comparison value of a timer of the serial timer.

■ STMCRn(n=0 to 11) : Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] TC15-0: Comparison Bits

These bits set a comparison value for serial timer.

These bits will be compared with the Serial Timer Register (STMR) and when these bits and the values of the STMR matched at the update timing of the STMR, the STMR will be set to "0". At this timing, the timer interrupt flag (SACSR:TINT) will be set to "1".

The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0).

- SACSR:TINT is set to "1".

Notes:

- When $(0000)_H$ is set to this register, the Serial Timer Register still indicates "0".
- With "0000_H" set to this register, the timer interrupt flag (SACSR:TINT) will be fixed to "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000_B" while the timer is running.
- This register can be changed only when serial timer is disabled (SACSR:TMRE="0").

4.2.8. Baud rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

■ BGRn(n=0 to 11) : Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
EXT	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	bit
BGR[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] EXT (EXTernal clock): EXTernal clock select bit

This bit selects whether to use an internal clock source or an external clock source for the internal reload counter for baud rate generation. When setting EXT=0, the internal clock source will be used. When setting EXT=1, the external clock source will be used.

[bit14 to bit0] BGR14 to BGR0 (Baud rate GeneratoR): Baud rate generator bits

- These bits set division ratio of the serial clock.
- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15’h00". Thus, if you wish to validate a new setting value immediately, execute programmable clear (UPCL) after you have change the setting value of BGR.
- When the reload value is an even number, As for the "H" width and "L" width of the reception serial clock, the "L" is longer by 1 cycle of the bus clock. When the reload value is an odd number, the widths of "H" and "L" become the same.
- Use a value 4 or greater to set to BGR. However, correct data may not be received depending on the baud rate error and reload setup value.
- When you change to the external clock setting (EXT=1) while baud rate generator is running, write "0" to the Baud Rate Generator (BGR) and perform programmable clear (UPCL), then set to the external clock (EXT=1).

4.3. Registers for CSIO

Registers for CSIO are shown.

4.3.1. Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission/reception interrupts, disable/enable transmission idle interrupt, disable/enable transmission and reception. Setup for connecting SPI and CSIO reset are also allowed.

■ SCRn(n=0 to 11) : Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R,W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of CSIO. When this bit is set to "1": · Directly reset CSIO (software reset). In this case, the register settings will be retained. Note that any active transmission or reception will be cut off immediately. · Baud rate generator restarts by reloading the setting value of the BGR register. · All the transmissions/receptions and status interrupt factors (SSR:TDRE, TBI, RDRF, ORE, TINT, CSE) will be initialized. · All serial chip select pins become inactive. When this bit is set to "0": No effect on the operation. A read always results in "0". Notes: · Execute a programmable clear after disabling interrupts. · When using FIFO, disable FIFO (FCR0:FE2, FE1=0) before you execute a programmable clear. · Even if programmable clear is executed (SCR:UPCL=1), the value of serial timer register (STMR) is not cleared.
bit6	MS: Master/slave function select bit	This bit selects master or slave mode. When this bit is set to "0": Master mode When this bit is set to "1": Slave mode Notes: · If SMR:SCKE=0 when the slave mode is selected, an external clock will be input directly. · Set this bit when transmission and reception are disabled (TXE=RXE=0). · After MS bit is set, set reception enable (RXE=1).

bit5	SPI:
	SPI support bit
		This bit is used to execute a SPI communication.
		When chip select is used in master mode (SCR: MS=0), this bit is used for serial chip select pin 0 communication.
		
		When this bit is set to "0": Normal synchronous communication
		When this bit is set to "1": SPI communication supported.
		
		Notes:
		· Set this bit when transmission and reception are disabled (TXE=RXE=0).
		· This bit is used in one of cases below.
		  · Chip select pin is disabled (SCSCR:CSEN3-0="0000_B")
		  · When mode is in slave (SCR:MS=1)
		  · When data format of chip select is disabled (ESCR:CSFE=0)
		  · When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

bit4	RIE: Reception interrupt enable bit	· When setting to "0": Reception interrupts are disabled. · When setting to "1": Reception interrupts are enabled. · This bit enables or disables the output of reception interrupt request to the CPU. · When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bit (ORE) is set to "1", a reception interrupt request will be output.

Bit name		Function
bit3	TIE: Transmission interrupt enable bit	· When setting to "0": Transmission interrupts are disabled. · When setting to "1": Transmission interrupts are enabled. · This bit enables or disables the output of transmission interrupt request to the CPU. · When the TIE bit and the SSR:TDRE bit are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	· When setting to "0": Transmission bus idle interrupts are disabled. · When setting to "1": Transmission bus idle interrupts are enabled. · This bit enables or disables the output of transmission bus idle interrupt request to the CPU. · When the TBIE bit and SSR:TBI bit are set to "1", a transmission bus idle interrupt request will be output.
bit1	RXE: Reception enable bit	This bit enables/disables the reception of CSIO. · If this bit is set to "0", data frame reception is disabled. · If this bit is set to "1", data frame reception is enabled. Notes: · If you disable reception (RXE=0) while a reception is in progress, it immediately stops the reception. · After MS and SMR:SCINV bits are set, set reception enable (RXE=1).
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of CSIO. · If this bit is set to "0", data frame transmission is disabled. · If this bit is set to "1", data frame transmission is enabled. Note: If you disable transmission (TXE=0) while a transmission is in progress, it immediately stops the transmission.

4.3.2. Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag as well as to clear the reception error flag.

■ SSRn(n=0 to 11) : Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved		AWC	ORE	RDRF	TDRE	TBI	
0	0	0	0	0	0	1	1	Initial value
R0,W	R0,W0	R/W0	R/W	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears ORE flag of the serial status register (SSR). · To clear an error flag, write "1" to this bit. · Writing "0" does not affect the operation. A read always results in "0".
bit6 ,bit5	Reserved bit	Always set these bits to "0".
bit4	AWC: Access width control bit	This bit selects 16-bit or 32-bit accesses for accessing transmission data register (TDR) or reception data register (RDR). · "0" is set: 16-bit access · "1" is set: 32-bit access Note: This bit can be changed only if transmission/reception is disabled (SCR:TXE=RXE=0) and TDR and RDR are empty (SSR:TDRE=1, SSR:RDRF=0). · To perform the DMA transfer while the transmit FIFO is not used, set "1" to this bit. · When the data length is 20, 24 or 32 bits, set "1" to this bit. · To use the SPI mode with the slave mode while the transmit FIFO is enabled (FCR0.FE2,FE1=0), set this bit to "1".
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error "1" Read: There is an overrun error · If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). · When the ORE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.

Bit name		Function
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: The received data register (RDR) contains data. · The flag indicates the state of the receive data register (RDR). · When received data is loaded in RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". · When the RDRF bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. · While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. · In the case where all the conditions below are met while using reception FIFO, when reception idle continues for more than 8 baud rate clocks, RDRF will be set to "1". · Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" · The reception FIFO contains data without receiving the specified number of data sets If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. · While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register is empty · The flag indicates the state of the transmit data register (TDR). · When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. · When the TDRE bit and the SCR:TIE bit are set to "1", a transmission interrupt request will be output. · When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". · For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmission is in progress. "1" Read: No transmission operation · This bit indicates CSIO has no transmission in progress. · When transmission data has been written to the transmit data register (TDR), this bit will become "0". · When the transmit data register (TDR) is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". · When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". · When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output. Note: This bit becomes "1" when transmission data register (TDR) is empty (TDRE=1) and serial chip select error (CSE=1) is generated.

4.3.3. Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended communication control register (ESCR) is used to set the transmission/reception data length as well as to fix the serial output at the "H" level.

Setting to serial chip select pins without assignment to external pins is disabled.
For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below.

Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)
Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)
Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)
Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)
Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

■ ESCRn(n=0 to 11) : Address Base + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
SOP	L[3]	CSFE	WT[1:0]		L[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	SOP: Serial output pin set bit	· This bit sets the serial output pin at the "H" level. When you write "1" to this bit, the SOT pin will be set to "H". However, you do not have to write "0" to this bit. · A read always results in "0". Note: Do not set this bit during serial data transmission.
bit5	CSFE: Serial chip select format enable bit	Enable or disable format setting for each serial chip select pin. · When set to "0": Same data format and clock format will be set to all serial chip select pins · When set to "1": Data format and clock format will be set to serial chip select pins separately When this bit is set to "1", following setups will be set to serial chip select pins separately. · Inactive level for serial chip select · Mark level of serial clock · SPI transfer/normal transfer selection · Direction of serial data transfer · Serial data length Note: · This bit setting will be invalid when any of following is met. · When chip select pin is disabled (SCSCR:CSEN3-0="0000_B") · When mode is in slave (SCR:MS=1) · When this bit is set to "1", set the following settings: · Enable reception FIFO · Set the hold delay to 2 or larger (setting CSHD7-0 bit in SCSTR1 register to 2 or larger) · Set the length of each serial chip select data to 9 bits or smaller, or 10 bits or larger to communicate with two or more slave devices · Disable setting the serial chip select pins separately to 9 bits and smaller, and 10 bits and larger (Examples of disabled settings) serial chip select 0 = 9 bits serial chip select 1 = 10 bits (Examples of enabled settings) serial chip select 0 = 16 bits serial chip select 1 = 10 bits

Bit name		Function
bit4, bit3	WT1, WT0: Data transmission/reception wait select bits	In the master mode, these bits set the number of waits for a successive data transmission or reception. Operation in the slave mode is "00". · "00": SCK will be output sequentially. · "01": SCK will be output after waiting for 1-bit time. · "10": SCK will be output after waiting for 2-bit time. · "11": SCK will be output after waiting for 3-bit time. Note: If this register is used when all of the following conditions are satisfied, set WT1 and WT0 to "00". · Chip select is used. · The SPI mode (SCR:SPI=1) is used. · "01_H" is set in TBYTE register. · The SCAM bit in SCSCR register is set to "1".
bit6, bit2 to bit0	L3, L2, L1, L0: Data length select bits	These bits specify the data length of transmission/reception data. When chip select is used in master mode (SCR:MS=0), these bits are used for serial chip select pin 0 communication. · "0000_B": Data length will be set to 8 bits. · "0001_B": Data length will be set to 5 bits. · "0010_B": Data length will be set to 6 bits. · "0011_B": Data length will be set to 7 bits. · "0100_B": Data length will be set to 9 bits. · "0101_B": Data length will be set to 10 bits. · "0110_B": Data length will be set to 11 bits. · "0111_B": Data length will be set to 12 bits. · "1000_B": Data length will be set to 13 bits. · "1001_B": Data length will be set to 14 bits. · "1010_B": Data length will be set to 15 bits. · "1011_B": Data length will be set to 16 bits. · "1100_B": Data length will be set to 20 bits. · "1101_B": Data length will be set to 24 bits. · "1110_B": Data length will be set to 32 bits. Notes: · Settings other than those listed above are prohibited. · This bit is used in one of cases below. · When chip select pin is disabled (SCSCR:CSEN3-0="0000_B") · While in slave mode (SCR:MS=1) · When data format of chip select is disabled (ESCR:CSFE=0) · When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

4.3.4. Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

■ Read

■ RDR1n-0n(n=0 to 11) : Address Base addr + 04_H (Access: Half-word, Word)

31	30	29	28	27	26	25	24	bit
D31	D30	D29	D28	D27	D26	D25	D24	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

23	22	21	20	19	18	17	16	bit
D23	D22	D21	D20	D19	D18	D17	D16	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is a 32-bit data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).
  Depending on the data length, received data will be filled from the lower bit and other bits become "0".

  Example: When the data length is 8 bits and "45_H" is received: D7-D0="45_H", D31-D8=0
- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When reception interrupts are enabled (SSR:RIE=1), a reception interrupt request will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the receive data register (RDR) has been read out.
- In case a reception error occurs (SSR:ORE is "1"), data in the receive data register (RDR) will become invalid.
- When you read RDR, accesses must be made with following methods.
  - SSR:AWC=0: 16-bit access to lower 16 bits of RDR
  - SSR:AWC=1: 32-bit access
- SSR:AWC=1 allows one-time read for any data length.
- SSR:AWC=0 allows one-time read for any data length from 5 to 16 bits.

Notes:

- When you use reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to RDRF.
- When you are using reception FIFO, if the reception FIFO becomes empty, RDRF will be cleared to "0".
- If a reception error occurs (SSR:ORE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.
- When SSR:AWC=0, each of the read values of D31-D16 is undefined.

■ Write

■ TDR1n-0n(n=0 to 11) : Address Base addr + 04_H (Access: Half-word, Word)

31	30	29	28	27	26	25	24	bit
D31	D30	D29	D28	D27	D26	D25	D24	
0	0	0	0	0	0	0	0	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute
23	22	21	20	19	18	17	16	bit
D23	D22	D21	D20	D19	D18	D17	D16	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute
15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute
7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the 32-bit data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output from the serial data output pin (SOT Pin).
- Depending on the data length, the transmitting data will be stored from the lower bit and other bits will become "invalid".
  Example: When you transmit "45_H" with 8 bits data length, D7-D0="45_H" and D31-D8 will become invalid.
- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- You will be able to write transmission data when the transmission data empty flag (SSR:TDRE) is set to "1". If the transmission interrupt is enabled, a transmission interrupt will occur. Writing next transmission data should be

performed after the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".

- You will not be able to write transmission data when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.
- When you write to TDR, accesses must be made with following methods.
  - SSR:AWC=0: 16-bit access to lower 16 bits of TDR
  - SSR:AWC=1: 32-bit access
- SSR:AWC=1 allows one-time write for any data length.
- SSR:AWC=0 allows one-time write for any data length from 5 to 16 bits.

■ Relationship between Transmit Data Register (TDR) and Transmission Data Empty Flag

In the 16-bit access mode (SSR:AWC=0), the TDR register has the 16-bit boundary, and transmission data is stored as 16-bit data for each writing. In addition, when 32-bit transmission data exists in the TDR register, the transmit data empty flag (SSR:TDRE) will be "0".

In the 32-bit access mode (SSR:AWC=1), the TDR register has the 32-bit boundary, and transmission data is stored as 32-bit data for each writing.

Table 4-7 Relationship between Transmit Data Register (TDR) and Transmission Data Empty Flag

Data access width	Data length	TDR register The storage data value	TBI flag	TDRE flag	Transmission
16-bit access (SSR:AWC=0)	5-16-bits	0 bit	1	1	Disabled
		16 bits	0		Enabled
		32 bits		0	
32-bit access (SSR:AWC=1)	All data length	0 bit	1	1	Disabled
		32 bits	0	0	Enabled

Notes:

- Transmit data register is write-only register and receive data register is read-only register. The value written is different from the read value since the transmit/receive registers are located at the same address. Therefore, instructions such as INC/DEC instructions which perform read-modify-write (RMW) operations cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR: TDRE) when using the transmission FIFO, see "7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

4.3.5. Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

The serial aid control status register (SACSR) allows you to control serial test operations, enable/disable timer interrupts, enable/disable synchronous transmission, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

Setting to serial chip select pins without assignment to external pins is disabled.
For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below.

Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)
Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)
Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)
Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)
Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

■ SACSRn(n=0 to 11) : Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	Reserved	TBEEN	CSEIE	CSE	Reserved		TINT	
0	0	0	0	0	0	0	0	Initial value
R,W	R0,W0	R/W	R/W	R(RM1), W	RX,W0	RX,W0	R(RM1), W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	TSYNE	Reserved	TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R,W	RX,W0	R,W	R,W	R,W	R,W	R/W	Attribute

[bit15] STST: Serial TeST bit

This bit is used to enable or disable the serial test mode.
When the serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed.
When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input into the SIN pin will be ignored.

STST	Serial test bit
0	Serial test mode disabled
1	Serial test mode enabled

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=0, SCR:RXE=0).
- Set this bit to "0" in slave mode (SCR:MS=1).

[bit14] Reserved

Read: The read value is "0".
Write: Always write "0" to this bit.

[bit13] TBEEN: Transfer Byte Error ENable bit

If one of the following conditions applies in the master mode (SCR:MS="0") and if no valid transmission data is available (SSR:TDRE="1") for the transmission data register (TDR) when one frame has been transmitted while the number of frames that are being transmitted is smaller than the setting value of TBYTE, this bit is used to enable/disable occurrence of serial chip select errors.

- Chip select is used
- Synchronous transmission of the serial timer is used

TBEEN	Transfer Byte Error ENable bit
0	Occurrence of chip select errors in the master mode (SCR:MS=0) disabled
1	Occurrence of chip select errors in the master mode (SCR:MS=0) enabled

Note:

This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

[bit12] CSEIE: Chip select error interrupt enable bit

This bit is used to enable/disable chip select error interrupt request output.
When the CSEIE bit and the chip select error flag bit (CSE) are set to "1", a transmission interrupt request will be output.

CSEIE	Chip select error interrupt enable bit
0	Chip select error interrupt disabled
1	Chip select error interrupt enabled

[bit11] CSE: Chip select error flag bit

In the master mode (SCR:MS=0) with transfer byte error enabled (TBEEN=1), this bit is set to "1" in either of the following cases, if there is no valid transmit data (SSR:TDRE=1) in the transmit data register (TDR) when 1-frame transmit is completed while the number of transmitted frames is less than the TBYTE setting value.

- Chip select is used
- Transmission synchronized with the serial timer is used

In the slave mode (SCR:MS="1"), if a serial chip select pin becomes inactive during the transmission operation (SSR:TBI=0), this bit will be set to "1".
When this bit is set to "1" and the chip select error interrupt enable bit (CSEIE) is set to "1", a transmission interrupt request will be output.
Writing "0" to this bit will reset it to "0".
Writing "1" to this bit has no effect.

CSE	Chip select error flag bit
0	No chip select errors
1	Chip select errors

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.
- When serial chip select is unused (SCSCR:CSEN0="0") in the slave mode (SCR:MS=1), this bit will not be set to "1".
- If a chip select error occurs (CSE=1), write "0" to this bit after disabling the transmission (SCR:TXE=0). To resume transmission, write "0" to this bit, and then enable the transmission (SCR:TXE=1) and write the transmission data to the transmission data buffer (TDR).
- If the serial chip select input noise of one bus clock or more occurs during the slave mode transmission, this bit may be set to "1". In this case, resume transmission after the master transmission is completed.

[bit10, bit9] Reserved bits

Always set these bits to "0".

[bit8] TINT: Timer interrupt flag

When the serial timer register (STMR) matches the serial timer compare register (STMCR), the serial timer register (STMR) will be set to "0", and this bit will be set to "1".

When this bit is set to "1" and the timer interrupt enable bit (TINTE) is set to "1", a status interrupt request will be output.

Writing "0" to this bit will reset it to "0".

Writing "1" to this bit has no effect.

TINT	Description
0	No timer interrupt request
1	Timer interrupt request

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.
- When the synchronous transmission enable bit (TSYNE) is set to "1", this bit will not be set to "1".

[bit7] TINTE: Timer interrupt enable bit

This bit is used to enable/disable timer interrupts to the CPU.

When this bit is set to "1" and the timer interrupt flag (TINT) is set to "1", a status interrupt request will be output.

TINTE	Description
0	Interrupts by the serial timer disabled
1	Interrupts by the serial timer enabled

[bit6] TSYNE: Synchronous transmission enable bit

This bit is used to enable or disable synchronous transmission.

When this bit is set to "1", transmission will be activated if the serial timer register (STMR) matches the serial timer compare register (STMCR).

TSYNE	Description
0	Synchronous transmission disabled The serial timer will be used as a timer.
1	Synchronous transmission enabled The serial timer will not be used as a timer.

Notes:

- This bit can be changed only when the serial timer enable bit (TMRE) is set to "0".
- When synchronous transmission is enabled (TSYNE="1") and transmission is disabled (SCR:TXE="0"), transmission will not be activated, even if the serial timer register (STMR) matches the serial timer compare register (STMCR).
- In the slave mode (SCR:MS="1"), this bit will be internally fixed to "0".

[bit5] Reserved bit

Always set this bit to "0".

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	ϕ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	ϕ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

ϕ: Bus clock

Notes:

- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped. During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

Note:

To perform synchronous transmission by the serial timer change this bit under one of the following conditions:

- Transmission disabled (SCR:TXE="0")
- Transmission bus idle (SSR:TBI="1")

4.3.6. Serial Timer Register: STMR

This section explains the bit structure of the serial timer register.

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

■ STMRn(n=0 to 11) : Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bt15 to bit0] TM15-0: Timer data bits

These bits are used to indicate the timer value of the serial timer.

During timer operation, 1 will be added to the timer value of the serial timer for each timer operating clock (set by SACSR:TDIV3-0).

Note:

At the start of timer operation, these bits will be initialized to "0".

4.3.7. Serial Timer Compare Register: STMCR

This section explains the bit structure of the serial timer compare register.

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

■ STMCRn(n=0 to 11) : Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute
7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[bit15 to bit0] TC15-0: Compare bits

These bits are used to set compare values of the serial timer.

These bits will be compared with the serial timer register (SMTR), and when these bits match the value of the serial timer register immediately after the serial timer register (SMTR) is updated, they will set the serial timer register to "0". Then, if synchronous transmission is disabled (SACSR:TSYNE="0"), these bits will set the timer interrupt flag (SACSR:TINT) to "1", and if synchronous transmission is enabled (SACSR:TSYNE="1"), these bits will activate transmission.

The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0).

- SACSR:TINT is set to "1".
- Transmission is activated synchronizing with the serial timer.

Notes:

- When $(0000)_H$ is set to this register, the serial timer register will remain set to "0".
- When "0000_H" is set to this register with synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) will be fixed to "1", if the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B" during timer operation.
- This register can be changed only when the serial timer is disabled (SACSR:TMRE="0").

4.3.8. Serial Chip Select Control Status Register: SCSCR

This section explains the bit structure of the serial chip select control status register.

The serial chip select control status register (SCSCR) is used to select a start pin and an end pin for serial chip select, indicate an output pin for serial chip select, retain an active level of serial chip select, invert serial chip select, and enable/disable output of serial chip select pins.

Setting to serial chip select pins without assignment to external pins is disabled.
For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below.

Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)
Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)
Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)
Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)
Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

■ SCSCRn(n=0 to 11) : Address Base addr + $0E_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
SST1	SST0	SED1	SED0	SCD1	SCD0	SCAM	CDIV2	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R,WX	R,WX	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CDIV1	CDIV0	CSLVL	CSEN3	CSEN2	CSEN1	CSEN0	CSOE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15, bit14] SST1-0: Serial chip select start bits

These bits are used to select a pin where serial chip select starts.
If transmission disabled (SCR:TXE="0") status is changed to transmission enabled (SCR:TXE="1") status, the serial chip select pin set by these bits will first become active.

SST1	SST0	Start pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- If the same value is set to both the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0), only the serial chip select pin set by these bits will become active.
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Only the serial chip select pin with serial chip select enabled (CSEN="1") will become active.

[bit13, bit12] SED1-0: Serial chip select end bits

These bits are used to select a pin where serial chip select ends.

When the serial chip select pin set by these bits becomes active, the serial chip select pin specified by the serial chip select start bits (SST1, SST0) will become active next time.

SED1	SED0	End pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- If the same value is set to both the serial chip select start bits (SST1, SST0) and the serial chip select end bits (SED1, SED0), only the serial chip select pin set by these bits will become active.
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Only the serial chip select pin with serial chip select enabled (CSEN="1") will become active.

[bit11, bit10] SCD1-0: Serial chip select display bits

These bits are used to display a pin where serial chip select is active.

SCD1	SCD0	Display pin
0	0	SCS0
0	1	SCS1
1	0	SCS2
1	1	SCS3

Notes:

- If a serial chip select pin is inactive, the serial chip select pin which becomes active next time will be displayed.
- These bits will be set to "00_B" when the slave mode is selected (SCR:MS="1"), software reset is triggered (SCR:UPCL="1"), or transmission is disabled (SCR:TXE="0").

[bit9] SCAM: Serial chip select active retain bit

This bit is used to select whether to retain the active state of a serial chip select pin or not. When this bit is set to "1", the serial chip select pin will not become inactive, even if transmission operation has completed (SSR:TBI="1") after a serial chip select pin becomes active. If this bit is set to "0" when a serial chip select pin is active and this bit is set to "1", the serial chip select pin will become inactive after transmission has completed.

SCAM	Serial chip select active retain bit
0	Active state of a serial chip select pin not retained
1	Active state of a serial chip select pin retained

Notes:

- If transmission is disabled (SCR:TXE="0") and software reset is triggered (SCR:UPCL="1"), a serial chip select pin will become inactive regardless of the value of this bit.
- When a serial chip error occurs (SACSR:CSE=1), a serial chip select pin will become inactive regardless of the value of this bit.
- If this register is used when all of the following conditions are satisfied, set this bit to "0".
  - The master mode (SCR:MS=0) is used.
  - The chip select is used.
  - The SPI mode (SCR:SPI=1) is used.
  - "01_H" is set in TBYTE register.
  - A value other than "00" is set to WT1 and WT0 bits of ESCR register.

[bit8 to bit6] CDIV2-0: Serial chip select timing operating clock division bits

These bits are used to set the division ratio of a serial chip select timing operating clock.

CDIV2	CDIV1	CDIV0	Serial chip select timing operating clock						
			Division ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs

ϕ: Bus clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Settings other than those listed above are prohibited.

[bit5] CSLVL: Serial chip select level setting bit

This bit is used to select "H" or "L" for the level when a serial chip select pin is inactive.

This bit is used for communication of a chip select pin 0.

CSLVL	Serial chip select level setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- Setting this bit is used under one of the following conditions:
  - Slave mode (SCR:MS="1")
  - Data format of chip select is disabled (ESCR:CSFE="0")
  - Data format of chip select is enabled (ESCR:CSFE="1") and a serial chip select pin 0 is active

[bit4 to bit1] CSEN3-0: Serial chip select enable bits

These bits are used to enable or disable each serial chip select pin.
CSEN3 bit corresponds to SCS3 pin, CSEN2 bit to SCS2 pin, CSEN1 bit to SCS1 pin, and CSEN0 bit to SCS0 pin.
In the slave mode (SCR:MS="1"), only CSEN0 bit is used to enable or disable a serial chip pin.

CSEN	Serial chip select enable bit
0	Operation of a serial chip select pin disabled
1	Operation of a serial chip select pin enabled

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the master mode (SCR:MS="0"), if CSEN3-0 are set to "0000_B", transmission/reception operation will be performed regardless of a serial chip select pin.
- In the slave mode (SCR:MS="1"), if CSEN0 is set to "0", transmission/reception operation will be performed regardless of a serial chip select pin.

[bit0] CSOE: Serial chip select output enable bit

This bit is used to enable or disable output of serial chip select pins.

CSOE	Serial chip select output enable bit
0	Output of all serial chip select pins disabled
1	Output of all serial chip select pins enabled

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), set this bit to "0".

4.3.9. Serial Chip Select Timing Register: SCSTR3-0

This section explains the bit structure of the serial chip select timing register.

The serial chip select timing register (SCSTR3-0) is used to set the setup delay time for serial chip select, the hold delay time for serial chip select, and the deselect time for serial chip select.

Setting to serial chip select pins without assignment to external pins is disabled.
For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below.

Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)
Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)
Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)
Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)
Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

Note:

· In the master mode (SCR:MS=0) and the normal mode (SCR:SPI=0), set the setup delay time (CSSU7-0) or the hold

delay time (CSHD7-0) so that either of the following conditions is satisfied.
Baud rate/2[ns] < hold delay [ns] + 3 × bus clock [ns]
Hold delay + setup delay < baud rate - 2 × bus clock [ns]

■ SCSTR1n-0n(n=0 to 11) : Address Base addr + 12_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8] CSSU7-0: Serial chip select setup delay bits

These bits are used to set a time interval between the timing when a serial chip select pin becomes active and the timing when the serial clock is output. If "$00"_H$ is set to these bits, the timing when the serial clock is output will be the same as the timing when a serial chip select pin becomes active.

CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	Setup delay time
0	0	0	0	0	0	0	0	No setup delay time
0	0	0	0	0	0	0	1	1× Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2× Serial chip select timing operating clock
. .	. .	. .	. .	. .	. .	. .	. .	. .
1	1	1	1	1	1	1	0	254× Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255× Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- If these bits are set to "$00"_H$, the timing when a serial chip select pin becomes active will be the same as the timing when the serial clock outputs an edge for the first time.

[bit7 to bit0] CSHD7-0: Serial chip select hold delay bits

These bits are used to set a time interval between the timing when an output of the serial time clock is terminated the timing when a serial chip select pin becomes inactive. If "00"$_H$ is set to these bits, the timing when an output of the serial time clock is terminated will be the same as the timing when a serial chip select pin becomes inactive.

CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0	Hold delay time
0	0	0	0	0	0	0	0	No hold delay time
0	0	0	0	0	0	0	1	1× Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2× Serial chip select timing operating clock
. .	. .	. .	. .	. .	. .	. .	. .	. .
1	1	1	1	1	1	1	0	254× Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255× Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- If these bits are set to "00"$_H$, the timing when a serial chip select pin becomes active will be the same as the timing when the serial clock outputs an edge for the first time.

■ SCSTR3n-2n(n=0 to 11) : Address Base addr + 10_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] CSDS15-0: Serial chip deselect bits

These bits are used to set a minimum time interval between the timing when a serial chip select pin becomes inactive and the timing when the serial chip select pin becomes active next time.

CSDS 15	CSDS 14	CSDS 13	. . .	CSDS 2	CSDS 1	CSDS 0	Minimum deselect time
0	0	0	. . .	0	0	0	No minimum deselect time (5 bus clock time)
0	0	0	. . .	0	0	1	1× Serial chip select timing operating clock
0	0	0	. . .	0	1	0	2× Serial chip select timing operating clock
. .	. .	. .	. . .	. .	. .	. .	. .
1	1	1	. . .	1	1	0	65534× Serial chip select timing operating clock
1	1	1	. . .	1	1	1	65535× Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Regardless of the deselect time setting, once after the serial chip select pin becomes inactive, it will take at least five bus clock cycles to be active again.

4.3.10. Serial Chip Select Format Register: SCSFR2-0

This section explains the bit structure of the serial chip select format register.

The serial chip select format register (SCSFR2-0) is used to select an active level of chip select for each serial chip select, invert the serial clock, configure settings for connection with SPI, and set data direction and data length of serial data output.

Setting to serial chip select pins without assignment to external pins is disabled.
For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below.

- Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)
- Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)
- Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)
- Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)
- Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

■ SCSFR1n-0n(n=0 to 11) :Address Base addr + 16_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
CS2 CSLVL	CS2 SCINV	CS2 SPI	CS2 BDS	CS2 L3	CS2 L2	CS2 L1	CS2 L0	
1	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CS1 CSLVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0	
1	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] CS2CSLVL: Serial chip select level setting bit for chip select 2

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select the level when the serial chip select pin 2 is inactive.

CS2CSLVL	Serial chip select pin 2 Serial chip select setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit14] CS2SCINV: Serial clock invert bit for chip select 2

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to set the serial clock format when the serial chip select pin 2 is active.

When this bit is set to "0":
- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a rising edge of the serial clock in the normal transfer while it is sampled at a falling edge of the serial clock in the SPI transfer.

When this bit is set to "1":
- Serial clock output mark level is set to "L".
- Transmission data is output in synchronization with a rising edge of the serial clock in the normal transfer while it is output in synchronization with a falling edge of the serial clock in the SPI transfer.
- Reception data is sampled at a falling edge of the serial clock in the normal transfer while it is sampled at a rising edge of the serial clock in the SPI transfer.

CS2SCINV	Serial chip select pin 2 Serial clock invert bit
0	Mark level "H" format
1	Mark level "L" format

Notes:
- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit13] CS2SPI: SPI support bit for chip select 2

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to execute an SPI communication when the serial chip select pin 2 is active.
- When this bit is set to "0": Normal synchronous communication
- When this bit is set to "1": SPI communication supported.

CS2SPI	Serial chip select pin 2 SPI support bit
0	Normal synchronous transfer
1	SPI supported

Notes:
- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit12] CS2BDS: Transfer direction select bit for chip select 2

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select whether to transfer the transfer serial data from the least significant bit (LSB first, BDS="0") or from the most significant bit (MSB first, BDS="1") when the serial chip select pin 2 is active.

CS2BDS	Serial chip select pin 2 Transfer direction select bit
0	LSB first (transfer from the least significant bit)
1	MSB first (transfer from the most significant bit)

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit11 to bit8] CS2 L3, L2, L1, L0: Data length select bits for serial chip select pin 2

If data format of chip select is enabled (ESCR:CSFE="1"), these bits are used to specify the data length of transmission/reception data when the serial chip select pin 2 is active.

CS2L3	CS2L2	CSL1	CS2L0	Serial chip select pin 2 Data length select bits
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- Settings other than those listed above are prohibited.
- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting these bits has no effect.

[bit7] CS1CSLVL: Serial chip select level setting bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select the level when the serial chip select pin 1 is inactive.

CS1CSLVL	Serial chip select pin 1 Serial chip select setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit6] CS1SCINV: Serial clock invert bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to set the serial clock format when the serial chip select pin 1 is active.

When this bit is set to "0":

- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a rising edge of the serial clock in the normal transfer while it is sampled at a falling edge of the serial clock in the SPI transfer.

When this bit is set to "1":

- Serial clock output mark level is set to "L".
- Transmission data is output in synchronization with a rising edge of the serial clock in the normal transfer while it is output in synchronization with a falling edge of the serial clock in the SPI transfer.
- Reception data is sampled at a falling edge of the serial clock in the normal transfer while it is sampled at a rising edge of the serial clock in the SPI transfer.

CS1SCINV	Serial chip select pin 1 Serial clock invert bit
0	Mark level "H" format
1	Mark level "L" format

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit5] CS1SPI: SPI support bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to execute an SPI communication when the serial chip select pin 1 is active.

- When this bit is set to "0": Normal synchronous communication.
- When this bit is set to "1": SPI communication supported.

CS1SPI	Serial chip select pin 1 SPI support bit
0	Normal synchronous transfer
1	SPI supported

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit4] CS1BDS: Transfer direction select bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select whether to transfer the transfer serial data from the least significant bit (LSB first, BDS="0") or from the most significant bit (MSB first, BDS="1") when the serial chip select pin 1 is active.

CS1BDS	Serial chip select pin 1 Transfer direction select bit
0	LSB first (transfer from the least significant bit)
1	MSB first (transfer from the most significant bit)

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit3 to bit0] CS1 L3, L2, L1, L0: Data length select bits for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), these bits are used to specify the data length of transmission/reception data when the serial chip select pin 1 is active.

CS1L3	CS1L2	CS1L1	CS1L0	Serial chip select pin 1 Data length select bits
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- Settings other than those listed above are prohibited.
- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting these bits has no effect.

■ SCSFR2n(n=0 to 11) : Address Base addr + 15_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
CS3 CSLVL	CS3 SCINV	CS3 SPI	CS3 BDS	CS3 L3	CS3 L2	CS3 L1	CS3 L0	
1	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] CS3CSLVL: Serial chip select level setting bit for chip select 3

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select the level when the serial chip select pin 3 is inactive.

CS3CSLVL	Serial chip select pin 3 Serial chip select setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit6] CS3SCINV: Serial clock invert bit for chip select 3

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to set the serial clock format when the serial chip select pin 3 is active.

When this bit is set to "0":

- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a 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.

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.

CS3SCINV	Serial chip select pin 3 Serial clock invert bit
0	Mark level "H" format
1	Mark level "L" format

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit5] CS3SPI: SPI support bit for chip select 3

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to execute an SPI communication when the serial chip select pin 3 is active.

- When this bit is set to "0": Normal synchronous communication.
- When this bit is set to "1": SPI communication supported.

CS3SPI	Serial chip select pin 3 SPI support bit
0	Normal synchronous transfer
1	SPI supported

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit4] CS3BDS: Transfer direction select bit for chip select 3

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select whether to transfer the transfer serial data from the least significant bit (LSB first, BDS="0") or from the most significant bit (MSB first, BDS="1") when the serial chip select pin 3 is active.

CS3BDS	Serial chip select pin 3 Transfer direction select bit
0	LSB first (transfer from the least significant bit)
1	MSB first (transfer from the most significant bit)

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit3 to bit0] CS3 L3, L2, L1, L0: Data length select bits for chip select 3

If data format of chip select is enabled (ESCR:CSFE="1"), these bits are used to specify the data length of transmission/reception data when the serial chip select pin 3 is active.

CS3L3	CS3L2	CS3L1	CS3L0	Serial chip select pin 3 Data length select bits
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- Settings other than those listed above are prohibited.
- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting these bits has no effect.

4.3.11. Transfer BYTE register: TBYTE3-0

This section explains the bit structure of the transfer byte register.

The transfer byte register (TBYTE) is used to indicate the transfer data count when each serial chip select pin is active.

■ TBYTE3n-0n(n=0 to 11) : Address Base addr + 18_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TBYTE3[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TBYTE2[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
TBYTE1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TBYTE0[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8, bit7 to bit0] TBYTE: Transfer data count display bits

The transfer byte register allows you to set transfer data count when each serial chip select pin is active. After a serial chip select pin has become active, the serial chip select pin will become inactive when transfer of data count set to these bits has completed.

The serial chip select pin 0 (SCS0) corresponds to TBYTE0, the serial chip select pin 1 (SCS1) to TBYTE1, the serial chip select pin 2 (SCS2) to TBYTE2, and the serial chip select pin 3 (SCS3) to TBYTE3.

If one of the following conditions is satisfied, the transfer byte register 0 (TBYTE0) is used for synchronous transmission. When transmission operation is started with synchronous transmission, data count set to TBYTE0 will be transferred.

- Serial chip select is disabled (SCSCR:CSEN3-0="0000"b)
  If the value of these bits is changed during transmission operation (SSR:TBI=0), the changed transfer data count setting will take effect after the operation of transmitting transfer data count set before the change has completed.

TBYTE	Transfer byte register
Write	Writing to TBYTE
Read	Setting value of TBYTE

Notes:

- If "00_H" is set to these bits, transfer count is 8.
- If synchronous transmission is to be performed when chip select is used during the master operation (SCR:MS="0"), transfer count varies as follows:
  - Specified number of TBYTE0 if the chip select pin 0 is active
  - Specified number of TBYTE1 if the chip select pin 1 is active
  - Specified number of TBYTE2 if the chip select pin 2 is active
  - Specified number of TBYTE3 if the chip select pin 3 is active
- If this register is used when all of the following conditions are satisfied, set TBYTE register to a value other than "01_H".
  - The master mode (SCR:MS=0) is used.
  - The chip select is used.
  - The SPI mode (SCR:SPI=1) is used.
  - "1" is set to SCAM bit of SCSCR register.
  - A value other than "00" is set to WT1 and WT0 bits of ESCR register.

4.3.12. Baud rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock.

■ BGRn(n=0 to 11) : Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
RX,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	bit
BGR[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

No effect for writing operations.

[bit14 to bit0] BGR (Baud rate GeneratoR): Baud rate generator bits

- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.

- If the reload value is an even number, the "H" and "L" widths of the serial clock depend on the SCINV bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  - If SMR:SCINV="0", the "H" width of the serial clock is longer by one cycle of the bus clock.
  - If SMR:SCINV="1", the "L" width of the serial clock is longer by one cycle of the bus clock.
- Set the reload value to 3 or higher.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15'h00". Thus, if you wish to validate a new setting value immediately, execute CSIO reset (SCR:UPCL) after you have changed the setting value of BGR.
- To operate in the slave mode by setting "1" to the reception FIFO idle detection enable bit (FCR1:FRIIE) when you use reception FIFO, set the baud rate at the BGR.

4.4. Registers for LIN

Registers for LIN is shown.

4.4.1. Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission/reception interrupts, disable/enable transmission idle interrupts, and disable/enable transmissions and receptions. This register also has setups for generating LIN break field and resetting LIN interface reset (v2.1).

■ SCRn(n=0 to 11) : Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of LIN interface (v2.1). When this bit is set to "1": · Directly reset LIN interface (v2.1) (software reset). In this case, the register settings will be maintained. Note that any active transmission or reception will be cut off immediately. · Baud rate generator restarts by reloading the setting value of the BGR register. · All transmission and reception and status interrupt sources (SSR:TDRE, TBI, RDRF, FRE, ORE, LBD, TINT, and SFD) are initialized. · The baud rate setting flag (SACSR:BST) is initialized. When this bit is set to "0": No effect. For reading, "0" is always read out. Notes: · Execute a programmable clear after disabling interrupts. · When using FIFO, disable FIFO (FCR0:FE2,FE1=0) before you execute a programmable clear · Transmission/reception FIFO is not cleared by a programmable clear. · Serial timer register (SMTR) value will not be initialized when a programmable clear is executed (SSR:UPCL=1).
bit6	MS: Master/slave select bit	This bit selects master or slave mode. "0" is set: Master mode will be set. "1" is set: Slave mode will be set.
bit5	LBR: LIN break field setting bit (Functions only in the master operation)	· In case of LIN manual mode operation (LAMCR: LAMEN ="0"), when "1" is set to this bit, an LIN Break Field and an LIN Break Delimiter (set by the ESCR:LBL1/0 bit and ESCR:DEL1/0) are generated. · In case of LIN assist mode operation (LAMCR: LAMEN ="1"), when "1" is set to this bit, an LIN Break Field and an LIN Break Delimiter (set by the ESCR:LBL2/1/0 bit and ESCR:DEL1/0) are generated, and then a Sync Field and an ID Field are transmitted. Write: Writing "0": No effect. Writing "1": In case of LIN manual mode operation (LAMCR: LAMEN = "0"), generates LIN break field. In case of LIN assist mode operation (LAMCR: LAMEN = "1"), an LIN Break Field is generated, and then a Sync Field and an ID Field are transmitted. For reading, "0" will be always read out. Notes: · Functions only in the master operation (MS="0"). · To set an LIN Break Field during transmission of header or response, initialize the transmission/reception data. See "7.5.2 ●LIN Break Field retransmission processing in assist mode".
bit4	RIE: Reception interrupt enable bit	· This bit enables or disables the output of reception interrupt request to the CPU. · When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bits (SSR: FRE, ORE) is set to "1", a reception interrupt request will be output.

Bit name		Function
bit3	TIE: Transmission interrupt enable bit	· This bit enables or disables the output of transmission interrupt request to the CPU. · When the TIE bit and the SSR:TDRE are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	· This bit enables or disables the output of transmission bus idle interrupt request to the CPU. · When the TBIE bit and SSR:TBI bit are set to "1", a transmission bus idle interrupt request will be output.
bit1	RXE: Reception enable bit	This bit enables/disables the reception of LIN-interface (v2.1). · If this bit is set to "0", data frame reception is disabled. · If this bit is set to "1", data frame reception is enabled. Notes: · Even when you enable reception (RXE=1), does not start the reception until a falling edge of the start bit is input. · In the master operation mode, data will not be received even if receptions are enabled (RXE=1) during LIN break field transmission. · If data reception is disabled (RXE=0) during reception in manual mode (LAMCR:LAMEN=0), the current data reception is stopped immediately. · In assist mode (LAMCR:LAMEN=1), disable the reception (RXE=0) during header transmission/reception and response transmission. · Even if the reception operation is disabled (RXE=0) during header reception in assist mode (LAMCR:LAMEN=1), header reception operation does not stop. To stop this, disable the reception (RXE=0) and set to manual mode (LAMCR:LAMEN=0). · If data reception is disabled (RXE=0) during response reception in assist mode (LAMCR:LAMEN=1), the current data reception is stopped immediately. · If data reception is enabled (RXE=1) during LIN Break Field reception, a framing error will be detected (SSR:FRE=1).
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of LIN-interface (v2.1). · If this bit is set to "0", data frame transmission is disabled. · If this bit is set to "1", data frame transmission is enabled. Notes: · If data transmission is disabled (TXE=0) during transmission in manual mode (LAMCR:LAMEN=0), the current data transmission is stopped immediately. · Even if the transmission operation is disabled (TXE=0) during header transmission in assist mode (LAMCR:LAMEN=1), header transmission operation does not stop. To stop this, disable the transmission (TXE=0) and set to manual mode (LAMCR:LAMEN=0). · If data transmission is disabled (TXE=0) during response transmission in assist mode (LAMCR:LAMEN=1), the current data transmission is stopped immediately.

4.4.2. Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag and to detect the LIN break field as well as to clear the reception error flag.

■ SSRn(n=0 to 11) : Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved	LBD	FRE	ORE	RDRF	TDRE	TBI	
0	0	0	0	0	0	1	1	Initial value
R0,W	R0,W0	R(RM1), W	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears the FRE and ORE flags of the serial status register (SSR). · To clear an error flag, write "1" to this bit. · Writing "0" does not affect anything. A read always results in "0".
bit6	Reserved bit	Read: Always reads "0". Write: Always write "0".
bit5	LBD: LIN break field detection flag bit	This bit indicates that LIN break field is detected. When data with 11 or greater bits of "0" is input to serial input (SIN), LBD bit is set to "1". In this case, when "1" is set to the LIN break field interrupt enable bit (LBIE), a status interrupt will be generated. (When read) "1": LIN break field is detected. "0": LIN break field Is not detected. (When written) "0" is written: LBD bit will be cleared. "1" is written: No effect. Note: When a read-modify-write instruction is used, "1" will be read.

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 RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO. Notes: · If data reception is enabled (SCR:RXE=1) during LIN Break Field reception, a framing error will be detected before detecting LIN Break Field. However, reception of the header is not stopped. · In assist mode (LAMCR:LAMEN), if the master transmits another LIN Break between LIN Break Field detection and ID Field reception completion, a framing error will be detected at the "L" level of the tenth bit of the new LIN Break Field, even if data reception is disabled (SCR:RXE=0). However, reception of the header is not stopped.
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error "1" Read: There is an overrun error · If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). · When the ORE bit and RIE bit are set to "1", a reception interrupt request will be output. · If this flag is set, data contained in the receive data register (RDR) becomes invalid. · When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.

Bit name		Function
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: Receive data register (RDR) contains data. · The flag indicates the state of the receive data register (RDR). · When received data is loaded in RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". · When the RDRF bit and RIE bit are set to "1", a reception interrupt request will be output. · While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. · While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty. Note: In slave (SCR:MS=1) operation of assist mode (LAMCR:LAMEN=1), if LAMCR:LIDEN=0 is set to use the receive data register (RDR) for ID Field reception, the reception data full flag bit is set (RDRF=1) when loading the reception ID Field value to the receive data register (RDR). In addition, the LIN automatic header complete flag is also set (LAMSR:LAHC=1) at the same time.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register (TDR) is empty · The flag indicates the state of the transmit data register (TDR). · When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. · When the TDRE bit and the TIE bit are set to "1", a transmission interrupt request will be output. · When you set UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". · For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmitting "1" Read: No transmission operation · This bit indicates that LIN Interface (v2.1) has no transmission in progress. · When transmission data has been written to the transmit data register (TDR), this bit will become "0". · When the transmit data register (TDR) is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". · In manual mode (LAMCR:LAMEN=0) · When the LIN Break Field is set (SMR:LBR=1), this bit is set to "0". · If the Transmit Data Register becomes empty after the LIN Break Field has been transmitted, this bit is set to "1". · In assist mode (LAMCR:LAMEN=1) · During header transmission of master (SCR:MS=0), this bit is set to "0". · If the Transmit Data Register becomes empty after the header (ID Field transmission) has been transmitted, this bit is set to "1". · During response transmission, this bit is set to "0". · If the Transmit Data Register becomes empty after the response (checksum transmission) has been transmitted, this bit is set to "1". · When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output.

4.4.3. Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended serial control register (ESCR) is used to enable/disable LIN break field interrupt, detect LIN break field, set LIN break field length and Break delimiter length, and select stop bit length.

■ ESCRn(n=0 to 11) : Address Base addr + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	ESBL	LBL2	LBIE	LBL[1:0]		DEL[1:0]		
-	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	Reserved bit	"0" is read. Always write "0".

Bit name		Function
bit6	ESBL: Extended stop bit length select bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SBL="0" and ESCR:ESBL="0" are set: Stop bit is set to 1 bit. When SBL="1" and ESCR:ESBL="0" are set: Stop bit is set to 2 bits. When SBL="0" and ESCR:ESBL="1" are set: Stop bit is set to 3 bits. When SBL="1" and ESCR:ESBL="1" are set: Stop bit is set to 4 bits. Notes: · When receiving, only the first bit of the stop bits will always be detected. · This bit should be set when transmission is disabled (TXE=0). · In assist mode (LAMCR:LAMEN=1), set this bit before setting the LIN Break Field (SCR:LBR=1).
bit4	LBIE: LIN break field detection interrupt enable bit	This bit enables/disables LIN break field detection interrupt. A reception interrupt occurs when LIN break field detection flag (LBD) is set to "1" and interrupts are enabled (LBIE=1). Notes: · "0" is set: LIN break field detection interrupt is disabled · "1" is set: LIN break field detection interrupt is enabled
bit5, bit3, bit2	LBL[2:0]: LIN break field length select bits (Functions only in the master operation)	"000": 13-bit length "001": 14-bit length "010": 15-bit length "011": 16-bit length "100": 17-bit length "101": 18-bit length "110": 19-bit length "111": 20-bit length · These bits set the length of LIN break field generation time interval (in bits). · Before you set LBR bit in serial control register (SCR) to "1" (LIN break field send), set this bit. · The timing of LIN break field detect is always the 11th bit at slave operation, regardless of the set value of this bit. Notes: · This function is enabled only in the master operation (SMR:MS="0"). · Set this bit before setting the LIN Break Field (SCR:LBR=1). · In manual mode (LAMCR:LAMEN=0), this is operated between the 13-bit length and 16-bit length. Therefore, always set LBL2 to "0".
bit1, bit0	DEL[1:0]: LIN Break delimiter length select bits (Functions only in the master operation)	"00": 1-bit length "01": 2-bit length "10": 3-bit length "11": 4-bit length · These bits set the length of LIN Break delimiter (in bits). · Before you set LBR bit in serial control register (SCR) to "1" (LIN break field send), set this bit. Note: This function is enabled only in the master operation (SMR:MS="0").

4.4.4. Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

■ Read

■ RDR1n-0n(n=0 to 11) : Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).
- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When the reception interrupt is enabled (SSR:RIE=1), reception interrupt requests will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the serial receive data register (RDR) has been read out.
- In case a reception error occurs (either SSR:ORE or SSR:FRE is "1"), data in the receive data register (RDR) will become invalid.

Notes:

- If a reception error occurs, data in the Receive Data Register (RDR) is invalid.
- Operations in assist mode (LAMCR:LAMEN=1) are shown below.
  - In slave operation, if LAMCR:LIDEN=1 is set to use the LIN assist mode reception ID register for ID Field reception, the reception ID value is not stored in the receive data register (RDR) when receiving ID Field, and the reception data full flag bit (SSR:RDRF) is not set.
  - In slave operation, if LAMCR:LIDEN=0 is set to use the receive data register (RDR) for ID Field reception, the reception ID value is stored in the receive data register (RDR) when receiving ID Field. However, the reception data full flag bit (SSR:RDRF) is not set. Check the ID value by setting the LIN automatic header completion flag (LAMSR:LAHC=1).
  - Sync Field and checksum are not stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.
  - The transmission data for each field is not stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.

- If a reception error (SSR:FRE, ORE, LAMESR:LCSER, LSFER, LBSER, LPTER) occurs, the transmission/reception processing of assist mode is stopped. At this time, the response reception processing stops the reception data storing operation to the receive data register, regardless of data reception setting (SCR:RXE=1).

- When using reception FIFO, the operation is shown below.
  - When you use reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to SSR:RDRF.
  - When you are using reception FIFO, if the reception FIFO becomes empty, SSR:RDRF will be cleared to "0".
  - If a reception error occurs (either SSR:ORE or SSR:FRE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.

■ Write

■ TDR1n-0n(n=0 to 11) : Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
-	-	-	-	-	-	-	-	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output from the serial data output pin (SOT pin).
- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the serial transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- If the transmission data empty flag (SSR:TDRE) is "1", the next transmission data can be written. If the transmission interrupt is enabled, a transmission interrupt will occur. Writing the next transmission data should be performed by the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".
- You will not be able to write transmission data to the transmit data register (TDR) when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.

Notes:

- Transmit data register is write-only register and receive data register is read-only register. Because the two registers are located in the same address, write value and read value might be different. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operation cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR: TDRE) when using the transmission FIFO, see "7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

4.4.5. Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

The serial aid control status register (SACSR) allows you to control serial test operations, enable/disable timer interrupts, enable/disable synchronous transmission, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

■ SACSRn(n=0 to 11) : Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	BST	SFD	SFDE	AUTE	Reserved		TINT	
0	0	0	0	0	0	0	0	Initial value
R,W	R,WX	R(RM1), W	R/W	R,W	RX,W0	RX,W0	R(RM1), W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	Reserved		TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W0	R,W0	R,W	R,W	R,W	R,W	R/W	Attribute

[bit15] STST: Serial test bit

This bit is used to enable or disable the serial test mode.

When the serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed.

When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input into the SIN pin will be ignored.

STST	Serial test bit
0	Serial test mode disabled
1	Serial test mode enabled

Note:

This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

[bit14] BST: Baud rate setting flag

This bit indicates that automatic baud rate adjustment due to Sync Field reception was executed.

This bit is updated if Sync Field detects the fifth fall of the LIN bus.

BST	Baud rate setting flag	
	Write	Read
0	No influence	Automatic baud rate adjustment disabled
1		Automatic baud rate adjustment enabled

Notes:

· When automatic baud rate adjustment is disabled (AUTE=0), this bit will be fixed to "0".
· When a software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
· This bit is valid only when the sync field detection flag (SACSR:SFD) is "1".
· Writing to this bit has no effect.

[bit13] SFD: Sync Field detection flag

This bit is used to indicate that Sync Field was detected.
When the fifth falling edge of LIN bus is detected in Sync Field, this bit will be set to "1".
When this bit is set to "1" and the Sync Field detection interrupt enable bit (SFDE) is set to "1", a status interrupt request will be output.
Writing "0" to this bit will reset it to "0".

SFD	Sync Field detection flag	
	Write	Read
0	Clear	No Sync Field detected
1	No influence	Sync Field detected

Notes:

· When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
· Writing "1" to this bit has no effect.
· In both the master mode (SCR:MS="0") and the slave mode (SCR:MS="1"), this bit takes effect.
· A read with a read-modify-write instruction will read "1".

[bit12] SFDE: Sync Field detection interrupt enable bit

This bit is used to enable/disable Sync Field interrupts to the CPU.

When this bit is set to "1" and the Sync Field detection flag (SFD) is set to "1", a status interrupt request will be output.

SFDE	Sync Field detection interrupt enable bit
0	Interrupts by Sync Field detection disabled
1	Interrupts by Sync Field detection enabled

[bit11] AUTE: Automatic baud rate adjustment bit

This bit is used to enable/disable automatic baud rate adjustment.

AUTE	Automatic baud rate adjustment bit
0	Automatic baud rate adjustment disabled
1	Automatic baud rate adjustment enabled

Notes:

- In the master mode (SCR:MS="0"), this bit will be internally fixed to "0".
- When this bit is set to "1", the timer operating clock division bit (TDIV3-0) will be set to "3_H" (8 divisions).
- This bit can be changed from "0" to "1" only when the serial timer enable bit (TMRE) is set to "0".

[bit10, bit9] Reserved bits

Always set these bits to "0".

[bit8] TINT: Timer interrupt flag

When the serial timer register (STMR) matches the serial timer compare register (STMCR), the serial timer register (STMR) will be set to "0", and this bit will be set to "1".

When this bit is set to "1" and the timer interrupt enable bit (TINTE) is set to "1", a status interrupt request will be output.

Writing "0" to this bit will reset it to "0".

Writing "1" to this bit has no effect.

TINT	Description	
	Write	Read
0	Clear	No timer interrupt request
1	No influence	Timer interrupt request

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.

[bit7] TINTE: Timer interrupt enable bit

This bit is used to enable/disable timer interrupts to the CPU.

When this bit is set to "1" and the timer interrupt flag (TINT) is set to "1", a status interrupt request will be output.

TINTE	Description
0	Interrupts by the serial timer disabled
1	Interrupts by the serial timer enabled

[bit6,bit5] Reserved bit

Always set these bits to "0".

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	ϕ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	ϕ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

ϕ: Bus clock

Notes:

- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

4.4.6. Serial Timer Register: STMR

This section explains the bit structure of the serial timer register.

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

■ STMRn(n=0 to 11) : Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TM15-0: Timer data bits

These bits are used to indicate the timer value of the serial timer.

During timer operation, 1 will be added to the timer value of the serial timer for each timer operating clock (set by SACSR:TDIV3-0).

Note:

At the start of timer operation, these bits will be initialized to "0".

4.4.7. Serial Timer Compare Register: STMCR

This section explains the bit structure of the serial timer compare register.

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

■ STMCRn(n=0 to 11) : Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] TC15-0: Compare bits

These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, the timer interrupt flag (SACSR:TINT) is set to "1".

Notes:

- When $(0000)_H$ is set to this register, the serial timer register will remain set to "0".
- When "0000_H" is set to this register with synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) will be fixed to "1", if the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B" during timer operation.
- This register can be changed only when the serial timer is disabled (SACSR:TMRE="0").
- If all the following conditions are satisfied, the serial timer register (STMR) might be reset to $(0000)_H$ before baud rate adjustment is made. Therefore, when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1", set a larger value to these bits than the value set by the Sync Field upper limit bit (SFUR).
  - The automatic baud rate adjustment bit (SACSR:AUTE) is set to "1"
  - These bits have a smaller value than the value set by the Sync Field upper limit bit (SFUR)

4.4.8. Sync Field Upper limit Register: SFUR

This section explains the bit structure of the Sync Field upper limit register.

The Sync Field upper limit register (SFUR) is used to set the upper limit of the value which can be set to the baud rate generator register for automatic baud rate adjustment.

■ SFURn(n=0 to 11) : Address Base addr + $0E_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	TU14	TU13	TU12	TU11	TU0	TU9	TU8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TU7	TU6	TU5	TU4	TU3	TU2	TU1	TU0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

The read value is "0". Writing has no effect on the operation.

[bit14 to bit0] TU14-0: Upper limit bits

These bits are used to set the upper limit of the value which can be set to the baud rate generator register (BGR) for automatic baud rate adjustment.

When the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1" and the slave mode is selected (SCR:MS="1"), the value of the serial timer register (STMR) will be set to the baud rate generator register (BGR), if the value of the serial timer register (STMR) after Sync Field is received is smaller than these bits and larger than the Sync Field lower limit register (SFLR).

Note:

These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".

4.4.9. Sync Field Lower limit Register: SFLR

This section explains the bit structure of the Sync Field lower limit register.

The Sync Field lower limit register (SFLR) is used to set the lower limit of the value which can be set to the baud rate generator register for automatic baud rate adjustment.

■ SFLR1n-0n(n=0 to 11) : Address Base addr + 12_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	TL14	TL13	TL12	TL11	TL0	TL9	TL8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

The read value is "0". Writing has no effect on the operation.

[bit14 to bit0] TL14-0: Lower limit bits

These bits are used to set the lower limit of the value which can be set to the baud rate generator register (BGR) for automatic baud rate adjustment.

When the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1" and the slave mode is selected (SCR:MS="1"), the value of the serial timer register (STMR) will be set to the baud rate generator register (BGR), if the value of the serial timer register (STMR) after Sync Field is received is smaller than the Sync Field upper limit register (SFUR) and larger than these bits.

Note:

These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".

4.4.10. Baud Rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

■ BGRn(n=0 to 11) : Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
EXT	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

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

[bit15] EXT (EXTernal clock): External clock select bit

This bit selects whether to use an internal clock source or an external clock source for the internal reload counter for baud rate generation. When setting EXT="0", the internal clock source will be used. When setting EXT="1", the external clock source will be used.

[bit14 to bit0] BGR (Baud rate GeneratoR): Baud rate generator bits

- These bits set division ratio of the serial clock.
- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15’h00". Thus, if you wish to validate a new setting value immediately, execute programmable clear (UPCL) after you have changed the setting value of BGR.
- When the reload value is an even number, As for the "H" width and "L" width of the reception serial clock, the "L" width is longer by 1 cycle of the bus clock. When the reload value is an odd number, the widths of "H" and "L" for serial clock become the same.
- Use a value 3 or greater for reload values. However, correct data may not be received depending on the baud rate error and reload setup value.
- When you change to the external clock setup(EXT=1) while baud rate generator is running, write "0" to the Baud Rate Generator (BGR) and perform programmable clear (UPCL), then set to the external clock (EXT=1).

4.4.11. LIN Assist Mode Status Register: LAMSR

This section explains the bit structure of the LIN assist mode status register.

LIN assist mode status register (LAMSR) is used in order to check the status of automatic header transmission/reception and to check whether there is any error flag received.

■ LAMSRn(n=0 to 11) : Address Base addr + 10_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
LER	SER	RDRF	TDRE	TBI	LCSC	Reserved	LAHC	
0	0	0	1	1	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,W	R0,W0	R,W	Attribute

[bit7] LER: LIN representative error flag bit

When the following errors occur, this bit is set to "1". As for conditions on setting and clearing the error flag bit, see the explanations for each bit of the LIN assist mode error status register (LAMESR).

- LIN bus error flag bit (LBSER)
- LIN Sync Data error flag bit (LSFER)
- LIN ID parity error flag bit (LPTER)
- LIN checksum error flag bit (LCSER)

LER	LIN representative error flag
0	No error
1	Error

Note:

In manual mode (LAMCR:LAMEN=0), the read value of this bit is always "0".

[bit6] SER: Serial interface representative error flag bit

When the following errors occur, this bit is set to "1".As for conditions on setting and clearing the error flag bit, see the explanations for each bit of the serial status register (SSR); "Serial Status Register: SSR".

- framing error flag bit (FRE)
- overrun error flag bit (ORE)

SER	Serial interface representative error flag
0	No error
1	Error

[bit5] RDRF: Reception data full flag bit

This bit is the same as the reception data full flag bit (RDRF) of the serial status register (SSR). As for the explanation for this bit, see "Serial Status Register: SSR".

[bit4] TDRE: Transmission data empty flag bit

This bit is the same as the transmission data empty flag bit (TDRE) of the serial status register (SSR). As for the explanation for this bit, see "Serial Status Register: SSR".

[bit3] TBI: Transmission bus idle flag bit

This bit is the same as the transmission bus idle flag bit (TBI) of the serial status register (SSR). As for the explanation for this bit, see "Serial Status Register: SSR".

[bit2] LCSC: LIN checksum arithmetic operation completion flag bit

This bit is a flag that shows the completion of LIN checksum arithmetic operation.
In LIN assist mode (LAMCR:LAMEN="1"), when a chunk of data with the set length (LAMCR:LDL3-0) and its checksum are received, the LIN checksum arithmetic operation is completed and then this bit is set to "1".
When the LIN checksum arithmetic operation completion flag bit (LCSC) and the checksum arithmetic operation completion interrupt enable bit (LCSCIE) are "1", the status interrupt request is output.

- When reading
  "1": The checksum arithmetic operation completion has been detected.
  "0": The checksum arithmetic operation completion has not been detected.

- When writing
  "0": The LCSC bit is cleared.
  "1": No influence.

LCSC	LIN checksum arithmetic operation completion flag	
	Write	Read
0	Clear LCSC flag	Performing checksum arithmetic operation or Waiting for the start of checksum arithmetic operation
1	No influence	Checksum arithmetic operation completed

Note:

If the read-modify-write instruction is executed, "1" will be read out.
In manual mode (LAMCR:LAMEN="0"), the read value of this bit is always "0".

[bit1] Reserved bit

This bit is a reserved bit. The read value is always "0". Be sure to write "0" to this bit.

[bit0] LAHC: LIN automatic header completion flag bit

- This bit is a flag that shows the state of LIN automatic header.
- In assist mode (LAMCR:LAMEN=1), when the LIN header is received this bit is set to "1".
- When the LIN automatic header completion flag bit (LAHC) and the LIN automatic header completion interrupt enable bit (LAHCIE) are "1", the status interrupt request is output.
- When reading the LIN automatic header receive ID register (LAMRID) after "1" is set to this bit, "0" is set.
  - When reading:
    "1": The LIN automatic header completion has been detected.
    "0": The LIN automatic header completion has not been detected.
  - When writing:
    "0": The LAHC bit is cleared.
    "1": No influence.

LAHC	LIN automatic header completion flag	
	Write	Read
0	Clear LAHC flag	Performing LIN automatic header reception or Waiting for reception
1	No influence	LIN automatic header reception completed

Note:

If the read-modify-write instruction is executed, "1" will be read out.
In manual mode (LAMCR:LAMEN="0"), the read value of this bit is always "0".

4.4.12. LIN Assist Mode Control Register: LAMCR

This section explains the bit structure of the LIN assist mode control register.

LIN assist mode control register (LAMCR) enables LIN automatic header processing, enables LIN ID register use, selects LIN checksum type, and clears TDR and sets LIN data length in LIN assist mode.

■ LAMCRn(n=0 to 11) : Address Base addr + 11_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
LDL3	LDL2	LDL1	LDL0	LTDRCL	LCSTYP	LIDEN	LAMEN	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R0, W	R/W	R/W	R/W	Attribute

[bit7 to bit4] LDL3, LDL2, LDL1, LDL0: LIN data length setting bit

These bits set 0 to 8 bytes as the LIN response data length.
Set the value of the data length for the setting value.
In transmission operation, after the data of this data length is transmitted, a checksum is generated and transmitted.
In reception operation, the checksum of the data, received after the previously received data with this data length, is

checked.

LDL3	LDL2	LDL1	LDL0	LIN data length setting bits
0	0	0	0	0 byte length
0	0	0	1	1 byte length
0	0	1	0	2 bytes length
0	0	1	1	3 bytes length
0	1	0	0	4 bytes length
0	1	0	1	5 bytes length
0	1	1	0	6 bytes length
0	1	1	1	7 bytes length
1	0	0	0	8 bytes length

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set these settings before transmitting/receiving the response part of the data.
- Be sure not to set "1001" to "1111" to the LDL3-0 bits.
- In a response transmission node, when the LIN data length is set to 0 byte length (LDL3-0="0000"), write the dummy value ("don't care") to the TDR register in order to transmit a checksum. At this time, the TDR setting value does not influence the checksum arithmetic operation.
- When the LIN data length is set to 0 bytes length (LDL3-0="0000"), value of the checksum is as follows.
  - When the standard checksum is set (LCSTYP=0), the value of the checksum is 0xFF.
  - When the extended checksum is set (LCSTYP=1), the value of the checksum is an inverted value of the ID Field.

[bit3] LTDRCL: Transmit data register clear bit

This bit clears the transmission data register (TDR).

"1": The transmission data register is reset.

"0": There is no influence on the operation.

LTDRCL	Transmission data register clear bit	
	write	read
0	No influence	The read value is always "0"
1	The transmission data register (TDR) is cleared	

Notes:

- A transmission FIFO is not reset even though the transmission data register is cleared.
- When the transmission FIFO is used, be sure to clear the transmission data register after clearing the transmission FIFO (FCR0:FCL1 or FCR0:FCL2).
- When the transmit data register (TDR) with data is cleared (LTDRCL=1) while reception FIFO is not used or is empty, the transmission data empty flag bit is set (SSR:TDRE=1 and LAMSR:TDRE=1).

[bit2] LCSTYP: LIN checksum type select bit

This bit selects the checksum type of LIN.

LCSTYP	LIN checksum type select bit
0	Standard checksum
1	Extended checksum

Notes:

- This function is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- For master, set this before generating an LIN Break field (LBR="1").
- For slave, fix this bit to "0".

[bit1] LIDEN: LIN ID register use enable bit

This bit enables the use of the LIN assist mode transmission/reception ID register (LAMTID/LAMRID).

- In master mode: (SCR:MS=0)
  "0": The transmission data register (TDR) is used in order to transmit the LIN ID Field.
  "1": The LIN assist mode transmission ID register (LAMTID) is used in order to transmit the LIN ID Field.

- In slave mode: (SCR:MS=1)
  "0": The receive data register (RDR) is used in order to receive the LIN ID Field.
  "1": The LIN assist mode reception ID register (LAMRID) is used in order to receive the LIN ID Field.

LIDEN	LIN ID register use enable bit	
	Master	Slave
0	Transmission data register (TDR) is used	Reception data register (RDR) is used
1	LIN assist mode transmission ID register (LAMTID) is used	LIN assist mode reception ID register (LAMRID) is used

Notes:

- This function is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set this setting before transmitting/receiving the header.
- When the reception FIFO is used and this setting is set to "1" (LAMRID is used), a received ID is not stored in the reception FIFO.

[bit0] LAMEN: LIN assist mode processing enable bit

This bit enables the LIN assist mode processing.

LAMEN	LIN assist mode enable bit
0	Manual mode
1	Assist mode

Notes:

- In manual mode, be sure to change this bit when transmission/reception is prohibited in LIN (SCR:RXE="0", SCR:TXE=0).
- In assist mode, be sure not to change this bit while LIN is operating, except for the change in setting by forced stop.
- If this bit is changed, perform software reset (SCR:UPCL=1).

4.4.13. LIN Assist Mode Interrupt Enable Register: LAMIER

This section explains the bit structure of the LIN assist mode interrupt enable register.

LIN assist mode interrupt enable register (LAMIER) enables/disables those interrupts of LIN checksum error, LIN ID parity error, LIN Sync Data error, LIN bus error, LIN checksum operation completion, and LIN automatic header completion,

■ LAMIERn(n=0 to 11) : Address Base addr + $1A_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	LCSERIE	LPTERIE	LSFERIE	LBSERIE	LCSCIE	Reserved	LAHCIE	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	Attribute

[bit7] Reserved bit

This bit is a reserved bit. The read value is "0". Be sure to write "0" to this bit.

[bit6] LCSERIE: LIN checksum error interrupt enable bit

This bit enables/disables the LIN checksum error interrupt request output to the CPU.

When the LCSERIE bit and the LAMESR:LCSER bits are set to "1", the reception interrupt request is output.

LCSERIE	LIN checksum error interrupt enable bit
0	Disable
1	Enable

[bit5] LPTERIE: LIN ID parity error interrupt enable bit

This bit enables/disables the LIN ID parity error interrupt request output to the CPU.

When the LPTERIE bit and the LAMESR:LPTER bits are set to "1", the reception interrupt request is output.

LPTERIE	LIN parity error interrupt enable bit
0	Disable
1	Enable

[bit4] LSFERIE: LIN Sync Data error interrupt enable bit

This bit enables/disables the LIN Sync Data error interrupt request output to the CPU.

When the LSFERIE bit and the LAHESR:LSFER bits are set to "1", the reception interrupt request is output.

LSFERIE	LIN Sync Data error interrupt enable bit
0	Disable
1	Enable

[bit3] LBSERIE: LIN bus error interrupt enable bit

This bit enables/disables the LIN bus error interrupt request output to the CPU.

When the LBSERIE bit and the LAMESR:LBSER bits are set to "1", the reception interrupt request is output.

LBSERIE	LIN bus error interrupt enable bit
0	Disable
1	Enable

[bit2] LCSCIE: LIN checksum arithmetic operations completion interrupt enable bit

This bit enables/disables the LIN checksum arithmetic operations completion interrupt request output to the CPU.

When the LCSCIE bit and the LAMSR:LCSC bits are set to "1", the status interrupt request is output.

LCSCIE	LIN checksum operations completion interrupt enable bit
0	Disable
1	Enable

[bit1] Reserved bit

This is a reserved bit. The read value is "0." Be sure to write "0" to this bit.

[bit0] LAHCIE: LIN automatic header completion interrupt enable bit

This bit enables/disables the LIN automatic header completion interrupt request output to the CPU.

When the LAHCIE bit and the LAMSR:LAHC bits are set to "1", the status interrupt request is output.

LAHCIE	LIN automatic header completion interrupt enable bit
0	Disable
1	Enable

4.4.14. LIN Assist Mode Transmission/Reception ID register: LAMTID / LAMRID

This section explains the bit structure of the LIN assist mode transmission/reception ID register.

LIN assist mode transmission/reception ID register (LAMTID/LAMRID) sets a value of the transmission LIN ID, indicates a parity value of the received LIN ID, and indicates a value of the received LIN ID.

■ LIN Assist Mode Transmission ID register (LAMTID)

■ LAMTIDn(n=0 to 11) : Address Base addr + $1B_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	Reserved	LID5	LID4	LID3	LID2	LID1	LID0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

[bit7, bit6] Reserved bit

This is a reserved bit. Be sure to write "0" to this bit.

[bit5 to bit0] LID5 to LID0: LIN ID setting bit

(when the LIN ID are written):

When the LIN assist mode is set as master mode and LIN ID register use enable bit (LIDEN) is enabled, these bits set a value of the transmission LIN ID.

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set this setting (SCR:LBR="1") before LIN Break activates.

■ LIN Assist Mode Reception ID register (LAMRID)

■ LAMRIDn(n=0 to 11) : Address Base addr + $1B_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
P1	P0	LID5	LID4	LID3	LID2	LID1	LID0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit7, bit6] P1, P0: LIN ID parity indication bit

(when parity is read):

In assist mode, these bits indicate a parity value of the received LIN ID.

[bit5 to bit0] LID5 to LID0: LIN ID setting bit

(when ID is read):

In assist mode, these bits indicate a value of the received LIN ID.

Note:

Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").

4.4.15. LIN Assist Mode Error Status Register: LAMESR

This section explains the bit structure of the LIN assist mode error status register.

LIN assist mode error status register (LAMESR) checks those flags of the LIN checksum error, the LIN ID parity error, and the LIN bus error.

■ LAMESRn(n=0 to 11) : Address Base addr + 18_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	LCSER	LPTER	LSFER	LBSER	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R,W	R,W	R,W	R,W	R0,W0	R0,W0	R0,W0	Attribute

[bit7] Reserved bit

This is a reserved bit. The read value is "0". Be sure to write "0" to this bit.

[bit6] LCSER: LIN checksum error flag bit

When the LIN checksum error occurs, this bit is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LCSER bit and the LCSERIE bits are set to "1", the reception interrupt request is output.

LCSER	LIN checksum error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.

[bit5] LPTER: LIN ID parity error flag bit

When the LIN ID parity error is generated, it is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LPTER bit and the LPTERIE bit are set to "1", the reception interrupt request is output.
When this flag is read and found to be "1", the reception ID data is invalid.

LPTER	LIN ID parity error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.

[bit4] LSFER: LIN Sync Data error flag bit

When the LIN Sync Data error occurs, it is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LSFER bit and the LSFERIE bits are set to "1", the reception interrupt request is output.

LSFER	LIN Sync Data error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read
- This function detects an error only at automatic baud rate adjustment disabled (SACSR:AUTE="0") in slave mode (SCR:MS=1).
- Even if a framing error is detected at automatic baud rate adjustment disabled (SACSR:AUTE="0"), the result of the Sync Field value (0x55) collation is indicated. The result, however, is not guaranteed.

[bit3] LBSER: LIN bus error flag bit

When the LIN bus error occurs, this bit is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LBSER bit and the LBSERIE bits are set to "1", the reception interrupt request is output.
When this flag is read and found to be "1" in ID Field and the data field, the data of reception data register (RDR) is invalid.

LBSER	LIN bus error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.
- When a bus error is detected in the ID Field, the LIN ID parity error is detected as well.
- When a bus error is detected by checksum, the LIN checksum error is detected as well.

[bit2 to bit0] Reserved bit

This is a reserved bit. The read value is "0". Be sure to write "0" to this bit.

4.4.16. LIN Assist Mode trouble Examination Register: LAMERT

This section explains the bit structure of the LIN assist mode trouble examination register.

LIN assist mode trouble examination register (LAMERT) sets a pseudo trouble of the framing error, the LIN bus error, the LIN Sync Field error, the LIN ID parity error, and the LIN checksum error by setting the key code control bit and the pseudo trouble setting bit.

■ LAMERTn(n=0 to 11) : Address Base addr + 19_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
KEY1	KEY0	Reserved	LCSERT	LPTERT	LSFERT	LBSERT	FRET	
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] KEY1, KEY0: Key code control bit

These key code bits make the following pseudo trouble settings effective:

- Framing error pseudo trouble setting bit (FRET)
- LIN bus error pseudo trouble setting bit (LBSERT)
- LIN Sync Field error pseudo trouble setting bit (LSFERT)

- LIN ID parity error pseudo trouble setting bit (LPTERT)
- LIN checksum error pseudo trouble setting bit (LCSERT)

When setting a pseudo trouble, write to these bits in accordance with the following procedures:

1. KEY1-0="00"+ the pseudo trouble setting value is written.
2. KEY1-0="01"+ the pseudo trouble setting value (the same value as before) is written.
3. KEY1-0="10"+ the pseudo trouble setting value (the same value as before) is written.
4. KEY1-0="11"+ the pseudo trouble setting value (the same value as before) is written.
5. When being written in the fourth time, the pseudo trouble setting value becomes effective.

When the above-mentioned setup procedures are not followed (when another register is written or read in the middle of the writing procedures, when the value written is incorrect, or when these two bits are read in the middle of the writing procedures), writing to these two bits becomes invalid.

When releasing the pseudo trouble setting, follow the procedures similar to those to set.
As for the read value, "0" is read.

Note:

When the following error occurs in assist mode, the assist mode will stop.

- LIN bus error
- LIN framing error
- LIN Sync Data error
- LIN ID parity error
- LIN checksum error

[bit5] Reserved bit

This is a reserved bit. The read value is "0". Be sure to write "0".

[bit4] LCSERT: LIN checksum error pseudo trouble setting bit

This bit controls occurrence of the LIN checksum error.
In assist mode, when this bit is set to "1" (errors occur) before the start bit of the checksum, the inverted checksum is outputted. When the inverted checksum is received, an LIN checksum error occurs and the flag bit (LAMESR:LCSER) is set to "1".
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and causes errors.

LCSERT	LIN checksum error pseudo trouble setting bit
0	Occurrence of error
1	Non occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- The framing error is detected in the data field when the response transmission is processed after a pseudo trouble of the framing error and the LIN checksum error is set, and an automatic transmission of checksum stops.
- Therefore, the LIN checksum error is not detected.

[bit3] LPTERT: LIN ID parity error pseudo trouble setting bit

This bit controls occurrence of the LIN ID parity error.

In assist mode, when this bit is set to "1" (errors occur), before the start bit of the ID Field , all the inverted bits of the ID parity are output. When ID Field of the inverted ID parity is received, the LIN ID parity error occurs, and the flag bit (LAMESR:LPTER) is set to "1".

Until the setting of this bit is released (="0"), the pseudo trouble function is effective and causes errors.

LPTERT	LIN ID parity error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- After a pseudo trouble of the framing error and the LIN ID parity error is set, when the automatic header transmission is done, a framing error is detected in the Sync Field, and then the automatic transmission stops. As a result, the LIN ID parity error is not detected.
- When a pseudo trouble of the LIN bus error and the LIN ID parity error is set, the LIN bus error is given priority. As a result, the LIN ID parity error is not detected

[bit2] LSFERT: LIN Sync Data error pseudo trouble setting bit

This bit controls the occurrence of the LIN Sync Data error.

When the LIN assist mode is set as master mode (SCR:MS= "0") and this bit is set to "1" (errors occur) before the start bit of the Sync Field, all bits of the LIN Sync Field are output inverted.

Until the setting of this bit is released (="0"), the pseudo trouble function is effective and the output from the Sync Field is being kept inverted.

LSFERT	LIN Sync Data error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- If setting this bit (LSFERT="1"), set it together with the LIN bus error pseudo trouble setting bit (LBSERT="1"). For the master, the completion of the LIN Sync Data error pseudo trouble transmission can be identified with the detection of a LIN bus error.

[bit1] LBSERT: LIN bus error pseudo trouble setting bit

This bit controls the occurrence of the LIN bus error.

When the LIN assist mode is set as master mode, the LIN bus error will occur and the flag bit (LAMESR:LBSER) will be set to 1 in the following case:

This bit is set to "1" (errors occur) before the stop bit of each field (Sync Field, ID Field, data, and checksum).

When the LIN assist mode is set as slave mode, if this bit is set to "1" (errors occur) before the stop bit of each field (data and checksum) that does the response transmission, the LIN bus error will occur, and then the flag bit (LAMESR:LBSER) is set to "1".

Until the setting of this bit is released (="0"), the pseudo trouble function is effective and generates the error.

LBSERT	LIN bus error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- LIN bus error pseudo trouble of LIN Break Field cannot be set.
- When the LIN bus error and the LIN checksum error or the LIN ID parity error are set as a pseudo trouble at the same time, only the LIN bus error is detected. Neither the LIN checksum error nor the LIN ID parity error is detected.
- When the LIN bus error and the LIN Sync Data error or the framing error is set as a pseudo trouble before the LIN transmission start is set (SCR:LBR="1") at the same time, the LIN bus error and the LIN Sync Data error or the framing error is detected at the same time.
- When the LIN bus error is detected, transmission and reception in the assist mode will stop.

[bit0] FRET: Framing error pseudo trouble setting bit

This bit controls the occurrence of the LIN framing error.

In the assist mode, if this bit is set to "1" (errors occur) before the stop bit of each field (Sync Field, ID field, data field, and checksum field), the stop bit will be output inverted. When the inverted stop bit is received, the framing error will occur, and then the flag bit (SSR:FRE) is set to "1".

Until the setting of this bit is released (="0"), the pseudo trouble function is effective and generates the error.

FRET	Framing error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- After a pseudo trouble of the framing error and the LIN ID parity error is set, when the automatic header transmission is done, a framing error is detected in the Sync Field, and then the automatic transmission stops. As a result, the LIN ID parity error is not detected.
- After a pseudo trouble of the framing error and the LIN checksum error is set, when the response transmission is processed, the framing error is detected in the data field, and then the automatic transmission of checksum stops. As a result, the LIN checksum error is not detected.

4.5. Registers for I^2C

Registers for I^2C are shown.

4.5.1. I²C Bus Control Register: IBCR

This section explains the bit structure of the I²C bus control register.

I²C bus control register (IBCR) indicates master/slave mode selection, generation of repeat start condition, acknowledge enable, interrupt enable setting, and display of interrupt flag.

■ IBCRn(n=3 to 8, 10, 11) : Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R/W	R/W	R/W	R/W	R,WX	R(RM1), W	Attribute

Bit name		Function
bit7	MSS: Master/slave select 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 IBSR:BB bit becomes "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. MSS bit will be cleared on the following conditions. (1) I²C interface operation 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) Write "0" to the MSS bit when DMA mode is enable (SSR:DMA="1") and SSR:TBI="1" The relation between MSS bit and ACT bit is as follows. MSS=0, ACT=0 idle MSS=0, ACT=1 slave address matches or responds ACK * 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) *: ACK response: indicates that SDA of I²C bus is "L" in the acknowledge interval. Notes: · If the DMA mode is prohibited (SSR:DMA=0) and the MSS bit is set to "1", the MSS bit can be changed to "0" when MSS bit ="1" and INT bit ="1". 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 permitted (SSR:DMA=1) and the MSS bit is set to "1", the MSS bit can be changed to "0" when MSS bit ="1", INT bit ="1" or the SSR:TBI bit is "1". 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
bit7	MSS: Master/slave select bit	· If "1" is written to MSS bit during slave operation, this bit becomes "1" and AL bit of IBSR register is set to "1". To activate master mode after completing the slave operation, perform the steps below. 1. Be sure that the slave operation has been completed with the detection of the stop conditions (IBCR:SPC=1). 2. Write the slave address and transmission data (only when using transmission FIFO). 3. Write "1" to this bit.
bit6	ACT/SCC: Operation flag/repeat start condition generation bit	This bit differs in meanings between read and write. Read: ACT bit Write: SCC 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 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, SCC bit will be read. · When both of conditions below are met, "1" will be set to INT bit and I^2C bus will be waited (SCL="L"). It is necessary to write "1" in the SCC bit again and clear the INIT bit to generate the repeat start condition. · "1" is written to the SCC bit during master mode interrupt (MSS="1", ACT="1", INT="1", WSEL="1") in the eighth bit · NACK is received in the ninth bit · When you generate the repeat start condition while DMA mode is enabled (SSR:DMA="1"), SSR:TBI bit is "1" and IBCR:INT bit is "0", follow the steps below. 1. Write "1" to the IBCR:INT bit. 2. Make sure that "1" has been set to the IBCR:INT bit. 3. Write a slave address to the TDR. 4. Set "1" to this bit.

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 acknowledge. · This bit may be changed only when any of following conditions is met. · DMA mode is disabled (SSR:DMA="0"), ACT="1" and INT bit is "1" · DMA mode is enabled (SSR:DMA="1"), ACT="1" and SSR:TBI bit is "1" · DMA mode is enabled (SSR:DMA="1"), ACT="1" and SSR:RDRF is "1" when in slave reception · ACT="0" This bit will be disabled on the following conditions. (1) Acknowledge 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"), 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", responds with ACK and when the SSR:TDRE bit is 1, responds with NACK. (5) When reception FIFO is enabled, WSEL is "0", reserved address is detected, and slave is transmitted (IBSR:RSA="1", IBSR:TRX="1", IBSR:FBT="1"), 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"). "0": Acknowledge is disabled "1": Acknowledge is enabled
bit4	WSEL: Wait select bit	· When the DMA mode is prohibited (SSR:DMA=0), this bit selects whether an interrupt is generated (INT="1") before or after acknowledge and I^2C bus is waited. · When the DMA mode is permitted (SSR:DMA=1), this bit selects whether an interrupt is generated (INT="1", SSR:TBI ="1" at transmission, SSR:RDRF ="1" at reception) before or after acknowledge and I^2C bus is waited. · WSEL bit will be disabled on the following condition. (1) When an interrupt to the first byte*[1] generated (INT=1) (2) When a reserved address detected (IBSR:FBT="1", IBSR: RSA="1") (3) While the data transfer is in progress using FIFO and when NACK response *[2] 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. "0": Wait after acknowledge (9 bits) "1": Wait after data transmission/reception is completed (8 bits)
bit3	CNDE: Condition detect 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. "0": Repeat start/stop condition interrupt is disabled "1": Repeat start/stop condition interrupt is enabled
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"). "0": Interrupt disabled "1": Interrupt enabled

Bit name		Function
bit1	BER: Bus error flag bit	This bit indicates that an error has been detected on I^2C bus. BER bit set conditions: (1) While the first byte* transferring, detects a start condition or a stop condition. (2) For the second byte or later, 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) I^2C interface disable (ISMK:EN bit="0") *: The first byte: indicates data after the (repeat) start condition. "0": No error "1": Error detected Note: Check this flag when interrupt flag (INT bit) turns "1". If it is "1", as normal transmission/reception operations could not be performed, send the data again.
bit0	INT: Interrupt flag bit	Sets this flag to "1" when in master or slave mode, after 8 or 9 bits (ACK) of the data transmission/reception, or upon a bus error. When the INT bit is "1", state of SCL becomes "L" and when "0", exits from "L" state except for bus errors. INT bit set conditions: <8th bit> <When 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 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 "1" is written to the INT bit in the second byte or later in master operation <9th bit> < When unrelated to the DMA mode > (1) When arbitration lost is detected in the first byte (2) When NACK received except for stop condition output 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 transmission direction in master or slave mode (IBSR:TRX=1)

Bit name		Function
bit0	INT: Interrupt flag bit	(2) When DMA mode is disabled (SSR:DMA=0), the SSR:TDRE bit is "1" when reception FIFO is disabled without detecting the reserved address in the first byte in the receiving direction in master or slave mode (IBSR:TRX=0) (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 is enabled, 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 transmission direction in slave mode (IBSR:TRX=1) (2) When DMA mode is enabled (SSR:DMA=1), the SSR:TDRE bit is "1" when reception FIFO is disabled without detecting the reserved address in the first byte in the receiving direction in slave mode (IBSR:TRX=0). (3) When DMA mode is enabled (SSR:DMA=1), WSEL is "0" and you write "1" in the INT bit 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 permitted (SSR:DMA=1), writes "1" in the INT bit and the master mode is operating when the SSR:TBI bit is "1" in the second byte or later, status interrupt (SIRQ="1") is not generated. · When you issue the repeat start condition when the DMA mode is permitted (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below. 1. Write "1" to IBCR:INT bit. 2. Make sure that "1" has been set to the IBCR:INT bit. 3. Write a slave address to the TDR. 4. Set "1" to the IBCR:SCC bit. · When "0" is written in the INT flag when the INT flag is set in "1", the waiting 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 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. · Write "1" to this bit when issuing the start conditions (IBCR:MSS=1).

4.5.2. Serial Status Register: SSR

This section explains the bit structure of the serial status register.

Serial status register (SSR) checks for the transmission/reception states.

■ SSRn(3 to 8, 10, 11) : Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI	
0	0	0	0	0	0	1	1	Initial value
R0,W	R0,W	R/W	R/W	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears the ORE bit of serial status register (SSR) · Writing "1" clears the ORE bit. · Writing "0" does not affect anything. A read always results in "0".
bit6	TSET: Transmit buffer empty flag set bit	This bit sets the TDRE bit in serial status register (SSR) · When writing "1", the TBI bit is set when the TDRE bit and DMA mode are enabled (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 to "1", it becomes an interrupt condition corresponding to the DMA Transfer. · When this bit is set to "0", it becomes an interrupt condition corresponding to the case without the DMA Transfer. See "Table 8-1 I^2C Interface Interrupt Control Bits and Interrupt Factors" for details. "0": DMA mode is disabled "1": DMA mode is enabled Note: When ISMK:EN=0 only, this bit can be changed.
bit4	TBIE: Transmission bus idle interrupt enabled bit (Only the DMA mode enabled is effective.)	· This bit enables/disables transmission bus idle interrupt request output to the CPU. · The transmission bus idle interrupt request will be output when DMA mode is enabled (DMA="1") and both TBIE bit and TBI bit are "1". · When DMA mode is disabled (DMA="0"), this bit becomes "0" and any writing operation will be ignored and "0" will be retained. "0": Transmission bus idle interrupt request is disabled "1": Transmission bus idle interrupt request is enabled

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 reception FIFO is used, 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 SMR: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. · In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1". · Reception FIFO idle detection enable bit (FCR:FRIIE) is "1" · Data count contained in the reception FIFO does not reach the transfer count · IBCR:BER bit is "0" If RDR is read while the counter is counting 8 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 "1" in the acknowledge interval. Note: In the case where all of the conditions below are met, SCL is made "L" after ACK is transmitted and SCL releases the state of "L" when the RDRF bit becomes "0". · Reception FIFO is unused · DMA mode is enabled (IBCR:DMA="1") · RDRF bit is "1" while receiving second or latter byte data (IBSR:TRX="0") · IBCR:WSEL="0" In the case where all of the conditions below are met, SCL is made "L" after 1-byte data is received and SCL releases the state of "L" when the RDRF bit becomes "0". · Reception FIFO is unused · DMA mode is enabled (IBCR:DMA="1") · RDRF bit is "1" while receiving second or latter byte data (IBSR:TRX="0") · IBCR:WSEL="1" In case of reception with the DMA mode enabled (DMA=1) and reception FIFO used, SCL is made "L" when reception FIFO becomes full and SCL releases the state of "L" when data is read out from RDR even once.

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 SMR:TIE bit are set to "1", a transmission interrupt request will be output. · When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. · Writing "1" to 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.)	"0" Read: Transmitting "1" Read: Not transmitting This bit indicates that I^2C doesn't do the transmission operation when the DMA mode is enabled (DMA=1). When SCL is made "L", and the TBI bit becomes "0" when the TBI bit becomes "1" in the 2nd or subsequent byte in DMA mode permission (DMA=1), the state of "L" of SCL is released. Set condition of TBI bit: <8th bit> (1) In the 2nd or subsequent byte, the SSR: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" while reservation address is not detected in the first byte and the master is operating (2) In the 2nd or subsequent byte, the SSR:TDRE bit is "1" while WSEL is "0" and the master is operating (3) In the 2nd or subsequent byte, the SSR:TDRE bit is "1" while WSEL is "0" and the slave is transmitting <Other> When transmission buffer empty flag set bit (TSET) is set to "1" Reset condition of TBI bit: When writing transmission data in transmit data register (TDR) When this bit is "1" and transmission bus idle interrupt is enabled (SCR:TBIE=1), this bit outputs the transmission interrupt request. · When the DMA mode is disabled (DMA="0"), this bit becomes undefined.

4.5.3. I²C Bus Status Register: IBSR

This section explains the bit structure of the I²C bus status register.

I²C bus status register (IBSR) indicates that repeat start, acknowledges, data directions, arbitration lost, stop conditions, I²C bus states, and bus errors have been detected.

■ IBSRn(n=3 to 8, 10, 11) : Address Base addr + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
FBT	RACK	RSA	TRX	AL	RSC	SPC	BB	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R(RM1), W	R(RM1), W	R,WX	Attribute

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) A (repeat) start condition detected FBT bit clear conditions: (1) Transmission/reception of the 2nd byte (2) A stop condition detected (3) I²C interface disabled (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 disabled (ISMK:EN bit="0") (3) Bus error detected (IBCR:BER bit="1")

Bit name		Function
bit5	RSA: Reserved address detect 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 reset condition (RSA="0") (1) A (repeat) start condition detected (2) A stop condition detected (3) I^2C interface disabled (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) becomes "1" and SCL becomes "L" at SCL falling edge of the 8th bit on the first byte, regardless of the FIFO enable/disable state. At that time, when you are planning to read the received data and make it operate as slave, set IBCR:ACKE to "1" and set interrupt flag (IBCR:INT) to "0". After that, if the TRX bit is "0", the data is received as a slave. When you are planning to not receive data along the way, 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 detect interrupt occurs and a slave transmission is identified, 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) A (repeat) start condition is transmitted in master mode (2) The 8th bit of the first byte is "1" in slave mode (transmission direction as a slave) TRX bit reset conditions: (1) Arbitration lost is generated (AL="1") (2) The 8th bit of the first byte is "0" in slave mode (reception direction as a slave) (3) The 8th bit of the first byte is "1" in master mode (reception direction as a master) (4) A stop condition detected (5) When a (repeat) start condition is detected 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) Output data and received data are different in master mode. (2) You set "1" to the IBCR: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 was disabled in master mode. (6) Trying to generate a stop condition was disabled in master mode. AL bit reset conditions: (1) Writing "1" to the IBCR:MSS bit (2) Writing "0" to the IBCR:INT bit (3) Writing "0" to SPC bit when AL="1" and SPC="1" (4) I^2C interface disabled (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 disabled (ISMK:EN bit="0") There will be no effect on the operation of writing "1" to this bit. Notes: · If an acknowledge response is not made when receiving data as the slave mode due to the detection of the reserved address, "1" will not be set to this bit the next time the detection of a repeat start condition occurred 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 disabled (ISMK:EN bit="0") Writing "1" to this bit does not effect on the operation. Notes: · If an acknowledge response is not made when receiving data as the slave mode due to the detection of the reserved address, "1" will not be set to this bit the next time the detection of a stop condition occurred 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 disabled (ISMK:EN bit="0") (3) Bus error detected (IBCR: IBER bit="1")

4.5.4. Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

Receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

■ Read

■ RDR1n-0n(n=3 to 8, 10, 11) : Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial data line (SDA pin) 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

■ TDR1n-0n(n=3 to 8, 10, 11) : Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-								
-	-	-	-	-	-	-	-	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the data buffer register for sending serial data.

- 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).
- Transmission data empty flag (SSR:TDRE) will be set to "1" when transferred to the transmit shift register.
- 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:

Transmission data register is write-only register and receive data register is read-only register. Because the two registers are located in the same address, write value and read value might be different. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operations cannot be used.

4.5.5. Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

The serial aid control status register (SACSR) allows you to enable/disable timer interrupts, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

■ SACSRn(n=3 to 8, 10, 11) : Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved							TINT	
0	0	0	0	0	0	0	0	Initial value
R0,W0	RX,W0	R0,W0	R0,W0	R0,W0	RX,W0	RX,W0	R(RM1), W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	-	Reserved	TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R0,WX	RX,W0	R,W	R,W	R,W	R,W	R/W	Attribute

[bit15 to bit9] Reserved bit

Always set these bits to "0".

[bit8] TINT: Timer interrupt flag

When the serial timer register (STMR) matches the serial timer compare register (STMCR), the serial timer register (STMR) will be set to "0", and this bit will be set to "1".

When this bit is set to "1" and the timer interrupt enable bit (TINTE) is set to "1", a status interrupt request will be output.

Writing "0" to this bit will reset it to "0".

Writing "1" to this bit has no effect.

TINT	Description
0	No timer interrupt request
1	Timer interrupt request

Note:

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

[bit7] TINTE: Timer interrupt enable bit

This bit is used to enable/disable timer interrupts to the CPU.

When this bit is set to "1" and the timer interrupt flag (TINT) is set to "1", a status interrupt request will be output.

TINTE	Description
0	Interrupts by the serial timer disabled
1	Interrupts by the serial timer enabled

[bit6] Undefined

The read value is "0". Writing has no effect on the operation.

[bit5] Reserved bit

Always set this bit to "0".

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	ϕ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	ϕ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	ϕ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

ϕ: Bus clock

Notes:

- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped. During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

4.5.6. Serial Timer Register: STMR

This section explains the bit structure of the serial timer register.

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

■ STMRn(n=3 to 8, 10, 11) : Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TM15-0: Timer data bits

These bits are used to indicate the timer value of the serial timer.

During timer operation, 1 will be added to the timer value of the serial timer for each timer operating clock (set by SACSR:TDIV3-0).

Note:

At the start of timer operation, these bits will be initialized to "0".

4.5.7. Serial Timer Compare Register: STMCR

This section explains the bit structure of the serial timer compare register.

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

■ STMCRn(n=3 to 8, 10, 11) : Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] TC15 to TC0: Compare bits

These bits are used to set compared values of the serial timer.

These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, these bits will set the timer interrupt flag (SACSR:TINT) to "1".

The interval of the following operations is (STMCR: TC+1) x timer operating clock (set to SACSR:TDIV3-0).

- SACSR:TINT is set to "1".

Notes:

- When $(0000)_H$ is set to this register, the serial timer register will remain set to "0".
- When "0000_H" is set to this register, the timer interrupt flag (SACSR:TINT) will be fixed to "1", if the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B" during timer operation.
- This register can be changed only when the serial timer is disabled (SACSR:TMRE="0").

4.5.8. 7-bit Slave Address Mask Register: ISMK

This section explains the bit structure of the 7-bit slave address mask register.

7-bit slave address mask register (ISMK) compares and configures bits of slave address.

■ ISMKn(n=3 to 8, 10, 11) : Address Base addr + $1E_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	
0	1	1	1	1	1	1	1	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] EN: I^2C interface enable bit

This bit enables/disables I^2C interface operation.

If this bit is set to "0", I^2C interface becomes disabled.

If this bit is set to "1", I^2C interface becomes enabled.

EN	I^2C-UART operation enable bit
0	Disabled
1	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 I^2C interface is disabled (EN="0"), transmission/reception becomes disabled immediately.
- When you disable the I^2C 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 I^2C bus.

[bit6 to bit0] SM6-0: Slave address mask bits

These bits configure whether to exclude the 7-bit slave address and received address as the comparison targets.
If these bits are set to "0": treat as matched
If these bits are set to "1": compare

SM6-0	7-bit slave address mask bits
0	Bits not compared
1	Bits compared

Note:

Configure this register when the EN bit is "0".

4.5.9. 7-bit Slave Address Register: ISBA

This section explains the bit structure of the 7-bit slave address register.

7-bit slave address register (ISBA) sets slave addresses.

■ ISBAn(n=3 to 8, 10, 11) : Address Base addr + $1F_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[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.

SAEN	Slave address enable bit
0	Disabled
1	Enabled

[bit6 to bit0] SA6-0: 7-bit slave address

- If the slave address detect is enabled (SAEN=1), the 7-bit slave address register (ISBA) compares with the 7-bit data received after a (repeat) start condition detected, 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.

SA	Slave address setting bits
6 to 0	7-bit slave address

Notes:

- The reserved address cannot be set.
- Set this register when the EN bit of the ISMK register is "0".

4.5.10. Baud rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock.

■ BGRn(n=3 to 8, 10, 11) : Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	BGR[14:8]							
-	0	0	0	0	0	0	0	Initial value
RX,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	bit
BGR[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

No effect for writing operations.

[bit14] to bit0 BGR (Baud rate GeneratoR): Baud Rate Generator Bit

- These bits set division rate of the serial clock.
- Capable of writing reload value to be counted and reading setup values.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- Configure the baud rate generator register when the EN bit of the ISMK register is "0".
- Configure baud rate regardless of the master mode or slave mode.
- Peripheral clock (PCLK) should be set with 8MHz or more in operating mode 4 (I^2C mode) and baud rate generator configured in 400kbps or more should not be used.

5. Operation of UART

This section explains operation of UART.

5.1 Interrupt of UART

5.2 Operation of UART

5.3 Setup Procedure and Program Flow

5.1. Interrupt of UART

Interrupt of UART is shown.

There are interrupts for both transmission and reception in UART. You can generate an interrupt request for the following factors.

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request

5.1.1. List of Interrupt of UART

This section explains the list of interrupt of UART.

The following table indicates how UART interrupt control bits relate to interrupt factors.

Table 5-1 Interrupt Control Bits and the Interrupt Factors of UART

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt factor	Interrupt factor enable bit	Interrupt request flag clear
			0	1			
Reception	RDRF	SSR	✓	✓	1-byte reception	SCR:RIE	Reading of receive data (RDR)
					Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
					Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	✓	✓	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	✓	✓	Framing error		
	PE	SSR	✓	-	Parity error		
Transmission	TDRE	SSR	✓	✓	Transmission register is empty	SCR:TIE	Writing to the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*[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	✓	✓	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or the transmission FIFO is full

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt factor	Interrupt factor enable bit	Interrupt request flag clear
			0	1			
Status	TINT	SACSR	✓	✓	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR:TINTE	Writing "0" to the timer interrupt flag bit (SACSR:TINT)

*1: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

✓: Operation mode effective
-: Operation mode non-effective

5.1.2. Reception Interrupts and Flag Setting Timing

This section explains the generation of reception interrupts and flag setting timing.

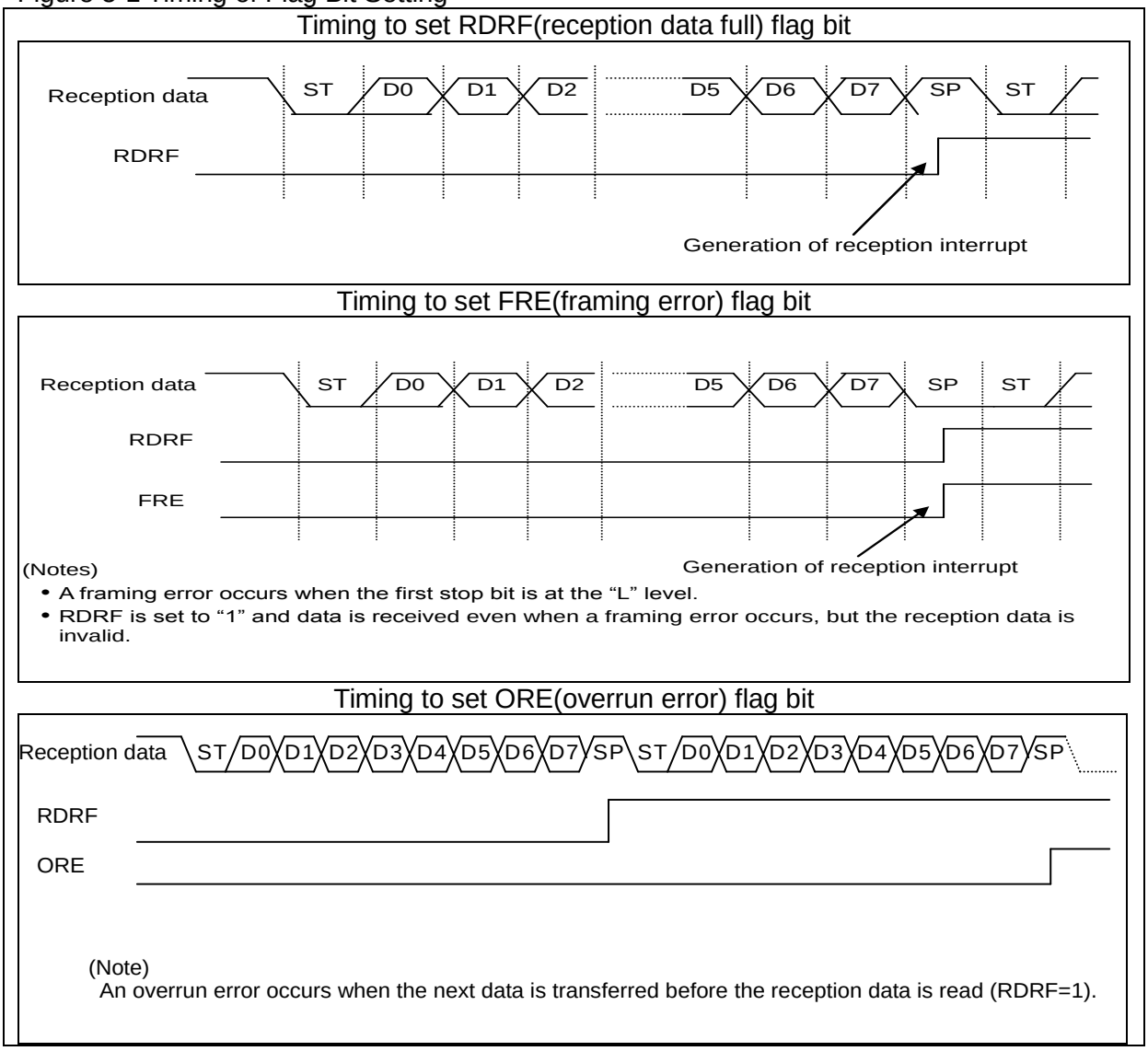
Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:PE, ORE, FRE).

When the first stop bit is detected, reception data is stored in the receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:PE, ORE, FRE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 5-1 Timing of Flag Bit Setting



Note:

When any of following conditions is detected while receiving at the same time of or 1 to 2 bus clocks before the sampling point for stop bit, its edge will be invalid and the next data may not be received correctly. To output frames continuously, some space is required between the frames.

- Trailing edge of serial data (when ESCR:INV="0")
- Rising edge of serial data (when ESCR:INV="1")

5.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using reception FIFO and flag setting timing.

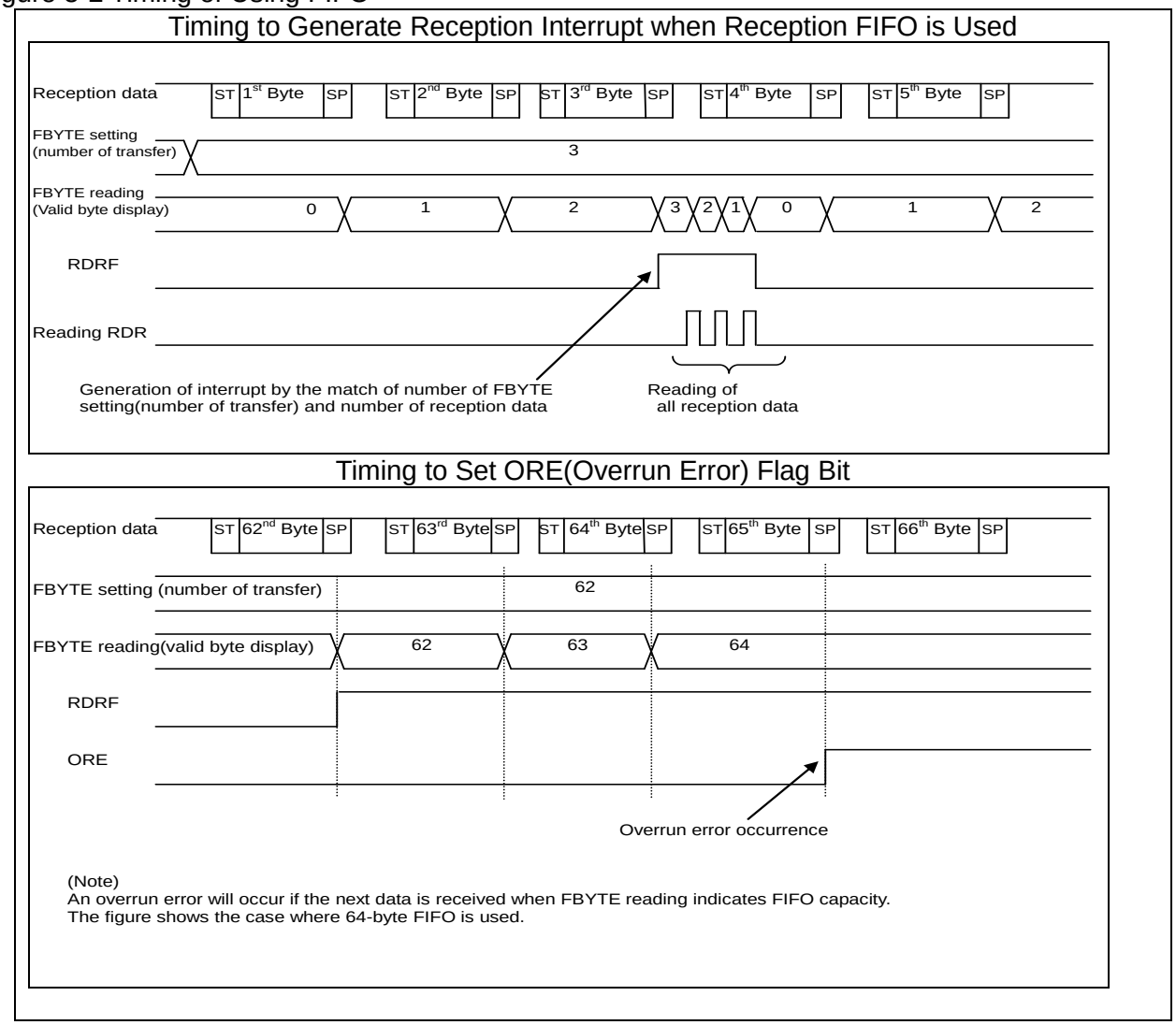
When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received. The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt is generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - Reception FIFO idle detection enable bit (FCR:FRIIE) is "1"
  - Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) is cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1) .

Figure 5-2 Timing of Using FIFO



5.1.4. Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

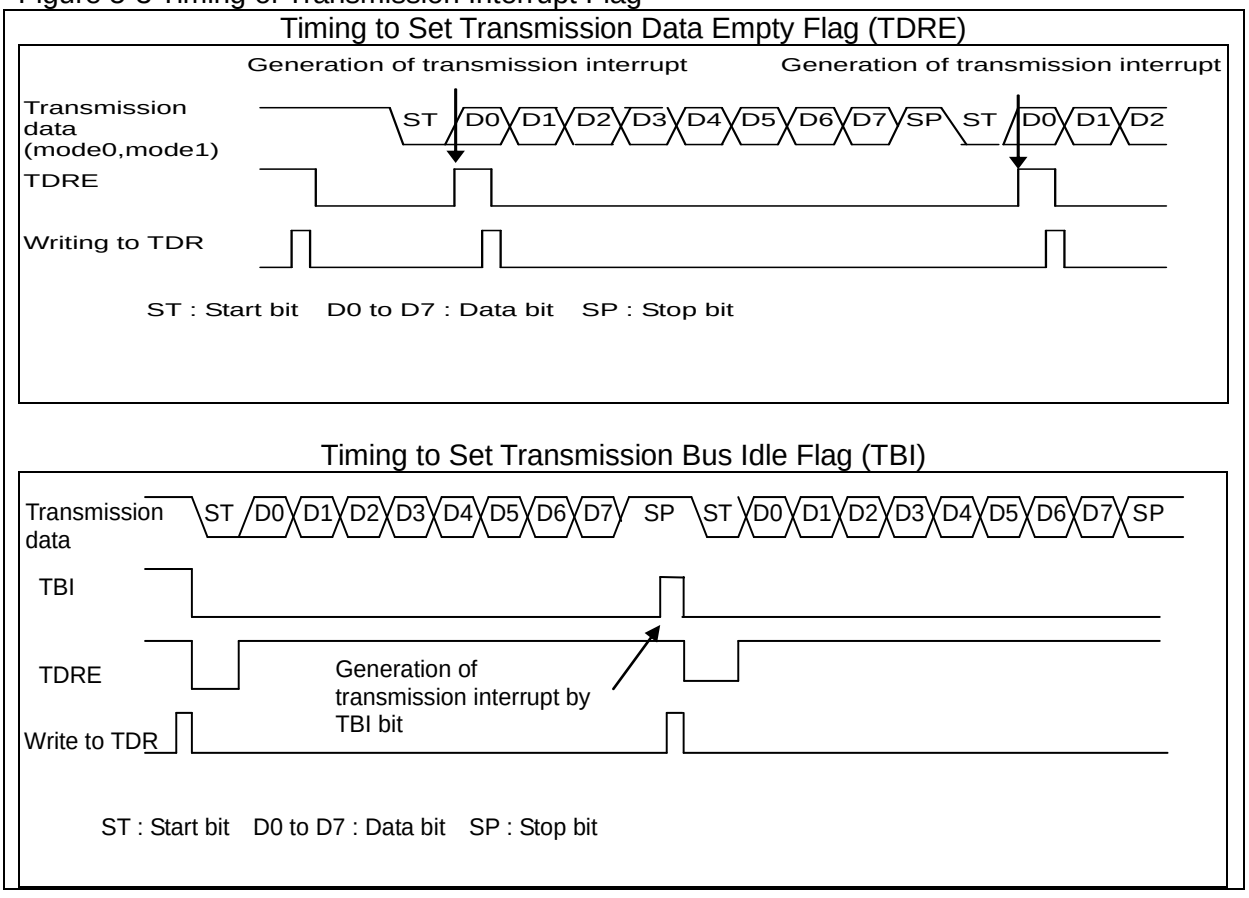
Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. The TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt will occur.

When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 5-3 Timing of Transmission Interrupt Flag



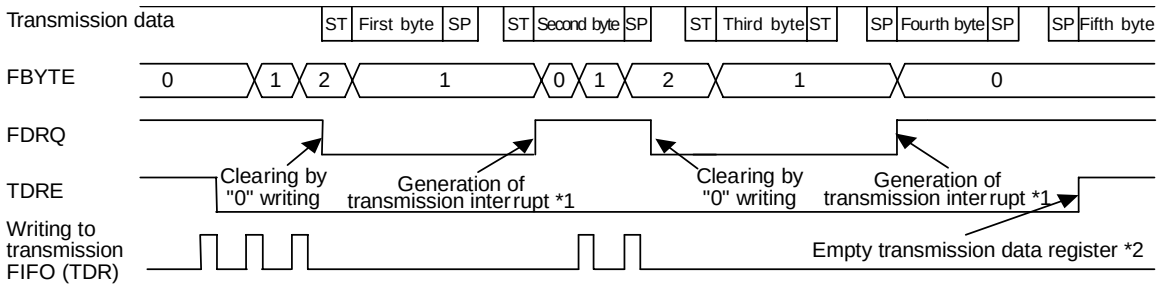
5.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and flag setting timing.

When the transmission FIFO is used, an interrupt is generated when the data count stored at the transmission FIFO is equal to or less than the count set for the FTICR register (FTICR).
When the transmission FIFO is used, the interrupt generation is decided depending on the FTICR register setting value.

- When the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1".
- If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
- When FBYTE=0x00 and FTICR=0x00, there is no data in the transmission FIFO.

Figure 5-4 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.

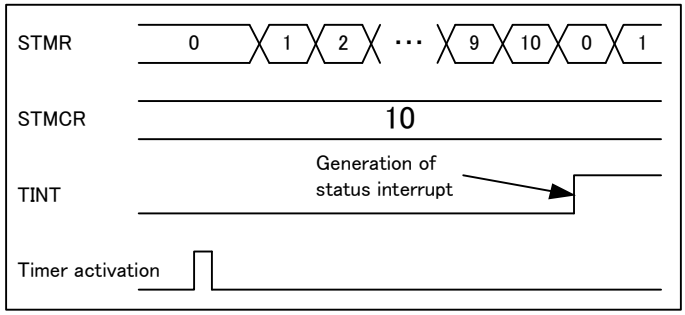
5.1.6. Timing of Timer Interrupt Generation and Flag Setting

This section explains the timing of timer interrupt generation and flag setting.

Timer interrupt is generated when Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR).

- When Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR), "1" will be set to timer interrupt flag (SACSR:TINT).
  At this time when the timer interrupt is enabled (SACSR:TINTE="1"), a status interrupt will be generated.

Figure 5-5 Timer Interrupt Generation Timing



5.2. Operation of UART

Operation of UART is shown.

UART operates with the mode 0 bidirectional serial asynchronous communication and the mode 1 master/slave multiprocessor communication.

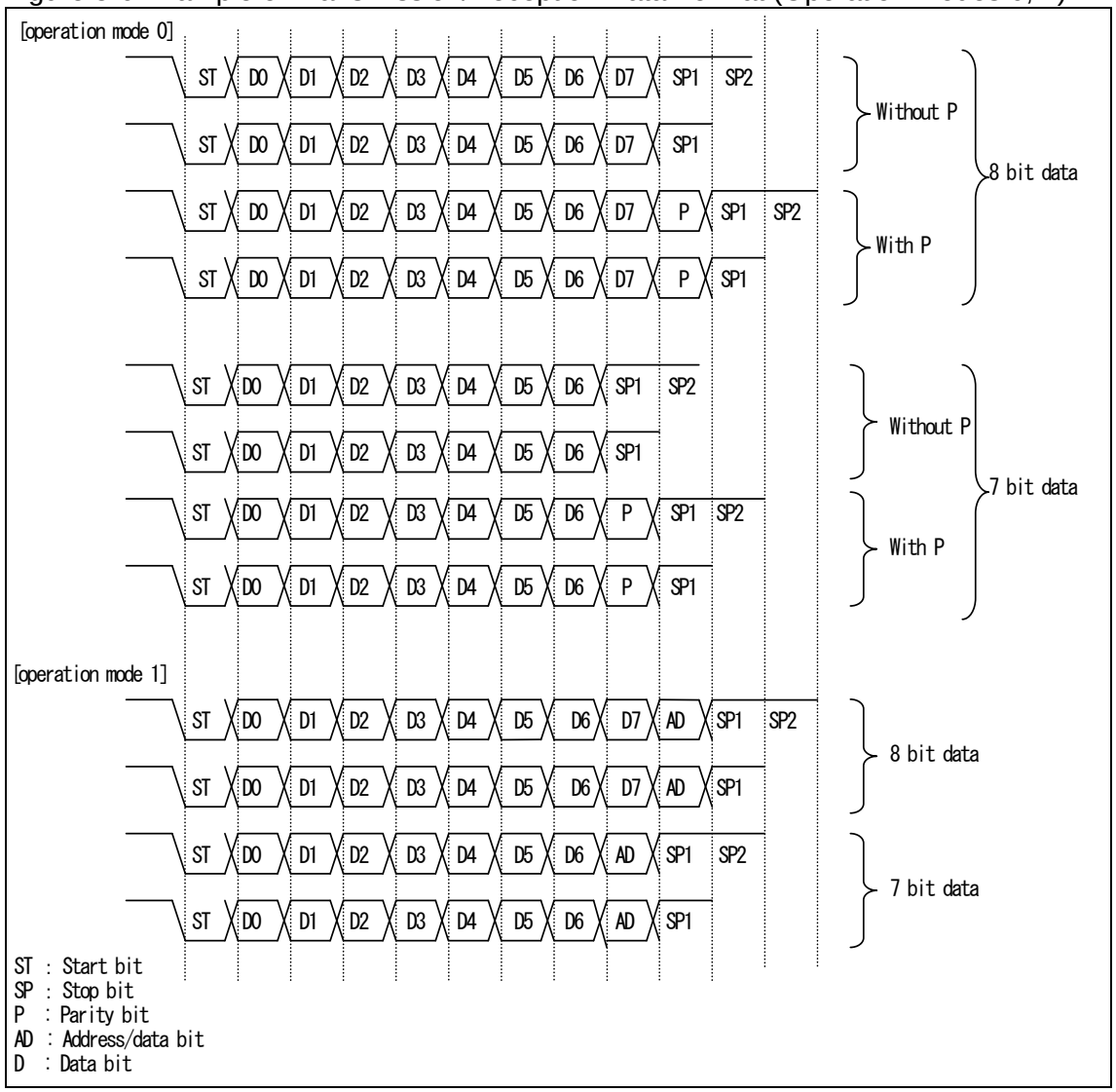
5.2.1. Transmission/Reception Data Format

This section explains the transmission/reception data format.

- The transmission/reception data always starts from the start bit and after the transmission/reception of data have taken place for the specified data bit length, ends at 1-bit or more length of stop bit.
- The direction of data transfer (LSB first or MSB first) is determined by the BDS bit of the serial mode register (SMR). If parity is used, the parity bit will always be placed between the last data bit and the first stop bit.
- In operation mode 0 (normal mode), you can select whether to use parity.
- In operation mode 1 (multiprocessor mode), the parity will not be added, instead AD bits will be added.

An example of transmission/reception data format (operation modes 0, 1) is shown in Figure 5-6:

Figure 5-6 Example of Transmission/Reception Data Format (Operation Modes 0, 1)



Notes:

- The Figure above shows the example of configurations with data length of 7 and 8 bits. (You can configure 5 to 9-bit data length in operation mode 0.)
- When you set "1" to the BDS bit of serial mode register (SMR) (MSB first), the bits will be processed in the order, D7, D6, D5, ..., D1, D0 (P).
- When you configure x bit of data length, the lower x bits on transmission/receive data register (RDR/TDR) will be enabled.

5.2.2. Transmission Operation

This section explains the transmission operation.

- If the transmission data empty flag bit (TDRE) of the serial status register (SSR) is "1", the transmission data can be written to the transmit data register (TDR). (If the transmission FIFO is enabled, transmission data can be written even if TDRE="0").
- When transmission data is written to the transmit data register (TDR), the transmission data empty flag bit (SSR:TDRE) becomes "0".
- When the transmission operation enable bit of the serial control register (SCR:TXE) is set to "1", the transmission data is loaded into the transmit shift register and the transmission starts from the start bit sequentially.
- When the transmission starts, the transmission data empty flag bit (SSR:TDRE) will be set to "1" again. If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. In interrupt processing, the next transmission data can be written to the transmit data register.

Notes:

- As soon as the transmission interrupt is enabled (SCR:TIE), a transmission interrupt occurs, because the transmission data empty flag bit (SSR:TDRE) has the initial value "1".
- As soon as the FIFO transmission interrupt is enabled (FCR1:FTIE=1), a transmission interrupt occurs, because the FIFO transmission data request bit (FCR1:FDRQ) has the initial value "1".

5.2.3. Reception Operation

This section explains the reception operation.

- When reception operation is enabled (SCR:RXE=1), the reception operation will start.
- When a start bit is detected, one frame data will be received according to the data format set in the extended communication control register (ESCR:PEN, P, L2, L1, L0) and serial mode register (SMR:BDS). The start bit is detected when the falling edge (at ESCR:INV="0") or the rising edge (at ESCR:INV="1") is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.
- When the reception of one frame data has completed, the reception data full flag bit (SSR:RDRF) will be set to "1". If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.
- Read reception data, after the one frame data reception has completed, and check for the state of error flag of the serial status register (SSR). When a reception error has detected, correct the error.
- After a read of reception data, the reception data full flag bit (SSR:RDRF) will be cleared to "0".
- When reception FIFO is enabled, if as many frames as set in the reception FBYTE have been received, the reception data full flag bit (SSR:RDRF) will be set to "1".
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (RDRF) will be set to "1".
  - Reception FIFO idle detection enable bit (FRIIE) is "1"
  - Data count contained in the reception FIFO does not reach the transfer count

  If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

  When the reception FIFO is enabled, if the error flag of the serial status register (SSR) is set to"1", the erroneous data will not be stored in the reception FIFO. Also, the reception data full flag bit (SSR:RDRF) at that time will not be set to "1". (However, when an overrun error does occur, the flag will be set to "1".) The reception FBYTE indicates the number of data items which have been successfully received before the error occurs. Unless the error flag of the serial status register (SSR) is cleared to "0", the reception FIFO will not be enabled.

- When the reception FIFO is enabled, if the reception FIFO has no more data, the reception data full flag bit (SSR:RDRF) will be cleared to "0".

Notes:

- The data on the receive data register (RDR) will be enabled when the receive data register full flag bit (SSR:RDRF) is set to "1" and a reception error does not occur (SSR:PE, ORE, FRE=0).
- When the noise passes the filter, the incorrect data is received though the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in. As measures against this, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, add the checksum of data at the end, and send it again if an error occurs).
- When any of following conditions is detected while receiving at the same time of or 1 to 2 bus clocks before the sampling point for stop bit, its edge will be invalid and the next data may not be received correctly. To output frames continuously, some space is required between the frames.
  - Trailing edge of serial data (when ESCR:INV="0")
  - Rising edge of serial data (when ESCR:INV="1")

5.2.4. Clock Selection

This section explains the clock selection.

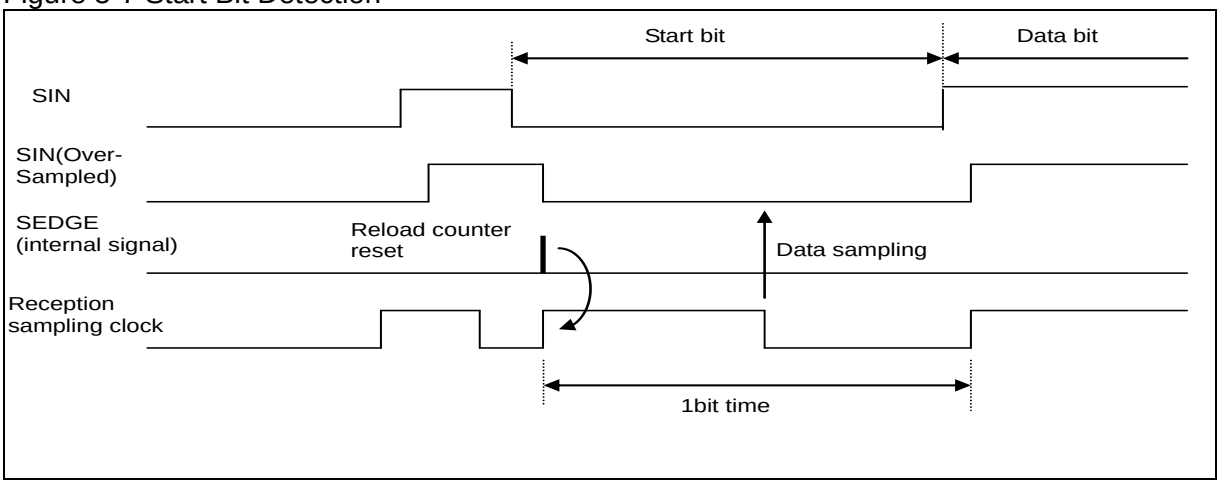
- Internal clocks or external clocks can be used.
- When you use an external clock, set BGR:EXT="1". In this case, the external clock is divided in the baud rate generator.

5.2.5. Start Bit Detection

This section explains the start bit detection.

- The start bit is recognized by the falling edge of the SIN signal in asynchronous mode.
  Therefore even if you enable reception operation (SCR:RXE="1"), the reception operation will not start unless the falling edge of the SIN signal is entered.
- When the falling edge of the start bit is detected, the reception reload counter of the baud rate generator will be reset, a reload will take place again, and the countdown will start. This will always launch a data sampling aimed at the center of the data.

Figure 5-7 Start Bit Detection



5.2.6. Stop Bit

This section explains the stop bit.

- You can select 1-4 bit length.
- The reception data full flag bit (SSR:RDRF) will be set to "1" when the first stop bit is detected.

5.2.7. Error Detection

This section explains the error detection.

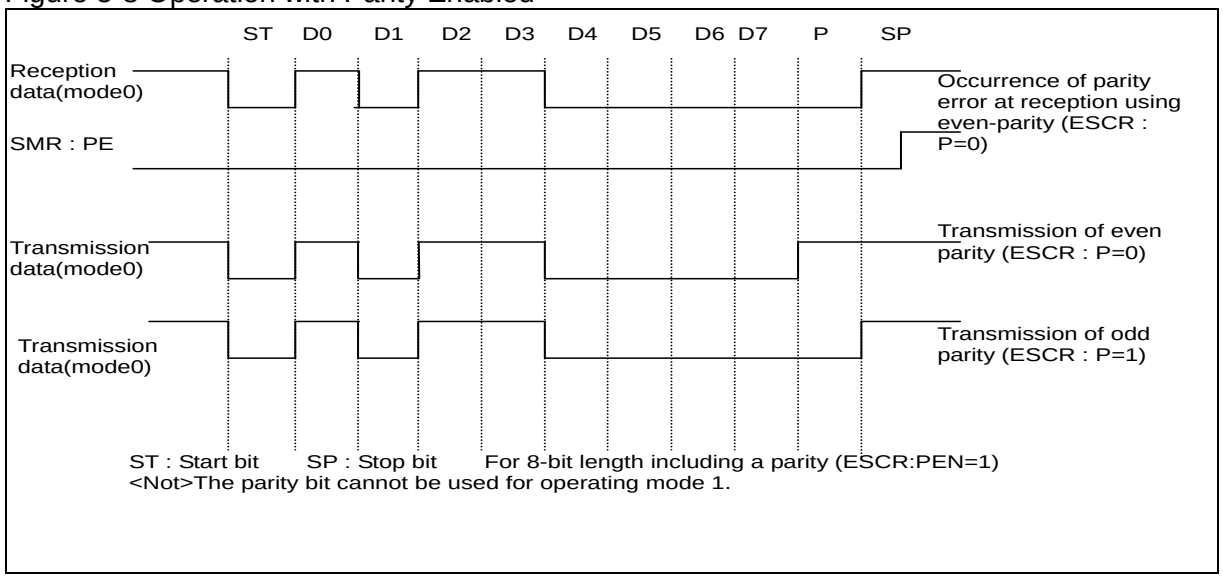
- In operation mode 0, parity errors, overrun errors, frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors cannot be detected.

5.2.8. Parity Bit

This section explains the parity bit.

- Parity bit can be added only in operating mode 0. The parity enable bit (ESCR:PEN) can specify whether to enable or disable the parity, and the parity selection bit (ESCR:P) can specify whether to use even parity or odd parity.
- Operation mode 1 does not use parity.

Figure 5-8 Operation with Parity Enabled

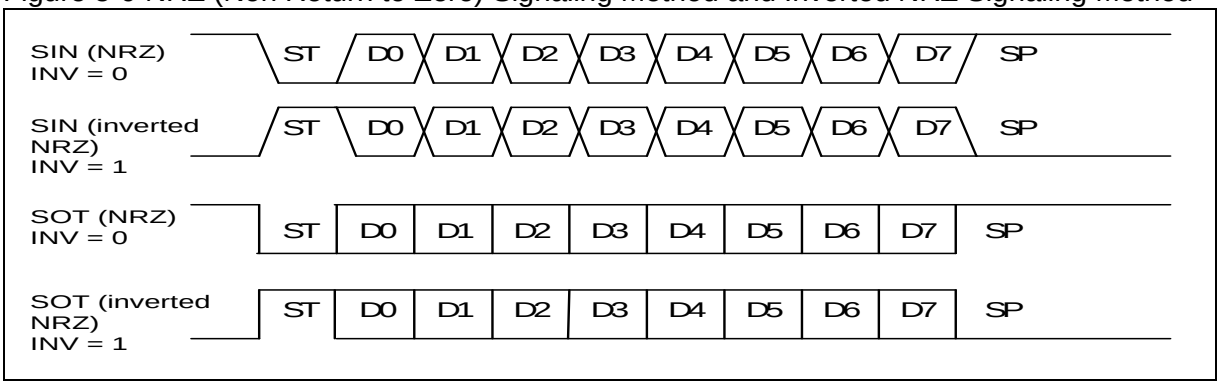


5.2.9. Data Signaling Method

This section explains the data signaling method.

The INV bit setting of the extended communication control register enables you to select the NRZ (Non Return to Zero) signaling method (ESCR:INV=0) or the inverted NRZ signaling method (ESCR:INV=1).

Figure 5-9 NRZ (Non Return to Zero) Signaling Method and Inverted NRZ Signaling Method



5.2.10. Operation of Serial Timer

This section explains the operation of the serial timer.

The serial timer can be used for either of the timer function or the synchronous transmission function.

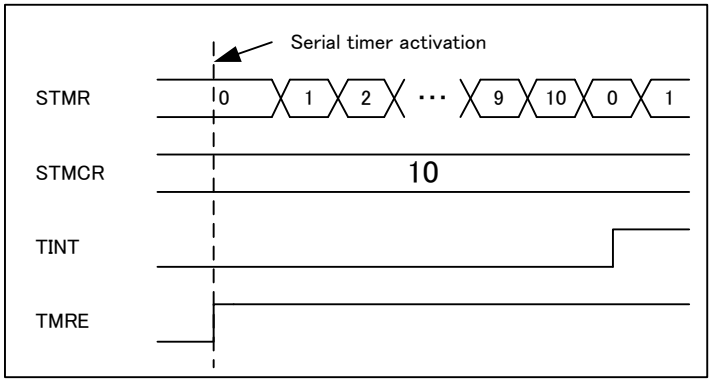
■ How to Start Serial Timer

To start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE).

Start by using the serial timer enable bit (SACSR:TMRE)

When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

Figure 5-10 Start by Using Serial Timer Enable Bit (STMCR="10")



■ How to Stop Serial Timer

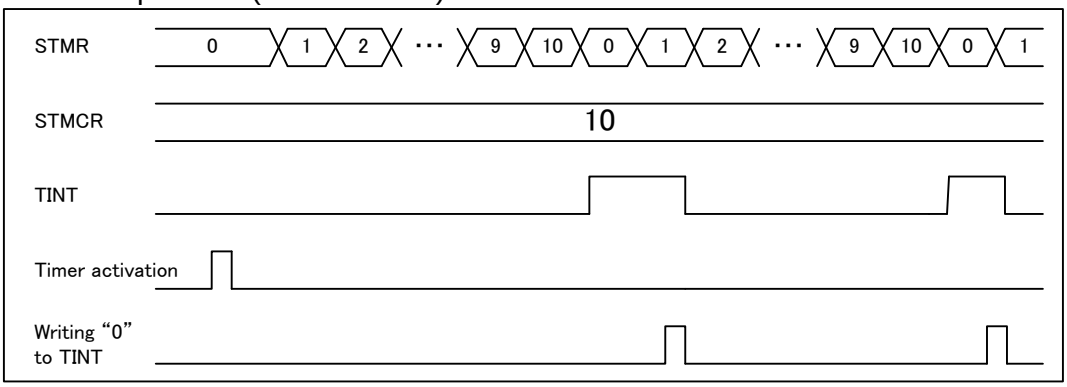
When the serial timer enable bit (SACSR:TMRE) is set to "0", the serial timer will stop. The value of the serial timer register (STMR) is retained.

■ Timer Operation

The serial timer operates as a timer.

If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 5-11 Timer Operation (STMCR="10")



Note:

When the timer comparison register (STMCR) is set to $(0000)_H$, the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".

5.2.11. Test Mode

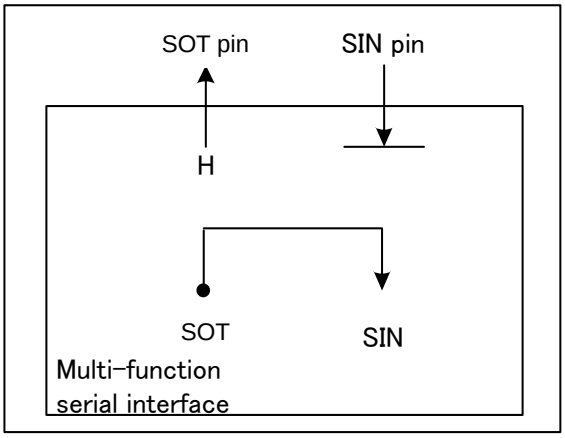
This section explains the test mode.

This section explains the operation of the test mode.

■ Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 5-12 Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

5.2.12. UART Baud Rate Selection/Setting

This section explains the UART baud rate selection/setting.

The UART transmission/reception baud rate generator can be configured for the settings below.

- **Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the internal clock**

  There are two internal reload counters that correspond to the transmission and reception serial clocks, respectively.
  The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).
  The reload counter divides the internal clock with the set value.
  To configure the clock source, select the internal clock (BGR:EXT=0).

- **Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the external clock**

  Use the external clock for the clock source of reload counter. The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).

  The reload counter divides the external clock with the set value.

  To configure the clock source, select the external clock and the baud rate generator clock (BGR:EXT=1).

  This mode is designed to accommodate the case where the division of an oscillator of a special frequency is used.

Notes:

- Configure the external clock (EXT=1) after stopping the reload counter (BGR=15´h00).
- When an external clock (EXT=1) has been set, the "H" width and "L" width of the external clock should be set to 2 bus clocks or more.

■ Baud Rate Calculation

Set two 15-bit reload counters in the baud rate generator register (BGR).
The baud rate calculation formulas are as follows:

(1) Reload value

$V = \phi / b - 1$

V: Reload value
b: Baud rate
ϕ: Bus clock frequency, external clock frequency

(2) Example of calculation

Reload values when setting the bus clock frequency at 16 MHz, usage of internal clock, and baud rate at 19200 bps are as follows:

Reload value:

$V = (16 \times 1{,}000{,}000) / 19200 - 1 = 832$

The baud rate is:
b = (16 × 1,000,000) / (832+1) = 19208 bps

(3) Baud rate error
The baud rate error can be obtained using the following formula:

Error (%) = (calculated value - desired value) / desired value × 100

(Example) Bus clock 20MHz, Target baud rate value 153600 bps
Reload value = (20 × 1,000,000) / 153600 - 1 = 129
Baud rate (calculated value) = (20 × 1,000,000) / (129 + 1) = 153846 bps
Error (%) = (153846 - 153600)/ 153600 × 100 = 0.16(%)

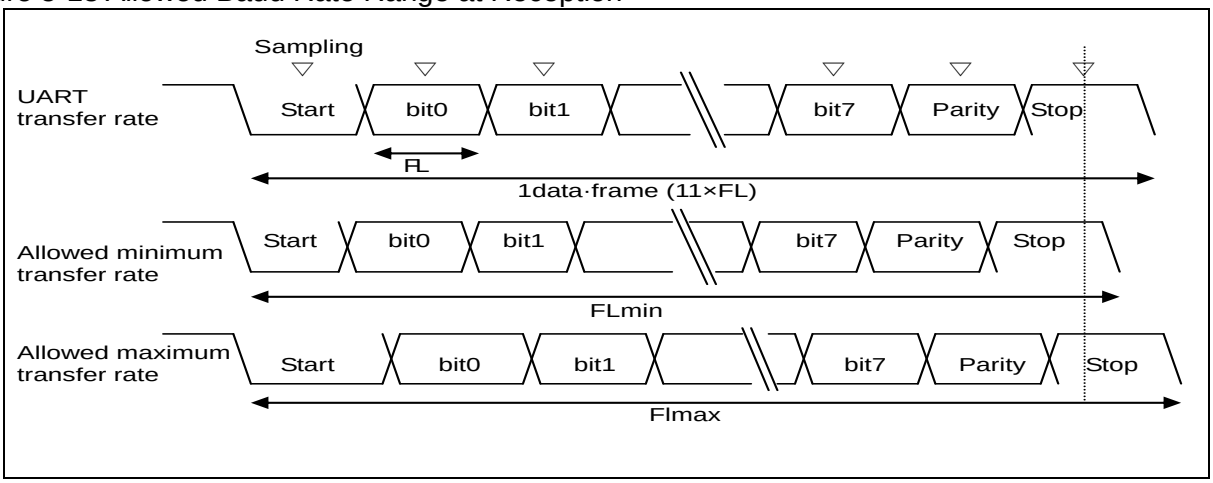
Notes:

- Set the reload value to "0" to stop the reload counter.
- If the reload value is an even number, the "L" width of the reception serial clock is 1 bus clock longer than "H" width. If it is an odd number, the "H" and "L" widths of the serial clock are equal.
- Set the reload value to 4 or higher. A normal data reception operation, however, could not be achieved due to some baud rate error and reload value settings.

■ Allowed Baud Rate Error Range at Reception

This section explains the amount of the destination baud rate error that can be allowed at reception.
The baud rate error at reception should be set within the allowed error range by using following formula.

Figure 5-13 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 within this sampling timing.

In theory, the following is expected when this is applied to 11-bit reception.

If the margin of sampling timing is 1 clock of bus clock (ϕ), the allowed minimum transfer rate (FLmin) would be calculated as follows.

FLmin = (11bit × (V+1) – (V+1) / 2 +2) / ϕ = (21V+25) / 2ϕ (s)

V: Reload value ϕ: Bus clock

Therefore, the allowed maximum baud rate (BGmax) at the destination would be calculated as follows.

BGmax = 11/FLmin = 22ϕ / (21V+25) (bps)

V: Reload value ϕ: Bus clock

When receiving data at the allowed maximum transfer rate (FLmax), sampling is done in the starting point of receive data in the 11th bit.

Therefore, the allowed maximum transfer rate (FLmax) is as follows.

10/11 × FLmax = (11bit × (V+1) – (V+1) / 2) / ϕ
Flmax = (21/20 × 11 × (V + 1)) / ϕ (s)
V: Reload value ϕ: Bus clock

When margin (ϕ) of the sampling timing is made two clocks, the allowed maximum transfer rate (FLmax) is as follows:

FLmax = (21/20 × 11 × (V+1) – 2) / ϕ = (231V+191) / 20ϕ (s)

V: Reload value ϕ: Bus clock

Therefore, the allowed minimum baud rate (BGmin) at the destination would be calculated as follows.

BGmin = 11 / FLmax = 220ϕ / (231V+191) (bps)

V: Reload value ϕ: Bus clock

The allowed baud rate errors at UART and the destination can be obtained from above minimum/maximum baud rate calculation formulas, the result of which are as follows.

Table 5-2 Allowed Baud Rate Error

Reload value	Allowed maximum baud rate error	Allowed minimum baud rate error
3	0%	0%
10	2.98%	-3.24%
50	4.37%	-4.44%
100	4.56%	-4.60%
200	4.66%	-4.68%
32767	4.76%	-4.76%

Note:

The accuracy of reception depends on the number of bits in a frame, bus clock, and the reload value. The higher the bus clock and the division ratio are, the more accurate it will become.

■ Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate

Table 5-3 Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	0	-	-	-	-	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.88	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: Setting value of the BGR register (decimal)
- ERR: Baud rate error (%)

■ External Clock

When the EXT bit of the baud rate generator register (BGR) is set to "1", the baud rate generator divides the external clock.

Note:

The external clock signals are synchronized with the internal clock by UART. If the external clock cannot be synchronized, therefore, the operation becomes unstable.

● Reload Counter Functions

Reload counters, including transmission and reception reload counters, serve as the dedicated baud rate generators. It consists of a 15-bit register for reload values and generates a transmission/reception clock from the external or internal clock.

● Count Start

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

● Restart

The reload counter restarts under one of the following conditions:

- Common to the transmission and reception reload counters
  - Programmable reset (SCR:UPCL bit)
- Reception reload counter
  - Detection of a start bit falling edge in asynchronous mode

5.3. Setup Procedure and Program Flow

Setup procedure and program flow is shown.

5.3.1. Operation Mode 0 (One-to-One Connection)

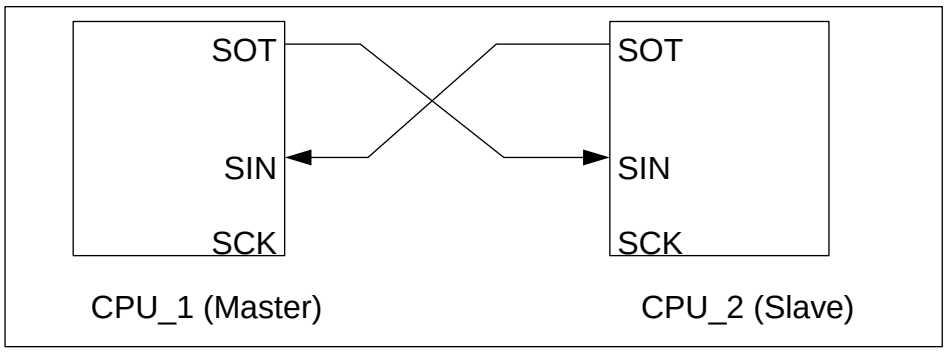
This section explains the operation mode 0 (one-to-one connection).

In operation mode 0, asynchronous serial bidirectional communications can be performed.

■ CPU Interconnection

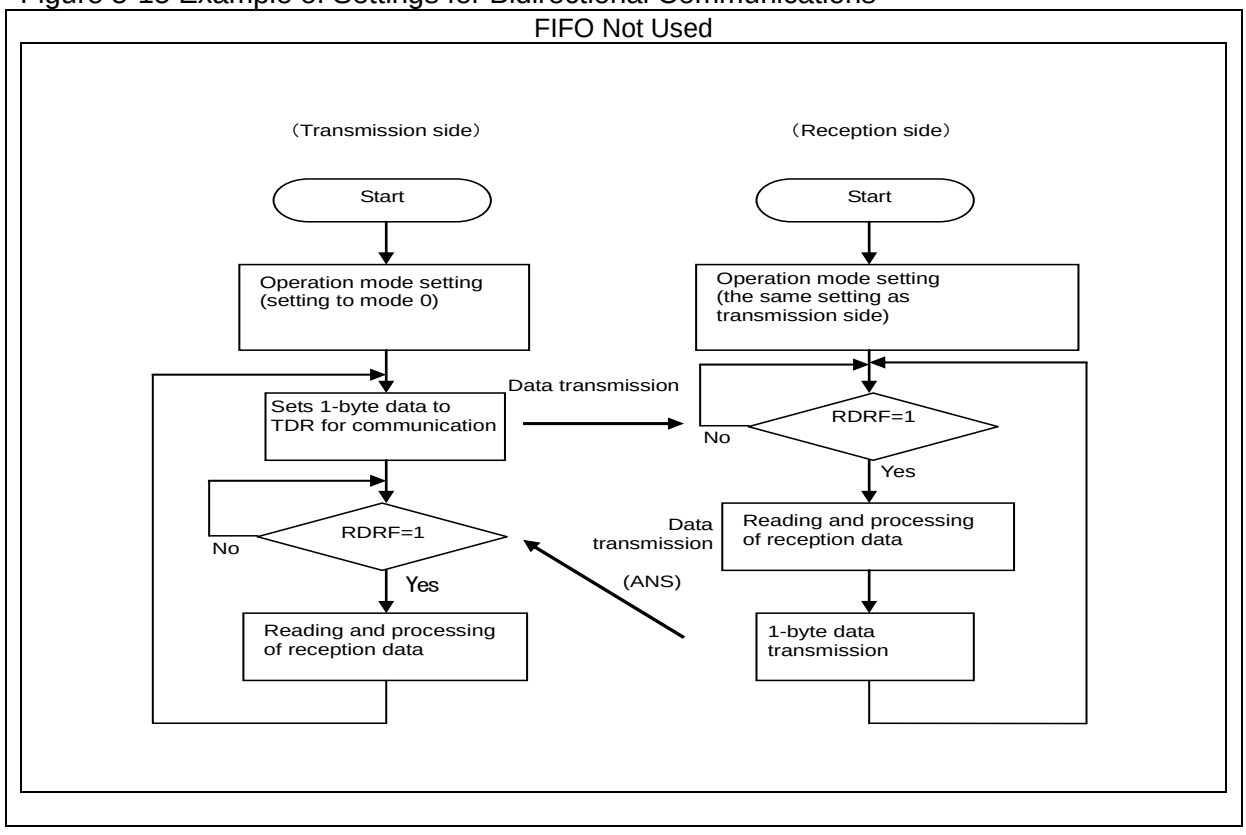
In operation mode 0 (normal mode), select bidirectional communications. Two CPUs are inter-connected as shown in Figure 5-14:

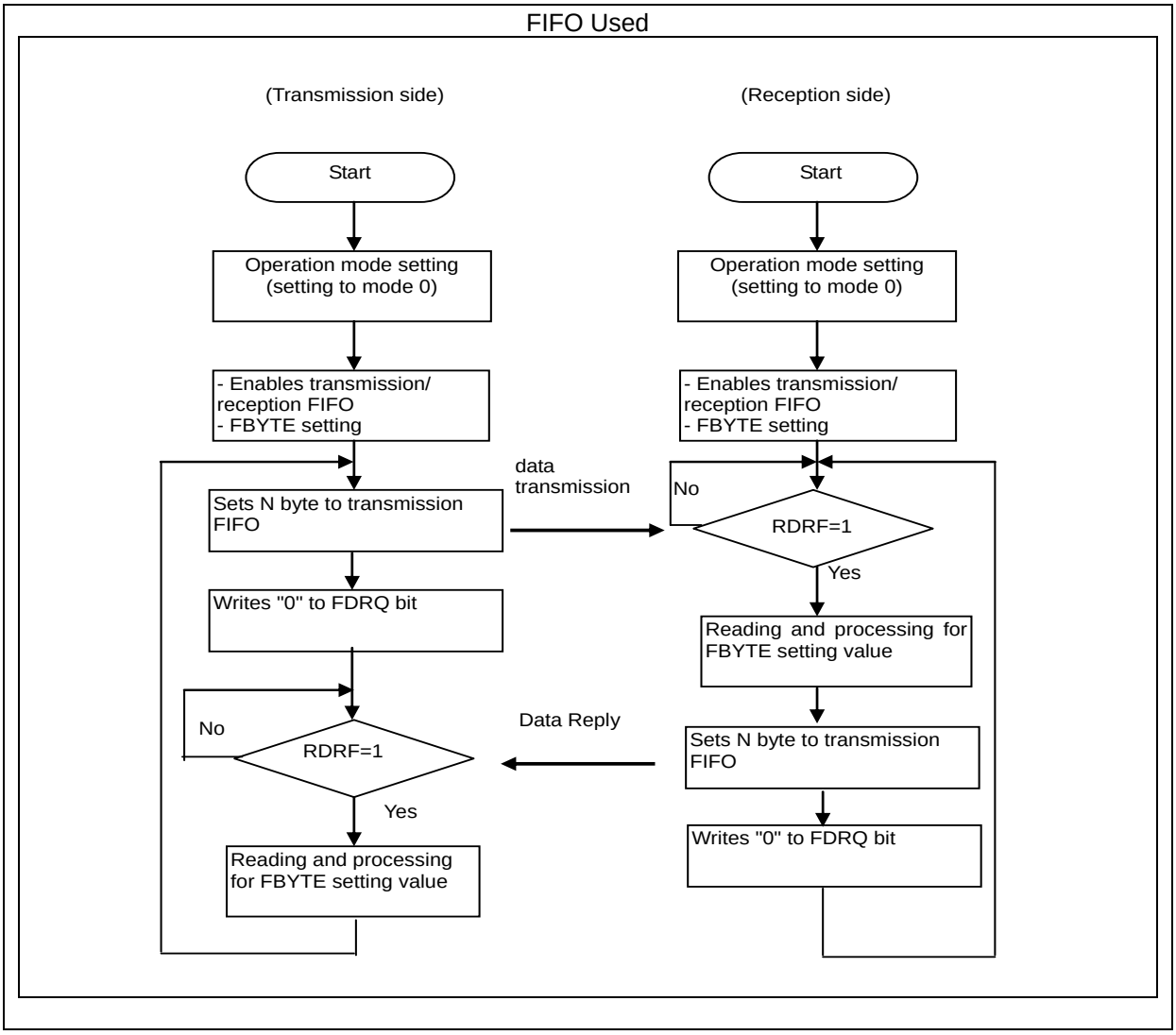
Figure 5-14 Example of Connection for Bidirectional Communications in UART Operation Mode 0



■ Flowchart

Figure 5-15 Example of Settings for Bidirectional Communications





5.3.2. Operation Mode 1 (One-to-N Connection)

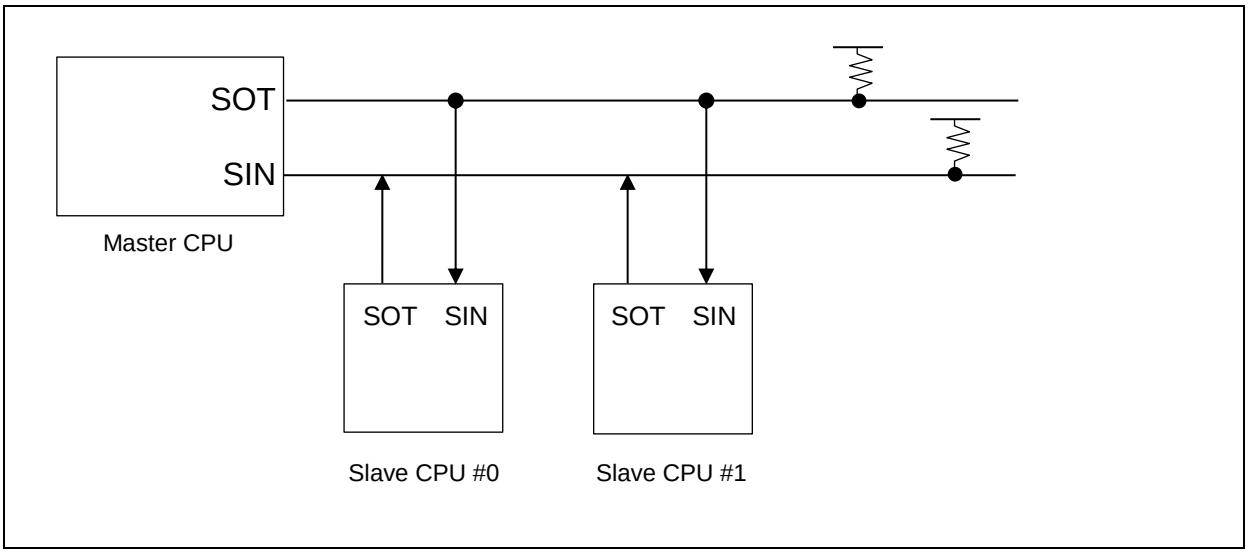
This section explains the operation mode 1 (one-to-n connection).

In operation mode 1 (multi-processor mode), communications can be performed via master-slave connection between multiple CPUs. UART can be used either as a master or slave.

■ CPU Interconnection

For master-slave communications, a communication system can be configured as one master CPU and multiple slave CPUs connected to two common communication lines as shown in the Figure below. UART can be used either as a master or slave.

Figure 5-16 Example of Connection for Master-Slave Communications of UART



■ Function Selection

For master-slave communications, select an operation mode and a data transfer method as shown in Table 5-4 Selection of Master-Slave Communication Function:

Table 5-4 Selection of Master-Slave Communication Function

	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission and reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = "1" + 7 or 8-bit Address	None	1 bit or 2 bits	LSB or, MSB First
Data transmission and reception			AD = "0" + 7 or 8-bit Data			

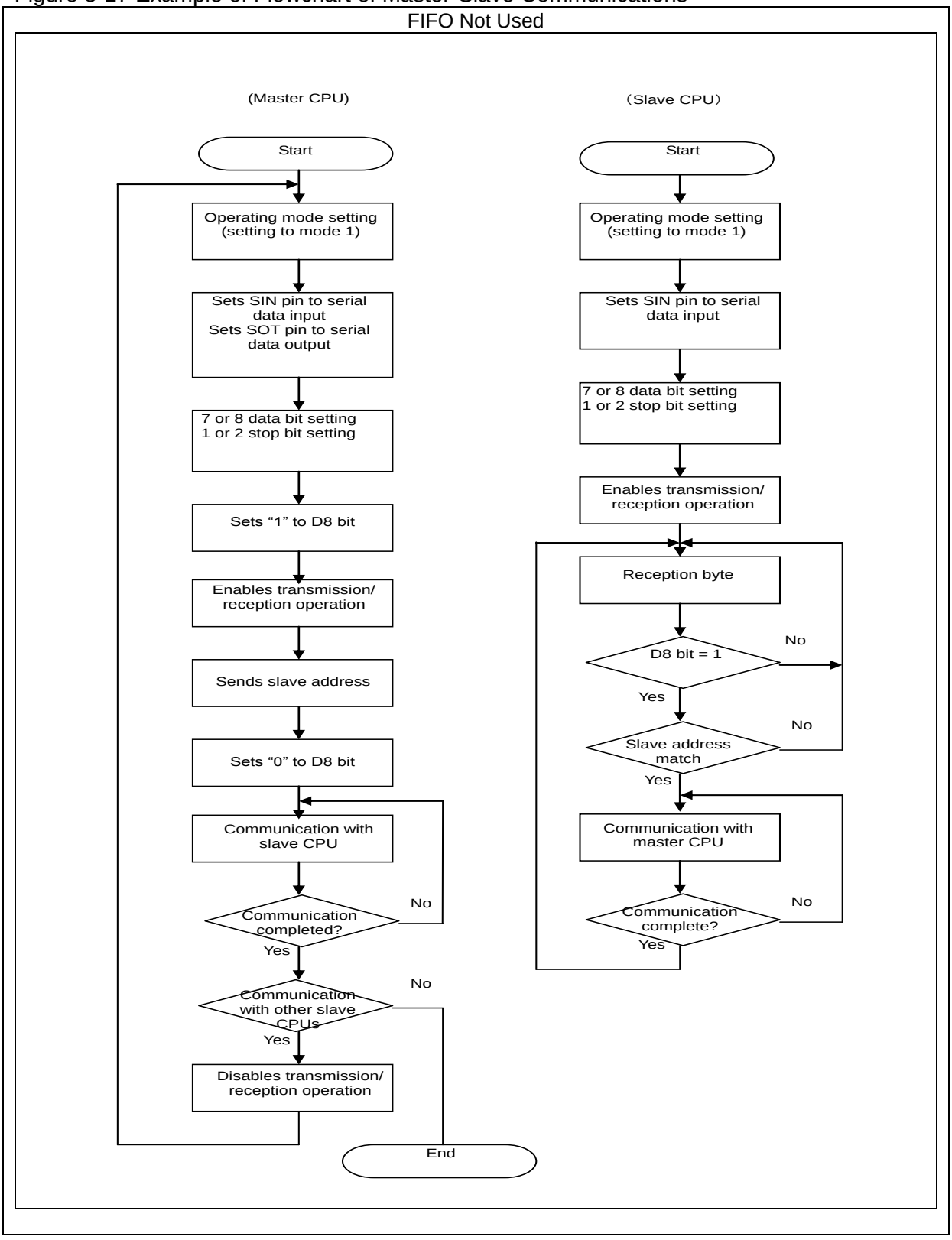
Note:

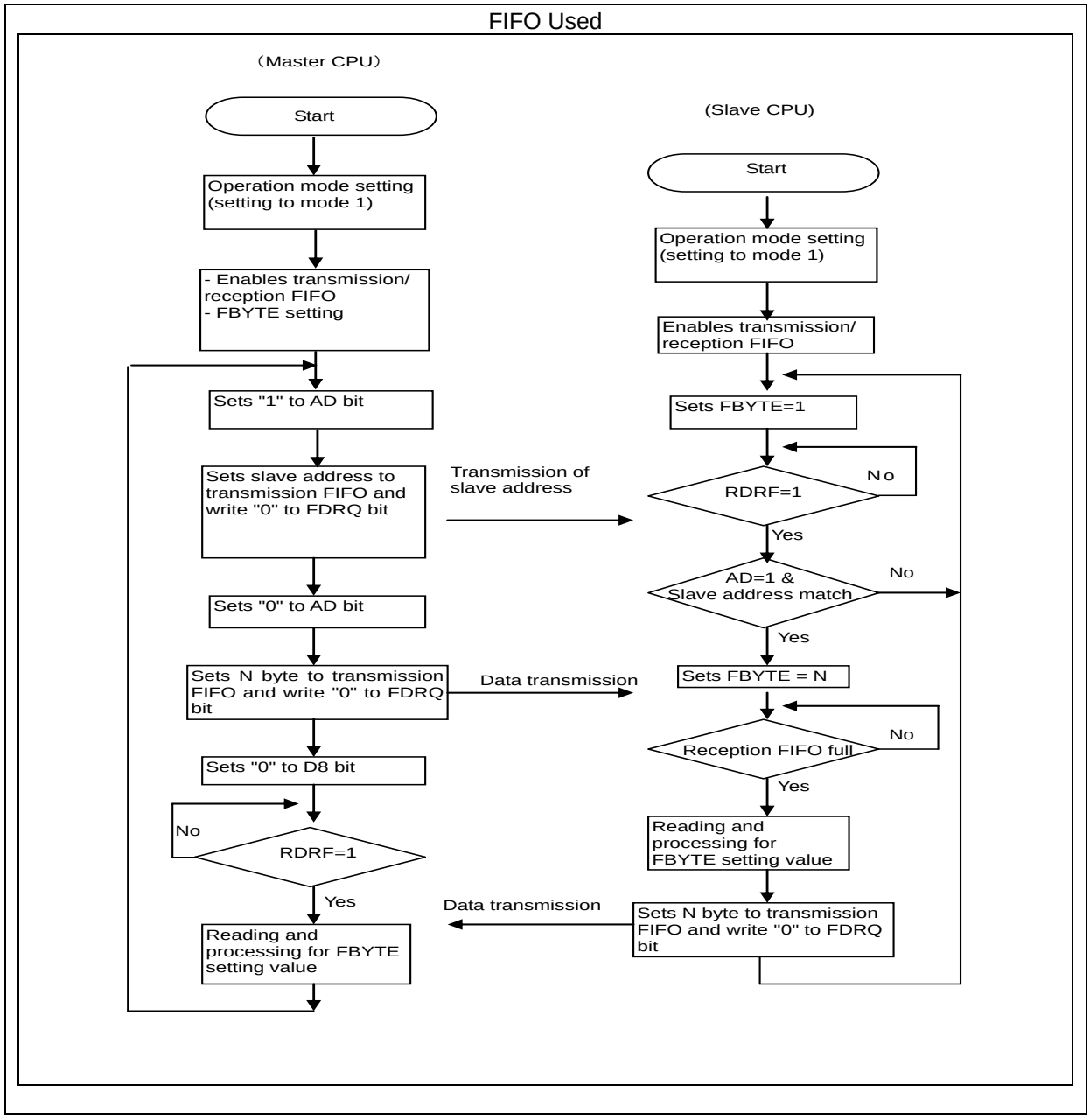
Have word access to transmitted and received data (TDR/RDR) in operation mode 1.

■ Communication Procedure

Communications start when the master CPU transmits address data. Address data refers to data with the D8 bit set to "1" and is used to select a slave CPU as the communication destination. Slave CPUs interpret address data via a program and the one with a matching address performs communications (normal data) with the master CPU. Figure 5-17 shows a flowchart of master-slave communications (multi-processor mode).

Figure 5-17 Example of Flowchart of Master-Slave Communications





6. Operation of CSIO

This section explains operation of CSIO.

6.1 Interrupts of CSIO

6.2 Operation of CSIO

6.3 Setup Procedure and Program Flow

6.1. Interrupts of CSIO

Interrupts of CSIO are shown.

The interrupts for the CSIO (clock synchronous serial interface) include reception and transmission interrupts. An interrupt request can be generated using the following factors:

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- Comparison value (STMCR) of the serial timer and serial timer value (STMR) match.
- Chip selection error generation

6.1.1. List of Interrupts of CSIO

This section explains the list of interrupts of CSIO.

Table 6-1 Interrupt Control Bits and Interrupt Factors of CSIO

Inter-rupt Type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Recep-tion	RDRF	SSR	1-byte reception	SCR: RIE	Reading of receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)

Inter-rupt Type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Trans-mission	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	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1: FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or transmission FIFO is full
Trans-mission	CSE	SACSR	In slave mode (SCR:MS="1"), when serial chip select pin is in inactive master mode (SCR:MS=0) during the transmission operation, the transmission count is equal to or less than the count set at TBYTE and the next transmission data is not written to TDR (SSR:TDRE=1). In master mode (SCR:MS=0), because the number of the transmission times is below the set value of TBYTE, and the next transmission data is not written to TDR (SSR:TDRE=1)	SACSR: SCEIE	Writing "0" to the serial chip select flag bit (SACSR:CSE)
Status	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to the timer interrupt flag bit (SACSR:TINT)

*1: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

6.1.2. Reception Interrupts and Flag Setting Timing

This section explains the generation of reception interrupts and flag setting timing.

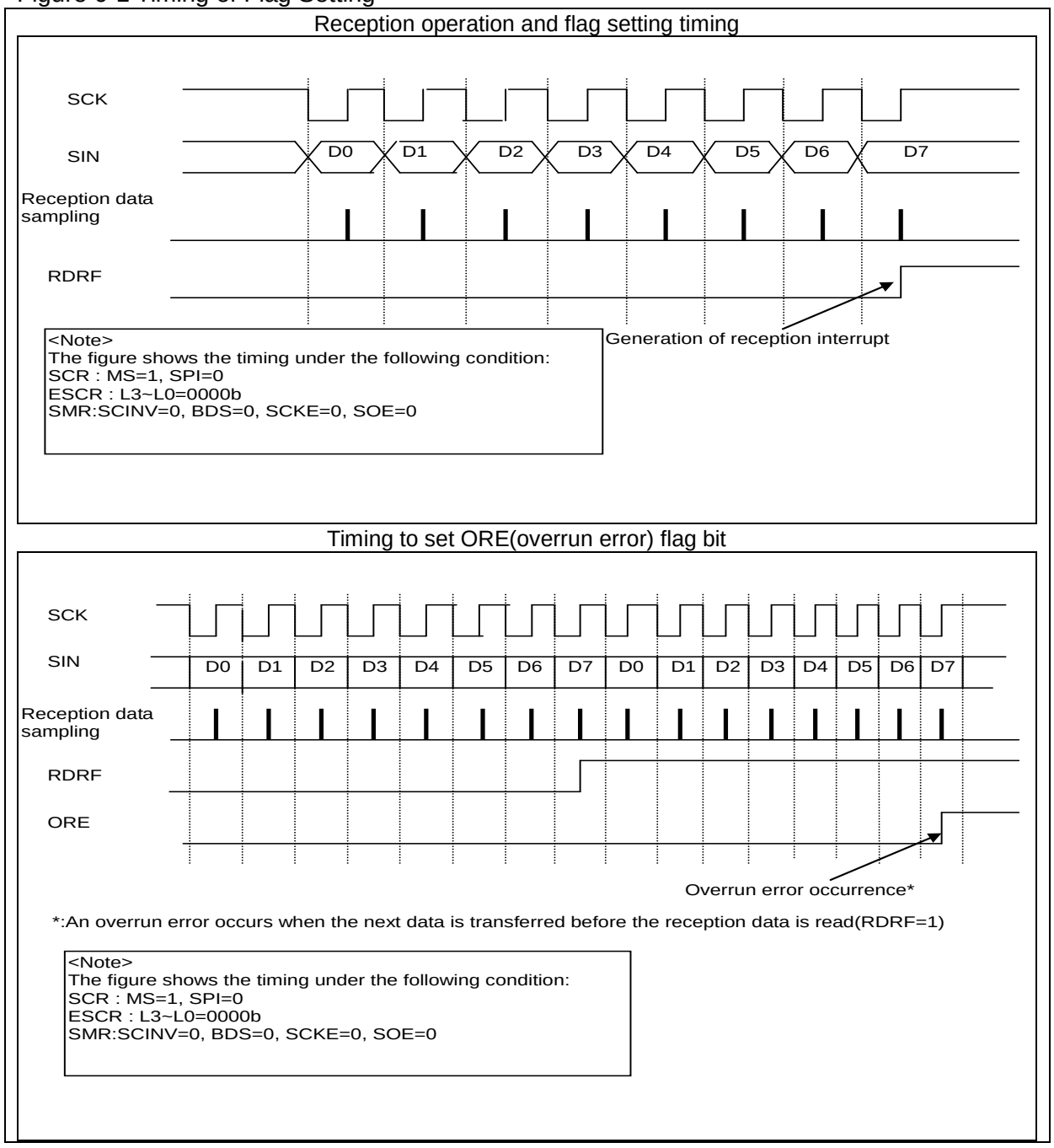
Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:ORE).

When the last data bit is detected, reception data is stored in the receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:ORE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 6-1 Timing of Flag Setting



6.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using the reception FIFO and flag setting timing.

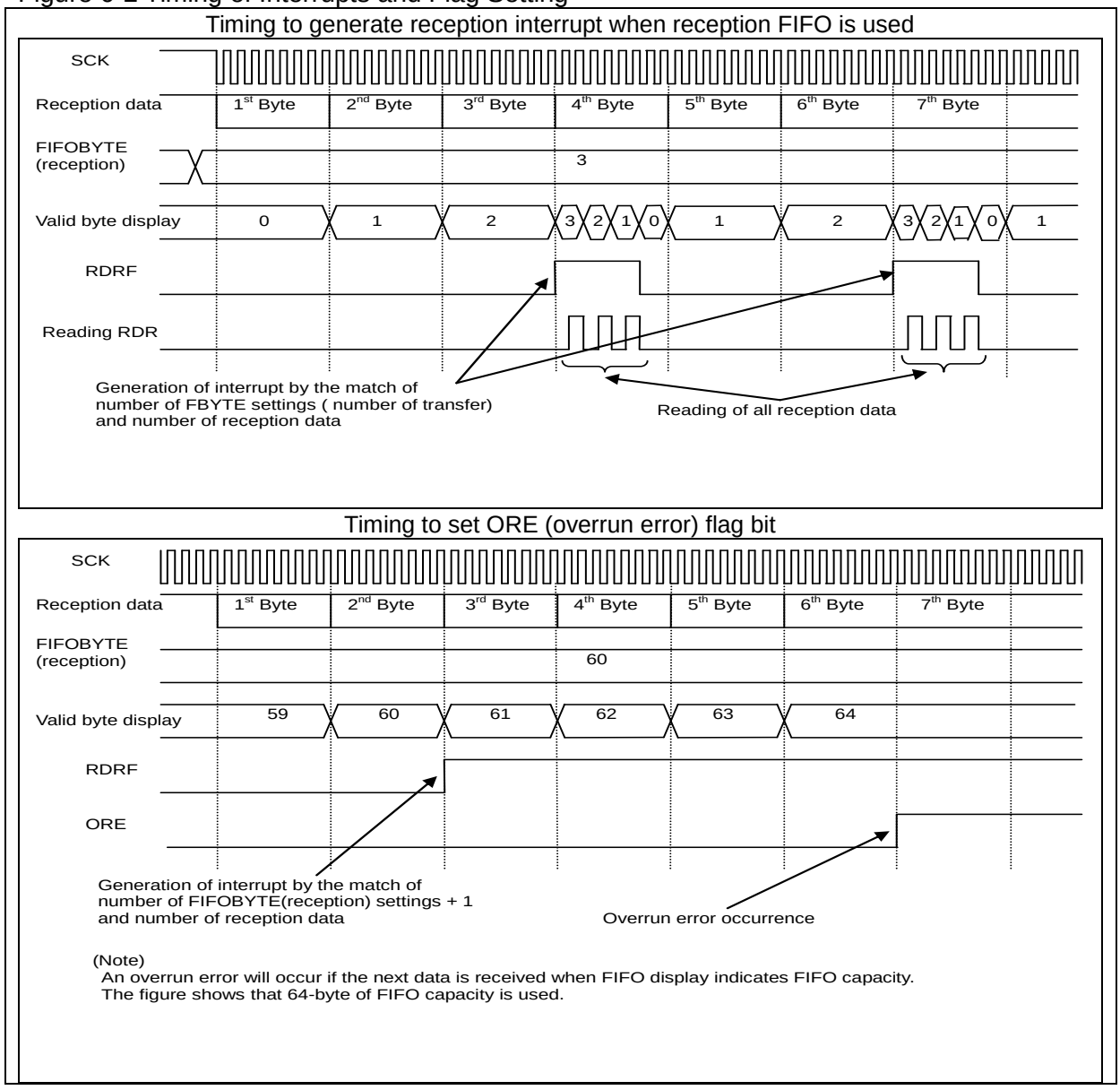
When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received. The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt will be generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (RDRF) will be set to "1".
  - Reception FIFO idle detection enable bit (FRIIE) is "1"
  - Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) will be cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1).

Figure 6-2 Timing of Interrupts and Flag Setting



6.1.4. Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

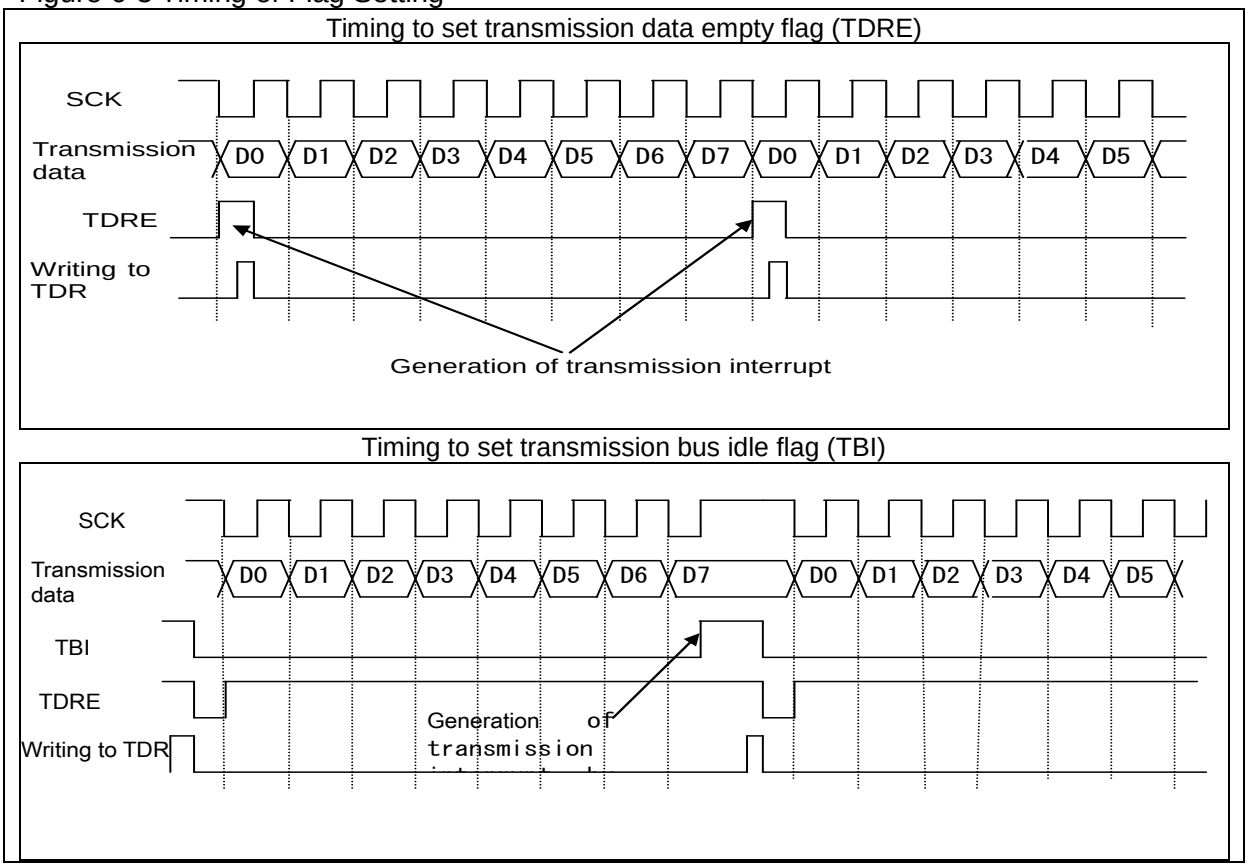
- **Timing of transmission data empty flag (TDRE) setting**

  When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. The SSR:TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

- **Timing of transmission bus idle flag (TBI) setting**

  When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 6-3 Timing of Flag Setting



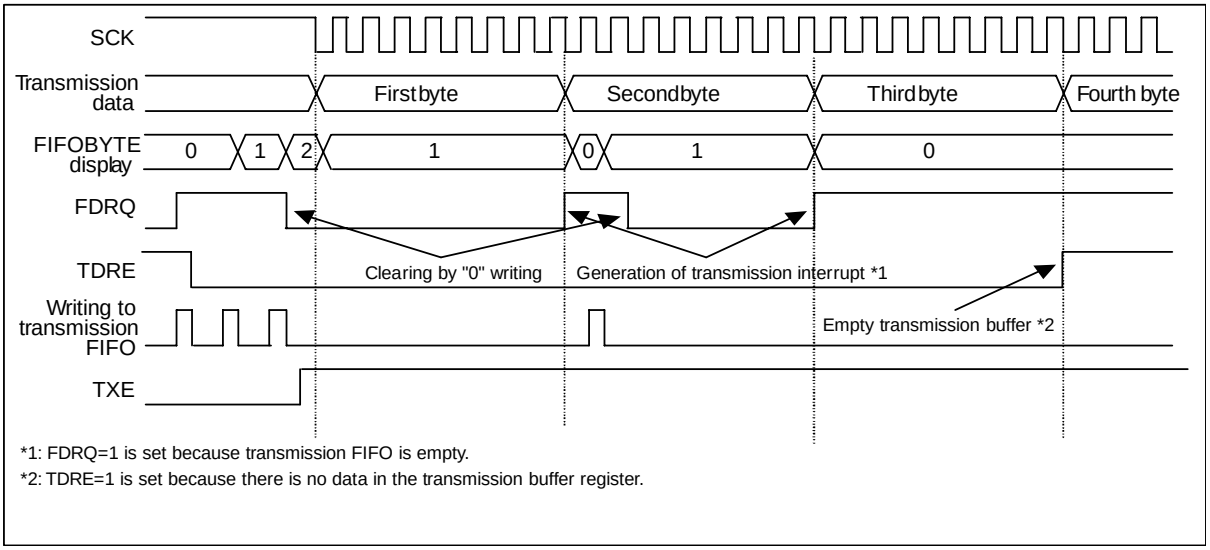
6.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and flag setting timing.

When the transmission FIFO is used, an interrupt generation when the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less.

- When the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  When FBYTE=0x00 and FTICR=0x00, there is no data in the transmission FIFO.

Figure 6-4 Timing of Interrupt Generation



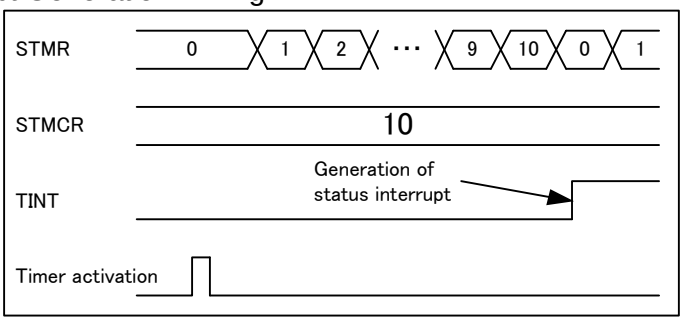
6.1.6. Timing of Timer Interrupt and Flag Setting

This section explains the timing of timer interrupt and flag setting.

Timer interrupt is generated when Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR).

- When Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR), "1" will be set to timer interrupt flag (SACSR:TINT).
  At this time when the timer interrupt is enabled (SACSR:TINTE="1"), a status interrupt will be generated.

Figure 6-5 Timer Interrupt Generation Timing



6.1.7. Timing of Chip Select Error Generation and Flag Setting

This section explains the timing of chip select error generation and flag setting.

Chip select error will be generated when the number of frames which have been transmitted is less than the setup value specified by the TBYTE and no valid data is present in the transmit data register (TDR) (SSR:TDRE="1") after one frame is transmitted while in master mode (SCR:MS="0"). This error will also be generated when chip select pin becomes inactive while transmitting in slave mode (SCR:MS=1).

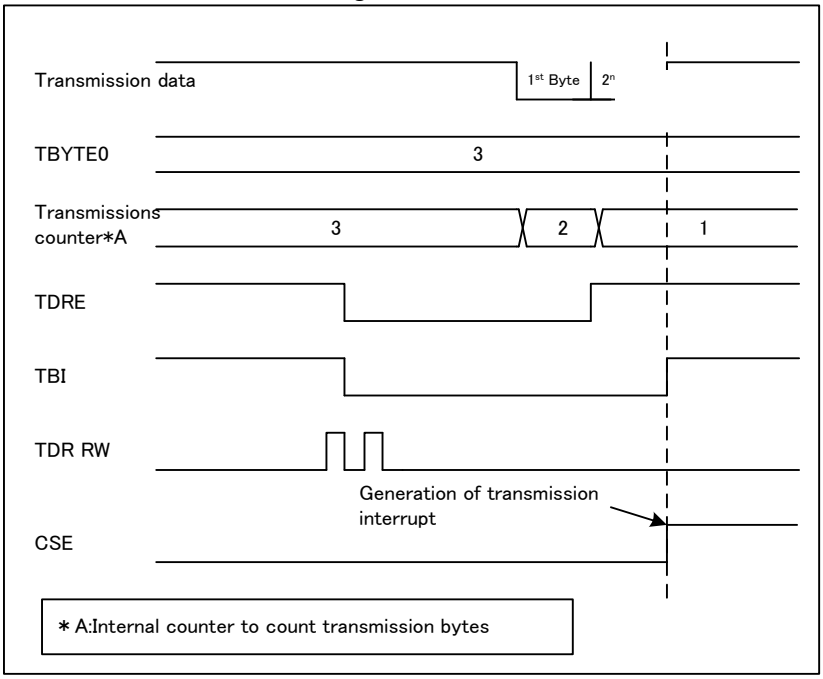
● Master Mode (SCR:MS="0")

Chip select error will be generated with transfer byte error enabled (TBEEN="1") and any of following events when no valid data is present in the transmission data register (TDR) (SSR:TDRE="1") before transmitting data frame specified by the TBYTE.

- Chip select is used
- Synchronous transmission with the serial timer is used

In this case, when chip select error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 6-6 Chip Select Error Generation Timing



Notes:

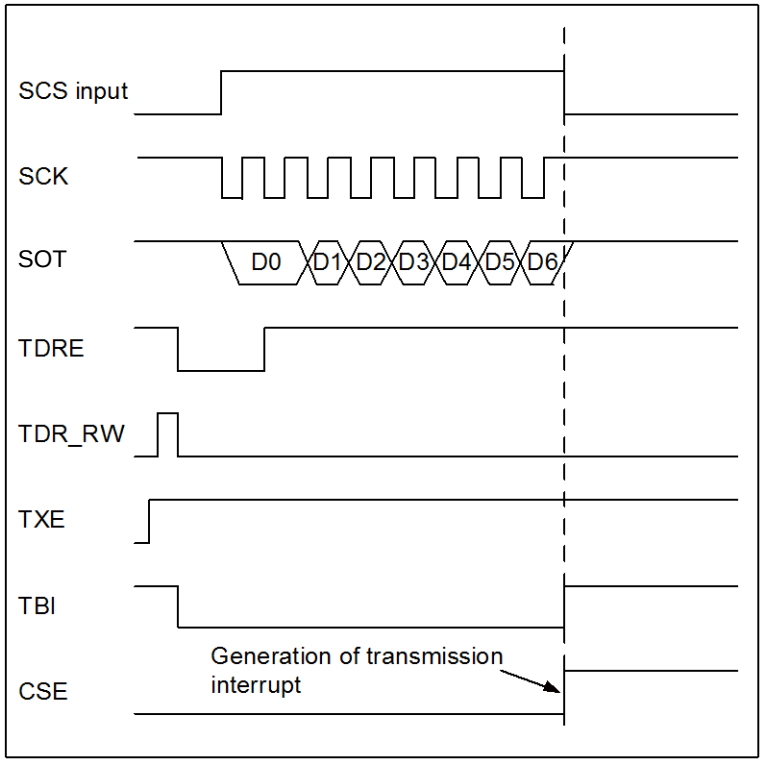
- When chip select is used, chip select error flag (SACSR:CSE) will be set to "1" after hold delay time is passed after a chip select error is generated and at the same time, the serial chip select pin will become inactive. No transmission will be started even if transmit data is written to the transmit data register (TDR) during hold delay time, and the chip select error flag (SACSR:CSE) will be set to "1" after hold delay time is passed.
- While the chip select error flag (SACSR:CSE) is set to "1", no transmission will be started even if transmission data is written to the transmit data register (TDR).
- When the chip select error flag (SACSR:CSE) is set to "1" while synchronous transmission is used with the serial timer, no transmission will be started even if serial timer register (STMR) matched the serial timer comparison register.

● Slave Mode (SCR:MS="1")

Chip select error will be generated when chip select pin becomes inactive while transmitting (SSR:TBI="0").

In this case, when chip select error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 6-7 Chip Select Error Generation Timing



6.2. Operation of CSIO

Operation of CSIO is shown.

6.2.1. Normal Transfer (I)

This section explains the normal transfer (I).

■ 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	to 5-16, 20, 24, 32 bits

■ Register Settings

The following table lists the register settings required for normal transfer (I).

SCR:SPI=0*[1], SMR:MD2=0, MD1=1, MD0=0, SCINV=0*[1]
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

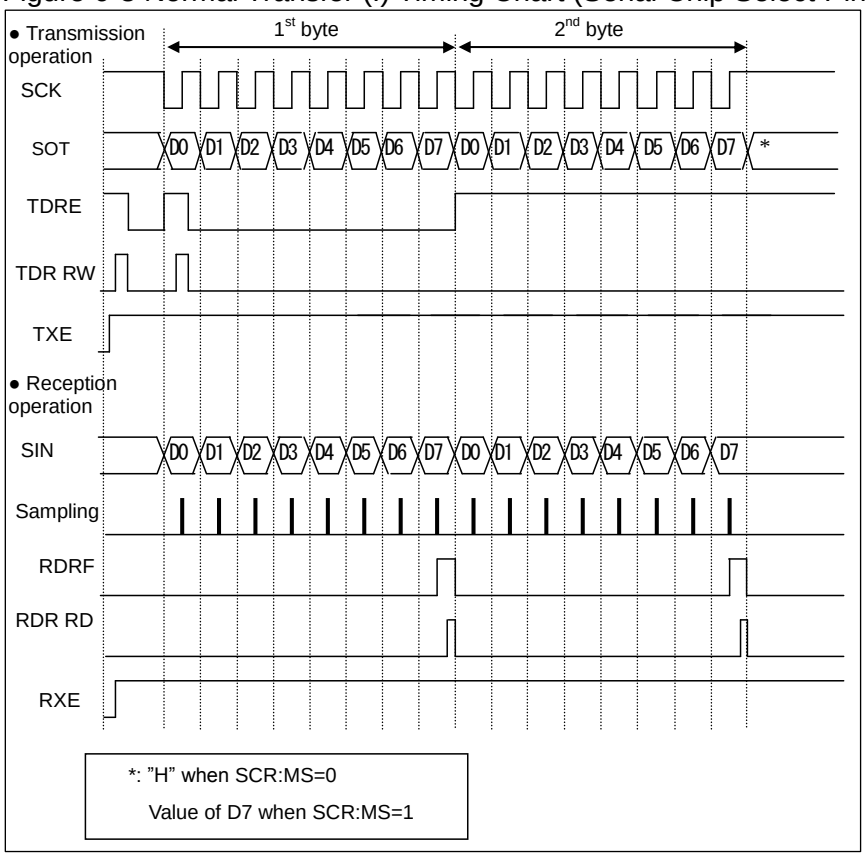
*1: Bit settings depend on the condition. See Table 6-2 for details.

Note:

Use proper usage for setting the registers other than the bits above.

■ Normal Transfer (I) Timing Chart (Serial Chip Select Pin Unused)

Figure 6-8 Normal Transfer (I) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSEN3-0="0000"b)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1. When the transmission interrupt is enabled

(SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a rising edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

● Transmission/Reception operation

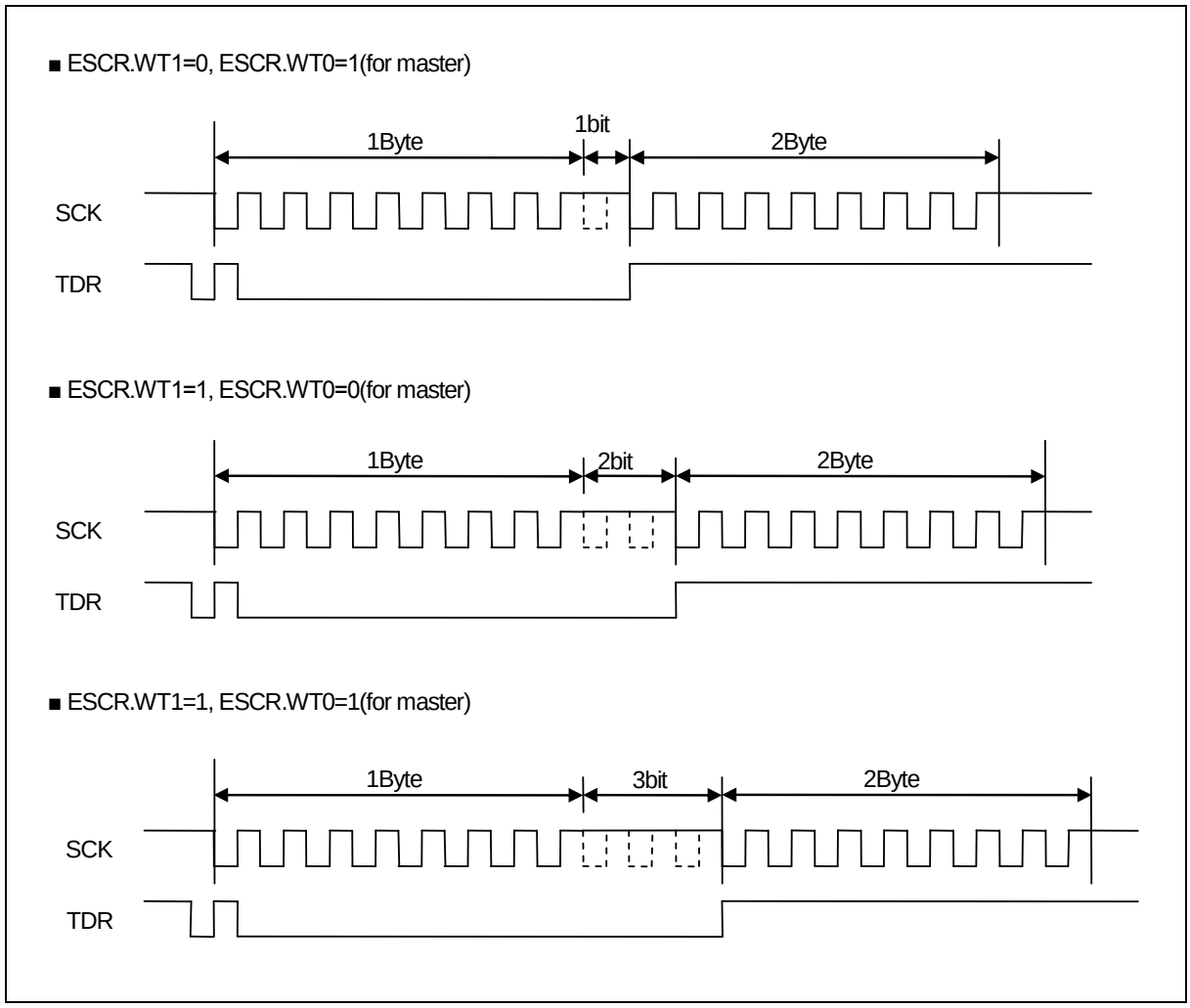
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

● Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. This results in output the transmission data in synchronization with a falling edge of the serial clock (SCK) input.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1 and, if the transmission interrupt is enabled (SCR:TIE=1), outputs a transmission interrupt request. At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

● Transmission/Reception operation

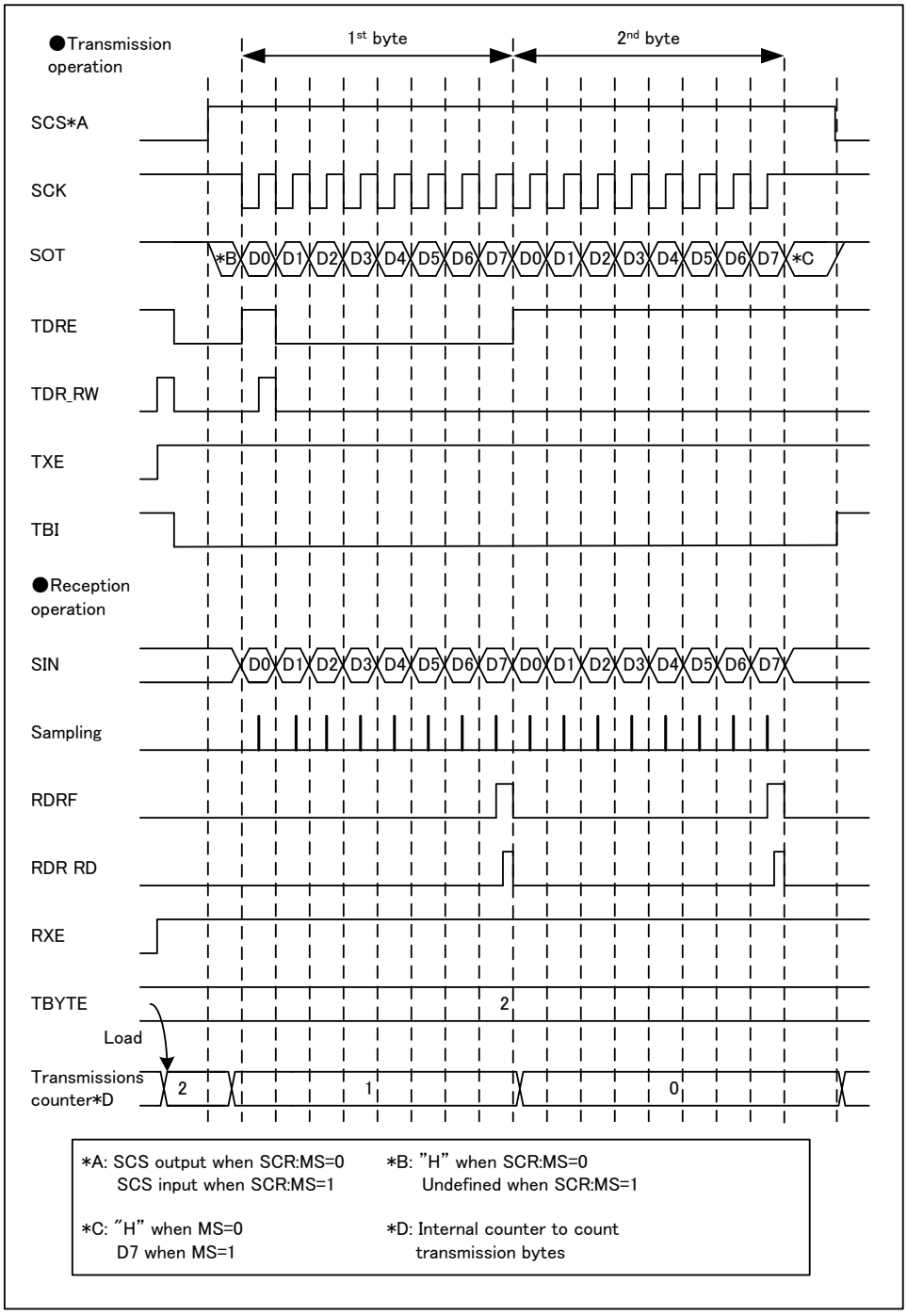
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

■ Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used)

Figure 6-9 Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: "n" shows the number of serial chip select pin used.

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active, transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a rising edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active and transmission/reception will be started after the setup time has passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a rising edge of the serial clock output (SCK). Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

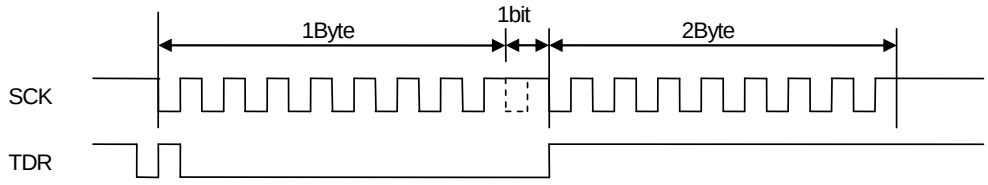
(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

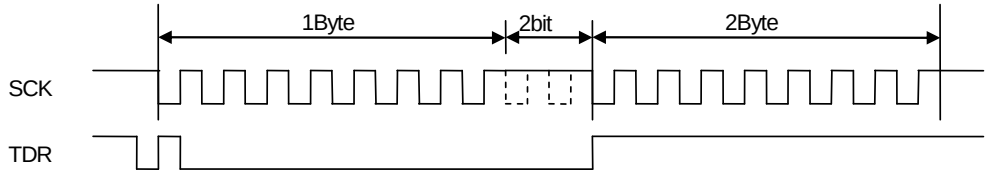
● Continuous Data Transmission or Reception Wait Operation

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.

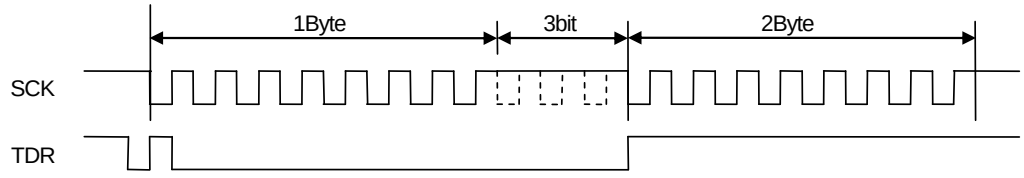
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started when serial chip select pin (SCS) becomes active, the transmission data will be output in synchronization with a falling edge of the serial clock (SCK) input.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception operation will be started and the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a rising edge of the serial clock input (SCK). Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOT) becomes "H".

6.2.2. Normal Transfer (II)

This section explains the normal transfer (II).

■ 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 16, 20, 24, 32 bits

■ Register Settings

The following table lists the register settings required for normal transfer (II).

SCR:SPI*[1]=0, SMR:MD2=0, MD1=1, MD0=0, SCINV*[1]=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

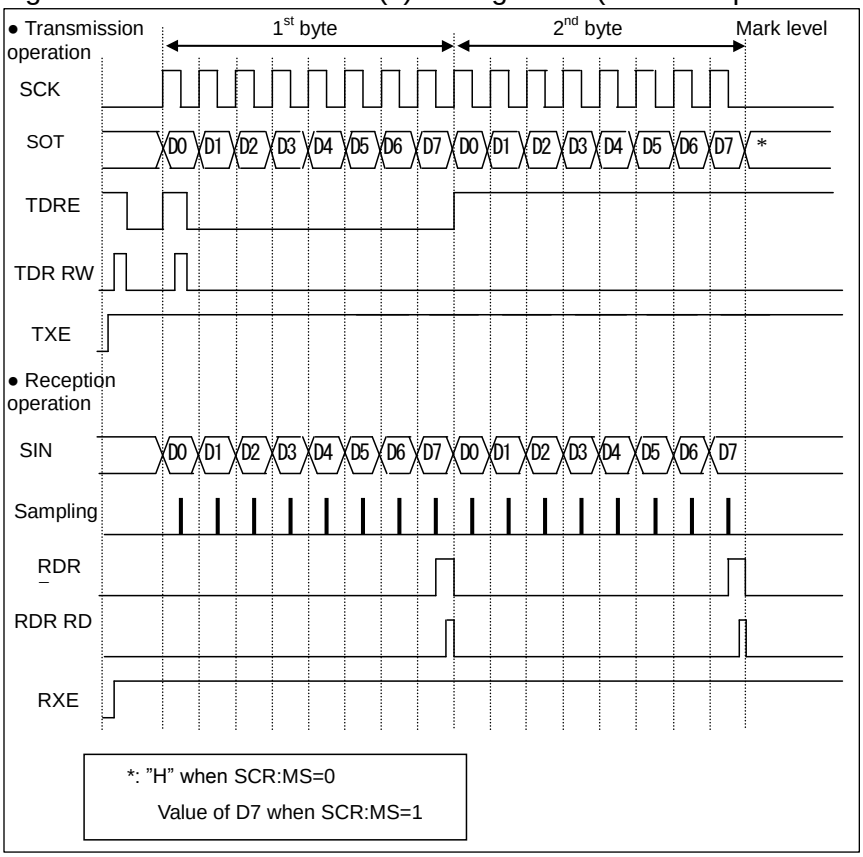
*1: Bit settings depend on the condition. See Table 6-2 for details.

Note:

Use proper usage for setting the registers other than the above.

■ Normal Transfer (II) Timing Chart (Serial Chip Select Pin Unused)

Figure 6-10 Normal Transfer (II) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1. When the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be output. At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

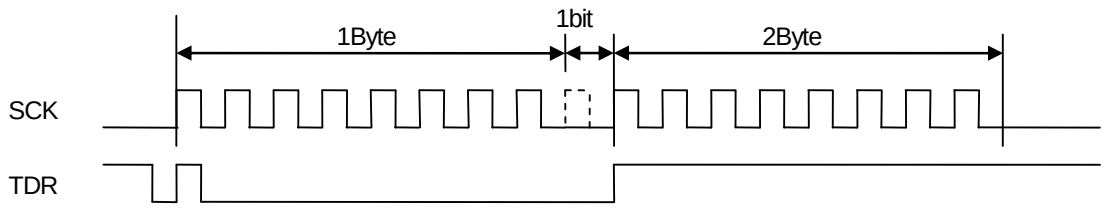
(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

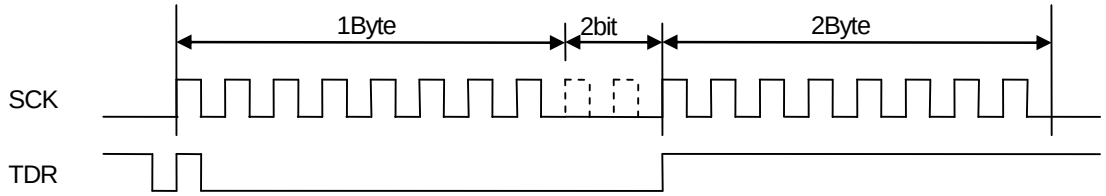
● Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

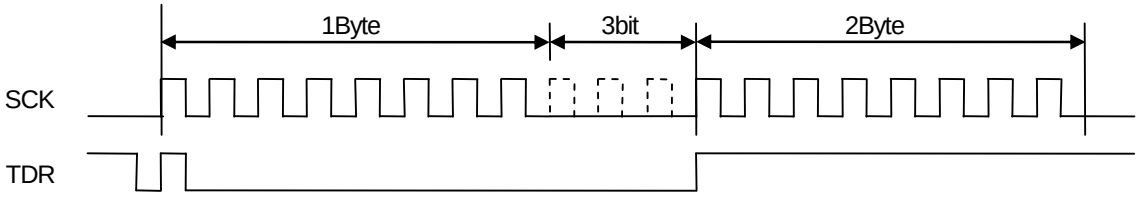
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) input.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be output. At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

● Transmission/Reception operation

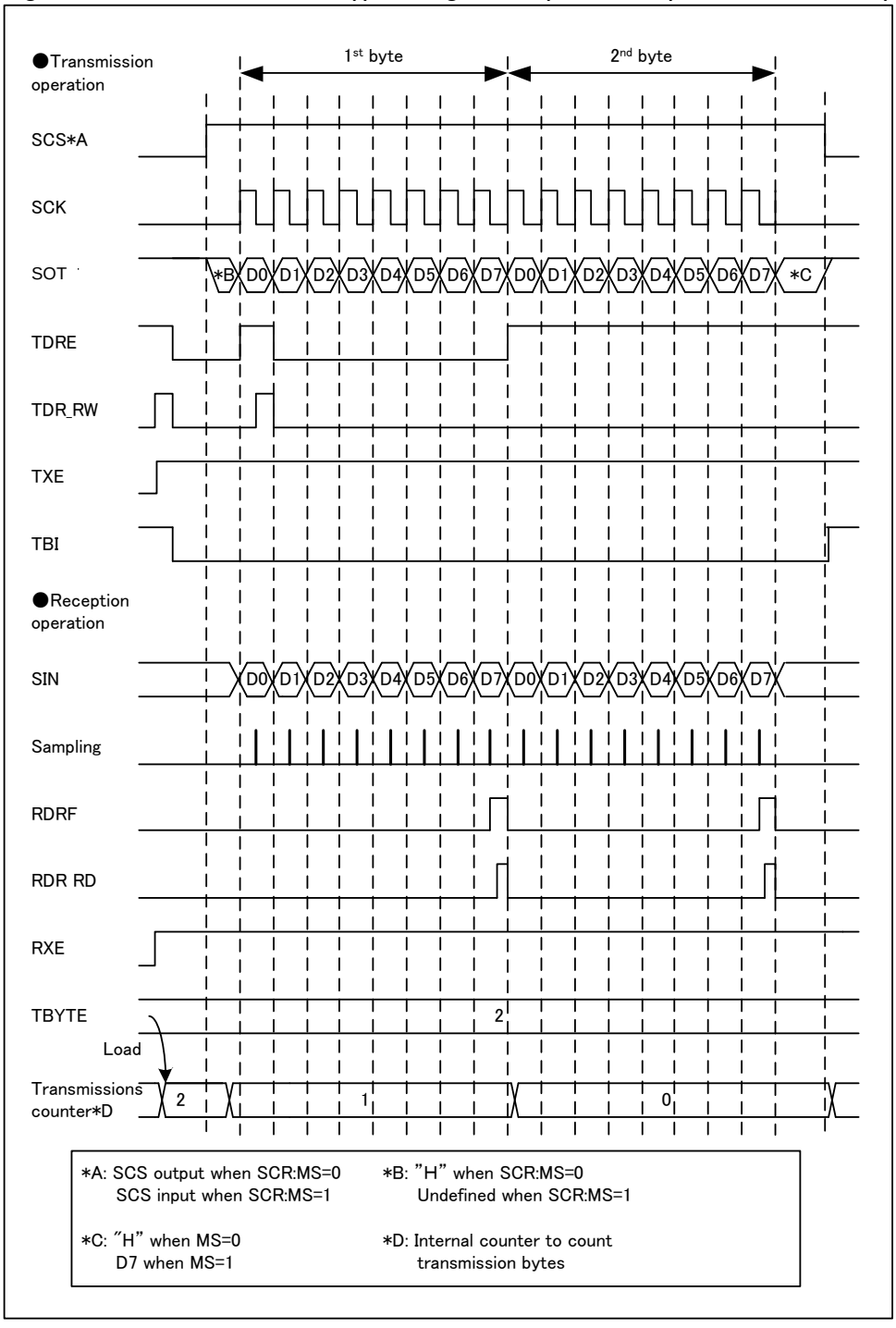
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3)Receive data is sampled by the falling edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

■ Normal Transfer (II) Timing Chart (Serial Chip Select Pin Used)

Figure 6-11 Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: "n" shows the number of serial chip select pin used.

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active, transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1, and if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, SSR:RDRF is cleared to "0".

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active and transmission/reception will be started after the setup time is passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a falling edge of the serial clock (SCK) output. Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

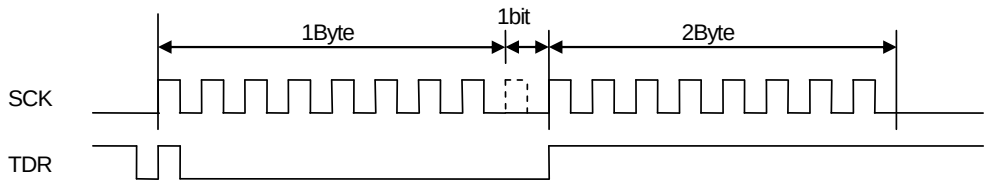
(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is has elapsed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

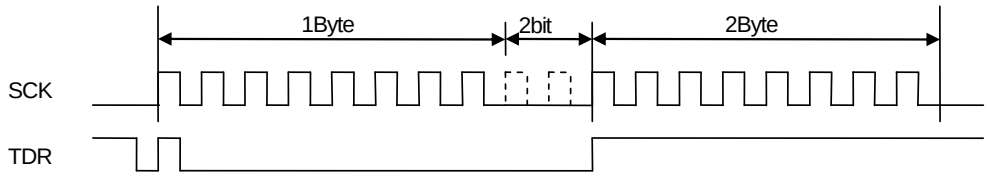
● Continuous Data Transmission or Reception Wait Operation

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.

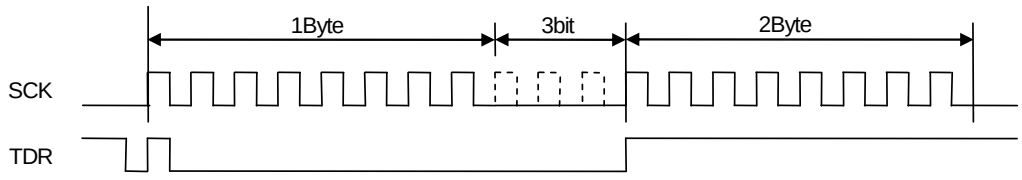
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started when serial chip select pin (SCS) becomes active, the transmission data will be output in synchronization with a rising edge of the serial clock (SCK) input.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception operation will be started and the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a falling edge of the serial clock input (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) The transmission/reception operation will be started when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOT) becomes "H".

6.2.3. SPI Transfer (I)

This section explains the SPI transfer (I).

■ 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).

SCR:SPI*[1]=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*[1]=0
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

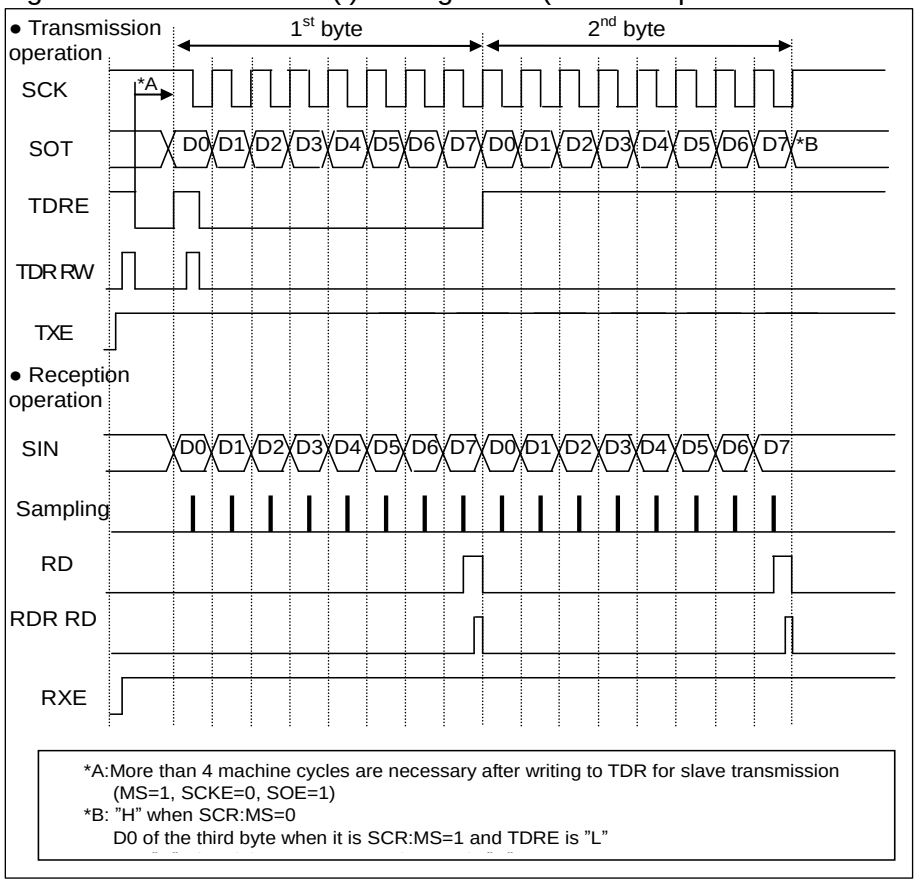
*1: Bit settings depend on the condition. See Table 6-2 for details.

Note:

Use proper usage for setting the registers other than the above.

■ SPI Transfer (I) Timing Chart (Serial Chip Select Pin Unused)

Figure 6-12 SPI Transfer (I) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSEN3-0="0000_B")

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0 and outputs the first bit. Then, the transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1. This results in generating a transmission interrupt request when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

● Transmission/Reception operation

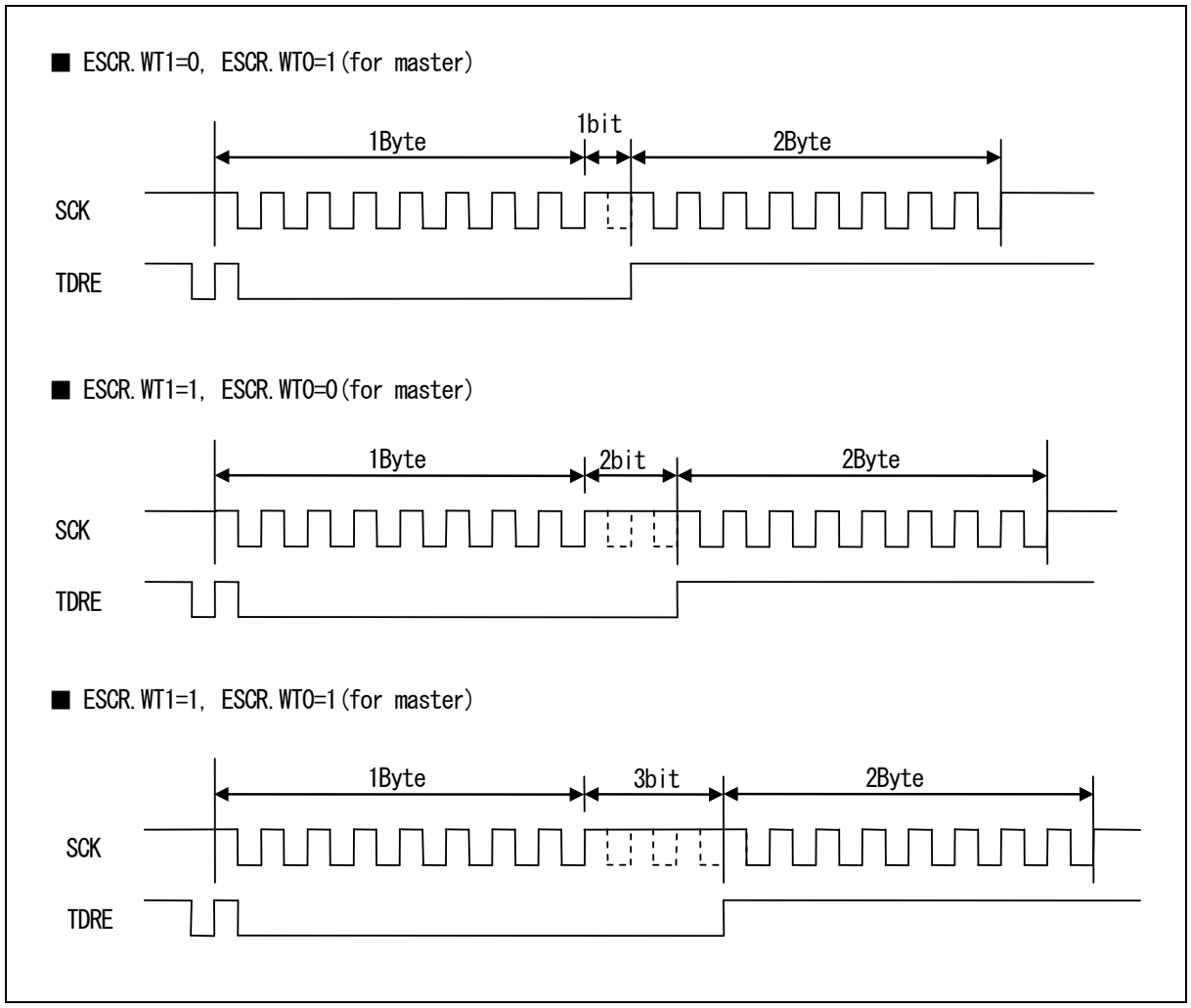
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

● Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation(Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. Therefore, the first bit is output. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) When the first bit of the transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

Note:

After transmission operation is enabled (SCR:TXE=1), when you write first transmission data to the TDR while serial clock (SCK) is at a level other than mark level, the first bit of the data will not be output and correct transmission operation will not be performed. After transmission operation is enabled (SCR:TXE=1), writing the first transmitting data to the TDR must be made while serial clock (SCK) is at mark level.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be output.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

● Transmission/Reception operation

To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3)Receive data is sampled by the falling edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

● Continuous change from reception operation to transmission operation

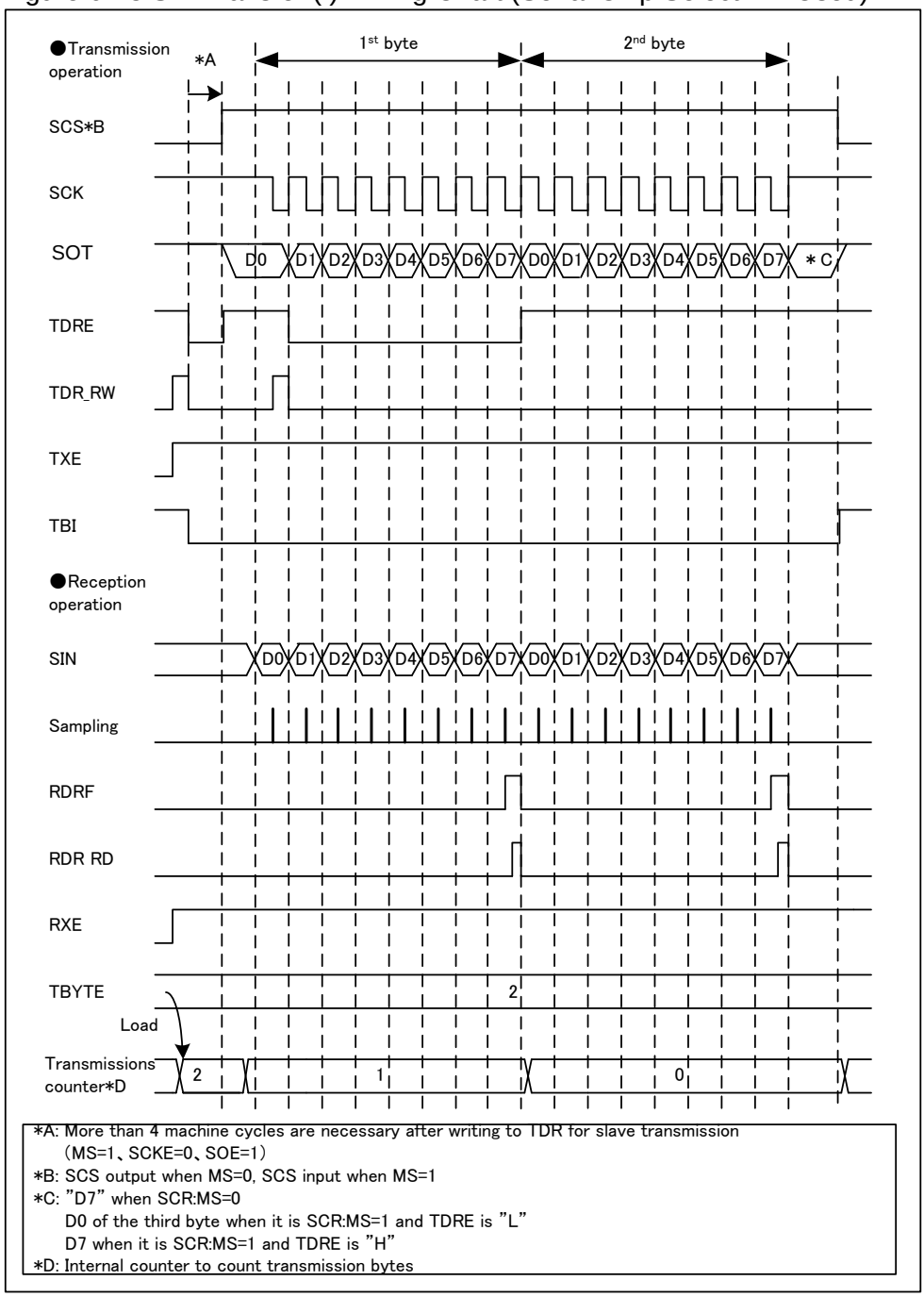
(1) Disable serial data output (SMR:SOE=0), enable reception interrupt (SCR:RIE=1), enable reception operation (SCR: RXE=1), and enable transmission operation (SCR:TXE=1). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the falling edge of the serial clock input (SCK).

(2) To continue reception operation, write dummy data to TDR between the reception interrupt request and the rising edge of the next serial clock (SCK).

(3) To switch from reception operation to transmission operation, enable serial data output (SMR:SOE=1), disable reception interrupt (SCR:RIE=0), and disable reception operation (SCR:RXE=0) between the reception interrupt request and the rising edge of the next serial clock (SCK), and after transmission data is written to TDR and reception operation finishes, transmission data will be output in synchronization with the rising edge of the serial clock.

■ SPI Transfer (I) Timing Chart (Serial Chip Select Pin Used)

Figure 6-13 SPI Transfer (I) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*:"n" shows the number of serial chip select pin used.

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active at the same time of outputting the first bit and transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1 and a transmission interrupt request will be generated when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active at the same time of outputting the first bit and transmission/reception will be started after the setup time is passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) The reception data will be sampled at a falling edge of the serial clock output (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

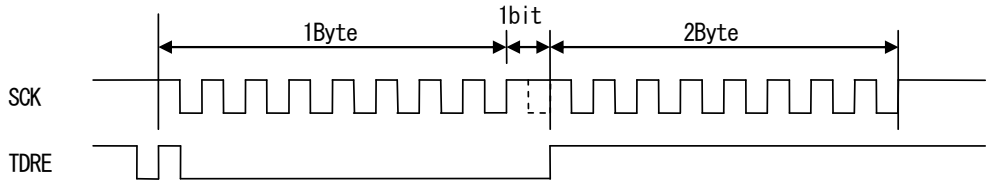
(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

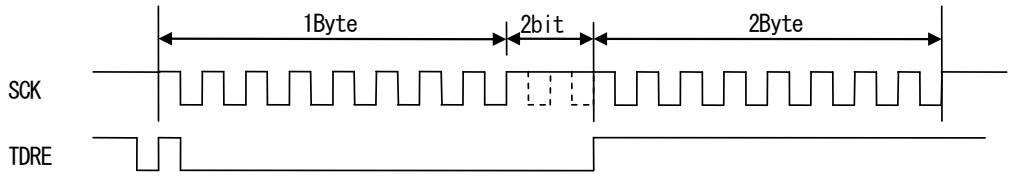
● Continuous Data Transmission or Reception Wait Operation

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous data transmission or reception, a wait will be inserted between frames.

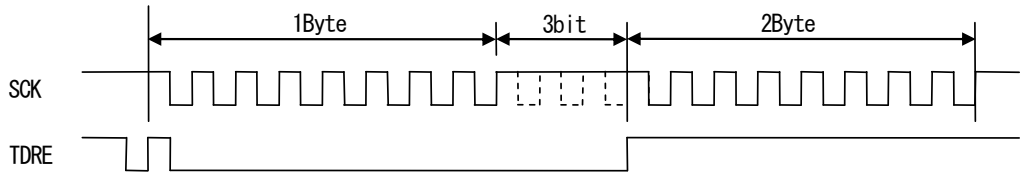
■ ESCR.WT1=0, ESCR.WTO=1(for master)



■ ESCR.WT1=1, ESCR.WTO=0(for master)



■ ESCR.WT1=1, ESCR.WTO=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started and the first bit will be output when serial chip select pin (SCS) becomes active. After transmission operation is started, the transmission data will be output in synchronization with a rising edge of the serial clock (SCK) output.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

Note:

After transmission operation is enabled (SCR:TXE=1), when you write first transmission data to the TDR while serial clock (SCK) is at a level other than mark level, the first bit of the data will not be output and correct transmission operation will not be performed. After transmission operation is enabled (SCR:TXE=1), writing the first transmission data to the TDR must be made while serial clock (SCK) is at mark level.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the received data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception will be started and the first bit will be output. When transmission/reception is started, the transmission data will be output in synchronization with the rising edge of serial clock (SCK) input. When the first bit of transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) The reception data will be sampled at a falling edge of the serial clock input (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

6.2.4. SPI Transfer (II)

This section explains the SPI Transfer (II).

■ 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 16, 20, 24, 32 bits

■ Register Settings

The following table lists the register settings required for SPI Transfer (II).

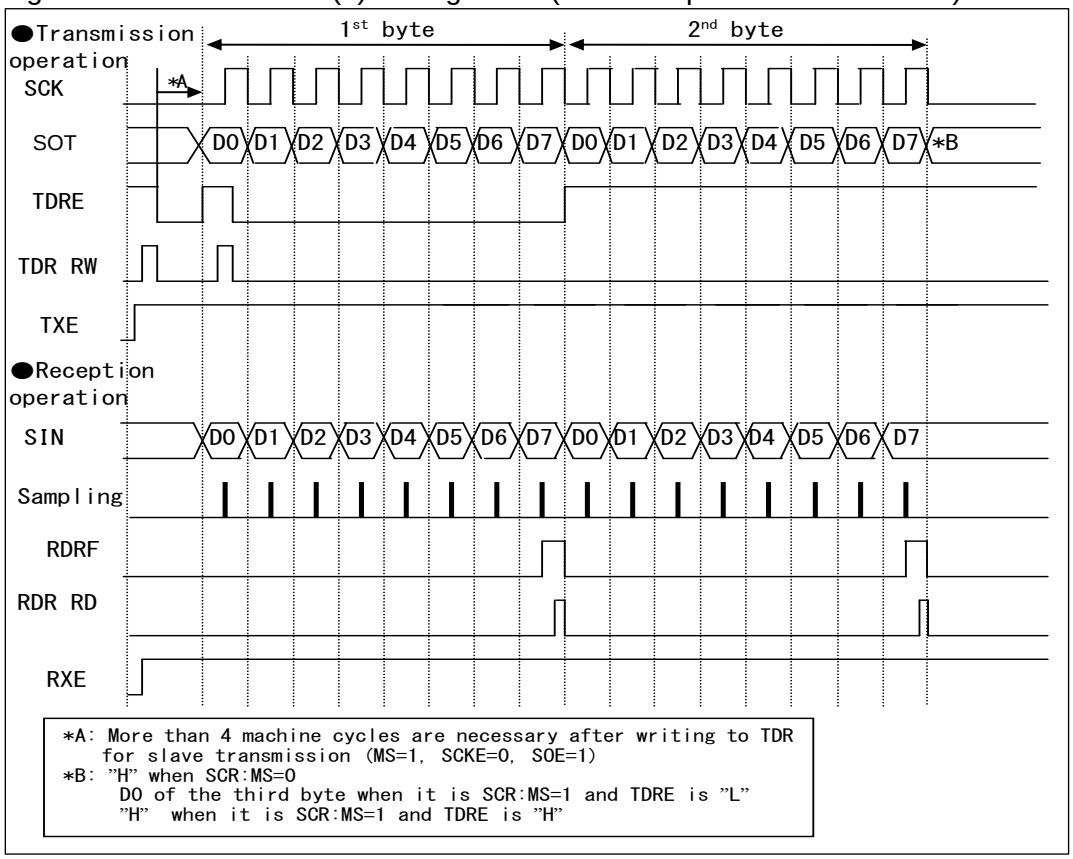
SCR:SPI*[1]=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*[1]=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0
*1: Bit settings depend on the condition. See Table 6-2 for details.

Note:

Use proper usage for setting the registers other than the above.

■ SPI Transfer (II) Timing Chart (Serial Chip Select Pin Unused)

Figure 6-14 SPI Transfer (II) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Half a cycle before a rising edge of the first serial clock (SCK), SSR:TDRE is set to 1. This results in generating a transmission interrupt request when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the receive data at a rising edge of the serial clock (SCK) output.

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

● Transmission/Reception operation

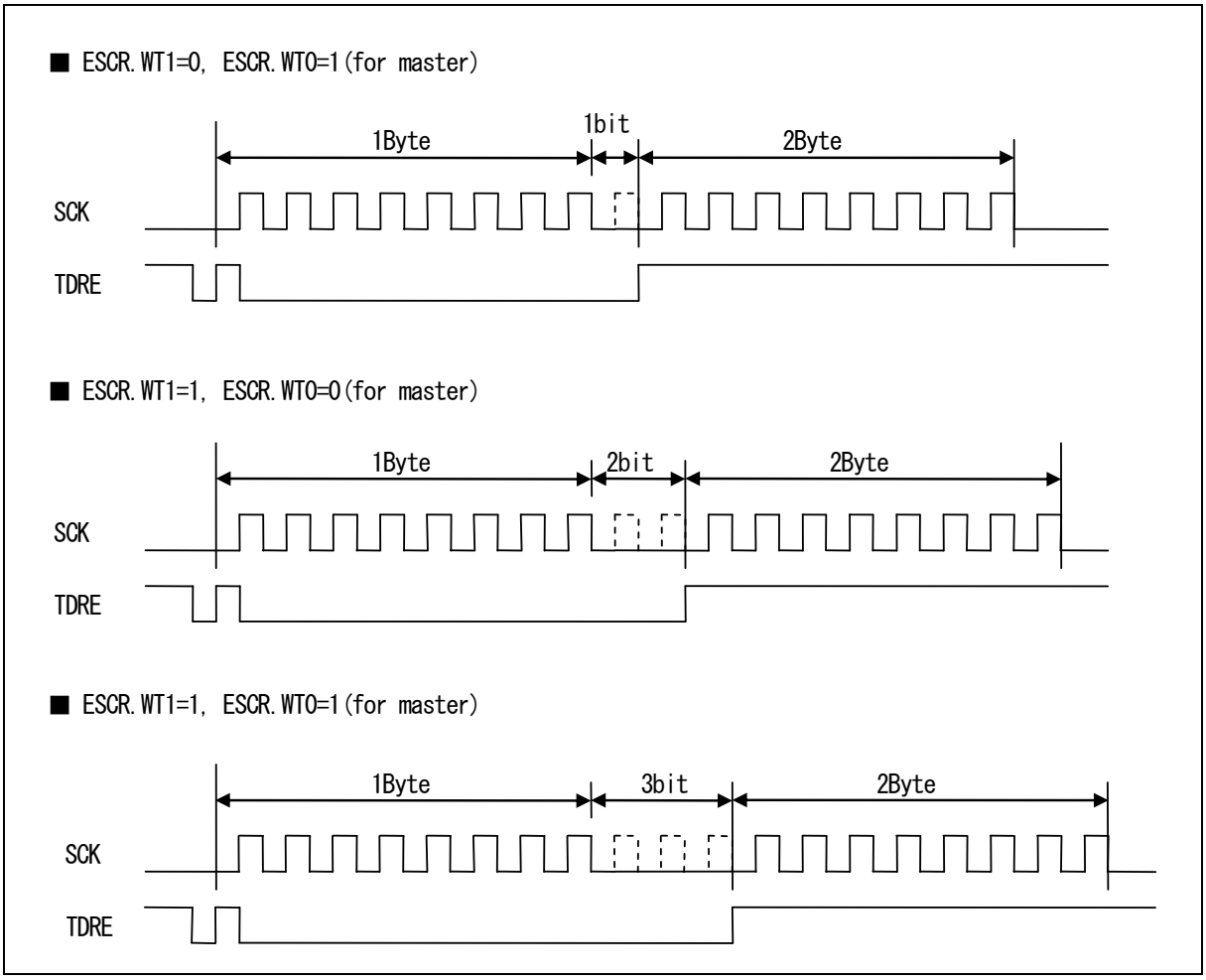
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

● Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0.)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. Therefore, the first bit is output. This results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) input.

(2) When the first bit of the transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

Note:

After the transmission operation is enabled (SCR:TXE=1), if transmission data is written to the first TDR except when serial clock (SCK) is at the mark level, the first bit of data is not output and transmission is not operated normally. After the transmission operation is enabled (SCR:TXE=1), write transmission data to the first TDR when serial clock (SCK) is at the mark level.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

● Continuous change from reception operation to transmission operation

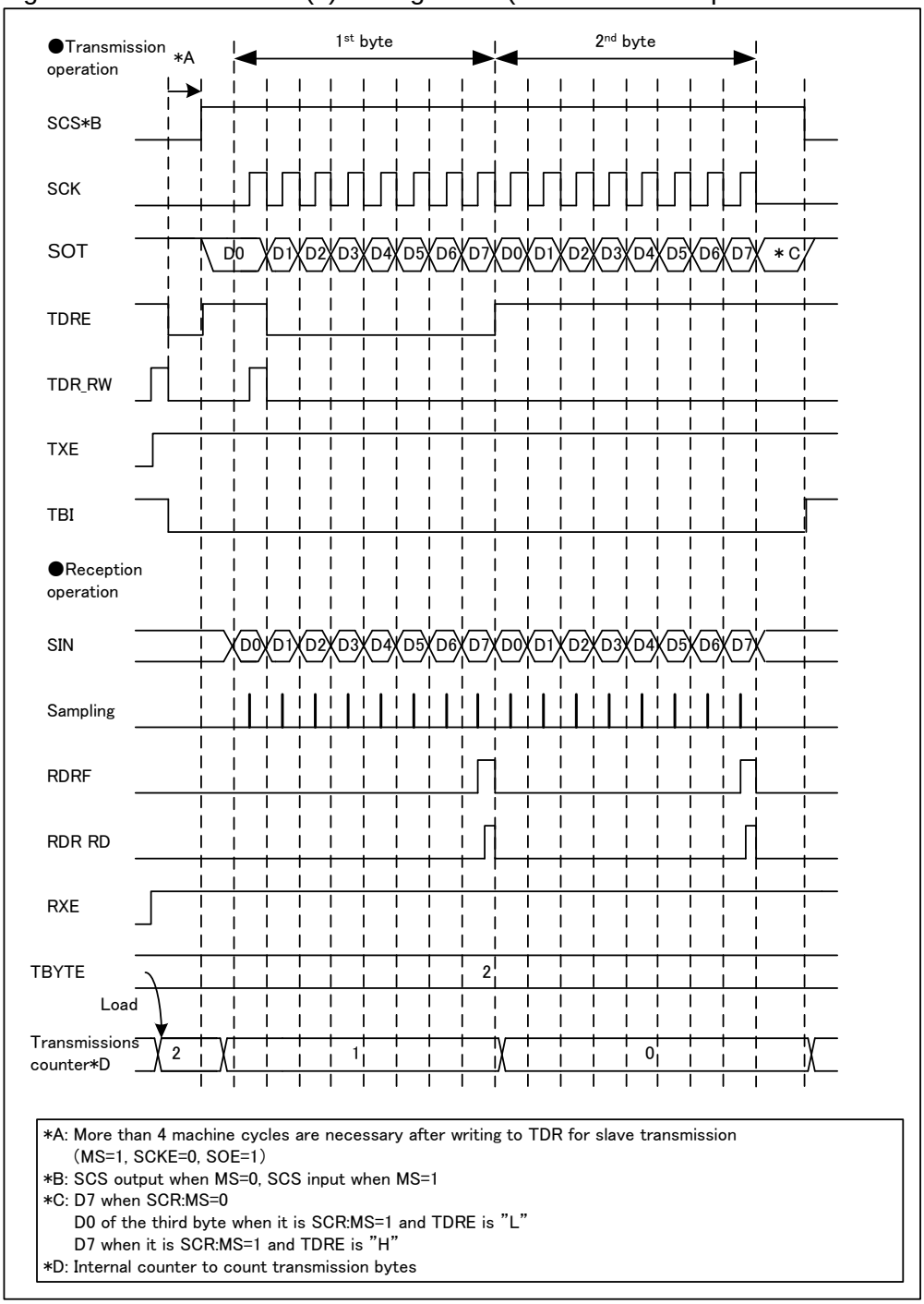
(1) Disable serial data output (SMR:SOE=0), enable reception interrupt (SCR:RIE=1), enable reception operation (SCR: RXE=1), and enable transmission operation (SCR:TXE=1). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the falling edge of serial clock (SCK) input.

(2) To continue reception operation, write dummy data to TDR between the reception interrupt request and the rising edge of the next serial clock (SCK).

(3) To switch from reception operation to transmission operation, enable serial data output (SMR:SOE=1), disable reception interrupt (SCR:RIE=0), and disable reception operation (SCR:RXE=0) between the reception interrupt request and the rising edge of the next serial clock (SCK), and after transmission data is written to TDR and reception operation finishes, transmission data will be output in synchronization with the rising edge of the serial clock.

■ SPI Transfer (II) Timing Chart (When Serial Chip Select Pin Used)

Figure 6-15 SPI Transfer (II) Timing Chart (When Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: n is the serial chip select pin number to be used.

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. After that, the serial chip select pin (SCS) becomes active at the same time the first bit is output, and the transmission operation starts after the setup time of the serial chip select pin has passed. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) The transmission operation is terminated after the data transmission is completed as many as the number of times set with TBYTE.

(4) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and starts the reception operation after the setup time of that pin has passed. Starting the reception operation samples the receive data at a rising edge of the serial clock (SCK) output.

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be output.
At this time, the reception data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) The reception operation is terminated after the data reception is completed as many as the number of times set with TBYTE.

(5) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. After that, the serial chip select pin (SCS) becomes active at the same time the first bit is output, and the transmission/reception operation starts after the setup time of the serial chip select pin has passed. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock, SSR:TDRE is set to 1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) Reception data is sampled by the rising edge of the serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, the reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

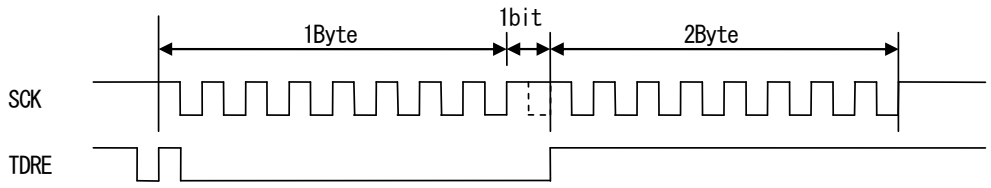
(4) The transmission/reception operation is terminated after the data transmission/reception is completed as many as the number of times set with TBYTE.

(5) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

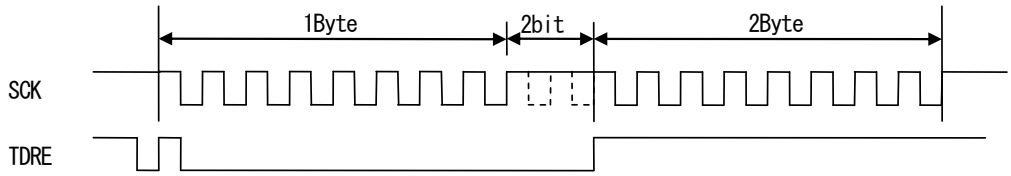
● Successive data transmission or reception wait operation

(1) If setting other than (ESCR.WT1, ESCR.WT0)=(0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

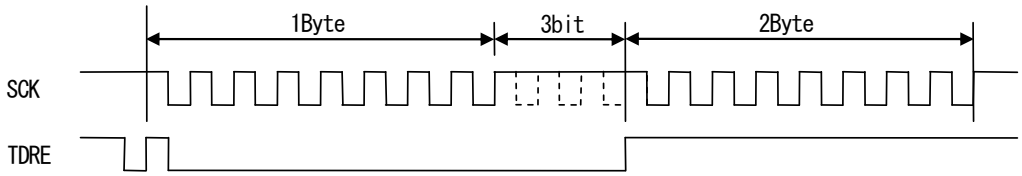
■ ESCR.WT1=0, ESCR.WTO=1(for master)



■ ESCR.WT1=1, ESCR.WTO=0(for master)



■ ESCR.WT1=1, ESCR.WTO=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0)

● Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) When the serial chip select pin (SCS) becomes active, the transmission operation is started and the data of the first bit is output. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.

(3) When the transmission data of the first bit is output, SSR:TDRE=1 is set, and when the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(4) When the serial chip select pin (SCS) becomes inactive, the transmission operation is terminated and the serial output pin (SOT) becomes "H".

Note:

After the transmission operation is enabled (SCR:TXE=1), if transmission data is written to the first TDR except when serial clock (SCK) is at the mark level, the first bit of data is not output and transmission is not operated normally. After the transmission operation is enabled (SCR:TXE=1), write transmission data to the first TDR when serial clock (SCK) is at the mark level.

● Reception operation

(1) With serial data output disabled (SMR:SOE=0), and reception operation enabled (SCR:RXE=1), the reception operation starts when the serial chip select pin (SCS) becomes active, and then the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the reception data (RDR) can be read.

(3) Reading the reception data (RDR) clears SSR:RDRF to "0".

(4) When the serial chip select pin (SCS) becomes inactive, the reception operation is terminated.

● Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When the serial chip select pin (SCS) becomes active, the transmission/reception operation is started and the data of the first bit is output. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the transmission data of the first bit is output, SSR:TDRE=1 is set, and when the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) Reception data is sampled by the rising edge of the serial clock (SCK) input. When the last bit of reception data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, the reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

(4) When the serial chip select pin (SCS) becomes inactive, the transmission/reception operation is terminated and the serial output pin (SOT) becomes "H".

6.2.5. Operation of Serial Timer

This section explains the operation of the serial timer.

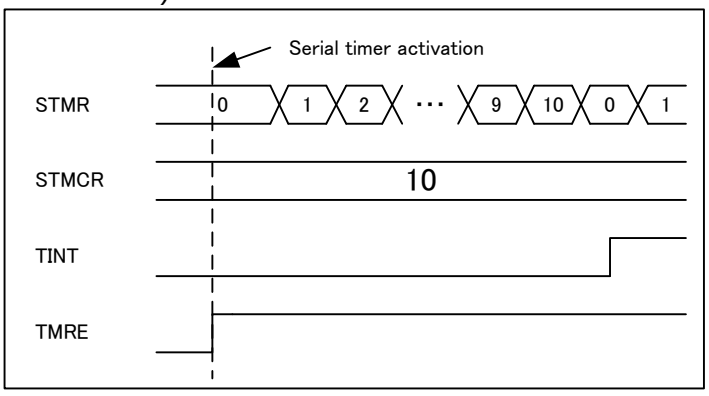
The serial timer can be used for either of the timer function or the synchronous transmission function.

● How to Start Serial Timer

To start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE).

- Start by using the serial timer enable bit (SACSR:TMRE)
  When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

Figure 6-16 Start by Using Serial Timer Enable Bit
(STMCR="10", SACSR:TSYNE="0")



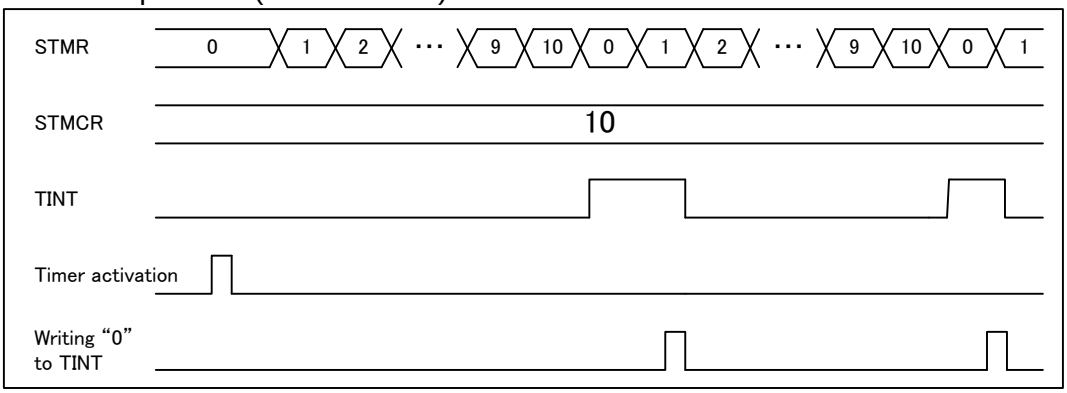
● How to Stop Serial Timer

When the serial timer enable bit (SACSR:TMRE) is set to "0", the serial timer will stop. The value of the serial timer register (STMR) is retained.

Timer Operation

When the synchronous transmission enable bit (SAGSR:TSYNE) is set to "0", the serial timer operates as a timer. If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 6-17 Timer Operation (STMCR="10")



Note:

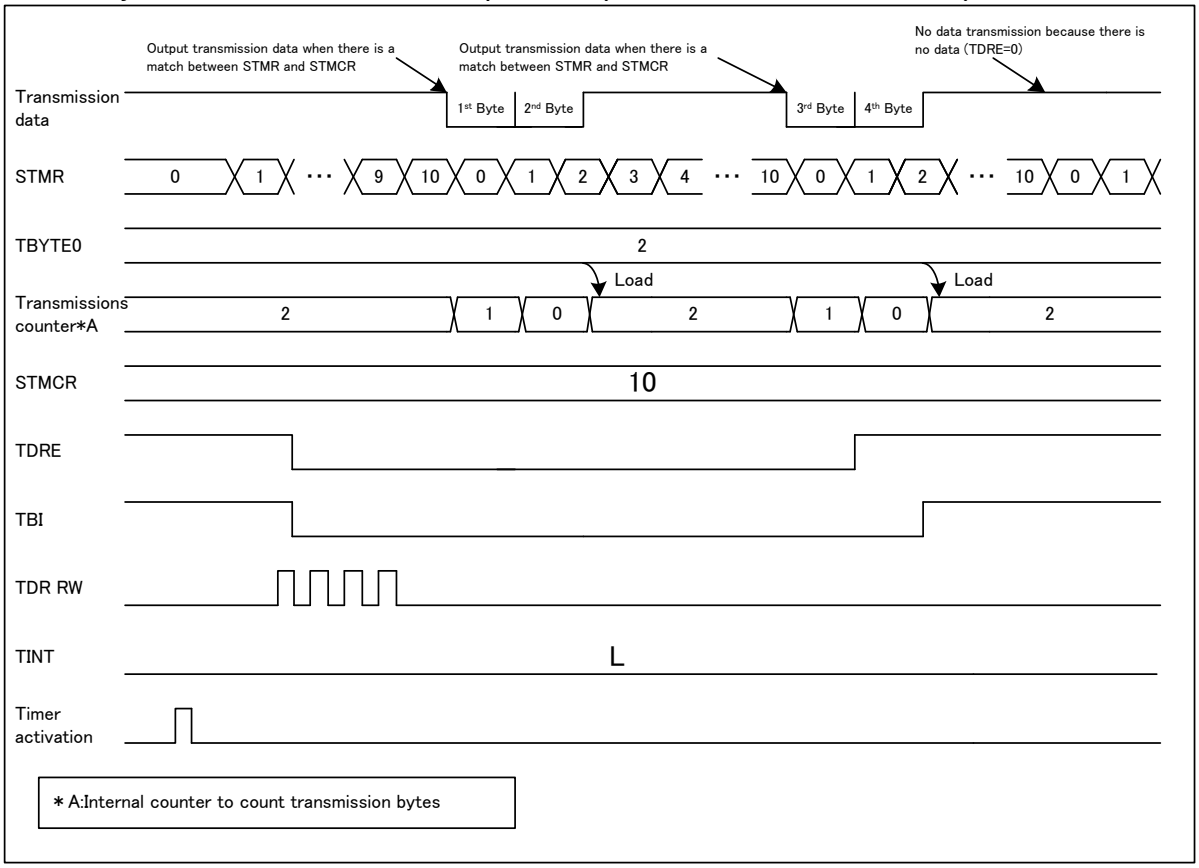
When the timer comparison register (STMCR) is set to (0000) H with the synchronous transmission disabled (SACSR:TSYNE=0), the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".

Synchronous Transmission Operation

When the synchronous transmission enable bit (SAGSR:TSYNE) is set to "1", the serial timer is used for a synchronous transmission.
The synchronous transmission operates as shown below:

1. If the transmission data register contains data (SSR:TDRE="0") and the serial timer register (STMR) matches the serial timer comparison register (STMCR), the transmission operation starts and the serial timer register (STMR) is reset to "0". It continues to send the same number of data as the number set to TBYTE0.
2. If the data transmission is completed as many as the number of data set to TBYTE0, then, the transmission operation stops until the serial timer register (STMR) matches the serial timer comparison register (STMCR).

Figure 6-18 Synchronous Transmission Operation (STMCR="10", TBYTE0="2")



If the synchronous transmission is enabled (SACSR:TSYNE="1") and the serial timer register (STMR) matches the serial timer comparison register (STMCR), the transmission will not start under the following conditions.

- When transmission is disabled (SCR:TXE="0")
- In slave mode (SCR:MS="1")
- When the chip select error (SACSR:CSE="1") occurred
- When there is no valid data in the transmission data register (SSR:TDRE="1")

However, if the transmission data register contains no valid data (SSR:TDRE=1), the synchronous transmission is enabled (SACSR:TSYNE="1"), and the serial timer register (STMR) matches the serial timer comparison register (STMCR), writing the transmission data to transmission data register starts a transmission immediately.
If the transmission data register (TDR) has valid transmission data (SSR:TDRE="0") after the data transmission is completed as many as the number of data set to TBYTE, then, the transmission data will not be sent until the serial timer register (STMR) matches the serial timer comparison register (STMCR).
However, if the synchronous transmission is enabled (SACSR:TSYNE="1") and the serial timer register (STMR) matches the serial timer comparison register (STMCR) under the following conditions, the transmission will not be stopped and the next transmission will be started after sending as much data as TBYTE0.

- Transmission operation in progress (SSR:TBI="0")

If synchronous reception operation is performed, disable serial data output (SMR:SOE="0"), enable transmission operation (SCR:TXE="1"), enable reception operation (SCR:RXE="1"), and write dummy data to TDR as many as the number of receptions.

Note:

When the transmit data register (TDR) has no valid transmission data (SSR:TDRE="1") before sending the data frame of TBYTE setting, the following operation will be performed.

- When the transfer byte error is enabled (TBEEN="1"), a chip select error (SACSR:CSE="1") occurs. When the chip select error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmit data register (TDR) does not start transmission operation.
- When the transfer byte error is disabled (TBEEN="0"), transmission operation will be stopped until transmission data is written to the transmit data register (TDR). Transmission operation will be restarted when transmission data is written to the transmit data register (TDR).

6.2.6. Operation of Serial Chip Select

This section explains the operation of serial chip select.

This section explains the operation of the serial chip select.

● Operation of master mode (SCR:MS="0")

In master mode (SCR:MS=0), the serial chip select pin operates as shown below:

(1) With the serial chip select operation enabled (SCSCR:CSENn="1") and transmission enabled (SCR:TXE="1"), writing transmission data makes the serial chip select pin active.

(2) The transmission/reception operation starts after the setup time of the serial chip select pin has passed.

(3) The transmission/reception operation is terminated after the data transmission/reception operation is repeated as many as the number of times set with TBYTE.

(4) Then, after the hold time of serial chip select pin has passed, the serial chip select pin becomes inactive.

Figure 6-19 Operation of Serial Chip Select (Master Transmission (MS="0"), Normal Transfer (SPI="0"), SCINV="0")

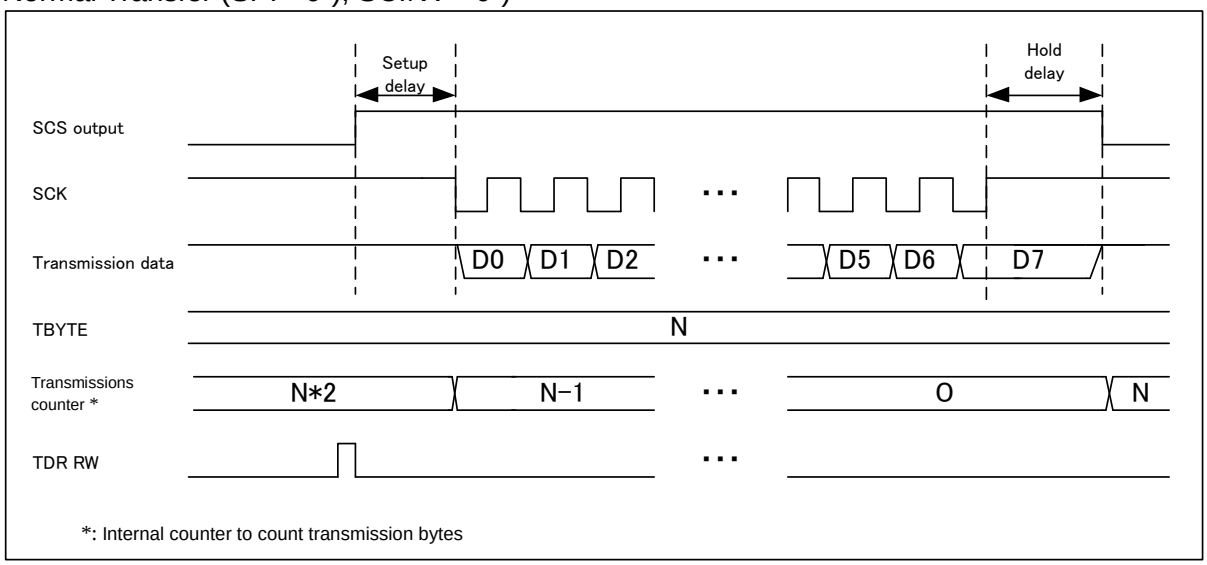
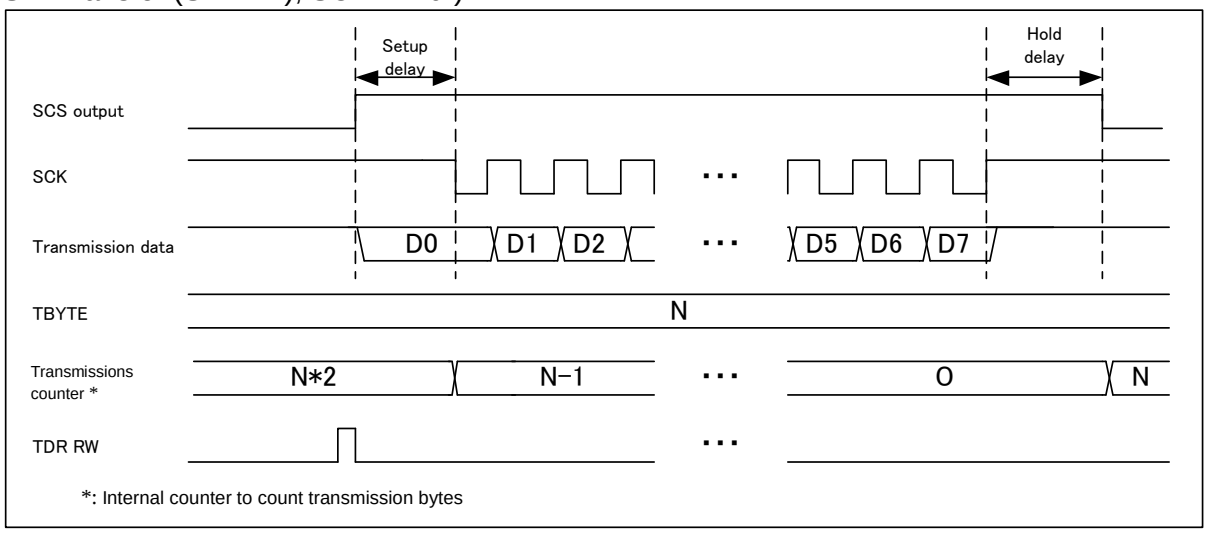


Figure 6-20 Operation of Serial Chip Select (Master Transmission (MS="0"), SPI Transfer (SPI="1"), SCINV="0")



Notes:

- If the transmission is disabled (SCR:TXE="0") and the software reset is set (SCR:UPCL=1) when the serial chip select pin is active, the serial chip select pin becomes inactive.
- If the transmission data is written to the hold delay time of the serial chip select pin, the pin does not become inactive and the next transmission data is sent.
- When the serial chip select pin does not maintain its active state (SCSCR:SCAM=0) and becomes inactive, transmission bus idle (SSR:TBI=1) is set.
- If SCSCR:CSEN3 to CSEN0 is set to "0000b" in master mode (SCR:MS=0), transmission/reception operation is performed regardless of the serial chip select pin.
- If the transmit data register (TDR) has no valid transmission data (SSR:TDRE=1) when 1 frame transmission is completed when the number of the frames sent is less than the number set to TBYTE, the following operation will

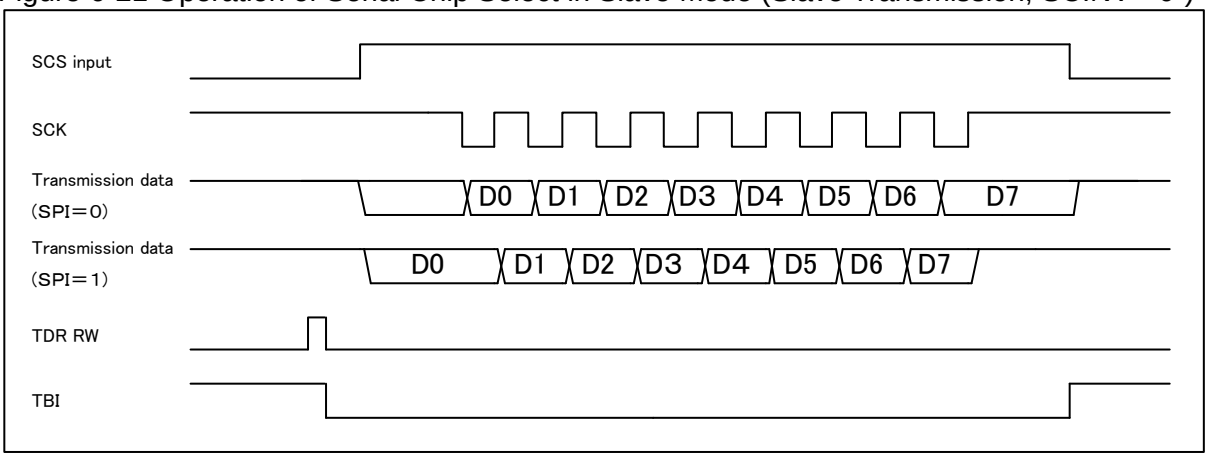
be performed.

- When the transfer byte error is enabled (TBEEN=1), a chip select error (SACSR:CSE=1) occurs. After the hold delay time has passed since the chip select error (SACSR:CSE=1) occurred, the serial chip select pin becomes inactive. When the chip select error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmit data register (TDR) does not start transmission operation.
- When the transfer byte error is disabled (TBEEN=0), transmission operation will be stopped until transmission data is written to the transmission data register (TDR). At this time, the serial chip select pin is active. Transmission operation will be restarted when transmission data is written to the transmit data register (TDR).

● Operation of slave mode (SCR:MS="1")

When the serial chip select pin 0 (SCS0) is enabled (SCSCR:CSEN0="1") and the serial chip select pin input becomes active, the transmission or reception operation is performed in synchronization to the serial clock (SCK). Then, when the serial chip select pin input becomes inactive, the transmission or reception operation is terminated.

Figure 6-21 Operation of Serial Chip Select in Slave Mode (Slave Transmission, SCINV="0")



Notes:

- It does not operate even if the serial clock is input when the serial chip select pin input is inactive.
- If the serial chip select input becomes inactive before sampling a bit at the end during reception operation, the data being received is deleted.
- If the serial chip select input becomes inactive during transmission operation, the data being transmitted is deleted and the chip select error occurs (SACSR:CSE).
- When the serial chip select pin input becomes inactive, the transmission bus idle (SSR:TBI=1) is set.
- If SCSCR:CSEN0 is set to "0" in slave mode (SCR:MS=1), transmission/reception operation is performed regardless of the serial chip select pin.

● Timing Adjustment of Serial Chip Select

When the serial chip select operation is enabled (SCSCR:CSENn="1") in master mode (SCR:MS=0), the setup delay time, hold delay time, and deselection time can be adjusted by adjusting the serial chip select timing register (SCSTR3 to SCSTR0).

- Setup delay time
  Time from the instant when the serial chip select pin becomes active to the instant when the serial clock is output.

See Figure 6-22 and Figure 6-23 for the specification of the setup delay time.
It can be adjusted by the chip select setup delay bit (SCSTR0:CSSU7 to CSSU0).

- Hold delay time
  Time from the instant when the serial clock output is completed to the instant when the serial chip select pin becomes inactive. See Figure 6-22 and Figure 6-23 for the specification of the hold delay time.
  It can be adjusted by the chip select hold delay bit (SCSTR1:CSHD7 to CSHD0).

- Deselection time
  Minimum time from the instant when the serial chip select pin becomes inactive to the next instant when the pin turns active. Even if transmission data is written to the transmit data register (TDR) during the deselection time, the serial chip select pin does not become active until the deselection time end. See Figure 6-22 and Figure 6-23 for the specification of the deselection time.

Figure 6-22 Timing Adjustment (Normal Transfer (SPI="0"), SCINV="0")

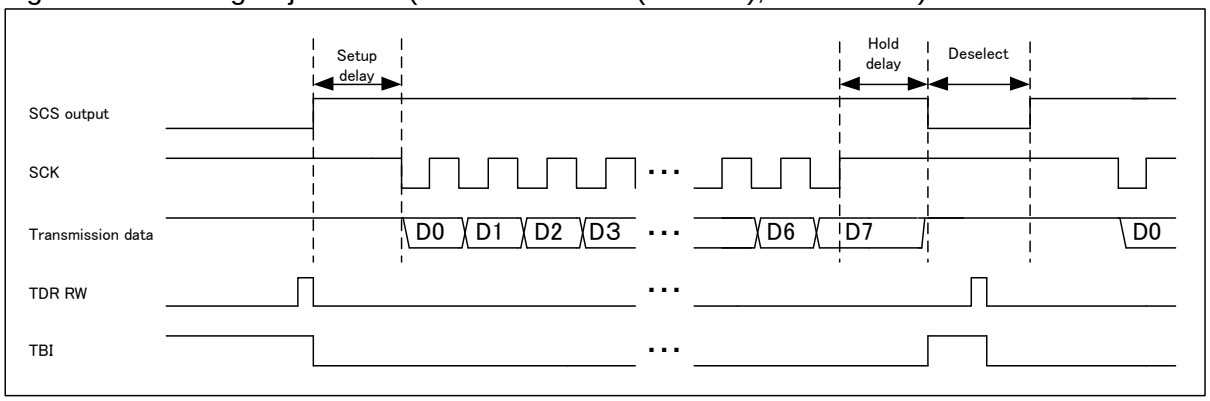
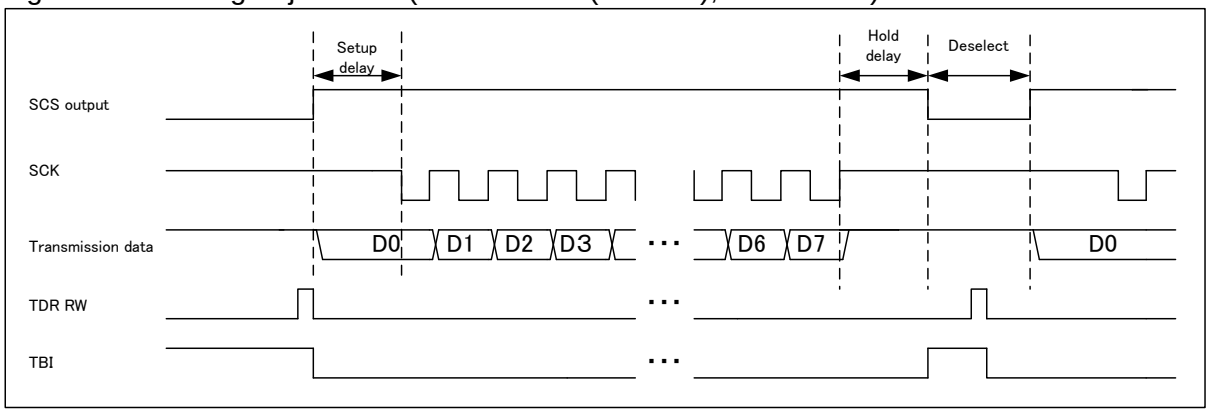


Figure 6-23 Timing Adjustment (SPI Transfer (SPI="1"), SCINV="0")



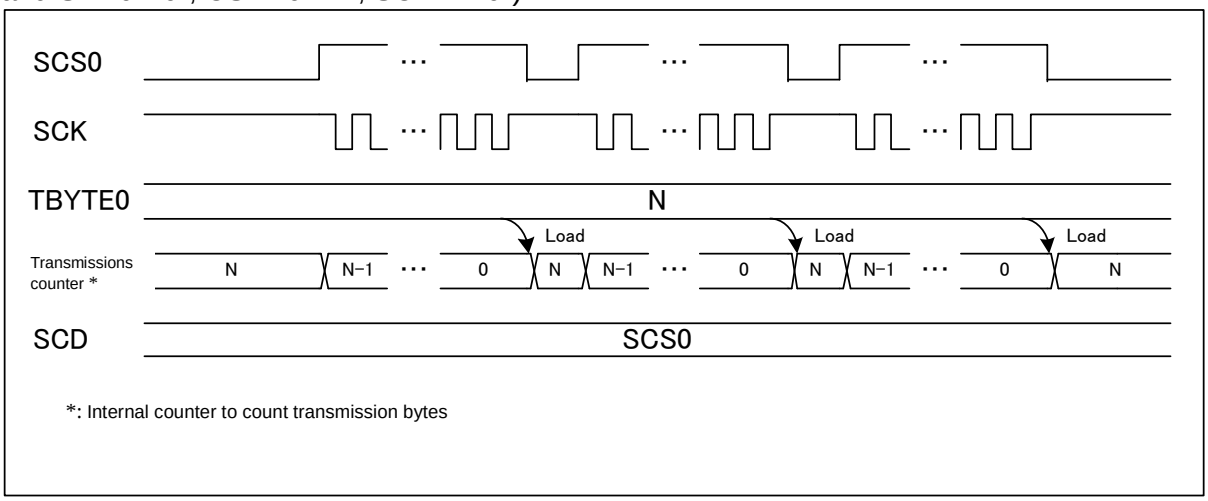
Note:

With the normal transfer (SPI=0) and no hold delay time (SCSTR1:CSHD7 to CSHD0="00_H"), the chip select pin may become inactive before the sampling of the last bit. In such cases, adjust the timing by increasing the value of SCSTR1:CSHD7 to CSHD0.

Single Operation of Chip select pin (Only Valid in Master Mode (SCR:MS="0"))

If the serial chip select start bit (SCSCR:SST1 and SST0) and the serial chip select end bit (SCSCR:SED1 and SED0) are equal, only the specified serial chip select pin operates.
When the serial chip select does not maintain its active state (SCSCR:SCAM=0), the serial chip select pin becomes inactive for each the data transmission/receptions for the number of times set with TBYTE.
See "Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))" for the operation of the serial chip select pin when the serial chip select maintains its active state (SCSCR:SCAM=0).

Figure 6-24 Single Operation of Chip Select (SST1 and SST0="0", SED1 and SED0="0", CSEN0="1", SCAM="0")



Note:

At the single operation, timing adjustment of the serial chip select pin (setup time, hold time, deselection time) is valid.

Rounding Operation of Chip Select Pin (Only Valid in Master Mode (SCR:MS=0))

(1) With the serial chip select output enabled (SCSCR:CSOE="1") and transmission enabled (SCR:TXE="1"), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST1 and SST0) becomes active first when transmission data is written.

(2) When the serial chip select does not maintain its active state (SCSCR:SCAM=0), the serial chip select pin becomes inactive after the data transmission/receptions are completed as many as the number of times set with TBYTE. Then, the serial chip select pin becomes active, which has the number added 1 to the number of the serial chip select pin that became active previously.
However, if the serial chip select pin to become active next is disabled (SCSCR;CSENn=0), the serial chip select pin will not become active and be skipped.

(3) If the active serial chip select pin number and the one specified by the serial chip select end bit (SCSCR:SED1 and SED0) are matched, the serial chip select pin to become active next is the one specified by the serial chip select start bit (SCSCR:SST1 and SST0).

See "Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode

(SCR:MS=0))" for the operation of the serial chip select pin when the serial chip select maintains its active state (SCSCR:SCAM=0).
Figure 6-25 shows a timing chart when the start pin of serial chip select pin is SCS0 (SST1 and SST0=0) and the end pin is SCS3 (SED1 and SED0=3).

Figure 6-25 Round Operation of Chip Select (SST1 and SST0=0, SED1 and SED0=3, CSEN3=1, CSEN2=1, CSEN1=1, CSEN0=1, SCAM=0)

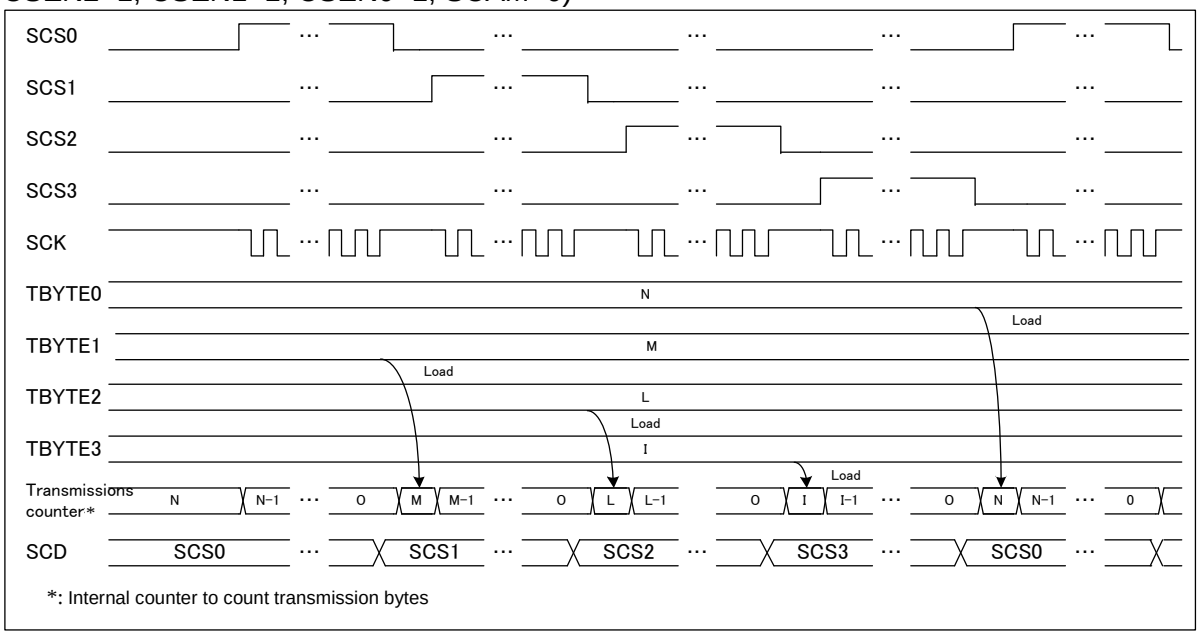


Figure 6-26 shows a timing chart when the start pin of serial chip select pin is SCS1 (SST1 and SST0=1) and the end pin is SCS2 (SED1 and SED0=2).

Figure 6-26 Round Operation of Chip Select (SST1 and SST0=1, SED1 and SED0=2, CSEN3=0, CSEN2=1, CSEN1=1, CSEN0=0, SCAM=0)

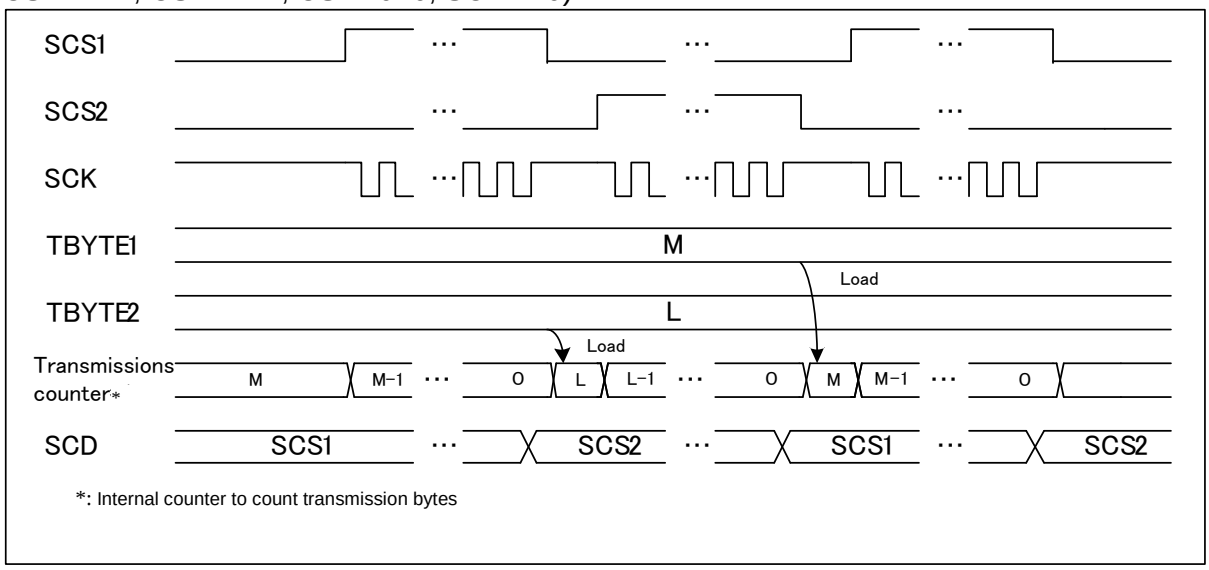
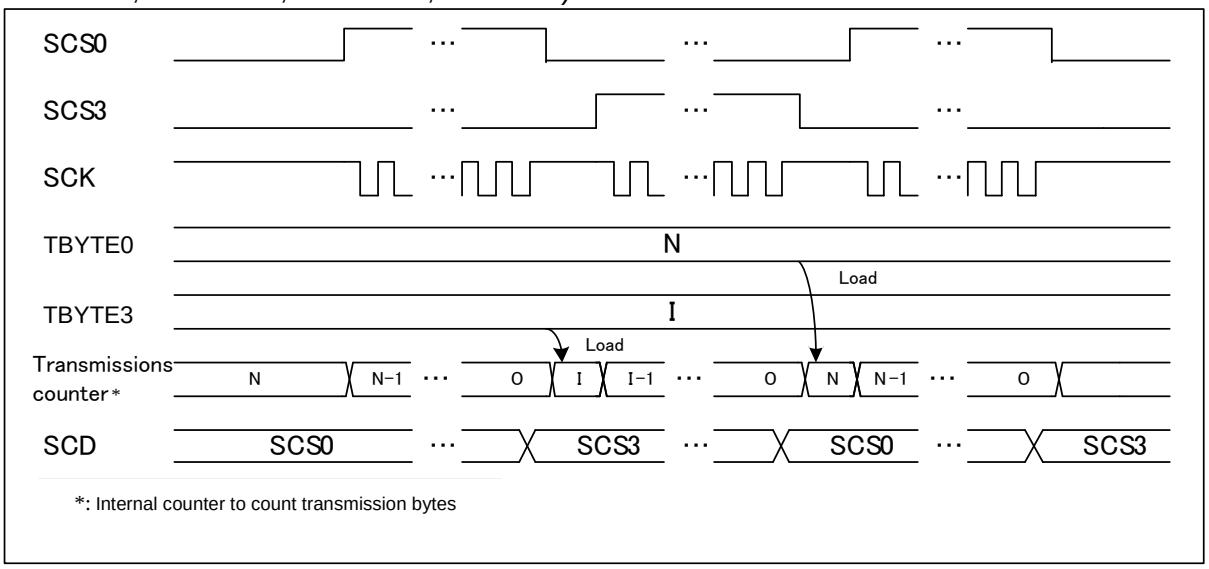


Figure 6-27 shows a timing chart when the start pin of serial chip select pin is SCS0 (SST1 and SST0=0), the end pin

is SCS3 (SED1 and SED0=3), and the chip select pin 1 and pin 2 is disabled (CSEN1 and CSEN2="00"b). After the serial chip select pin 0 becomes active, the pin 1 and pin 2 are skipped, and the pin 3 becomes active.

Figure 6-27 Round Operation of Chip Select (SST1 and SST0=0, SED1 and SED0=3, CSEN3=1, CSEN2=0, CSEN1=0, CSEN0=1, SCAM=0)



Notes:

- When transmission operation disabled (SCR:TXE=0) is changed to enabled (SCR:TXE=1), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST1 and SST0) becomes active first.
- When the serial chip select maintains its active level (SCSCR:SCAM=1), it does not transit to the next serial chip pin even if transmission data is written to the transmission bus idle (SCR:TBI=1).
- At the round operation, timing adjustment of the serial chip select pin (setup time, hold time, deselection time) is valid.
- After a software reset (SCR:UPCL=1), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST1 and SST0) becomes active first.

Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))

When the serial chip select active maintaining bit (SCSCR:SCAM) is set to "1" and transmission operation is started, the serial chip select pin is maintained to be active.
The value of the serial chip select active maintaining bit is checked for each transmission as many as the number of times set with TBYTE. After the data transmission/receptions are completed as many as the number of times set with TBYTE, the pin operates as shown below.

- If the serial chip select active maintaining bit is "0", the serial chip select pin turns inactive after the hold delay time has passed.
- If the serial chip select active maintaining bit is "1" and the serial timer synchronous transmission is used, the serial chip select pin is maintained to be active. Then, when the serial timer value (STMR) and the serial timer comparison value (STMCR) are matched, transmission operation is restarted. After that, the serial chip select pin is maintained to be active until the frame transmission is completed as many as the number of times set with TBYTE.

After that, the serial chip select pin is maintained to be active until the frame transmission is completed as many as the number of times set with TBYTE.

- If the serial chip select active maintaining bit is "1" and the serial timer synchronous transmission, the serial chip select pin is maintained to be active. At that time, if the transmit data register (TDR) contains the transmission data (SSR:TDRE=0), the transmission operation is continued, and the serial chip select pin is maintained to be active until the next time the frames as many as the number of times set with TBYTE are sent.

If the serial chip select active maintaining bit (SCSCR:SCAM) is written to "0", it operates as shown below.

- The serial chip select pin becomes inactive after the data transmission/receptions are completed as many as the number of times set with TBYTE and the hold delay time has passed.

Under the following conditions, the serial chip select pin becomes inactive when the serial chip select active maintaining bit (SCSCR:SCAM) is used.

- When SCSCR:SCAM=0 after as many transmissions as the TBYTE count are made
- When the chip select error occurred (SACSR:CSE=1)
- When transmission is disabled (SCR:TXE=0)
- When software reset is performed (SCR:UPCL=1)

Note:

If the transmit data register (TDR) is empty (SSR:TDRE=1) when the transfer byte error is enabled (SACSR:TBEEN=1) and the data transmission/reception is not completed as many as the number of times set with TBYTE, the serial chip select pin is not retained and becomes inactive after the hold delay time has passed, and the chip select error (SACSR:CSE=1) occurs.

● Format Setting of Serial Chip Select Pin

Active level of the chip select for each serial chip select pin, mark level of the serial clock, enabling/disabling SPI mode, and data direction/length of serial data output can be set by using bit shown in Table 6-2.

Table 6-2 Format Setting of Serial Chip Select Pin

Condition		Active level of chip select	Serial clock inversion	SPI setting	Data direction	Data length
Chip select format enabled (SCR:CSFE="1") and master mode (SCR:MS="0")	Serial chip select pin 0 output	SCSCR0: CSLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
	Serial chip select pin 1 output	SCSFR0: CS1CSLVL	SCSFR0: CS1SCINV	SCSFR0: CS1SPI	SCSFR0: CS1BDS	SCSFR0: CS1L3-0
	Serial chip select pin 2 output	SCSFR1: CS2CSLVL	SCSFR1: CS2SCINV	SCSFR1: CS2SPI	SCSFR1: CS2BDS	SCSFR1: CS2L3-0
	Serial chip select pin 3 output	SCSFR2: CS3CSLVL	SCSFR2: CS3SCINV	SCSFR2: CS3SPI	SCSFR2: CS3BDS	SCSFR2: CS3L3-0
Chip select format disabled (SCR:CSFE="0")		SCSCR0: SCLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Slave mode (MS="1")						
Chip select unused (CSEN3 to CSEN0="0000_B")						

6.2.7. Test Mode

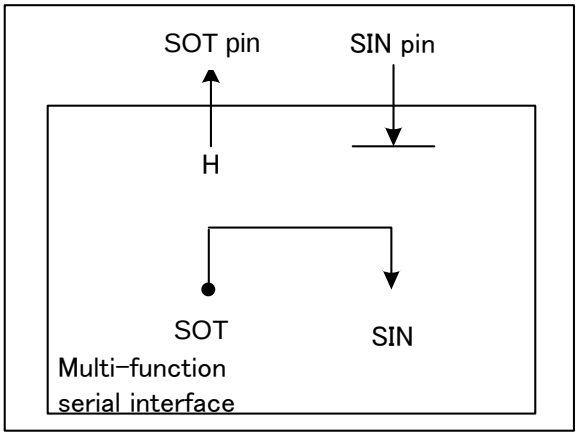
This section explains the test mode.

This section explains the operation of the test mode.

● Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 6-28 Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

6.2.8. Baud Rate Generation

This section explains the baud rate generation.

The dedicated baud rate generator works only in master operation. However, if the reception FIFO is to be used, set the dedicated baud rate generator even in slave operation.

The dedicated baud rate generator settings are different between the master and slave operations.

[1] Master operation

- The dedicated baud rate generator divides the internal clock and a baud rate is selected.
- There are two internal reload counters that correspond to the transmission and reception serial clocks, respectively. The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).
- The reload counter divides the internal clock with the setting value.

[2] Slave operation

The dedicated baud rate generator does not work in slave operation (SCR:MS=1). (The external clock entered from the clock input pin SCK is used without change.)

Note:

If the reception FIFO is to be used, set the dedicated baud rate generator even in slave operation.

■ Baud Rate Calculation

Set two 15-bit reload counters in the baud rate generator register (BGR). The baud rate calculation formulas are as follows:

(1) Reload value

$V = \phi : b - 1$

V: Reload value ϕ: bus clock frequency b: Baud rate

(2) Example of calculation

Reload values when setting the bus clock frequency at 16 MHz, usage of internal clock, and baud rate at 19200 bps are as follows:

Reload value:

$V = (16 \times 1{,}000{,}000) / 19200 - 1 = 832$

Therefore, the baud rate is

$b = (16 \times 1{,}000{,}000) / (832+1) = 19208$ bps

(3) Baud rate error

The baud rate error can be obtained using the following formula:

Error (%) = (calculated value – desired value) / desired value × 100

(Example) When you set bus clock at 20 MHz and target baud rate at 153600 bps:

Reload value = (20 × 1,000,000) / 153600 – 1 = 129

Baud rate (calculated) = (20 × 1,000,000) / (129 + 1) = 153846 bps

Error (%) = (153846 × 153600) / 153600 × 100 = 0.16(%)

Notes:

- Set the reload value to "0" to stop the reload counter.
- If the reload value is an even number, the "H" and "L" widths of the serial clock depend on the SCINV bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  If SMR:SCINV="0", the "H" width of the serial clock is longer by one cycle of the bus clock.
  If SMR:SCINV="1", the "L" width of the serial clock is longer by one cycle of the bus clock.
- Set the reload value to 3 or higher.

■ Reload values and Baud Rate for Bus Clock Frequencies

Table 6-3 Reload values and Baud Rate for Bus Clock Frequencies

Baud Rate (bps)	8MHz		10MHz		16MHz		20MHz		24MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	-	-	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	87	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: Setup values in the BGR register
- ERR: Baud Rate Error (%)

■ Reload Counter Functions

Reload counters, including transmission and reception reload counters, serve as the dedicated baud rate generators. It consists of a 15-bit register for reload values and generates a transmission/reception clock from the internal clock.

■ Count Start

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

■ Restart

The reload counter restarts under one of the following conditions:

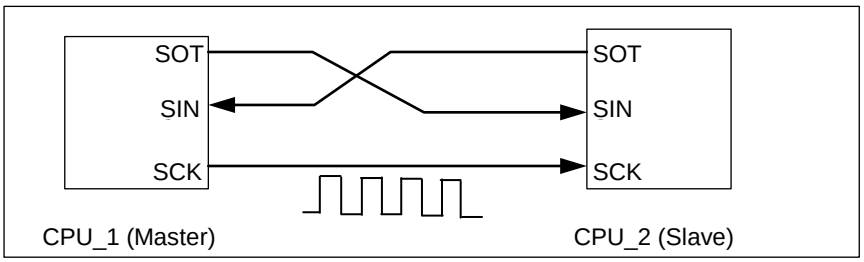
- **Common to the transmission and reception reload counters**
  Programmable reset (SCR:UPCL bit)

6.3. Setup Procedure and Program Flow

Setup procedure and program flow is shown.

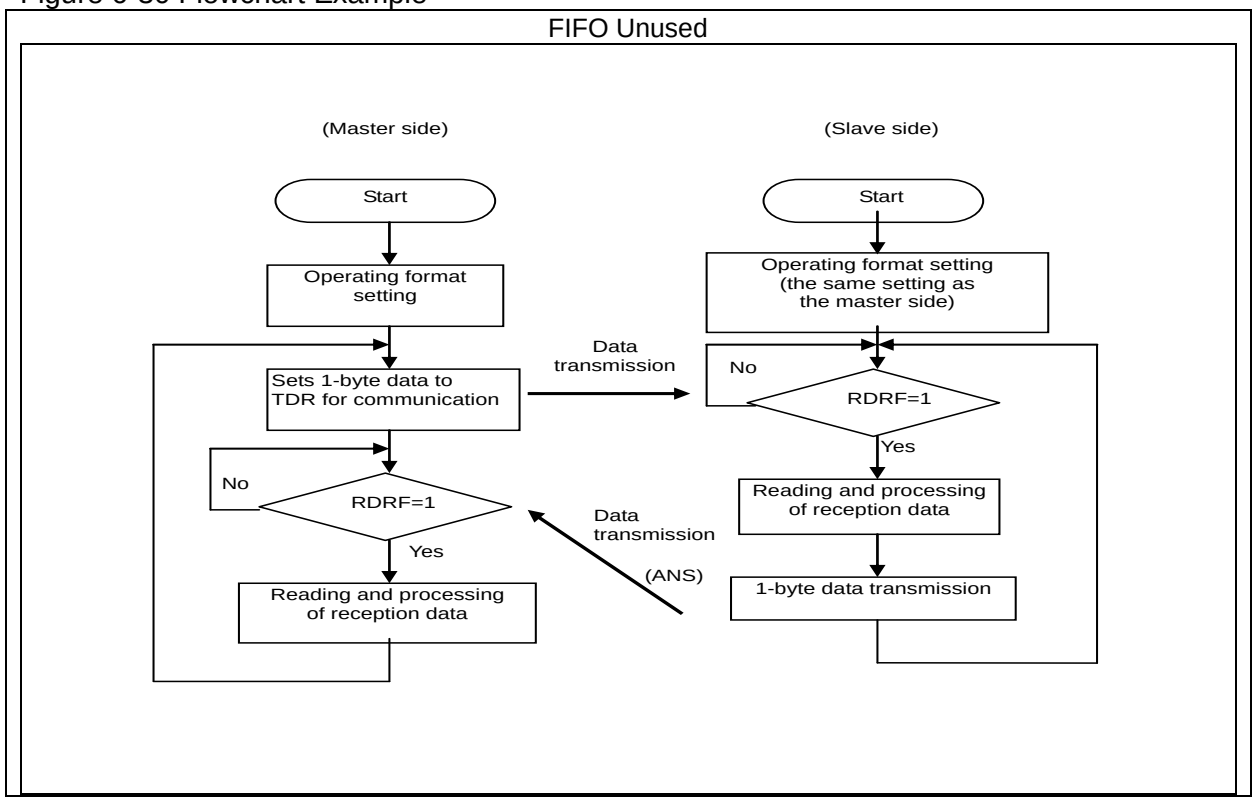
■ CPU Interconnection

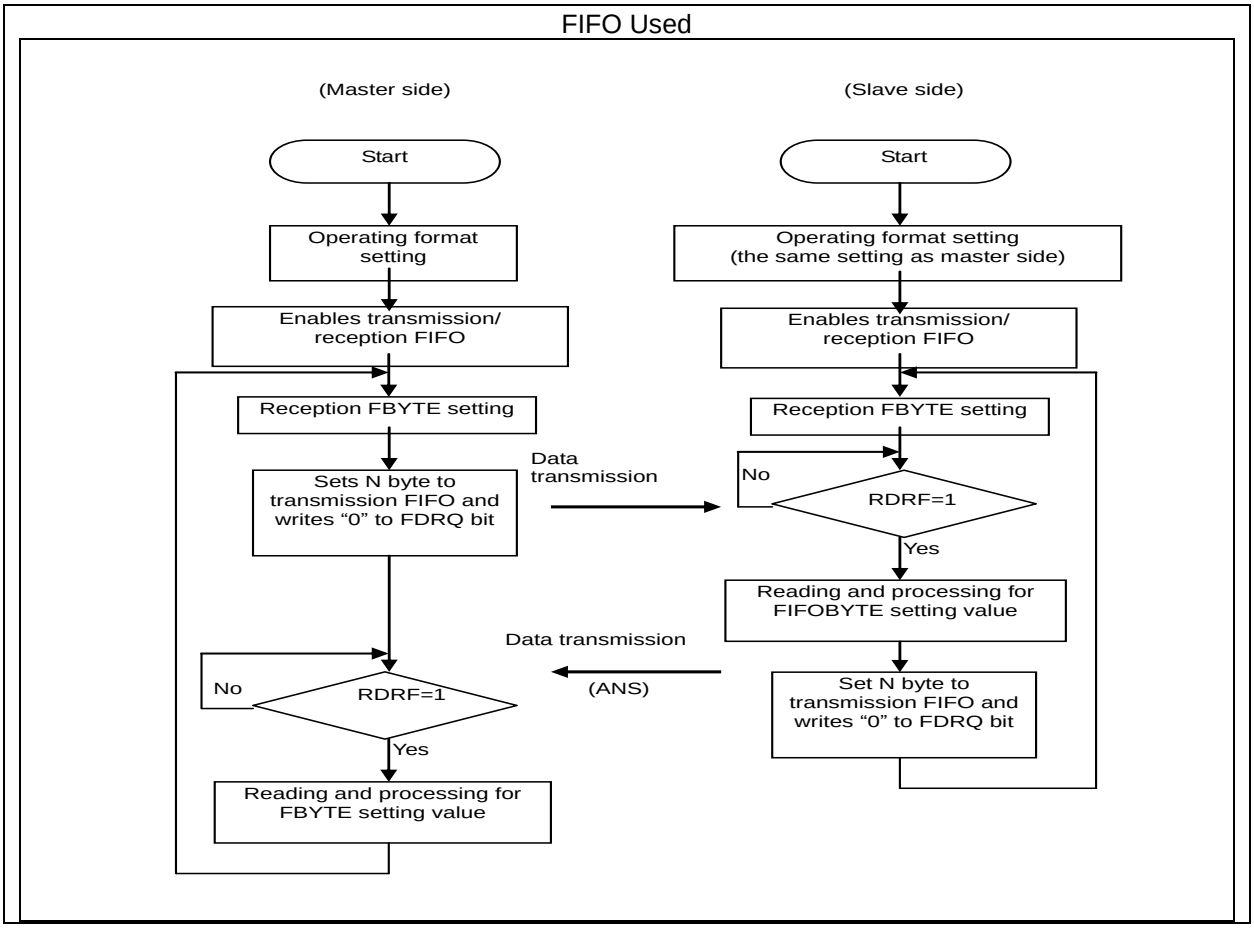
Figure 6-29 Example of Connection between CSIO Chips



■ Flowchart

Figure 6-30 Flowchart Example





7. Operation of LIN Interface (v2.1)

This section explains operation of the LIN Interface (v2.1).

The LIN communication function has two modes; manual mode for LIN header transmission/reception using the interrupt functions, and assist mode for automatic LIN header transmission/reception. Assist mode supports the master node of LIN protocol Revision1.X and LIN protocol Revision2.X, and the slave node of LIN protocol Revision1.X. To support the slave node of LIN protocol Revision2.X, use manual mode.

7.1 Interrupt of LIN Interface (v2.1) manual mode

7.2 Interrupts in LIN Interface (v2.1) Assist Mode

7.3 Operation of Serial Timer

7.4 Test Mode

7.5 Operation of LIN Interface (v2.1)

7.6 Setup Procedure and Program Flow

7.1. Interrupt of LIN Interface (v2.1) manual mode

Interrupt of LIN Interface (v2.1) manual mode is shown.

The LIN interface (v2.1) consists of reception interrupt and transmission interrupt. This interface can generate interrupt requests for the following factors:

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- LIN Break field detection

7.1.1. List of Interrupts of LIN Interface (v2.1) (manual mode)

This section explains the list of interrupts of the LIN interface (v2.1) (manual mode).

Table 7-1 shows the relationship between the LIN interrupt control bits and the interrupt factors in the manual mode.

Table 7-1 Interrupt Control Bits and Interrupt Factors for LIN Interface (v2.1) (Manual Mode)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request
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		

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request
Trans-mission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*1
	TBI	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), write "1" to the Lin break field setting bit(LBR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has valid data (retransmission)*1
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full
Status	LBD	SSR	Lin break field detection	ESCR:LBIE	Writing "0" to the SSR:LBD
	SFD	SACSR	Sync Field is detected	SACSR: SFDE	Writing "0" to Sync Field detection flag (SACSR:SFD)
	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to timer interrupt flag bit (SACSR:TINT)
Input capture	ICP	ICS	1st falling edge of Lin Synch Field	ICS:ICE0	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Synch Field		

*1: Wait for TDRE bit with "0" before setting "1" to TIE bit.

7.1.2. Reception Interrupts and Flag Setting Timing

This section explains the generation of reception interrupts and the flag setting timing.

Reception interrupts occur when the reception is completed (SSR:RDRF), when a reception error occurs (SSR:ORE, FRE), or when LIN break Field is detected.

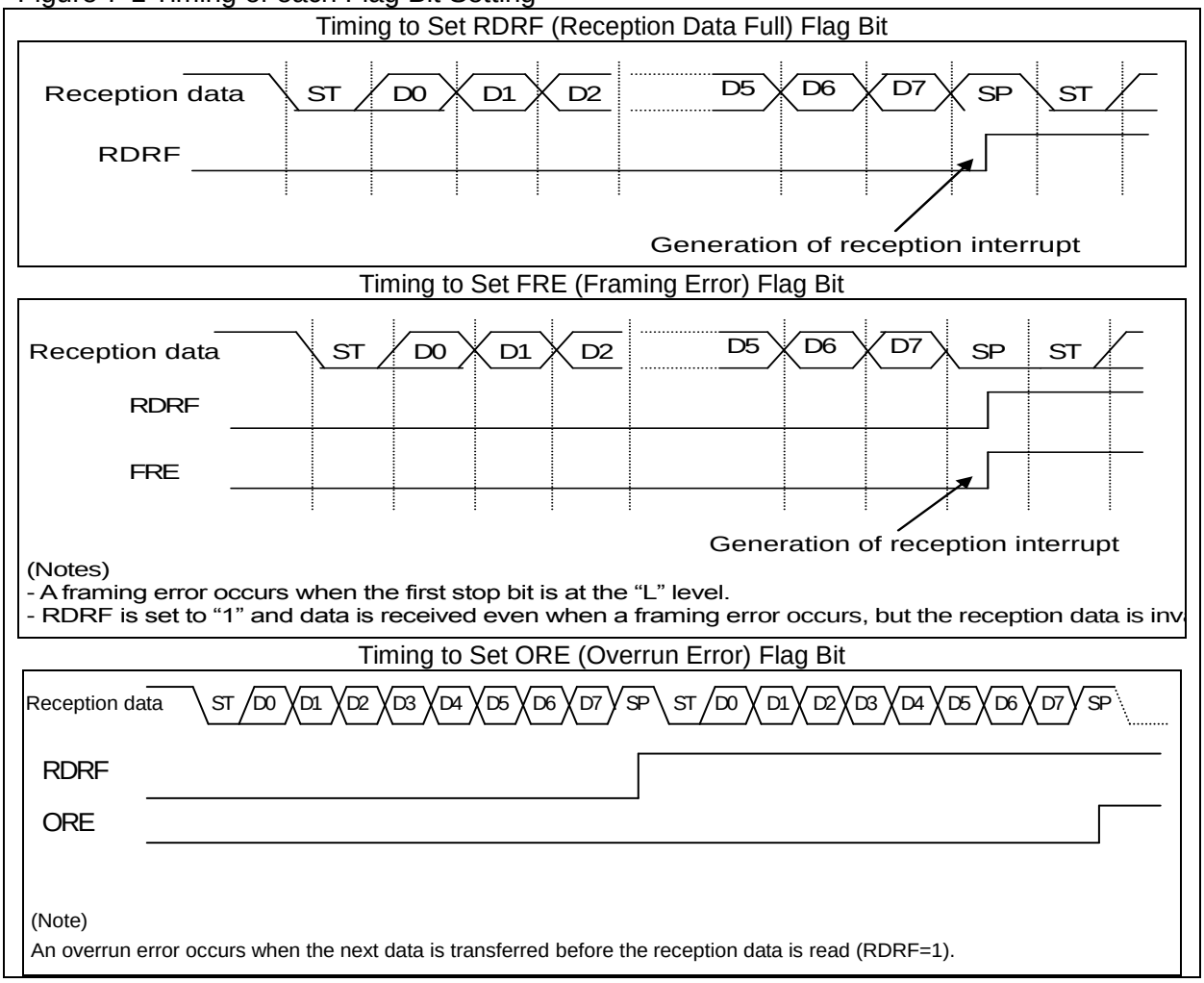
■ Reception Interrupts and Flag Setting Timing

When the first stop bit is detected, reception data is stored in the Receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:ORE, FRE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt will occur.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 7-1 Timing of each Flag Bit Setting



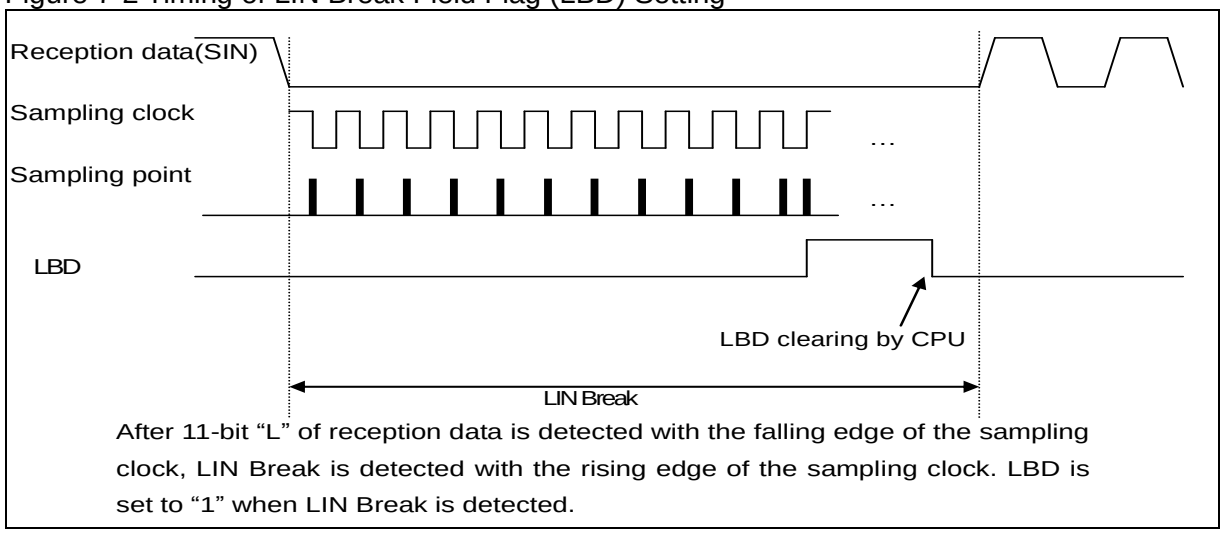
Note:

If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the data may not be received with the edge disabled. It is recommended to leave a space between frames if successive frames are to be output.

■ Timing of LIN break Field Detection Flag (LBD) Setting

LBD bit is set to "1" when the serial input (SIN) is "0" for more than 11-bit width. In this case when LIN break Field interrupt is enabled (ESCR:LBIE=1), a reception interrupt occurs.

Figure 7-2 Timing of LIN Break Field Flag (LBD) Setting



7.1.3. Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using reception FIFO and the flag setting timing.

When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received.

■ Reception Interrupts when Using Reception FIFO and Flag Setting Timing

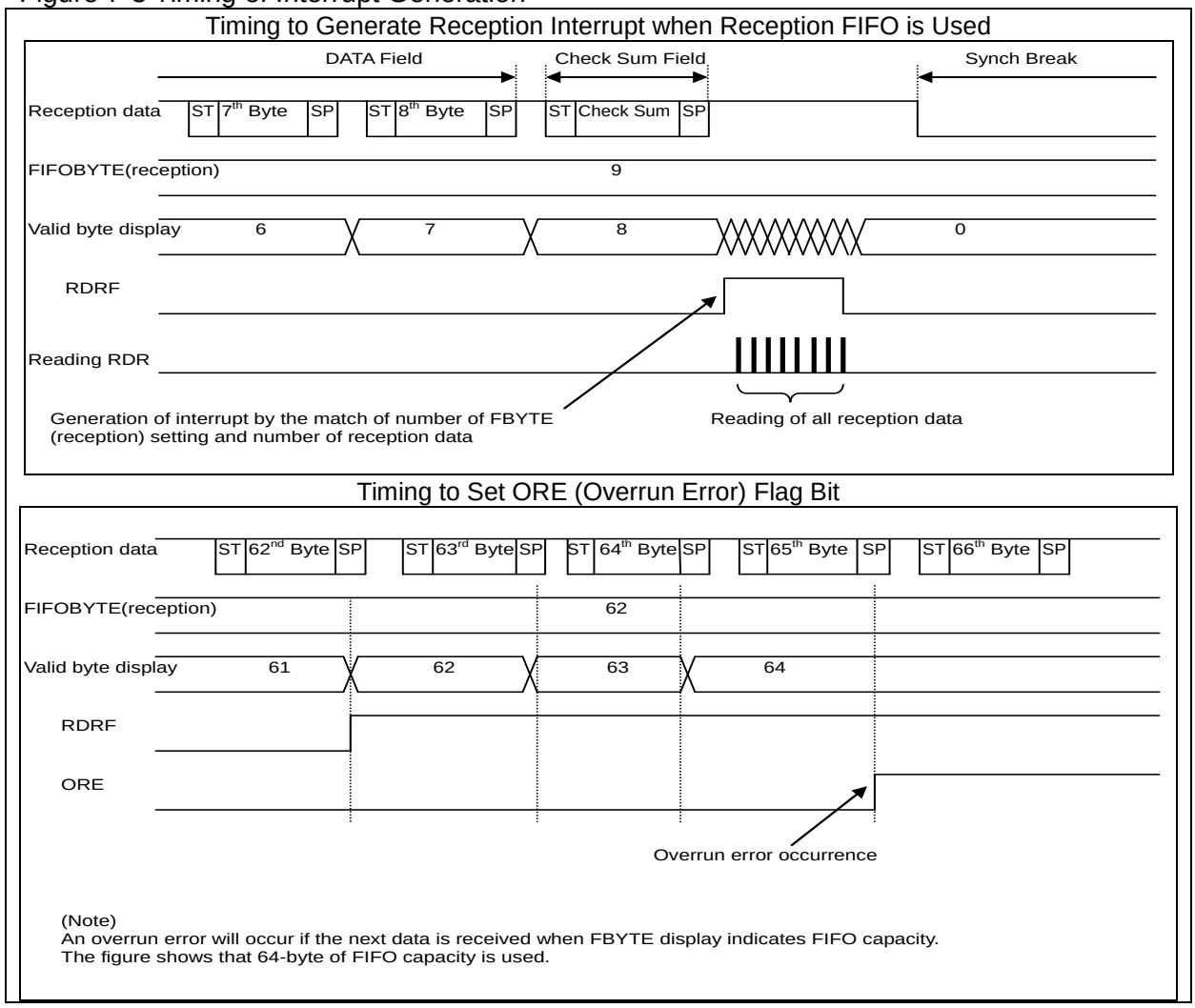
The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt will be generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - Reception FIFO idle detection enable bit (FCR:FRIIE) is "1"
  - Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) will be cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1).

Figure 7-3 Timing of Interrupt Generation



7.1.4. Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and the flag setting timing.

Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

■ Transmission Interrupts and Flag Setting Timing

● Timing of transmission data empty flag (TDRE) setting

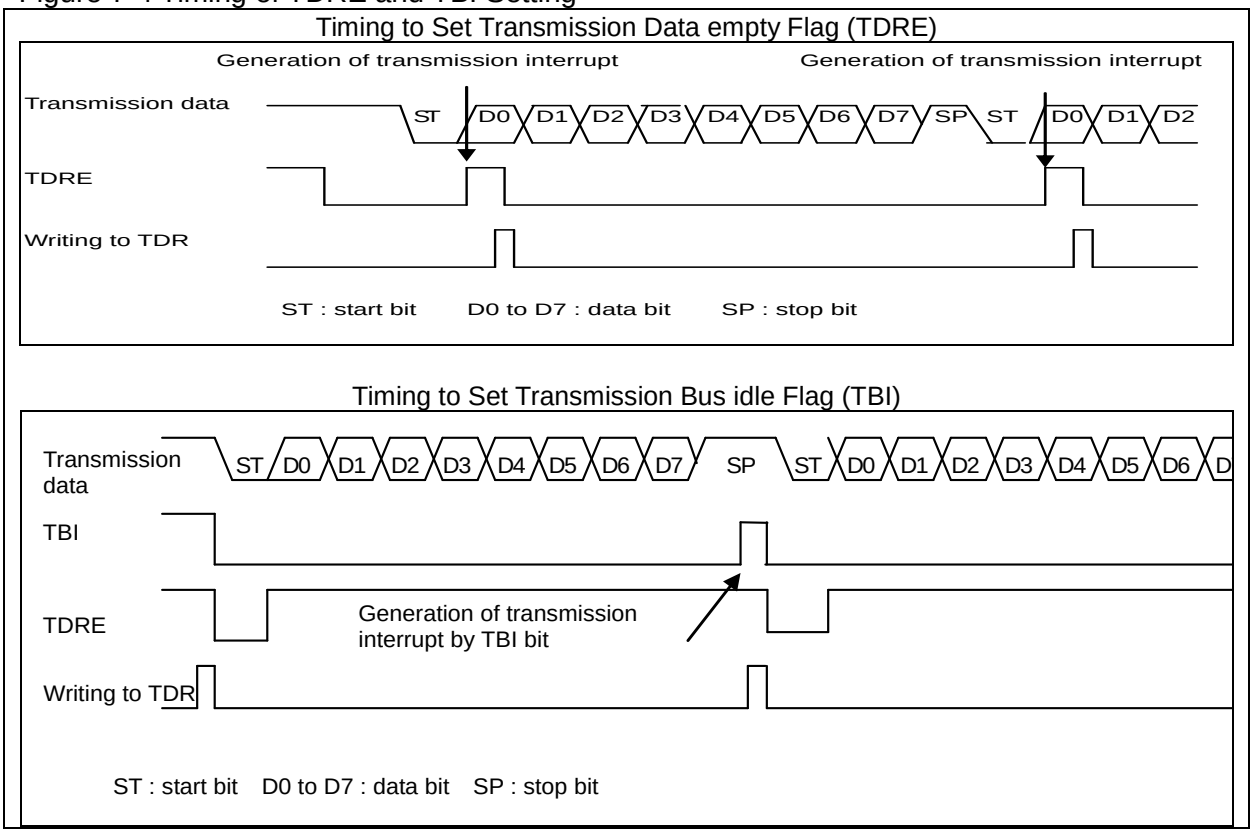
When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt

occurs. The TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

● Timing of transmission bus idle flag (TBI) setting

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the TBI bit and the transmission interrupt request will be cleared.

Figure 7-4 Timing of TDRE and TBI Setting



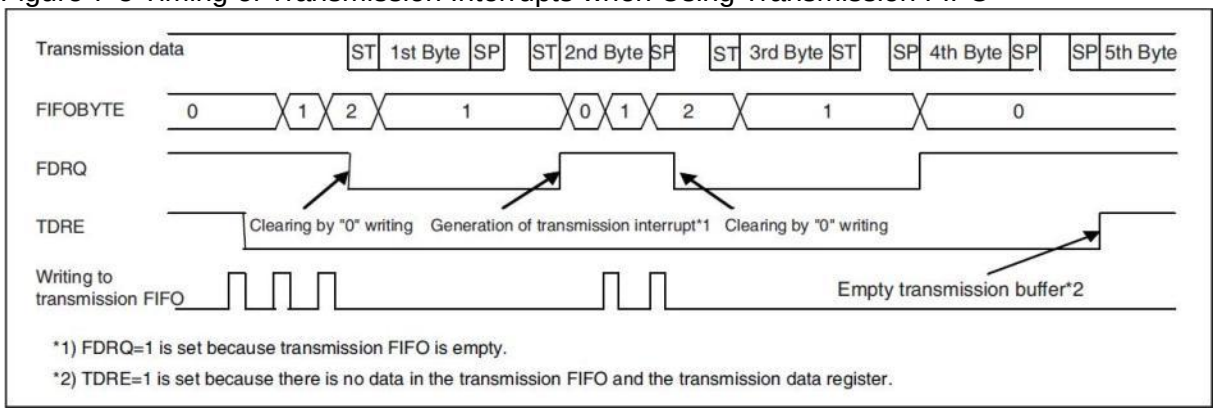
7.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and the flag setting timing.

When the transmission FIFO is used, an interrupt will be generated when the number of data items stored in the transmission FIFO is FTICR register (FTICR) setting value or less.

- When the number of data items stored in the transmission FIFO is FTICR register (FTICR) setting value or less. the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE=1) at this time, a transmission interrupt occurs.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  When FBYTE=0x00, there is no data in the transmission FIFO.

Figure 7-5 Timing of Transmission Interrupts when Using Transmission FIFO



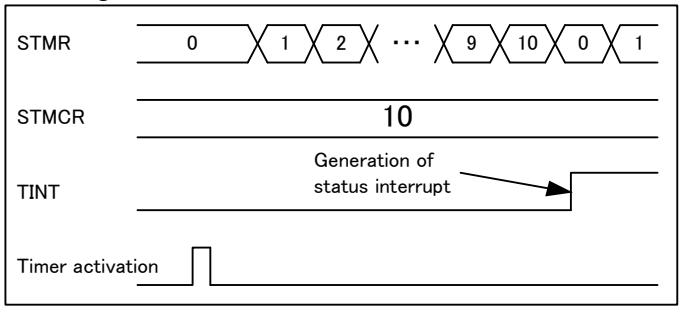
7.1.6. Timer Interrupt and Flag Setting Timing

This section explains the timer interrupt and the flag setting timing.

Timer interrupt occurs when the serial timer register (STMR) and the serial timer comparison register (STMCR) are matched.

- When the serial timer register (STMR) and the serial timer comparison register are matched, the timer interrupt flag (SACSR:TINT) is set to "1".
  If the timer interrupt is enabled (SACSR:TINTE="1") at this time, a status interrupt occurs.

Figure 7-6 Timer Interrupt Timing



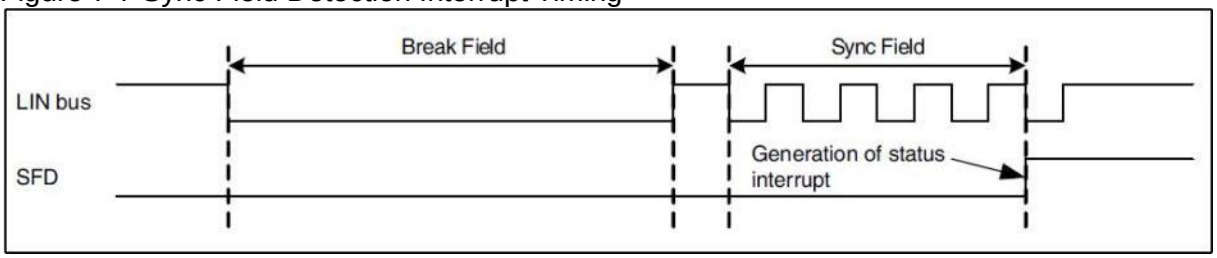
7.1.7. Sync Field Detection Interrupts and Flag Setting Timing

This section explains the generation of Sync Field detection interrupts and the flag setting timing.

Sync Field detection interrupt occurs when the detection of Sync Field is completed.

- When the auto baud rate adjustment is enabled (SACSR:AUTE="1") and the fifth falling of LIN bus on Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1".
  If the Sync Field interrupt is enabled (SACSR:SFDE="1") at this time, a status interrupt occurs.

Figure 7-7 Sync Field Detection Interrupt Timing



7.2. Interrupts in LIN Interface (v2.1) Assist Mode

Interrupts in LIN Interface (v2.1) Assist Mode is shown.

The LIN interface (v2.1) consists of reception interrupt, transmission interrupt, and status interrupt.
In assist mode, this interface can generate interrupt requests for the following factors:

- When reception data is stored in the receive data register (RDR) or when a reception error occurs.
- When transmission data is transmitted from the transmit data register (TDR) to the transmit shift register, and the transmission is begun.
- Transmission bus idle (There is no transmission operation)
- Transmission FIFO data request
- LIN Break Field detection
- LIN Sync Field detection

- Both the comparison value (STMCR) of the serial timer and the value (STMR) of the serial timer are equal.
- The LIN automatic header completion or the checksum arithmetic operations completion is detected.

7.2.1. List of Interrupts of LIN Interface (v2.1) (assist mode)

This section explains the list of interrupts of LIN interface (v2.1) (assist mode).

Table 7-2 shows the relationship between the LIN interrupt control bits and the interrupt factors in the LIN assist mode.

Table 7-2 Interrupt Control Bits and Interrupt Factors for LIN Interface (v2.1) (Assist Mode)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	How to clear the interrupt request
Recep-tion	RDRF	SSR	1-byte reception	SCR:RIE	Reading receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading receive data (RDR) until the reception FIFO becomes empty
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	Framing error		
	LBSER	LAMSER	LIN bus error detection	LAMIER: LASERIE	Writing "0" to the LAMESR:LBSER
	LPFER	LAMESR	LIN Sync Data error detection	LAMIER: LSFERIE	Writing "0" to the LAMESR:LSFER
	LPTER	LAMESR	LIN ID parity error detection	LAMIER: LPTERIE	Writing "0" to the LAMESR:LPTER
	LCSER	LAMESR	LIN checksum error detection	LAMIER: LCSERIE	Writing "0" to the LAMESR:LCSER

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	How to clear the interrupt request
Trans-mission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing the transmit data (TDR), writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*[1]
	TBI	SSR	No transmission operation	SCR:TBIE	Writing the transmit data (TDR), writing of "1" to the Lin break field set bit(LBR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has valid data (retransmission)*[1]
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or the value is empty.	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full
Status(Assist Mode)	LBD	SSR	Lin break field detection	ESCR:LBIE	Writing "0" to the SSR:LBD
	SFD	SACSR	Sync Field is detected	SACSR: SFDE	Writing "0" to Sync Field detection flag (SACSR:SFD)
	TINIT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to timer interrupt flag bit (SACSR:TINT)
	LAHC	LAMSR	Automatic header completion	LAMIER: LAHCIE	Writing "0" to the LAMSR:LAHC
	LCSC	LAMSR	Checksum arithmetic operation completion	LAMIER: LCSCIE	Writing "0" to the LAMSR:CRC
Input capture	ICP	ICS	1st falling edge of Lin Synch Field	ICS:ICE0	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Synch Field		

*1: Set the TIE bit to "1" after the TDRE bit becomes "0".

7.2.2. Reception Interrupts and Flag Setting Timing in Assist Mode

This section explains the generation of reception interrupts and the flag setting timing in assist mode.

Reception interrupts occur when the reception is completed (SSR:RDRF) and when a reception error occurs (SSR:ORE, FRE,LAMESR:LBSER,LSFER,LPTER,LCSER).

■ Reception Interrupts and Flag Setting Timing

In the LIN assist mode (LAMCR:LAMEN=1), data is stored in the receive data register (RDR) when the first stop bit of each of the following fields is detected. When the reception of data is completed (SSR:RDRF=1), the flag is set. If reception interrupt is enabled (SSR:RIE=1), a reception interrupt occurs.

- ID Field when slave mode is set (SCR:MS=1) and the data reception register (RDR) is set to receive the ID Field (LAMCR:LIDEN=0)
- Data Field for response

As for the timing of setting the reception data full flag bit (SSR:RDRF), the timing is similar to those described in "7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 7-1.

Notes:

- When the reception error occurs, the reception data stored in the receive data register (RDR) becomes invalid.
- When the LIN assist mode reception ID register is used in order to receive the ID Field (LAMCR:LIDEN=1), the received ID value is not stored in the receive data register (RDR) , and the reception data full flag bit (SSR:RDRF) is not set.
- Neither the Sync Field nor the checksum are stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.
- Those data transmitted from each field are not stored in the receive data register (RDR), and reception data full flag bit (SSR:RDRF) is not set.

■ Framing Error Interrupt and Flag Setting Timing

In assist mode (LAMCR:LAMEN=1), the framing error is detected and a flag of the framing error is set (SSR:FRE=1) when "L" level is detected in the stop bit of the Sync Field, the ID Field, the Data Field, and the Check Sum Field, respectively. If the reception interrupt is enabled (SSR:RIE=1), the reception interrupt will occur. Moreover, when the framing error is detected, transmission/reception of both the header and the response is stopped in the assist mode.

While the framing error flag is being set (SSR:FRE=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1=0 or FCR0:FE2=0).

As for the timing of setting the framing error flag bit (SSR:FRE), the timing is similar to those described in "7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 7-1.

■ Overrun Error Interrupt and Flag Set Timing

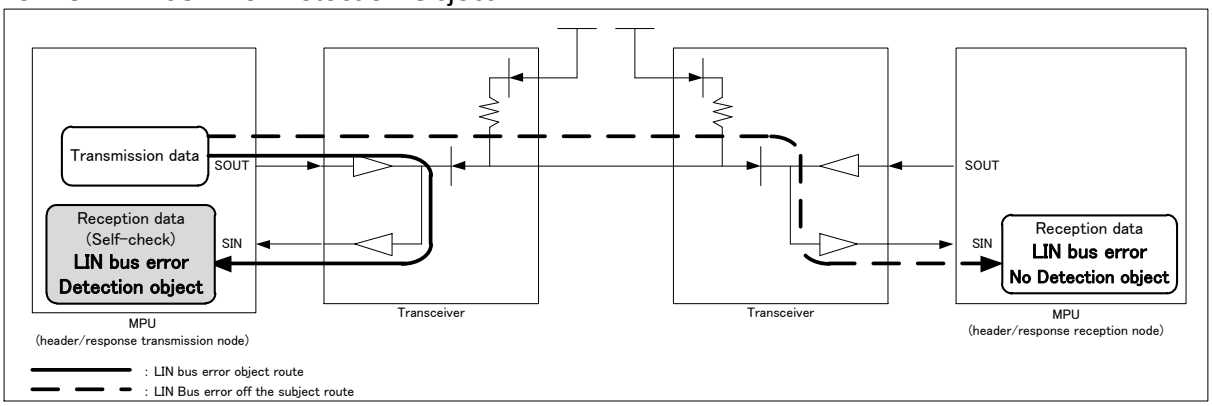
When reception of data is detected before the previously received data is read (RDRF=1), the overrun error is detected. After the reception of the next data is completed (SSR:RDRF=1), the overrun error flag is set (SSR:ORE=1). If the reception interrupt is enabled (SSR:RIE=1), the reception interrupt occurs.
Moreover, when the overrun error is detected, transmission/reception of both the header and the response is stopped in the assist mode.

While the overrun error flag is being set (SSR:ORE=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1=0 or FCR0:FE2=0).
As for the timing of setting the overrun error flag bit (SSR:ORE), the timing is similar to those described in "7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 7-1.

■ LIN Bus Error Detection Interrupt and Flag Setting Timing

The LIN bus error is detected by the self-check done on the side where a header/response is transmitted in assist mode (LAMCR:LAMEN=1). The LIN bus error cannot be detected on the side where header/response is received.
Figure 7-8 shows the LIN bus error detection object.

Figure 7-8 LIN Bus Error Detection Object



The range of detection of the LIN bus error is a start bit and byte data of LIN Break and Sync Field/ID Field/Data Field/Check Sum Field. The stop bit is outside the detection range of the LIN bus error. When the stop bit is detected as "L" level, the framing error is detected (SSR:FRE=1).
Moreover, when the LIN bus error is detected, transmission of the header and the response is stopped in the assist mode.
Even if the LIN bus error occurs when the ID Field transmission is completed, LIN automatic header completion flag (LAMSR:LAHC=1) is set.

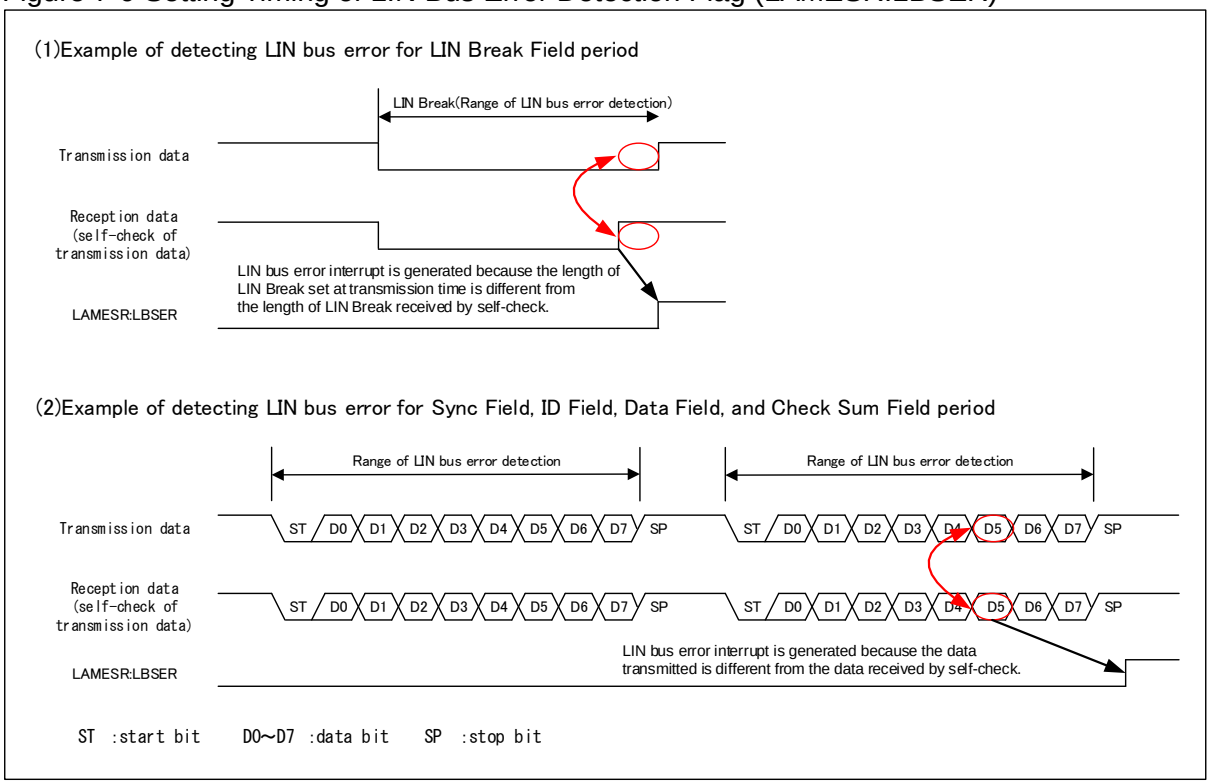
● LIN bus error detection interrupt and flag setting timing on master side

On master side (SCR:MS=0), the LIN bus error is detected when a header/response is transmitted.
The LIN bus error is detected and the flag is set (LAMESR:LBSER=1) if an error is detected as a result of the comparison between transmission LIN Break length and reception LIN Break length or between the transmission data and the reception data. When this interrupt is set to be enabled (LAMIER:LBSERIE=1), the reception interrupt will occur.

● LIN bus error detection interrupt and flag setting timing on slave side

On slave side (SCR:MS=1), the LIN bus error is detected when the response is transmitted.
The LIN bus error is detected and the flag is set (LAMESR:LBSER=1) if an error is detected as a result of the comparison between the transmission data and the reception data. When the interrupt is set to be enabled (LAMIER:LBSERIE=1), the reception interrupt will occur.

Figure 7-9 Setting Timing of LIN Bus Error Detection Flag (LAMESR:LBSER)



■ LIN Sync Data Error Detection Interrupt and Flag Setting Timing

The LIN Sync Data error is detected when automatic baud rate adjustment is disabled (SACSR:AUTE=0), in slave node (SCR:MS=1) which is set to assist mode (LAMCR:LAMEN=1).

The range of detection of the LIN Sync Data error is a start bit and byte data of the Sync Field. The stop bit is off the target of the LIN Sync Data error detection. When the stop bit is detected as "L" level, the framing error is detected (SSR:FRE=1).

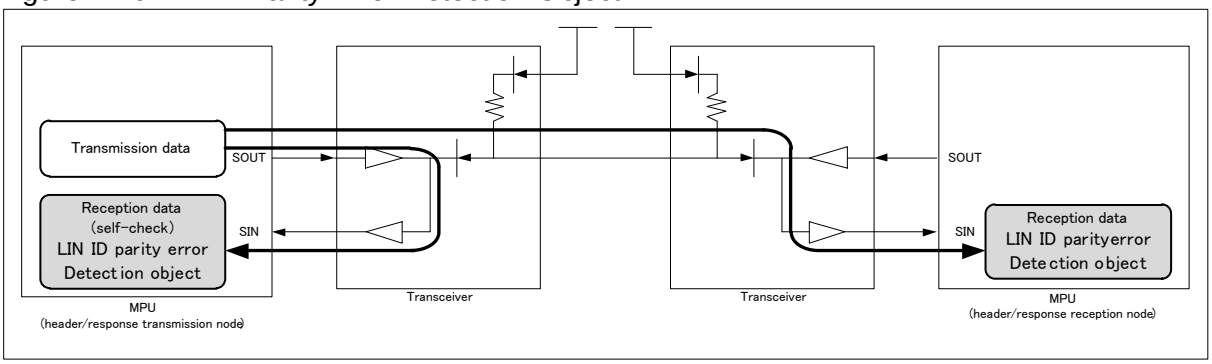
● LIN Sync Data error detection interrupt and flag setting timing when automatic baud rate self adjustment is prohibited

On slave node (SCR:MS=1) which is set to automatic baud rate adjustment disabled (SACSR:AUTE=0), the LIN Sync Data error is detected and the flag is set (LAMESR:LSFER=1) if the value other than 0x55 is detected as a result of checking the data value of Sync Field. If this interrupt is set to be enabled (LAMIER:LSFERIE=1) at that time, the reception interrupt will occur.

■ LIN ID Parity Error Detection Flag Interrupt and Flag Setting Timing

The LIN ID parity error detection in the assist mode (LAMCR:LAMEN=1) is done by the master and slave. Here, the master transmits the ID Field and does a self-check, and the slave receives ID Field.
Figure 7-10 shows the LIN ID parity error detection object.

Figure 7-10 LIN ID Parity Error Detection Object



The range of detection of the LIN ID parity error is ID data and byte data of a parity. The start bit and the stop bit are outside the LIN ID parity error detection ranges. When "L" level is detected on the stop bit, the error is the framing error (SSR:FRE=1).

In the LIN assist mode (LAMCR:LAMEN=1), when the LIN ID parity error occurs during the transmission of an automatic header, the automatic header transmission completion flag is set (LAMSR:LAHC=1).
When the LIN ID parity error occurs while receiving an automatic header, the automatic header (reception) completion flag is set (LAMSR:LAHC=1) as well.

When the LIN ID parity error is detected, transmission/reception of the response stops in the assist mode.
While the LIN ID parity error flag is set (LAMESR:LPTER=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1 or FCR0:FE2=0).

Note:

Even though the framing error is detected in the ID Field, the result of ID parity arithmetic operation is indicated. The result at this time, however, is not guaranteed.

● LIN ID parity error detection interrupt and flag setting timing on master side

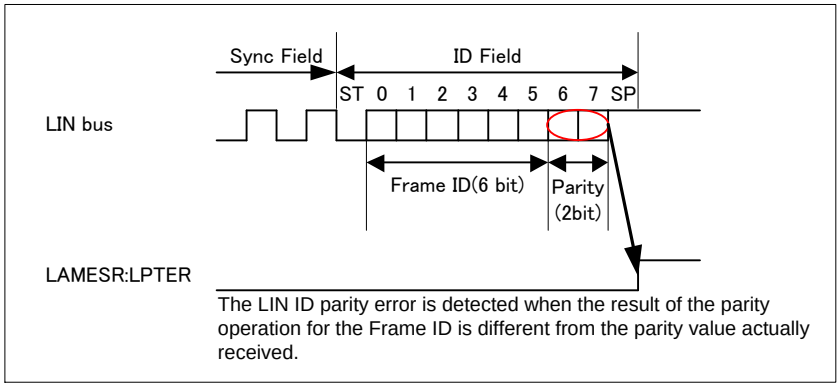
On master side (SCR:MS=0) set in the assist mode (LAMCR:LAMEN=1), detection of the LIN ID parity error is performed when the ID Field is transmitted. The master does the parity arithmetic operation for the six-bit Frame ID which is set by the transmit data register (TDR) or the LIN assist mode transmission ID register (LAMTID). Then the ID Field automatically generated is transmitted.

When the master receives the ID Field by self-check, and if there is a difference of parity between the result of the arithmetic operation for Frame ID and received value, the LIN ID parity error is detected and the flag is set (LAMESR:LPTER=1).
At that time, the interrupt occurs if the interrupt is set to be enabled (LAMIER:LPTERIE=1).

● LIN ID parity error detection interrupt and flag setting timing on slave side

On slave side (SCR:MS=1) set in the assist mode (LAMCR:LAMEN=1), detection of the LIN ID parity error is performed when the ID Field is received. If the result of parity arithmetic operation for the Frame ID value in the received ID Field is different from the received parity value, the LIN ID parity error is detected and the flag is set (LAMESR:LPTER=1).
At that time, the reception interrupt occurs if the interrupt is set to be enabled (LAMIER:LPTERIE=1).

Figure 7-11 Setting Timing of LIN ID Parity Error Detection Flag (LAMESR:LPTER)

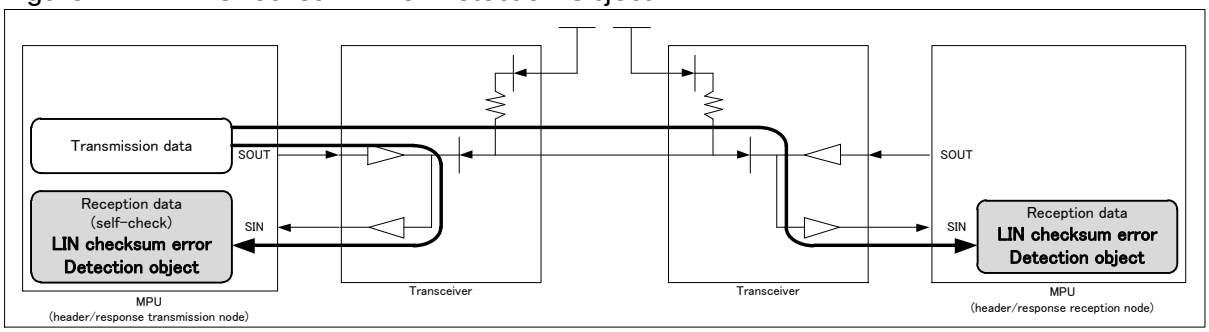


■ LIN Checksum Error Detection Flag Interrupt and Flag setting Timing

The LIN checksum error detection is done on both the side where the checksum is self-checked and transmitted in the assist mode (LAMCR:LAMEN=1) and the side where the checksum is received

Figure 7-12 shows the LIN checksum error detection object.

Figure 7-12 LIN Checksum Error Detection Object



The method of arithmetically operating the checksum transmitted automatically can be selected from standard (The object: data)/extended (The object: ID Field + data).

Even if the LIN checksum error occurs when the checksum transmission is completed, the transmission of data is not stopped.

● LIN checksum error detection interrupt and flag setting timing on the setting for calculating standard checksum

When the standard checksum arithmetic operation is set (LAMCR:LCSTYP=0), on the side who sends the response (data and checksum), the checksum of the transmission data for set LIN data length (LAMCR:LDL) is calculated and transmitted automatically after final data is transmitted.

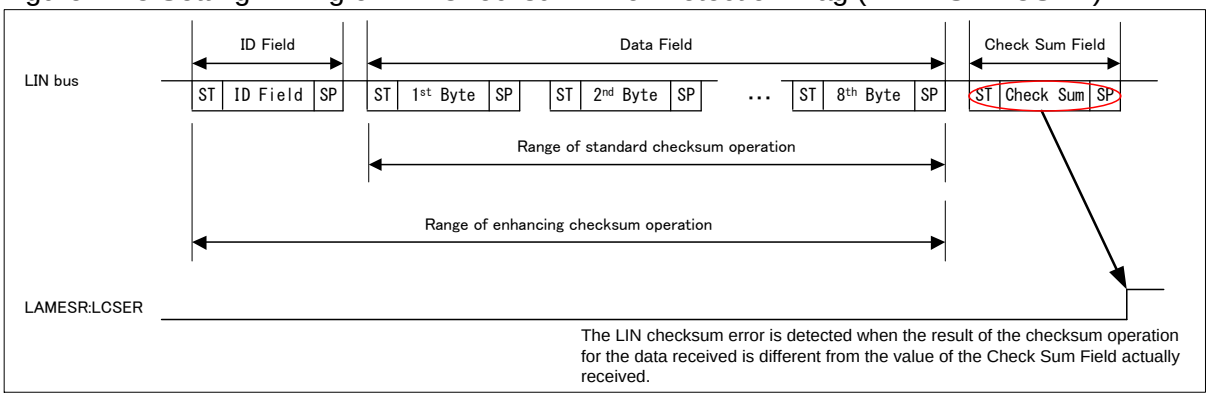
On another side where the response (data and checksum) is received, the checksum of the received data for set LIN data length (LAMCR:LDL) is calculated. If the received checksum and the calculated value are different from each other, the LIN checksum error is detected and the flag is set (LAMESR;LCSER=1).

At that time, the reception interrupt occurs if the interrupt is set to be enabled (LAMIER:LCSCIE=1).

- **LIN checksum error detection interrupt and flag setting timing on the setting for calculating extended checksum**

  When the extended checksum arithmetic operation is set (LAMCR:LCSTYP=1), on the side who sends the response (data and checksum), the checksum of the ID Field and transmitted data for set LIN data length (LAMCR:LDL) is calculated and transmitted automatically after final data is transmitted.
  On another side where the response (data and checksum) is received, the checksum of the received data for set LIN data length (LAMCR:LDL) is calculated. If the received checksum and the calculated value are different from each other, the LIN checksum error is detected and the flag is set (LAMESR:LCSER=1).
  At that time, the reception interrupt occurs if the reception interrupt is set to be enabled (LAMIER:LCSCIE=1).

Figure 7-13 Setting Timing of LIN Checksum Error Detection Flag (LAMESR:LCSER)



Note:

When the error (LIN bus error, LIN ID error, LIN Sync Data error, and framing error) is detected in the data of the header and the response part, processing in the assist mode stops and the checksum operation is not executed, regardless of standard/extended checksum arithmetic operation.

7.2.3. Reception Interrupts and Flag Setting Timing when using Reception FIFO

This section explains the generation of interrupts and the flag setting timing when using reception FIFO.

The description is similar to those in "7.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing" in the manual mode.

7.2.4. Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

A transmission interrupt occurs when the transmission data is transmitted from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) and then the transmission is started, and when the transmission operation is not done.

■ Transmission Interrupts and Flag Setting Timing

● Setting timing of transmission data empty flag (TDRE)

The timing is similar to those described in "7.1.4 Transmission Interrupt and Flag Setting Timing" in the manual mode.

● Setting timing of transmission bus idle flag (TBI)

When either of the following transmission operation is not done, the transmission bus idle flag bit (SSR:TBI) is set to "1". At this time, if transmission bus idle interrupt is enabled (SCR:TBIE=1), the transmission interrupt occurs.

- The empty flag of the transmission data is set (TDRE=1) and transmission processing is not done.
- In master operation (SCR:MS=0) in assist mode (LAMCR:LAMEN=1), the header transmission processing is not done.
- In assist mode (LAMCR:LAMEN=1), the response transmission processing is not done.

Moreover, the transmission bus idle flag bit (SSR:TBI) and the transmission interrupt request are cleared by the following factors:

- Transmission data is written in the transmit data register (TDR).
- In master operation (SCR:MS=0) in assist mode (LAMCR:LAMEN=1), the header transmission is being processed (LIN Break Field, Sync Field, ID Field).
- In assist mode (LAMCR:LAMEN=1), the response transmission is being processed (data and checksum).

7.2.5. Interrupts and Flag Setting Timing when using Transmission FIFO

This section explains the generation of interrupts and flag setting timing when using transmission FIFO.

The description is similar to those in "7.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing".

7.2.6. Timer Interrupts and Flag Setting Timing

This section explains the generation of timer interrupts and the flag setting timing.

The description is similar to those in "7.1.6. Timer Interrupt and Flag Setting Timing" in the manual mode.

7.2.7. Status Interrupts and Flag Setting Timing in Assist Mode

This section explains the generation of status interrupts and the flag setting timing in assist mode.

The status interrupt in the assist mode occurs when LIN Break Field is detected (SSR:LBD), when the sink field is detected (SACSR:SFD), when an automatic header is completed (LAMSR:LAHC), and when the checksum arithmetic operations is completed (LAMSR:LCSC)

■ Setting Timing of LIN Break Field Detection Flag

When serial input (SIN) of "0" is inputted into the width of eleven bits or more, the LBD bit is set to "1". At this time, when the LIN Break Field interrupt is set enabled (ESCR:LBIE=1), the status interrupt occurs.

Notes:

- When the LIN Break Field is received, if the reception is enabled (SCR:RXE=1), a framing error is detected before the LIN Break Field is detected. However, it operates normally without stopping the header reception.
- In assist mode (LAMCR:LAMEN=1), if a new LIN Break is consecutively transmitted by the master during the time from the detection of the LIN Break Field until the completion of the ID Field reception, the framing error is detected at "L" level of the tenth bit of new LIN Break Field regardless of reception prohibition setting (SCR:RXE=0). However, it operates normally without stopping the header reception.

■ Sink Field Detection Interrupt and Flag Setting Timing

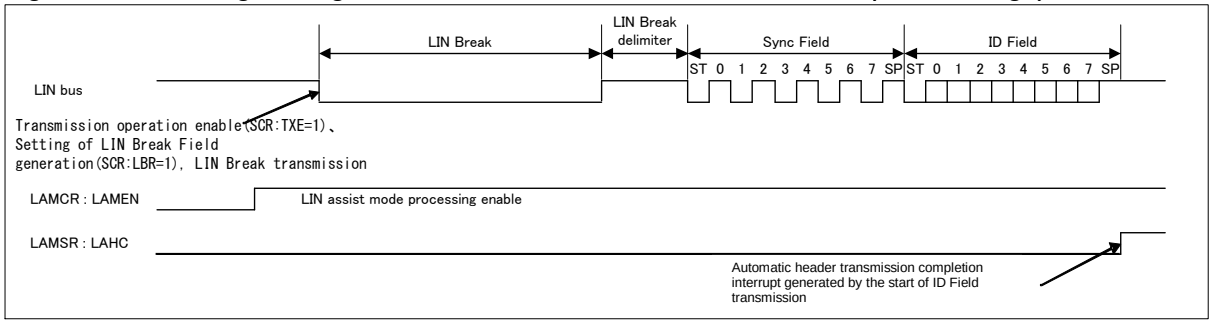
The description is similar to those in "7.1.7 Sink Field Detection Interrupt and Flag Setting Timing" in the manual mode.

■ Automatic Header Completion Interrupt and Flag Setting Timing under Transmission

On master side set in the LIN assist mode (LAMCR:LAMEN=1),when the header transmission from LIN Break to ID

Field is completed, the flag is set (LAMSR:LAHC=1). If the interrupt is set enabled (LAMIER:LAHCIE =1), the status interrupt occurs.
Even when LIN bus error/LIN ID parity error/framing error occurs in the LIN assist mode for the ID Field, period the automatic header completion flag is set (LAMSR:LAHC=1). However, reception/transmission of the response stops.

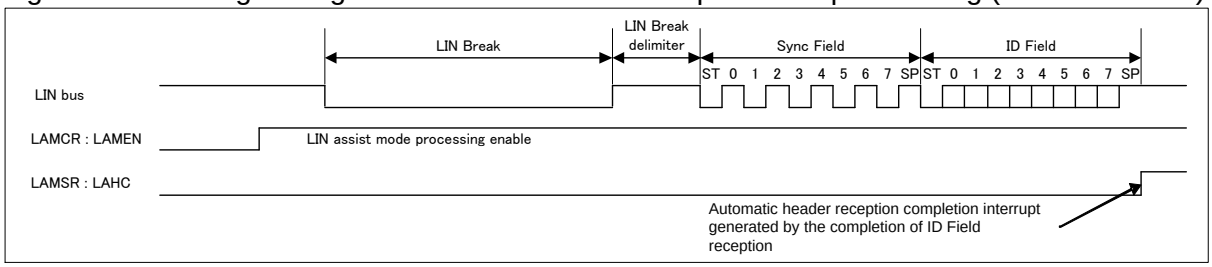
Figure 7-14 Setting Timing of Automatic Header Transmission Completion Flag (LAMSR:LAHC)



■ Automatic Header Completion Interrupt under Reception

On slave side set in the LIN assist mode (LAMCR:LAMEN=1),when the header reception from LIN Break to ID Field is completed, the flag is set (LAMSR:LAHC=1). If the interrupt is set enabled (LAMIER:LAHCIE =1), the status interrupt occurs.
Even when LIN bus error/LIN ID parity error/framing error occurs in the LIN assist mode for the ID Field, the automatic header completion flag is set (LAMSR:LAHC=1). However, reception/transmission processing of the response stops.

Figure 7-15 Setting Timing of Automatic Header Reception Completion Flag (LAMSR: LAHC)



■ LIN Checksum Detection Completion Flag Interrupt and Flag Setting Timing

In the assist mode (LAMCR:LAMEN=1), the LIN checksum detection is done on both the side where the checksum is self-checked and transmitted and the side where the checksum is received When the data of the setting length (LAMCR:LDL3-0) and the checksum are received, the checksum operation is completed and the flag is set (LAMSR:LCSC=1). When the interrupt is set enabled (LAMIER:LCSCIE=1), the status interrupt occurs.
When reception of the checksum is completed, neither the reception checksum value is stored in the RDR register nor the (SSR:RDRF) is set to "1". When FIFO is used, the received checksum value is not stored in the reception FIFO

Note:

The result of the checksum operation at this time is not guaranteed when the framing error is detected in the final data for the setting length (LAMCR:LDL3-0) or when the framing error is detected by the checksum.

7.3. Operation of Serial Timer

Operation of serial timer is shown.

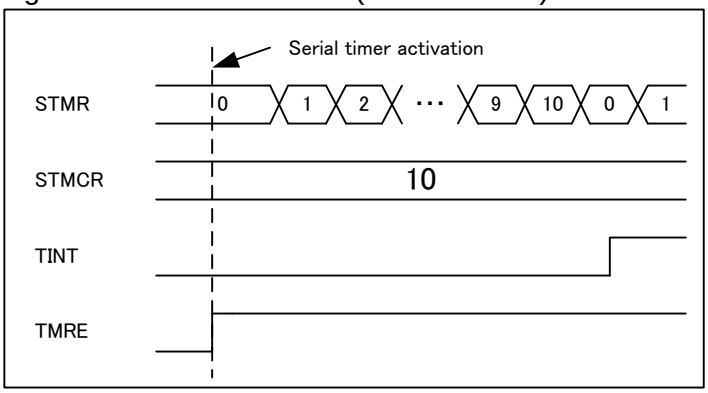
The serial timer can be used for the timer function.

● How to Start Serial Timer

There are two ways to start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE) and starting by the Sync Field.

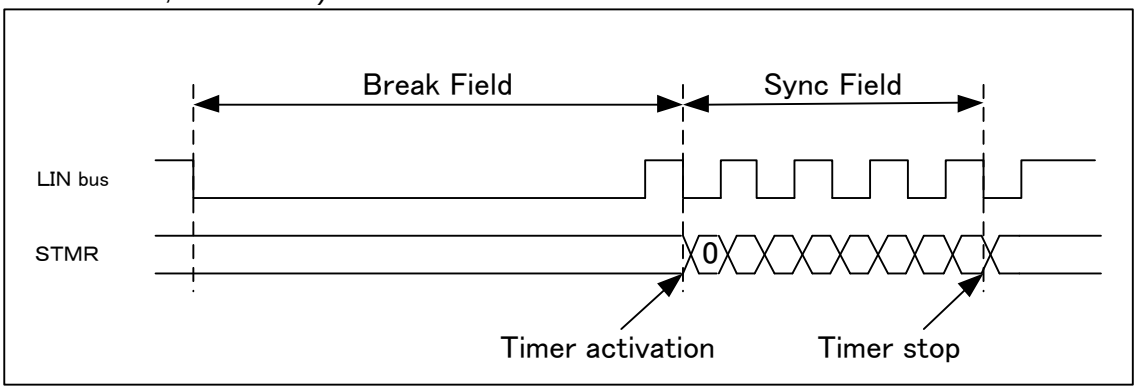
- Start by using the serial timer enable bit (SACSR:TMRE)
  When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

Figure 7-16 Start by Using Serial Timer Enable Bit (STMCR="10")



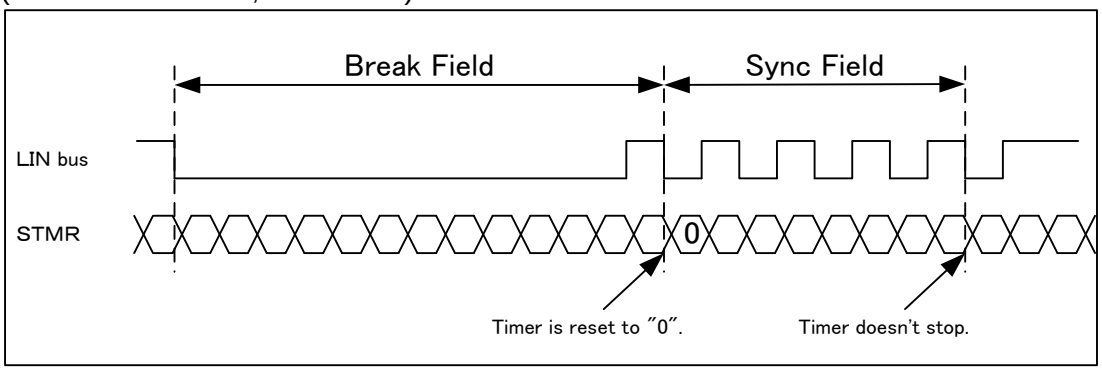
- Start by Sync Field reception
  When the serial timer is stopped and the auto baud rate adjustment bit (SACSR:AUTE) is "1", the serial timer starts and the serial timer register (STMR) starts counting from 0 if LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 7-17 Start by Sync Field Reception During Serial Timer Stopped (SACSR:AUTE="1", TMRE="0")



When the serial timer is operating and the auto baud rate adjustment bit (SACSR:AUTE) is "1", the serial timer register (STMR) starts counting from 0 if LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 7-18 Start by Sync Field Reception During Serial Timer Operation (SACSR:AUTE="1", TMRE="1")



How to Stop Serial Timer

The serial timer will stop under the following conditions.

- When the auto baud rate adjustment bit (AUTE) is "0", the serial timer will stop with resetting the serial timer enable bit (SACSR:TMRE) to "0". The value of the serial timer register (STMR) is retained.
- When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enable bit (SACSR:TMRE) is "1", the serial timer will stop with resetting the serial timer enable bit (SACSR:TMRE) to "0" not during Sync Field reception. The value of the serial timer register (STMR) is retained.
- When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enabled bit (SACSR:TMRE) is "0", the serial timer is stopped and the value of the serial timer register (STMR) is maintained if LIN interface (v2.1) detects the fifth falling edge of the Sync Field.

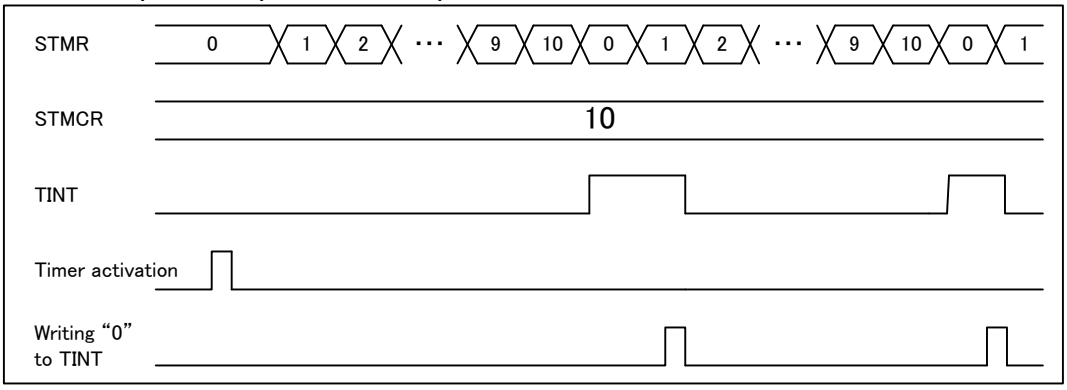
Note:

When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enabled bit (SACSR:TMRE) is "1", the serial timer is not stopped and continues the operation even if LIN interface (v2.1) detects the fifth falling edge of the Sync Field.

Timer Operation

If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 7-19 Timer Operation (STMCR="10")



Notes:

- When the timer comparison register (STMCR) is set to $(0000)_H$, the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".
- If the auto baud rate adjustment bit (SACSR:AUTE) is set to "1", the serial timer register (STMR) is reset to 0 when the Sync Field is received.

7.4. Test Mode

Test mode is shown.

This section explains the operation of the test mode.

7.4.1. Manual Mode

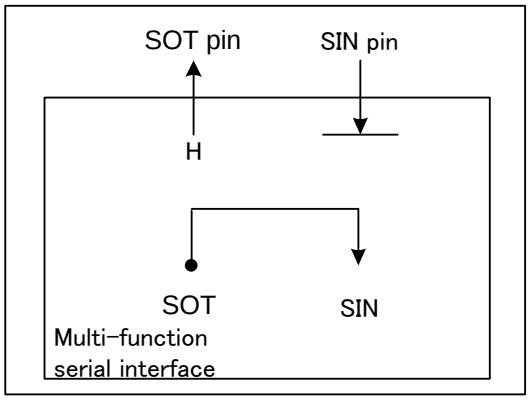
This section explains the manual mode.

■ Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.

When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 7-20 Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

7.4.2. Assist Mode

This section explains the assist mode.

■ Serial test mode

It is similar to the serial test mode in the manual mode.
However, only master node (SCR:MS=0) can execute the serial test in LIN assist mode (LAMCR:LAMEN=1). The result of the serial test can be confirmed with the transmission/reception flag and the status flag. Please refer to Table 7-2 for the transmission/reception flag and the status flag.

■ Pseudo error test mode

In assist mode (LAMCR:LAMEN=1), the LIN bus error, the LIN ID parity error, the LIN checksum error, and the framing error can be artificially caused. These errors can be caused simultaneously.

Moreover, the following self-diagnoses become possible by using them together with the serial test mode.

- Pseudo LIN bus error test mode
- Pseudo LIN ID parity error test mode
- Pseudo LIN checksum error test mode
- Pseudo framing error test mode

● Method of starting pseudo error test mode

It is necessary to write in key code control bit (LAMERT:KEY1,KEY0) according to the following procedure to start the pseudo error test mode, and to enable the pseudo trouble setting.

- KEY1-0="00"+ The pseudo trouble setting value is written.
- KEY1-0="01"+ The pseudo trouble setting value (the same value last time) is written.
- KEY1-0="10"+ The pseudo trouble setting value (the same value last time) is written.
- KEY1-0="11"+ The pseudo trouble setting value (the same value last time) is written.
- The pseudo trouble setting value becomes effective because of writing the fourth times.

If this procedure is not observed (When other registers are written/read during the writing procedure, the writing value is incorrect or this register is read during the writing procedure), writing becomes invalid.

The pseudo trouble setting can be released as well as setup procedure.

Note:

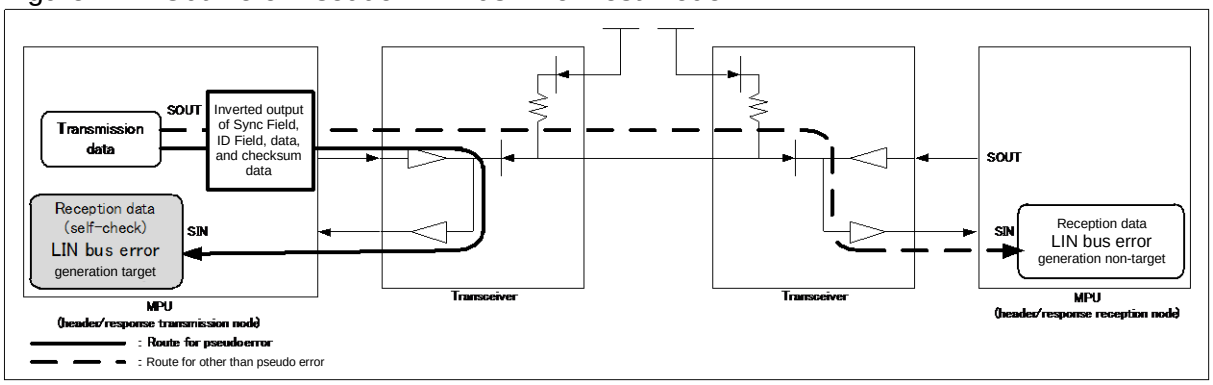
The assist mode stops when the following errors occur in the assist mode, and note the setting, please.

- LIN bus error
- LIN ID parity error
- Framing error

Overview of pseudo LIN bus error test mode

The pseudo LIN bus error test is executed by the self-check of the master/slave that transmits data. A pseudo LIN bus error cannot be detected by master/slave that receives the data.

Figure 7-21 Outline of Pseudo LIN Bus Error Test Mode



Please set the LIN bus error pseudo trouble setting bit by the method of starting the pseudo error test mode to start the pseudo LIN bus error test mode (LAMERT:LBSERT=1). The start of the pseudo LIN bus error test mode operates as follows.

· Master

Sync Field, ID Field, data, and checksum are transmitted.
Reception data is inverted according to the timing of the stop bit from when LIN bus error pseudo trouble was set (LAMERT:LBSERT=1), the LIN bus error is generated when self-checking it, and "1" is set to flag bit (LAMESR: LBSER).

· Slave

Data and checksum are transmitted.
Reception data is inverted according to the timing of the stop bit from when LIN bus error pseudo trouble was set (LAMERT:LBSERT=1), the LIN bus error is generated when self-checking it, and "1" is set to flag bit (LAMESR: LBSER).

The LIN bus error is generated until the pseudo LIN bus error test mode setting is released (LAMESR:LBSER=0).

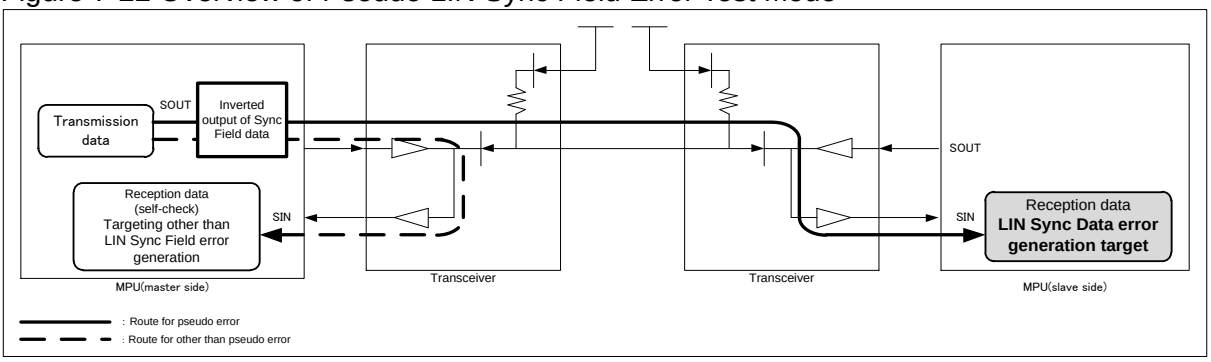
Note:

The transmission/reception processing of the header/response part of the assist mode stops by LIN bus error detection (LAMESR:LBSER=0).

● Overview of pseudo LIN Sync Data error test mode

The slave that checks Sync Field value (0x55) can execute the pseudo LIN Sync Data error test.
A pseudo LIN Sync Field error cannot be detected by the master that transmits Sync Field.

Figure 7-22 Overview of Pseudo LIN Sync Field Error Test Mode



It is necessary to set the LIN Sync Data error pseudo trouble setting bit to effective (LAMERT:LSFERT=1) by using the method of starting the pseudo error test mode to start the pseudo LIN Sync Data error test mode.
The values (0x55) are all inverted by the master set to pseudo LIN Sync Data error pseudo trouble setting (LAMERT:LSFERT=1) before the start bit of Sync Field when it transmits Sync Field. It does continuing this operation until the pseudo LIN Sync Data error test mode setting is released (LAMESR:LSFERT=0).

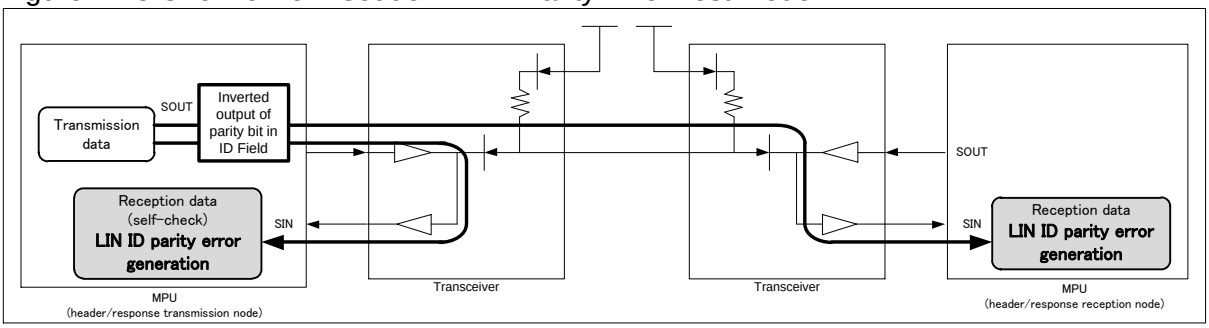
Notes:

- The detection of the LIN Sync Data error is detected in slave (SCR:MS=1) of assist mode (LAMCR:LAMEN=1).
- The header reception of the assist mode and the transmission/reception processing of the response stop by LIN Sync Data error detection (LAMESR:LBSER=1).

● Overview of pseudo LIN ID parity error test mode

The pseudo LIN ID parity error test can be executed by self-check of the master that transmits ID Field and the slave that receives ID Field.

Figure 7-23 Overview of Pseudo LIN ID Parity Error Test Mode



It is necessary to set the LIN ID parity error pseudo trouble setting bit to effective (LAMERT:LPTERT=1) by the method of starting the pseudo error test mode to start the pseudo LIN ID parity error test mode.
The master to which the pseudo LIN ID parity error trouble setting (LAMERT:LPTERT=1) was enabled before the start bit of ID Field inverts and outputs all parity values (2 bit) in the ID Field when the ID Field is transmitted.

The LIN ID parity error is generated when ID Field is received, and "1" is set by flag bit (LAMESR:LPTER).
LIN ID parity error continues to be generated until the pseudo LIN ID parity error test mode is disabled (LAMESR:LPTERT=0).

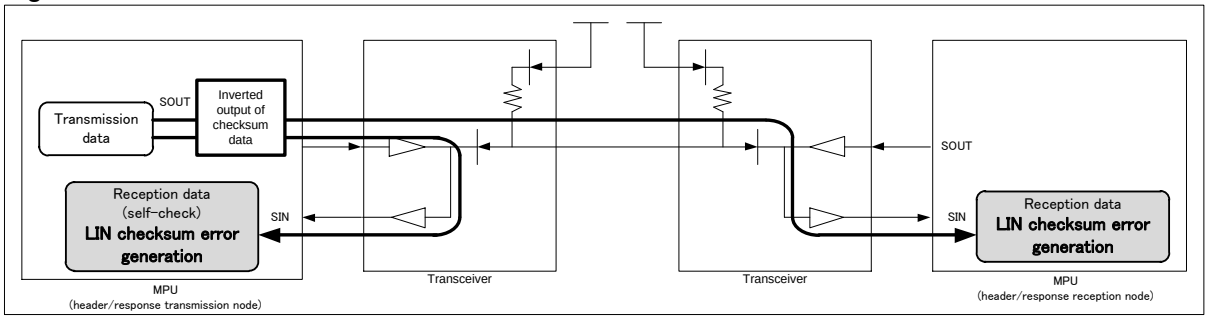
Note:

The transmission/reception processing of the response of the assist mode stops by LIN ID parity error detection (LAMESR:LPTER=1).

● Overview of pseudo LIN checksum error test mode

The pseudo LIN checksum error test can be executed on the side that transmits the response for self-check and on the side that receives the response.

Figure 7-24 Outline of Pseudo LIN Checksum Error Test Mode

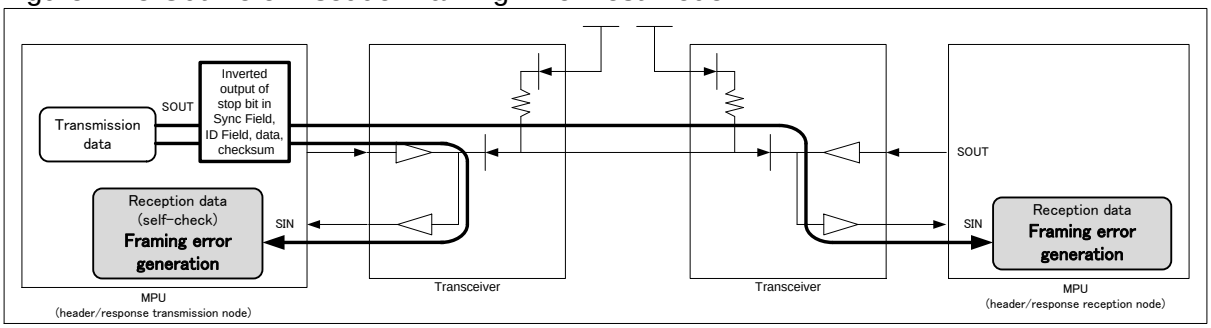


It is necessary to set the LIN checksum error pseudo trouble setting bit to effective (LAMERT:LCSERT=1) by the method of starting the pseudo error test mode to start the pseudo LIN checksum error test mode.
All the values are inverted when checksum is transmitted and the node to which the pseudo LIN checksum error pseudo trouble was set before the start bit of checksum (LAMERT:LCSERT=1) is output.
The LIN checksum error occurs when checksum is received, and "1" is set by flag bit (LAMESR:LCSER).
The LIN checksum error is generated until the pseudo LIN checksum error test mode setting is released (LAMESR:LCSERT=0).

● Overview of pseudo framing error test mode

The pseudo framing error test can be executed on the side that transmits data for self-check and on the side that receives data.

Figure 7-25 Outline of Pseudo Framing Error Test Mode



It is necessary to set the framing error pseudo trouble setting bit to effective (LAMERT:FRET=1) by the method of starting the pseudo error test mode to start the pseudo framing error test mode.
The start of the pseudo framing error test mode operates as follows.

· Master

When Sync Field, ID Field, data, and checksum are transmitted, the value of the stop bit (" H" level) is inverted to be output when the framing error pseudo trouble is set before the stop bit of each Field (LAMERT:FRET=1).
A framing error occurs at reception, and "1" is set to flag bit (LAMESR: FRE).

· Slave

When data and checksum are transmitted, the value of the stop bit ("H" level) is inverted to be output when the framing error pseudo trouble is set before the stop bit of each Field (LAMERT:FRET=1).
A framing error occurs at reception, and "1" is set to flag bit (LAMESR: FRE).
The framing error is generated until the pseudo framing error test mode setting is released (LAMESR:FRET=0).

Note:

The transmission/reception processing of the header/response part of the assist mode stops by framing error detection (LAMESR:FRE=1).

7.5. Operation of LIN Interface (v2.1)

Operation of LIN Interface (v2.1) is shown.

The LIN interface (v2.1) operates for the master/slave bidirectional LIN communication.

7.5.1. Manual mode

This section explains the manual mode.

■ Master Operations

● Selecting Master Operation

To make the LIN interface (v2.1) work as the master device, set the SCR:MS bit to "0".

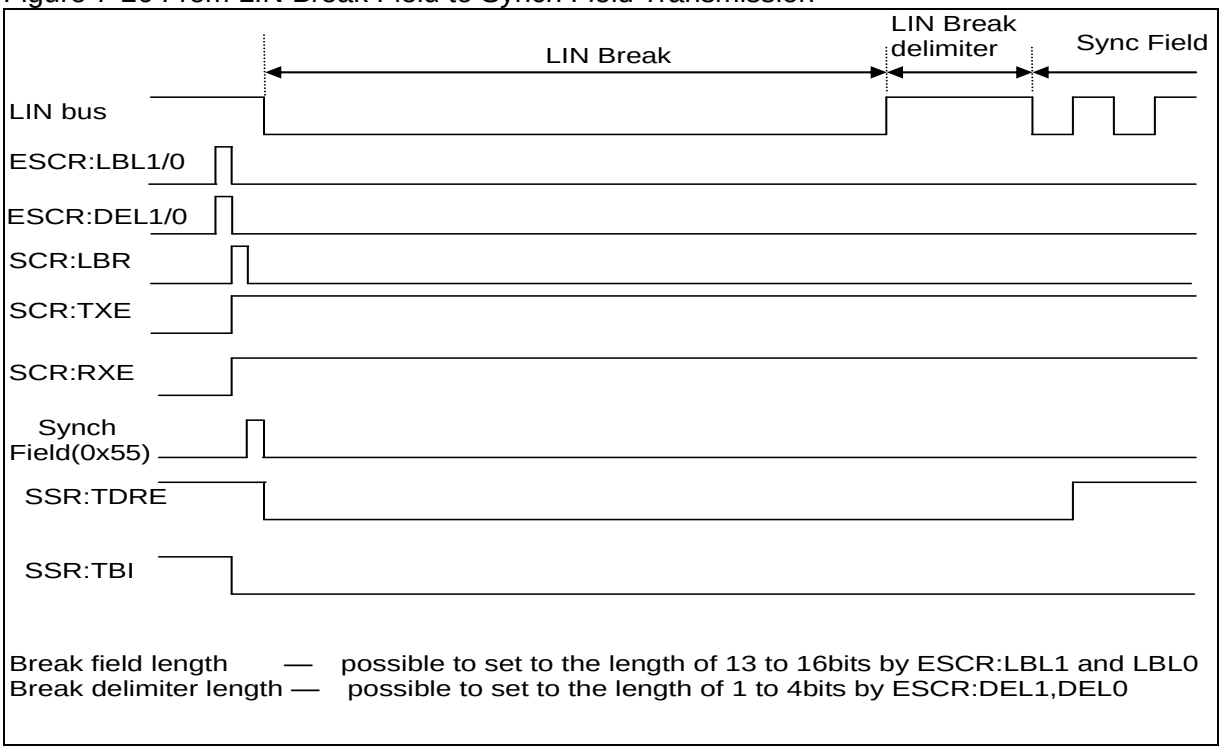
● From LIN Break Field Transmission to Synch Field Transmission

- Select of the LIN Break Field length (ESCR:LBL1, LBL0) and the Break Field delimiter length (ESCR:DEL1, DEL0).
- The LIN Break Field is transmitted by enabling transmission (SCR:TXE=1) and setting the SCR:LBR bit (LIN Break field setting bit) to "1".
- The Synch Field is transmitted by writing 0x55 in the transmit data register (TDR).

Notes:

- Set 0x55 in the transmit data register (TDR) after setting the SCR:LBR bit (LIN Break field setting bit) to "1".
- Even if the SCR:RXE bit (reception enable bit) is set to "1", the LIN Break Field part is not received.

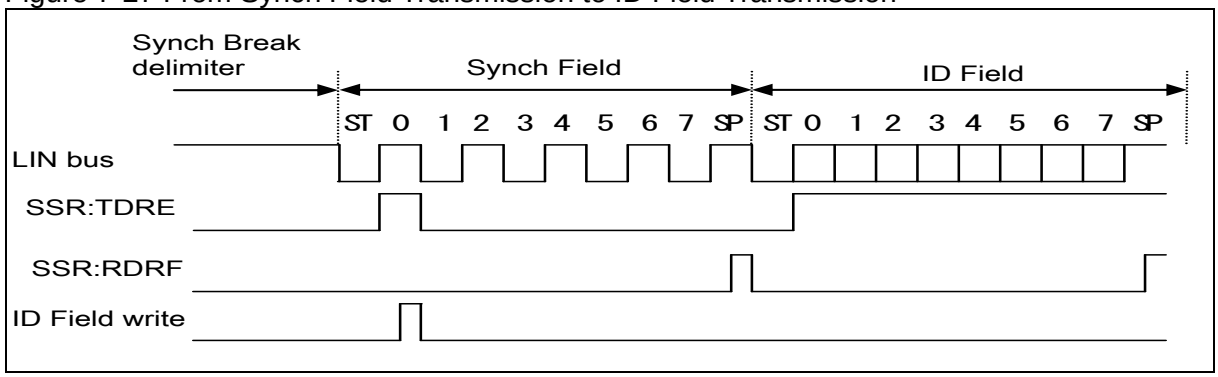
Figure 7-26 From LIN Break Field to Synch Field Transmission



From Synch Field Transmission to ID Field Transmission

- When the first bit of the Synch Field (0x55) is transmitted, the SSR:TDRE (transmission data empty) bit is set to "1". If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs.
- When this interrupt occurs, the ID Field can be written to the transmit data register (TDR).
- When a reception interrupt occurs, the reception data will be compared with the transmission data to confirm that no error has occurred.
- The ID Field is output in an LSB-first fashion with a data length of 8 bits.

Figure 7-27 From Synch Field Transmission to ID Field Transmission

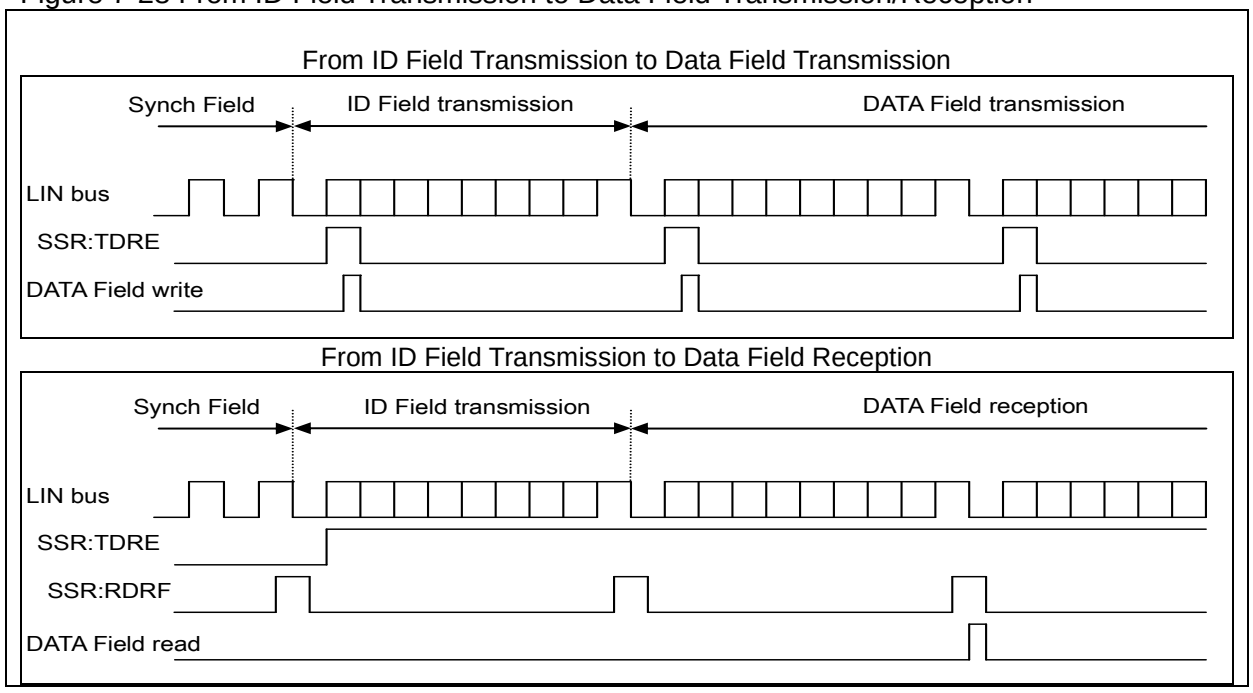


From ID Field Transmission to Data Field Transmission/Reception

Specify whether to transmit the Data Field to the slave device or receive it.

- In the case of Data Field transmission:
- When the first bit of the ID Field is transmitted, the SSR:TDRE bit is set to "1". Data can then be written in the Data Field.
- In the case of Data Field reception:
- When the first bit of the ID Field is transmitted, the SSR:TDRE bit is set to "1". However, do not write transmission data.
- Also, disable transmission interrupts (SCR:TIE=0).
- When the Data Field is received, the SSR:RDRF bit is set to "1". If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt will occur.
- The start bit is detected when the falling edge is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.

Figure 7-28 From ID Field Transmission to Data Field Transmission/Reception



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.

● Master Operation Timing Chart (FIFO Unused)

Figure 7-29 LIN Bus Timing (at the Time of Data Field Transmission without Using FIFO)

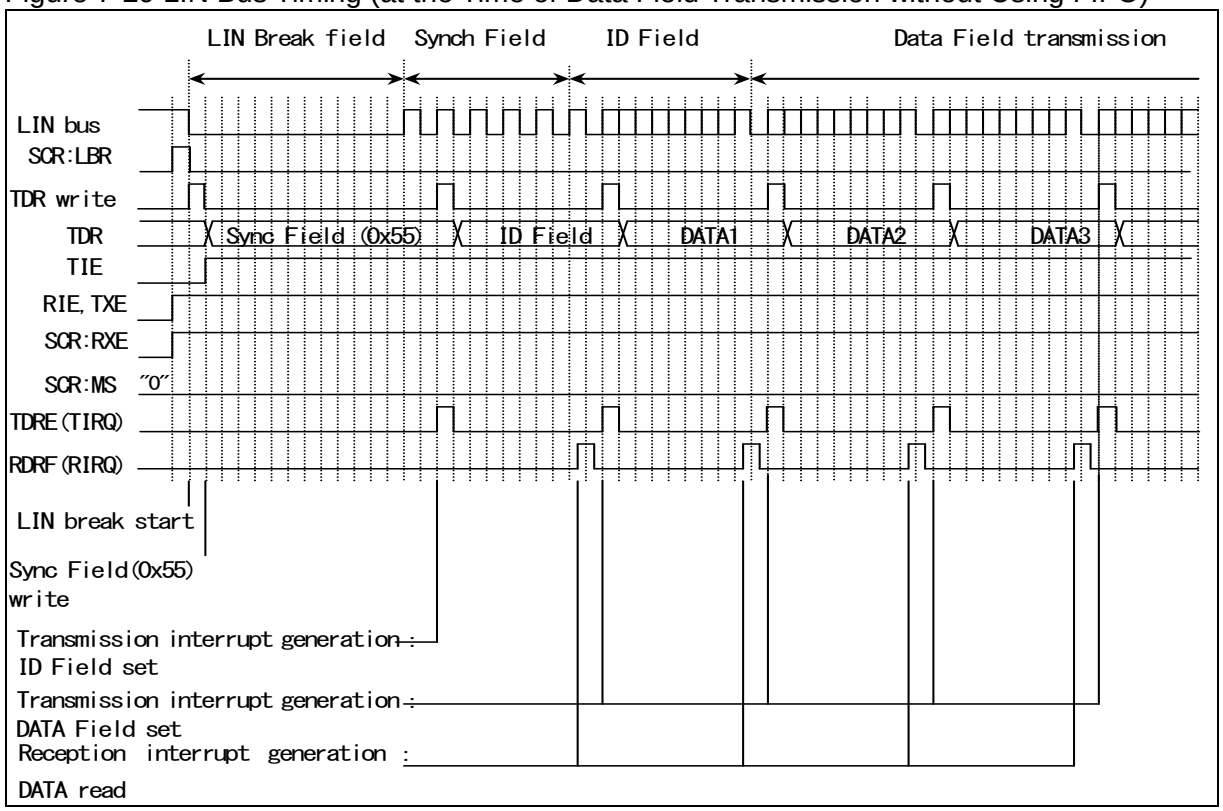
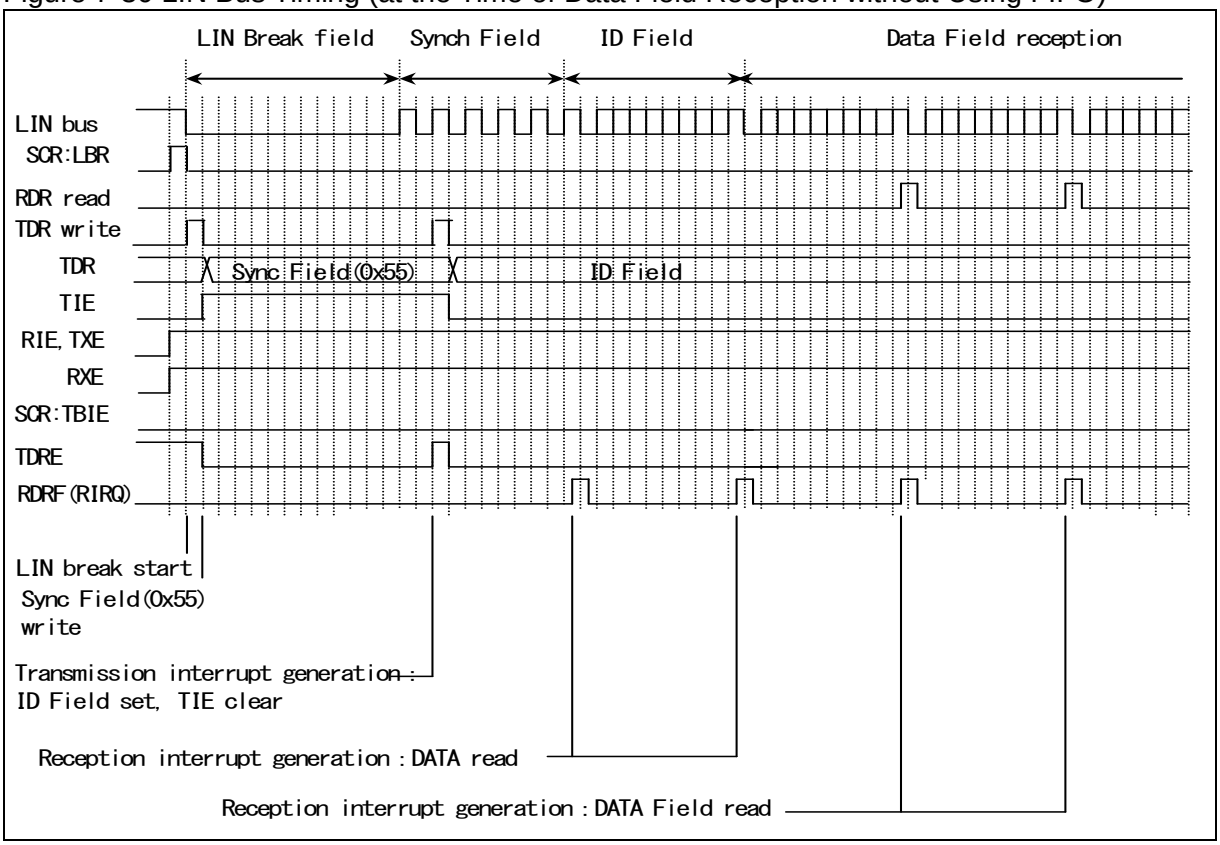


Figure 7-30 LIN Bus Timing (at the Time of Data Field Reception without Using FIFO)



● Master Device Operation Timing Chart (FIFO Used)

Figure 7-31 LIN Bus Timing (at the Time of Data Field Transmission when Using FIFO)

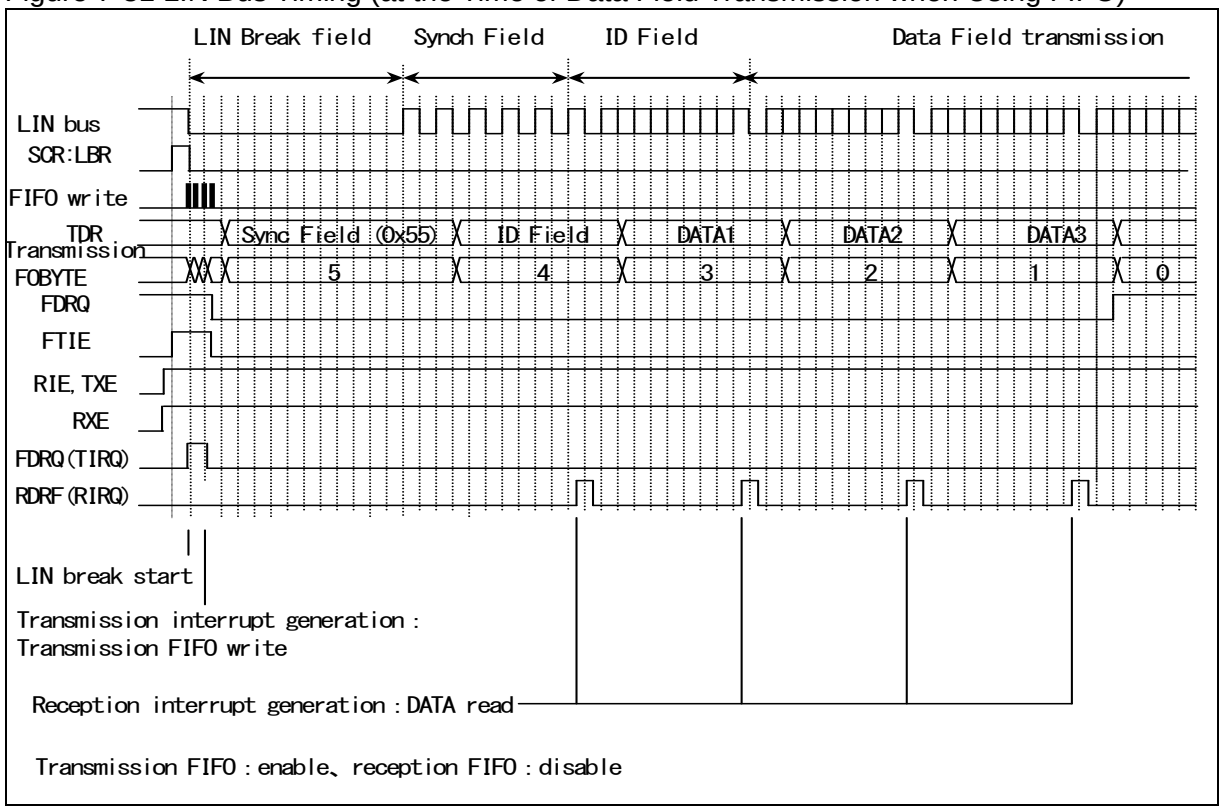
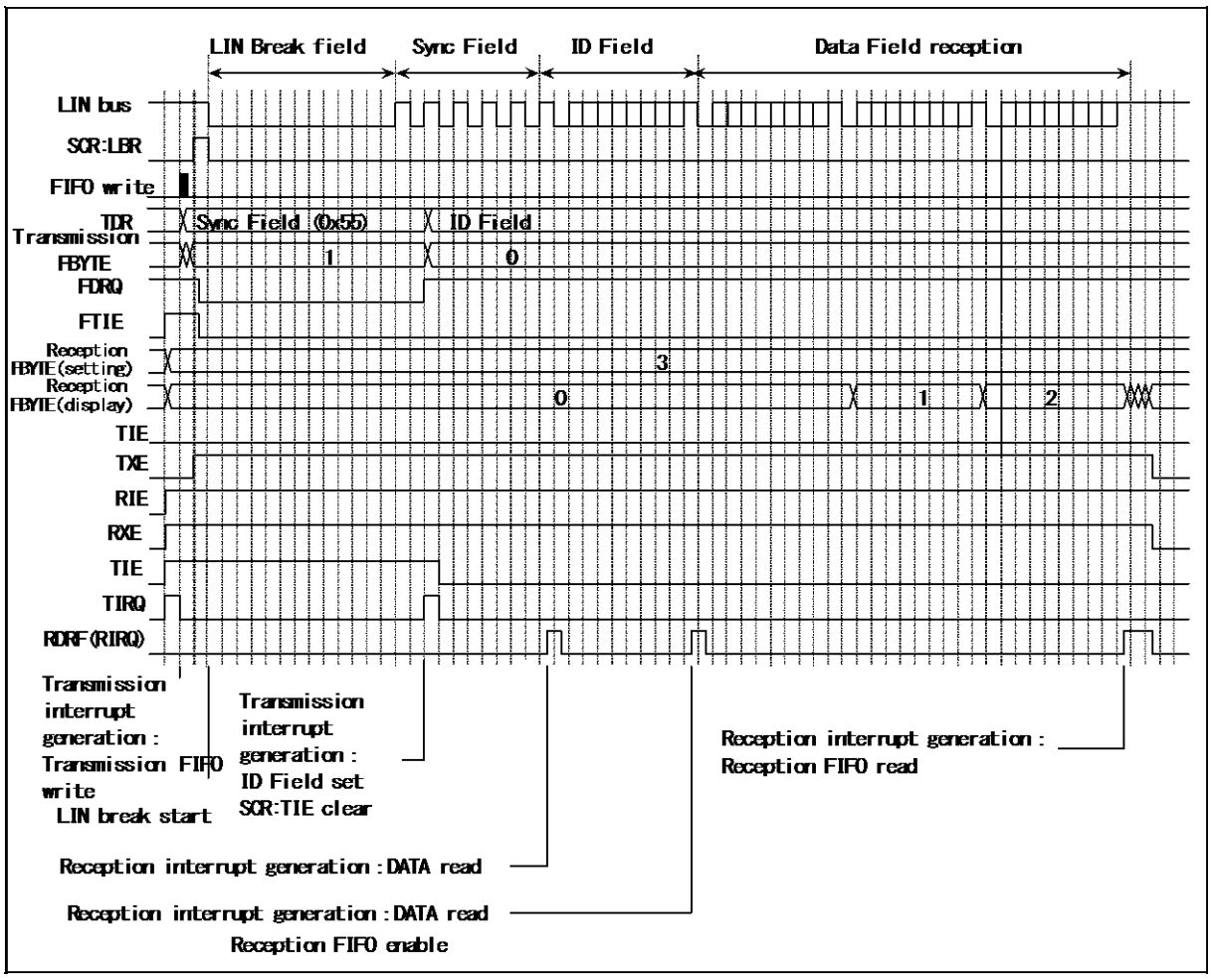


Figure 7-32 LIN Bus Timing (at the Time of Data Field Reception when Using FIFO)



■ Slave Operations

● Selecting Slave Operation

To make the LIN interface (v2.1) work as the slave device, set the SCR:MS bit to "1".

● From LIN Break Field Reception to Synch Field Reception

The method of confirming that the automatic baud rate adjustment was executed from LIN Break Field reception to Sync Field reception is as follows.

· Method of comparing BGR with STMR

Processing using this method is as follows.

1. Method of comparing BGR with STMR

(1) The automatic baud rate adjustment is set to effective (SACSR:AUTE=1).

(2) When LIN Break Field is input, LIN Break Field is detected in the 11th bit (SSR:LBD=1). At this time, if the ESCR:LBIE bit is set in "1", the status interrupt is generated. After LIN Break Field is detected (SSR:LBD=1), the serial timer is set to prohibition (SACSR:TMRE=0).

(3) When LIN interface (v2.1) detects the first falling edge of Sync Field, serial timer register (STMR) is initialized to "0".

(4) When the fifth falling edge of Sync Field is detected, Sync Field detection flag (SACSR:SFD) is set in "1". At this time, confirm whether the automatic baud rate adjustment was executed by checking the following.

· The reading value of baud rate generator register (BGR) becomes equal with serial timer register (STMR) at Sync Field detection (SACSR:SFD=1) when the automatic baud rate is adjusted.
· The reading value of baud rate generator register (BGR) is different from serial timer register (STMR) at Sync Field detection (SACSR:SFD=1) when the automatic baud rate adjustment is not adjusted.

Figure 7-33 From LIN Break Field Reception to Synch Field Reception (in Case where STMR is SFUR or smaller and SFLR or larger)

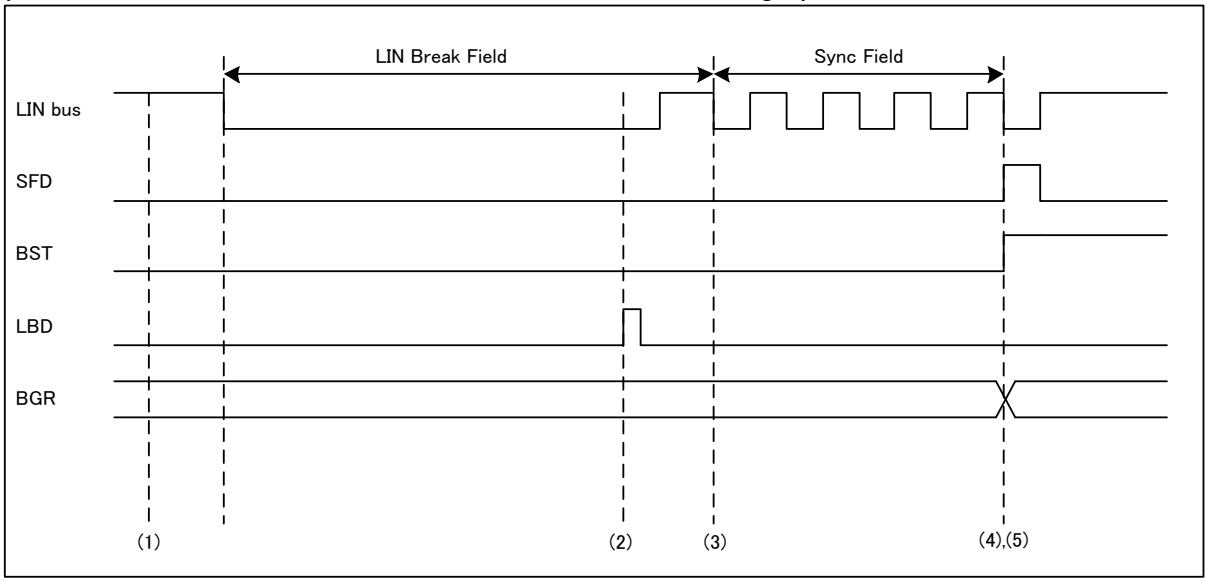
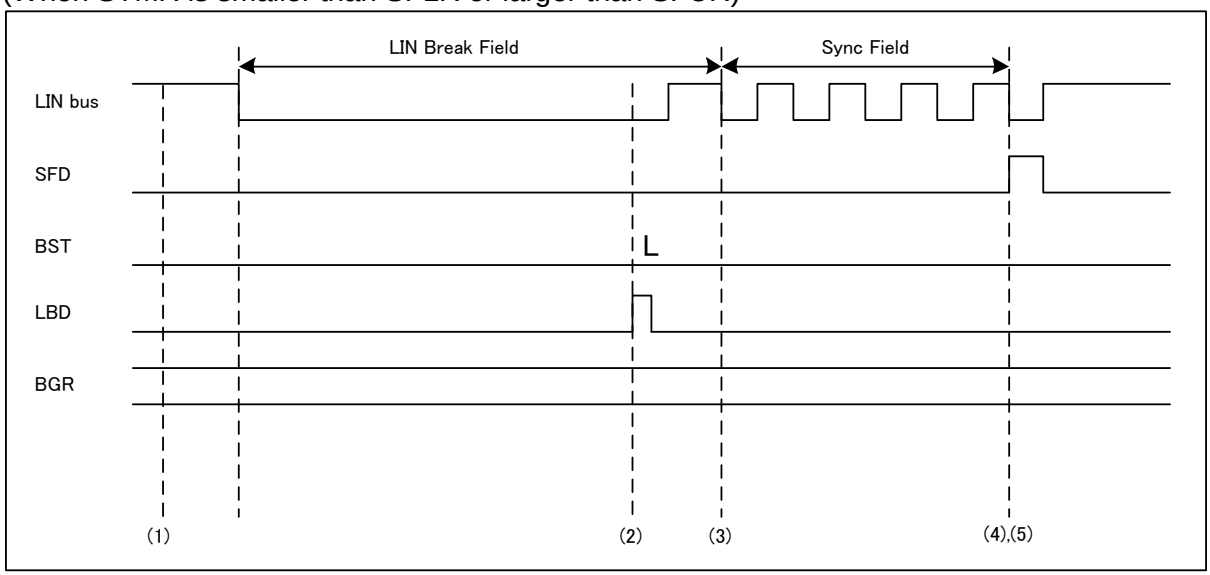


Figure 7-34 From LIN Break Field Reception to Synch Field Reception (When STMR is smaller than SFLR or larger than SFUR)



Note:

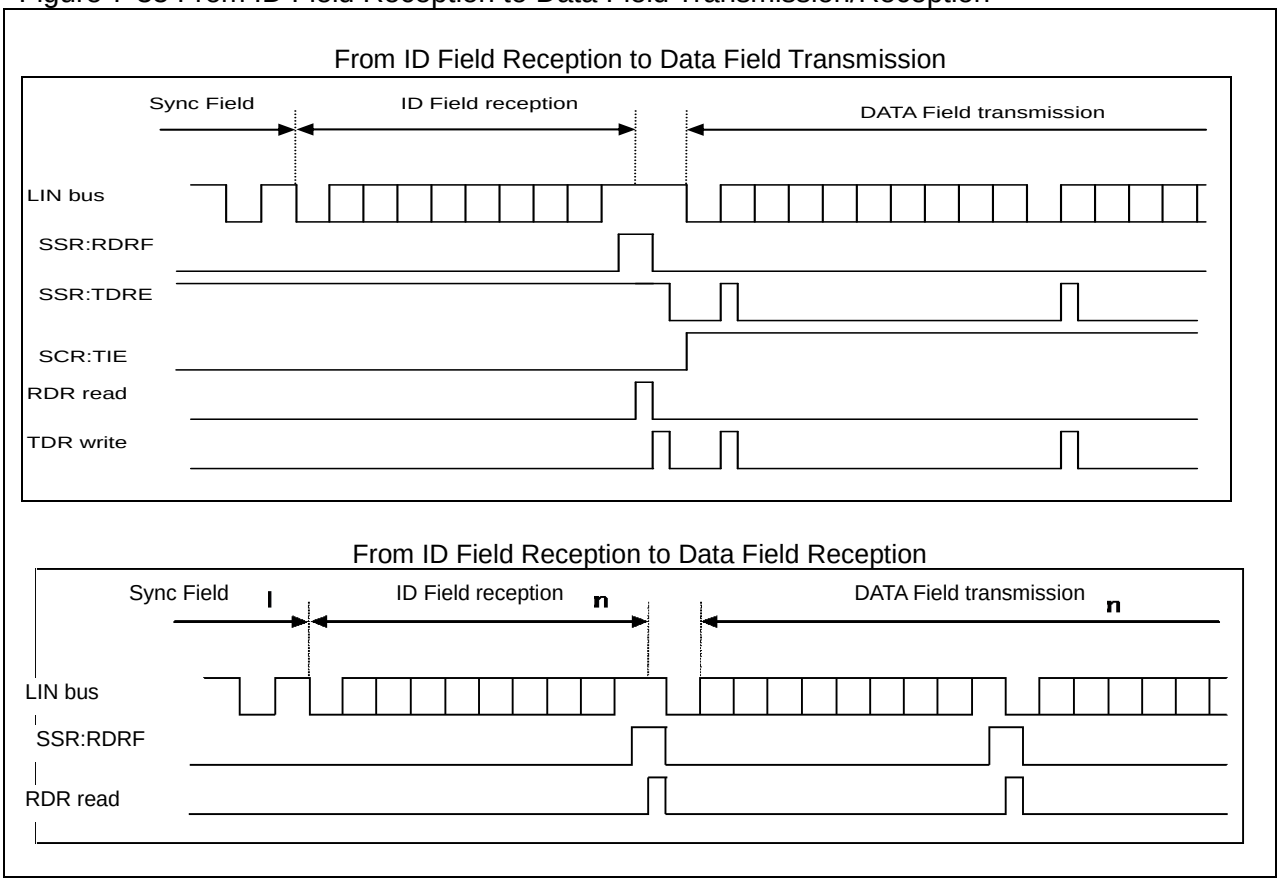
While in Break field and Sync field, set reception disabled (SCR:RXE=0).

From ID Field Reception to Data Field Transmission/Reception

After the ID Field is received, specify whether to transmit the Data Field to the master device or receive it.

- In the case of Data Field transmission:
  - After the ID Field is received, write data in the transmit data register (TDR). At this time, transmission interrupts must be enabled (SCR:TIE=1).
- In the case of Data Field reception:
  - For every Data Field reception, the SSR:RDRF bit is set to "1". If reception interrupts are enabled (SCR:RDRF=1) at this time, a reception interrupt occurs.
  - The start bit is detected when the falling edge is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.

Figure 7-35 From ID Field Reception to Data Field Transmission/Reception



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive data normally. Leave a space between frames if successive frames are to be output.

● Slave Operation Timing Chart

Figure 7-36 LIN Bus timing (DATA Field Transmitted: FIFO Unused, AUTE=1)

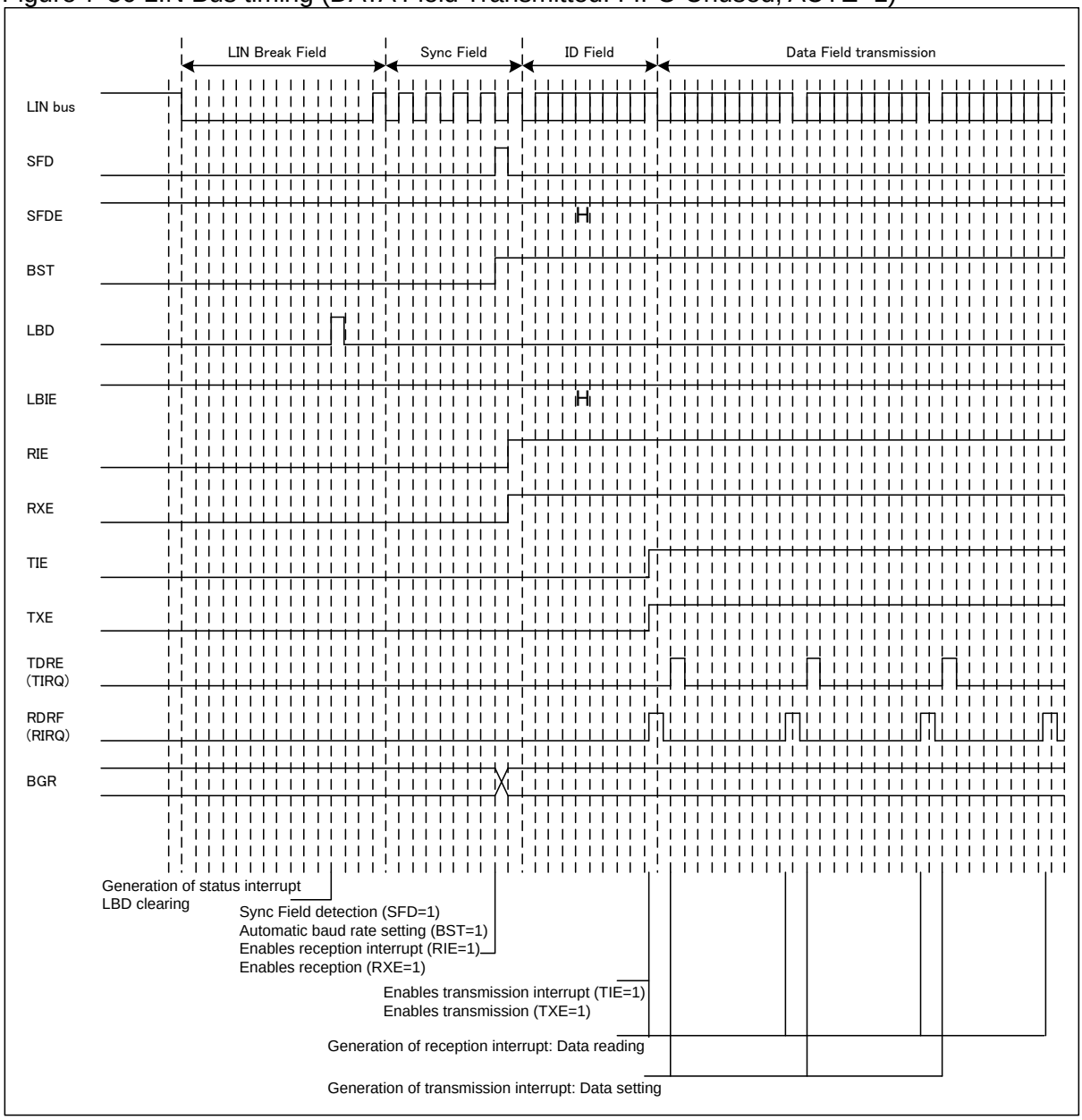
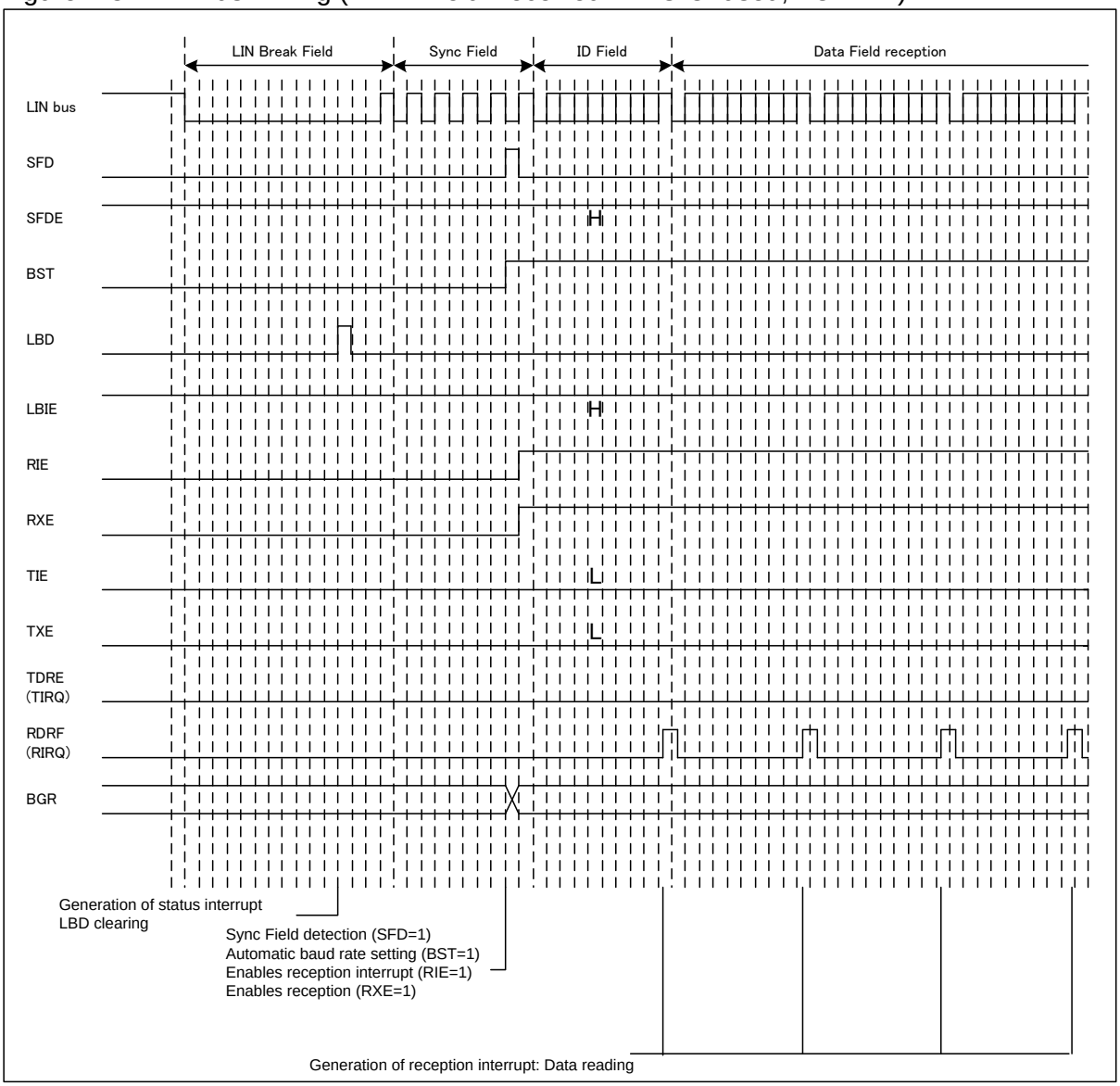


Figure 7-37 LIN Bus Timing (DATA Field Received: FIFO Unused, AUTE=1)



● FIFO Used

Figure 7-38 LIN Bus Timing (DATA Field Transmitted: FIFO Used, AUTE=1)

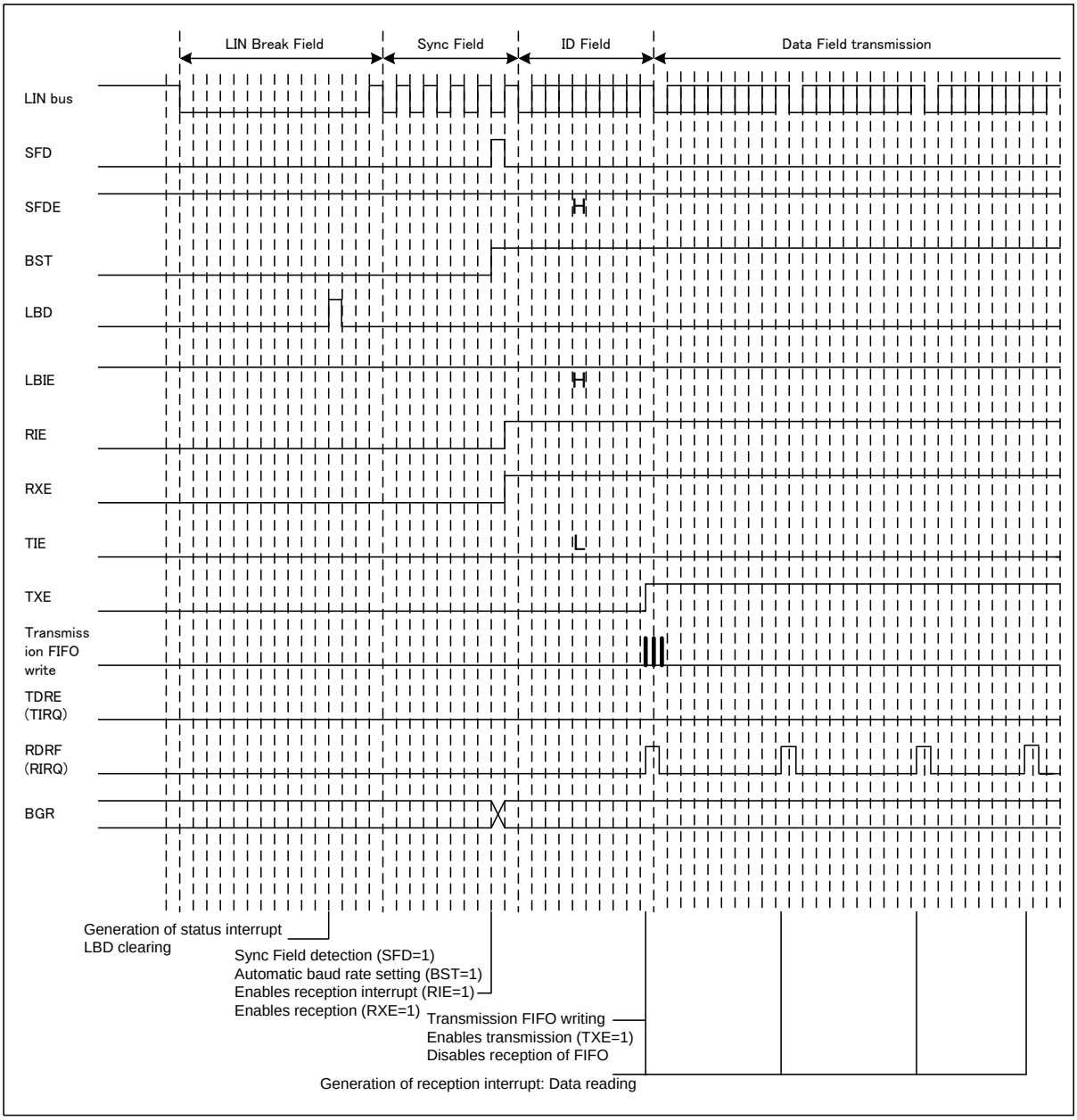
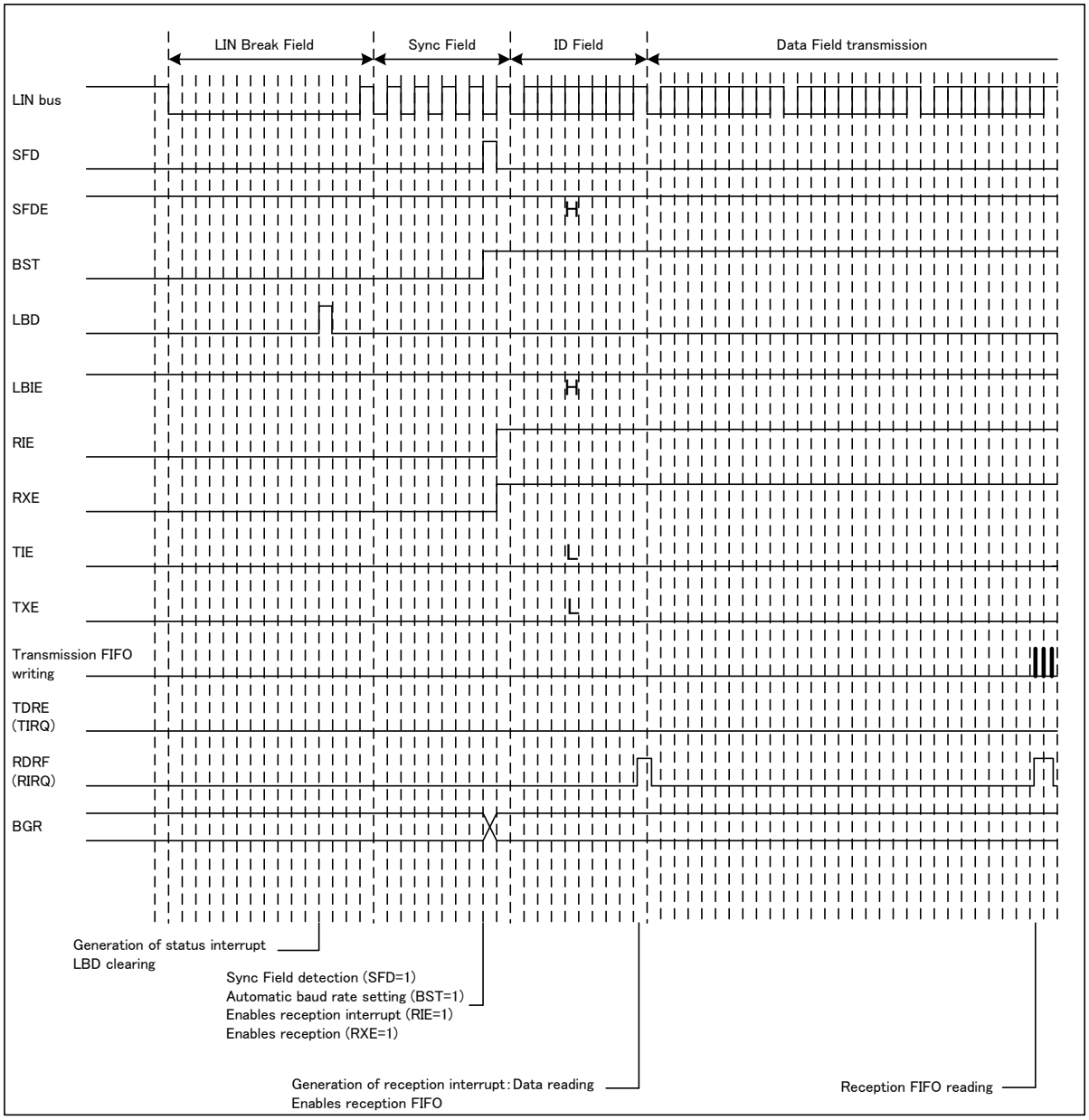


Figure 7-39 LIN Bus Timing (DATA Field Received: FIFO Used, AUTE=1)



7.5.2. Assist Mode

This section explains the assist mode.

The assist mode has the function to transmit/receive the LIN automatic header, and generate and check the following.

- Parity generation and check on ID Field
- Generation and check on checksum

■ Master operation

● Automatic header transmission setting

To transmit the automatic header in the assist mode, please set the SCR:LBR bit (LIN Break Field setting bit) to "1" after initial setting. "LIN Break Field - Sync Field - ID Field" is automatically transmitted by setting the SCR:LBR bit to "1". The transmission setting is shown as follows.

- Please set the SCR:MS bit (master/slave function select bit) to "0" to operate as the master.
- Please set the LAMCR:LAMEN bit (LIN assist mode processing enable bit) to "1".
- Please set the ID Field value before the LIN assist mode begins.
- Please set the LAMCR:LIDEN bit (LIN ID register enable bit) to "1" when you use LIN assist mode transmission ID register (LAMTID).
- Please set the LAMCR:LIDEN bit to "0" when you use data transmit register (TDR).
- Please set the ID Field data to LAMTID (LIN assist mode transmission ID register) when you set the LAMCR:LIDEN bit to "1".
- Please set selection of the LIN break field length (ESCR:LBL2, LBL1, LBL0) and selection of the LIN Break delimiter length (ESCR:DEL1, DEL0).
- Please set the selection of the stop bit length (SMR:SBL and ESCR:ESBL).
- LIN Break Field transmitted on the master side is detected also on the master side. The SSR:LBD bit is set in "1" when detected. At this time, if ESCR:LBIE is set to "1", the status interrupt is generated. Please set ESCR:LBIE to "0" and change the interrupt to the prohibition setting for the LIN assist mode.
- The Sync Field value transmitted on the master side is detected also on the master side. The SACSR:SFD bit is set in "1" when detected. At this time, if the SACSR:SFDE bit is set to "1", the interrupt is generated. Please set SACSR:SFDE to "0" for the LIN assist mode and prohibit interrupting.
- Please set transmission operation enable bit (SCR:TXE) to "1" (transmission enable).

● From LIN Break Field to ID Field transmission

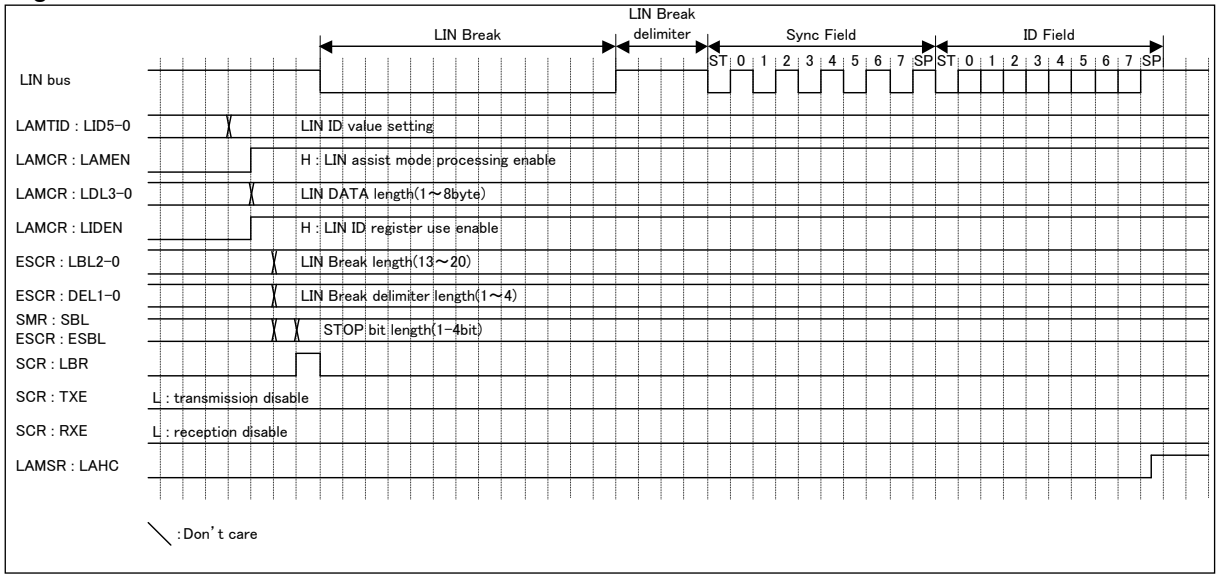
- Please set LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation).
- LIN Break Field set with ESCR:LBL2 to LBL0 is transmitted.
- Please write the ID Field data in data transmission register (TDR) when you do not use LIN assist mode transmission ID register (LAMTID).
- LIN Break Field that the master transmitted is received on the master side, and the bus error is checked.
- After LIN Break Field is transmitted, the LIN Break Field delimiter set with ESCR:DEL1 and DEL0 is transmitted.
- After the LIN Break Field delimiter is transmitted, Sync Field (0x55 fixation value) is transmitted.
- Sync Field that the master transmitted is received on the master side, and the bus error is checked.
- After Sync Field is transmitted, the set ID Field value is transmitted. When the LAMCR:LIDEN bit is "0", the value set to TDR is transmitted as ID FIeld value. When the LAMCR:LIDEN bit is "1", the value set to LAMTID is transmitted as ID FIeld value.

- The LIN assist mode does the parity arithmetic operations of ID Field automatically.
- When the first bit of ID Field is transmitted, SSR:TDRE (the transmission data empty) bit is set to "1". At this time, if transmission interrupt enable (SCR:TIE=1) is done, the transmission interrupt is generated.
- When transmission interrupt (TDRE) is generated, the transmission data can be written in transmit data register (TDR).
- ID Field is eight bits in the data length, and it is output with LSB first. The LIN parity in ID Field is operated automatically.
- When the transmission of ID Field is completed, the LIN automatic header transmission completion flag is set (LAMSR:LAHC="1"). At this time, when the LIN automatic header transmission completion interrupt enable bit has been enabled (LAMIER:LAHCIE="1"), the interrupt is generated.
- The transmission stops when the following errors are generated.
  - LIN bus error
  - LIN ID parity error
  - Framing error

Notes:

- The automatic header transmission stops when SCR:TXE is set to "0" after the LIN assist mode activation.
- Please do not change the automatic header transmission setting after the LIN assist mode activation.

Figure 7-40 From LIN Break Field to ID Field Transmission



LIN Break Field retransmission processing in assist mode

Only when transmission prohibition setting (SCR:TXE=0) and state (SSR:TBI=1) of the transmission bus idle, LIN Break Field can be set (SCR:TBR=1). Therefore, when other than the transmission prohibition setting (SCR:TXE=0) or the transmission bus idle (SCR:LBR=1), the LIN Break Field setting (SCR:LBR=1) can be set after initializing the state according to the following procedure.

- First of all, execute transmission prohibition setting (SCR:TXE=0) and reception prohibition setting (SCR:RXE=0).
- Clear the transmission data before retransmission.

  - When transmission FIFO is used, reset the transmission FIFO (FCR0:FCL1=1 or FCR0:FCL2=1) after prohibiting transmission FIFO operation (FCR0:FE1=0 or FCR0:FE2=0).
  - Then, execute the transmission data register clear (LAMCR:LTDRCL=1), and set the state to transmission bus idle.
- Clear the reception data before retransmission.
  - When reception FIFO is used, reset the reception FIFO (FCR0:FCL1=1 or FCR0:FCL2=1) after prohibiting reception FIFO operation (FCR0:FE1=0 or FCR0:FE2=0).
  - Then, read the RDR register to clear the reception data register
- When LIN assist mode transmission ID register (LAMTID) is used after the above-mentioned is processed, set the ID Field data to LAMTID (LIN assist mode transmission ID register).
  - The processing that follows is equal to the preceding section "From LIN Break Field to ID Field transmission".

● DATA Field transmission/reception

Whether DATA Field is transmitted or received to the slave device is selected.

(When DATA Field is transmitted)

- When LIN assist mode transmission register (LAMTID) is not used, SSR:TDRE is set to "1" if the first bit of ID Field is transmitted. At this time, DATA Field can be written.
- When LIN assist mode transmission register (LAMTID) is used, DATA Field can be written after LIN Break Field setting bit (SCR:LBR) is set to "1".
- Please set the transmission enable (SCR:TXE=1) during the period from the setting of LIN Break Field setting bit (SCR:LBR) to "1" to starting of the response transmission.
- The LIN assist mode does the checksum arithmetic operations automatically. The arithmetic operations of checksum can select the arithmetic operations method by LIN checksum type selection bit (LAMCR:LCSTYP).
- When the transmission of checksum is completed, the transmission bus idle flag (SSR:TBI) is set. At this time, when the transmission bus idle interrupt enable bit is set (SCR:TBIE="1"), the interrupt is generated.
- After the response transmission is completed (LAMSR:LCSC=1), transmission prohibition setting (SCR:TXE=0) is done.

Notes:

- The response transmission data (Data Field and checksum) when the assist mode operates cannot be stored in the RDR register.
- Please write data in FIFO after setting LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation bit) when you use FIFO.
- Please write the dummy value ("don't care") to the TDR register to operate checksum automatically and to transmit when you set the LIN data length to 0 byte length (LAMCR:LDL3-0="0000") in the response transmission. The TDR setting value at this time doesn't influence checksum.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes inverted ID Field.

Figure 7-41 From ID Field Transmission to DATA Field Transmission (FIFO Unused when ID Register is Used)

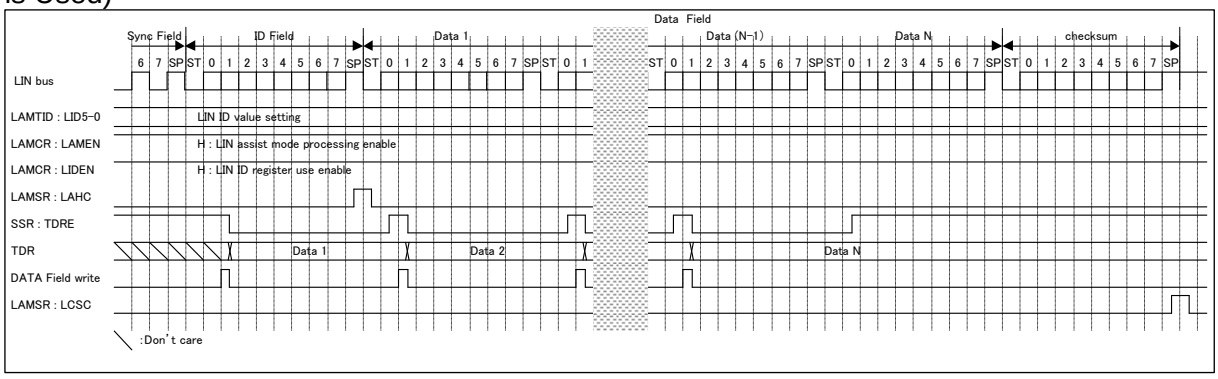
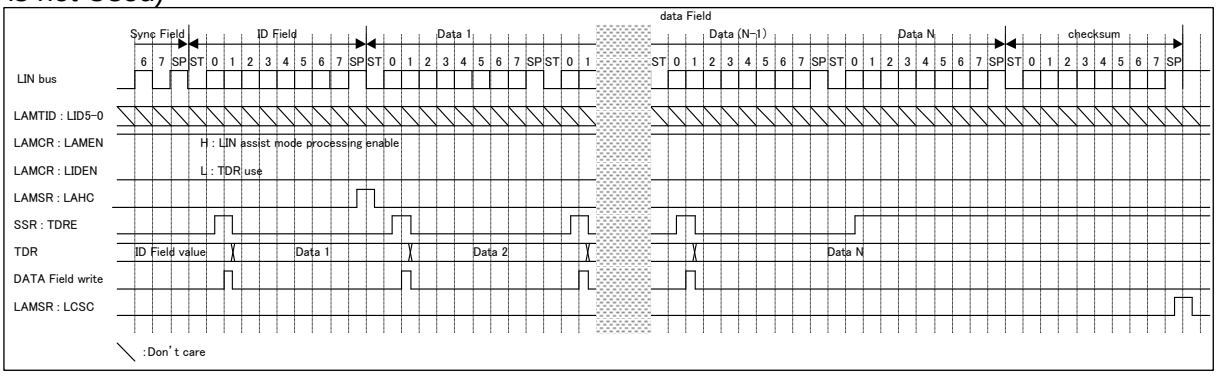


Figure 7-42 From ID Field Transmission to DATA Field Transmission (FIFO Unused when ID Register is not Used)



(When DATA Field is received)

- When LIN assist mode transmission register (LAMTID) is not used, please do not write data though SSR:TDRE is set to "1" when the first bit of ID Field is transmitted.
  Moreover, please set transmission interrupt prohibition (SCR:TIE="0").
- When LIN assist mode transmission register (LAMTID) is used, please do not write data though it is possible to write in DATA Field after LIN Break Field setting bit (SCR:LBR) is set to "1".
- Please set the reception enable (SCR:RXE=1) during the period from LIN Break Field detection (SSR:LBD=1) to the response reception starting.
- When DATA Field is received, SSR:RDRF is set to 1. At this time, if reception interrupt enable (SSR:RIE="1") is done, the reception interrupt is generated.
- When the reception of checksum is completed, the LIN checksum arithmetic operations completion flag is set (LAMSR:LCSC="1"). At this time, when the checksum arithmetic operations completion interrupt enable bit has been enabled (LAMIER:LCSCIE="1"), the interrupt is generated.
- After the checksum reception is completed (LAMSR:LCSC=1), reception prohibition setting (SCR:RXE=0) is done.
- Detection of the start bit is as follows; the falling edge is detected after passing through the noise filter (which samples serial data input in 3 bus clock and decides the value by majority), and the "L" is detected in the data after the noise filter at the sampling point.

Figure 7-43 From ID Field Transmission to DATA Field Reception (FIFO Unused when ID Register is Used)

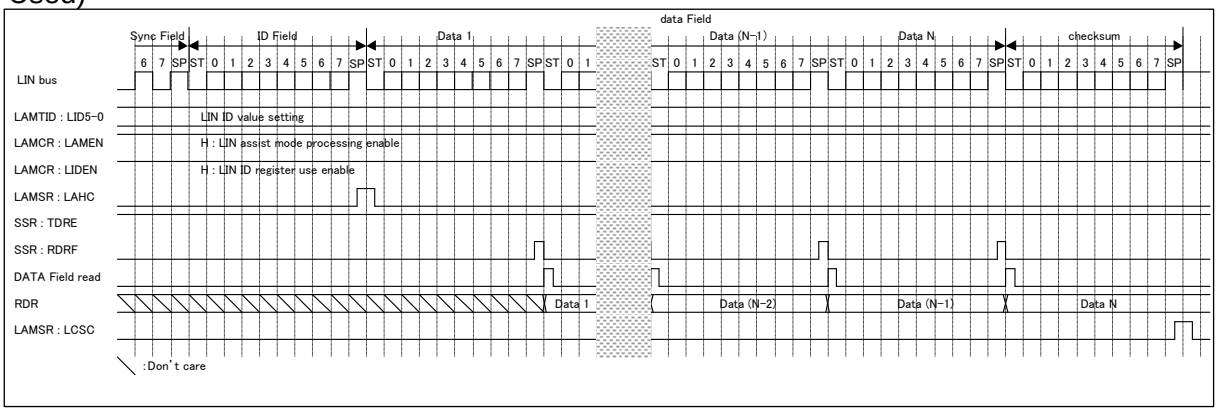
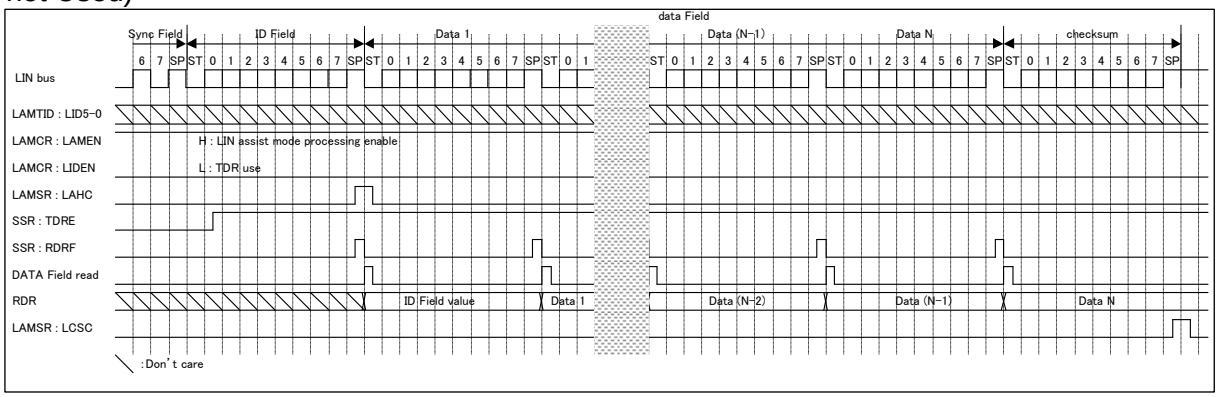


Figure 7-44 From ID Field Transmission to DATA Field Reception (FIFO Unused when ID Register is not Used)



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.
- The checksum value of the response reception when the assist mode operates is not stored in the RDR register.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes reversed ID Field.

- **Master operation timing chart (When FIFO unused).**

Figure 7-45 LIN Bus Timing (ID Register Use, DATA Field Transmission, and FIFO Unused)

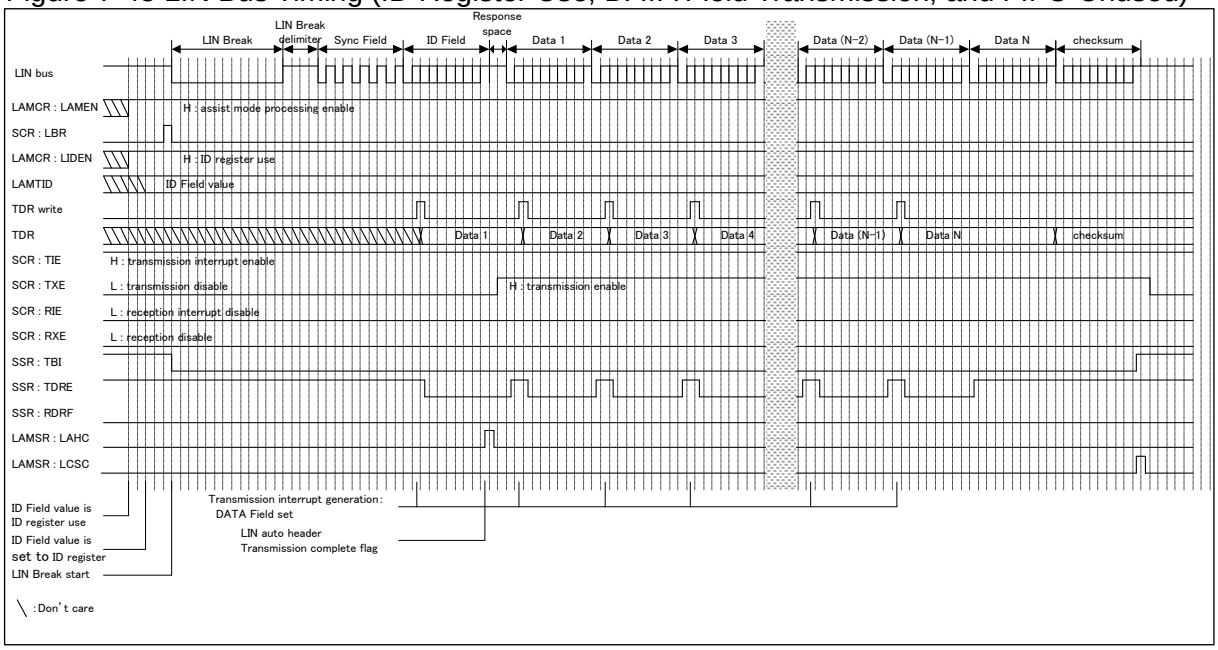


Figure 7-46 LIN Bus Timing (ID Register Unused, DATA Field Transmission, and FIFO Unused)

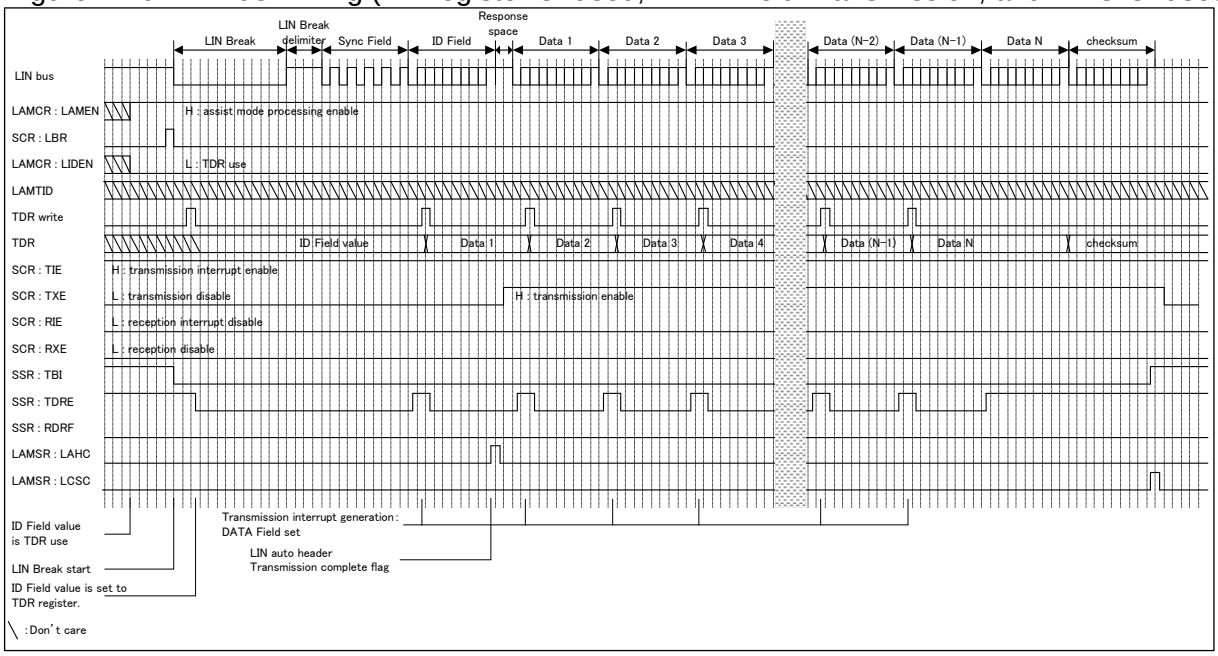


Figure 7-47 LIN Bus Timing (ID Register Use, DATA Field Reception, and FIFO Unused)

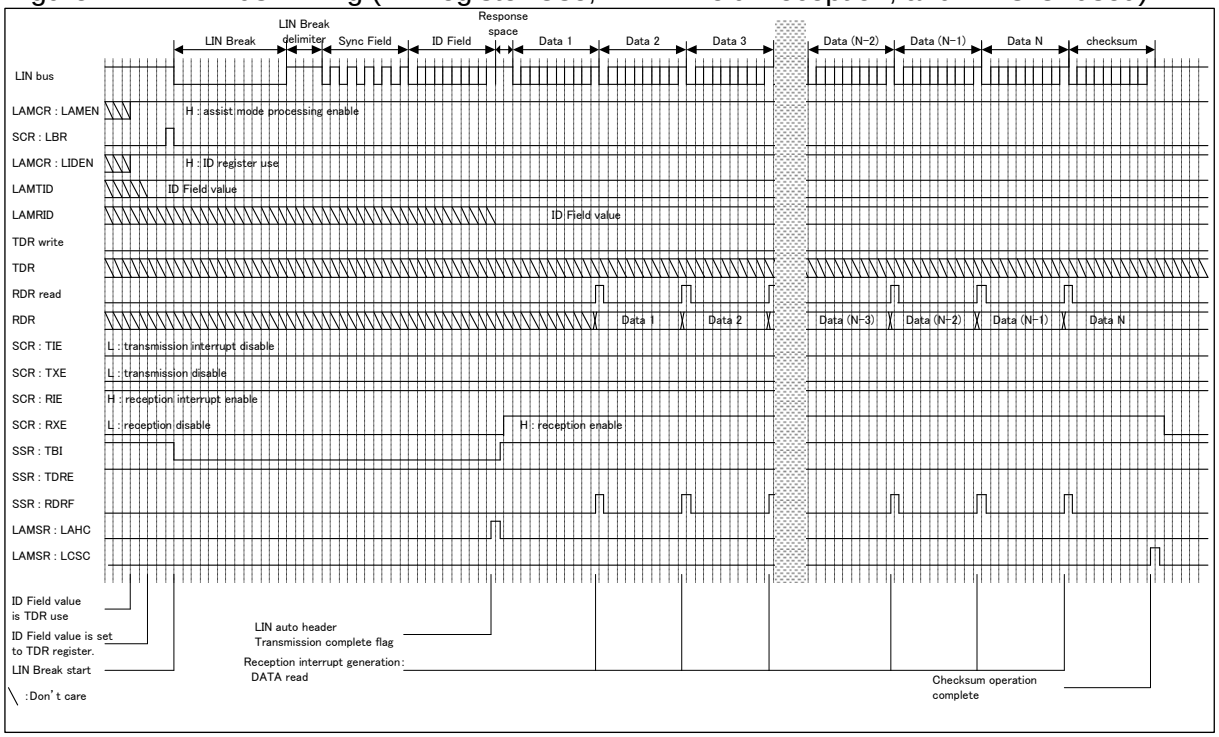


Figure 7-48 LIN Bus Timing (ID Register Unused, DATA Field Reception, and FIFO Unused)

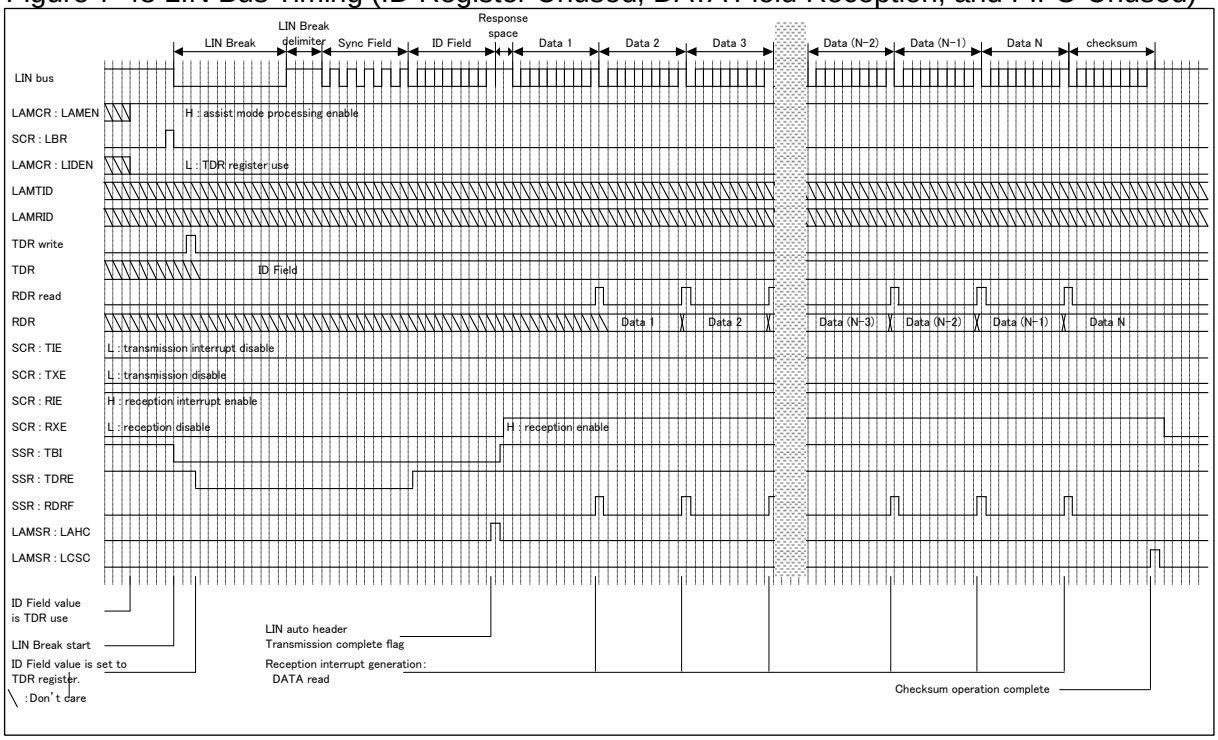


Figure 7-49 LIN Bus Timing (ID Register Use, DATA Field Transmission, and FIFO Use)

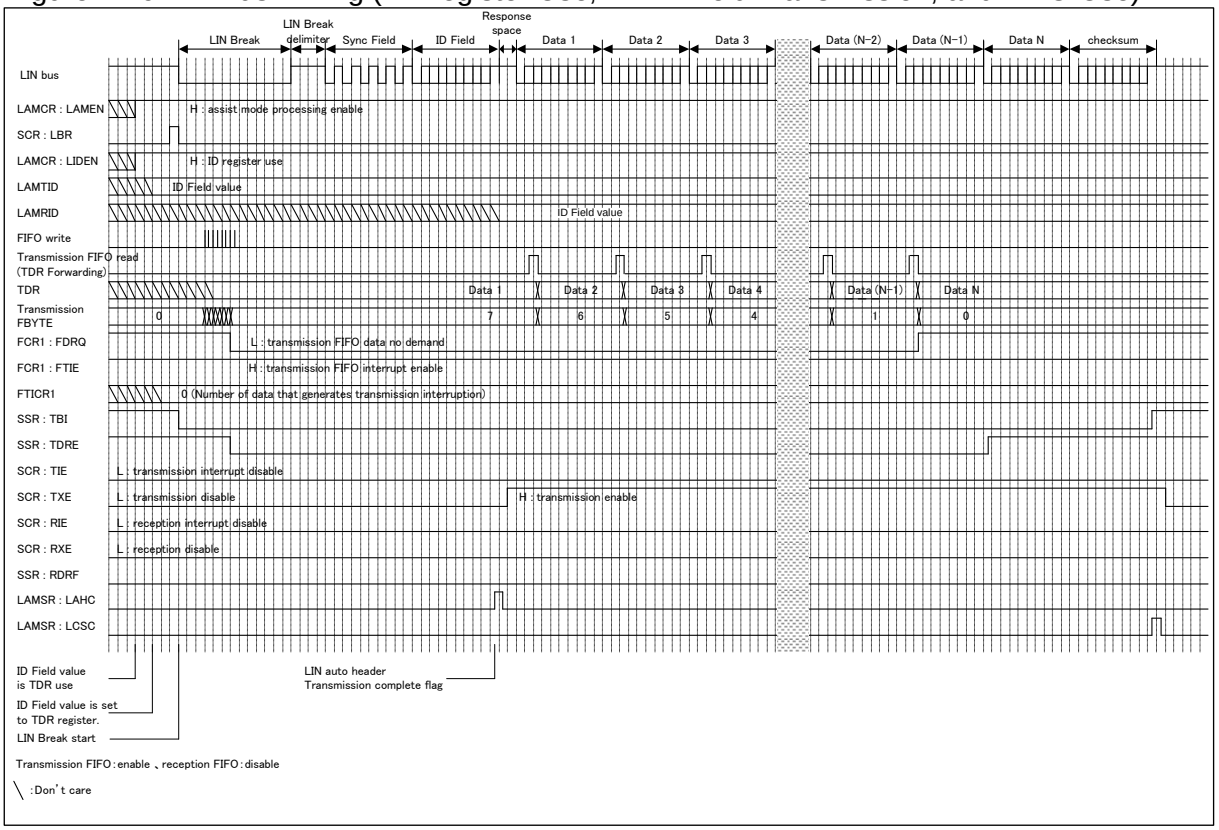


Figure 7-50 LIN Bus Timing (ID Register Unused, DATA Field Transmission, and FIFO Use)

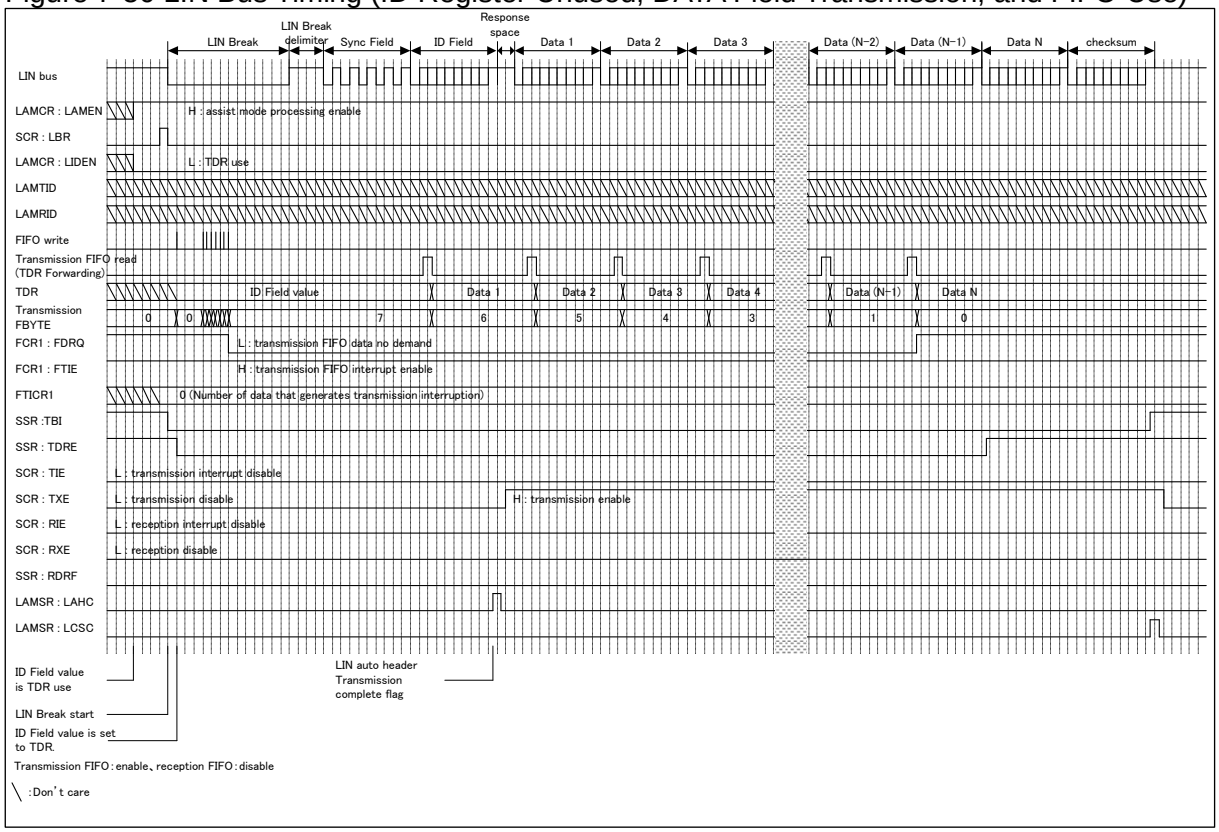


Figure 7-51 LIN Bus Timing (ID Register Use, DATA Field Reception, and FIFO Use)

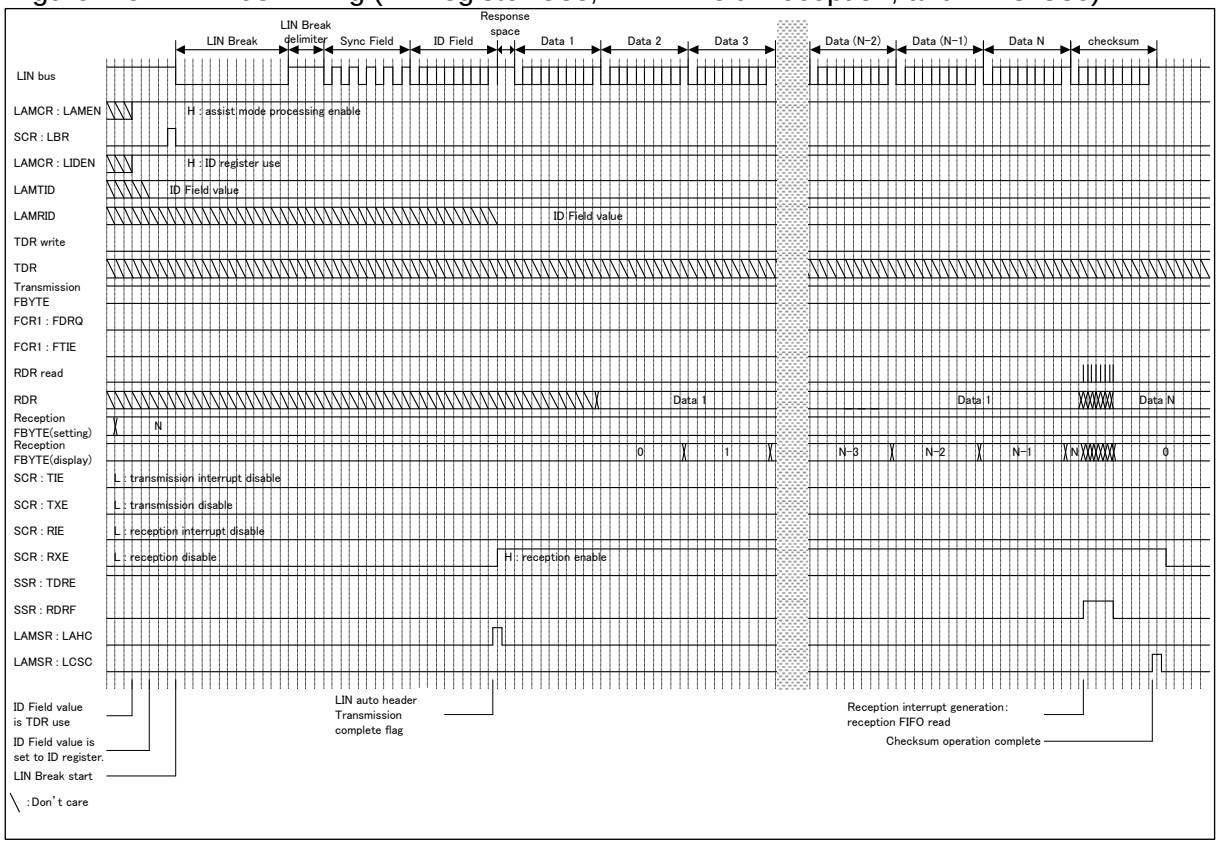
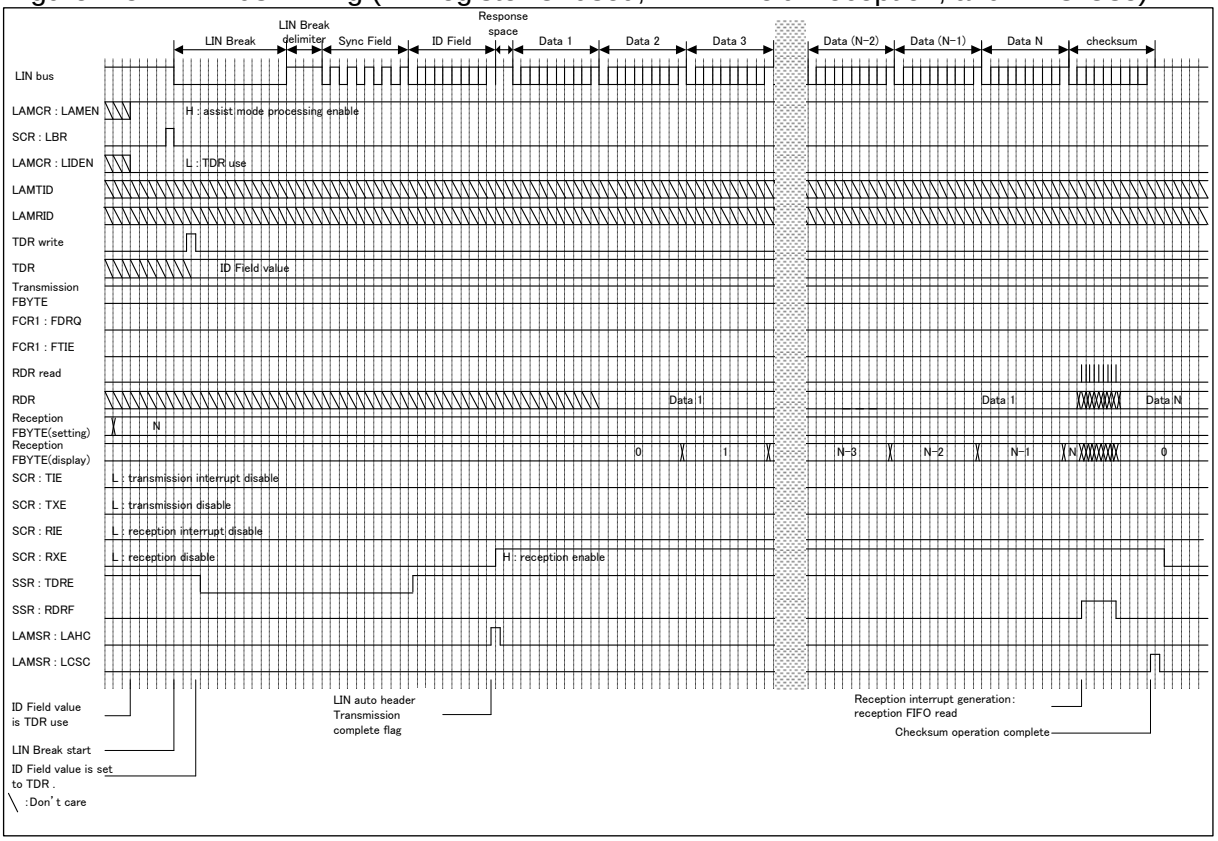


Figure 7-52 LIN Bus Timing (ID Register Unused, DATA Field Reception, and FIFO Use)



■ Slave operation

● Automatic header reception setting

Please set the following to receive an automatic header in the assist mode.

- Please set the SCR:MS bit to "1" to operate as a slave.
- Please set the LAMCR:LAMEN bit to "1" to operate as LIN assist mode.
- The ID Field value is written in LAMRID (LIN assist mode reception ID register) or RDR (receive data register). Please set the LAMCR:LIDEN bit to "1" when you write the value of ID Field in LAMRID. Please set the LAMCR:LIDEN bit to "0" when writing it in RDR.
- Please set the SACSR:AUTE bit to "1" when you do the baud rate adjustment automatically.
- Please set reception enable bit (SCR:RXE) to "1" (reception enable).

● From LIN Break Field reception to ID Field reception

1. When LIN Break Field is input, LIN Break Field is detected in the 11th bit (SSR:LBD=1). At this time, if the ESCR:LBIE bit is set to "1", the interrupt is generated. Please set the ESCR:LBIE bit to "0", and prohibit the status interrupt in the LIN assist mode.

Operation when the automatic baud rate is adjusted is shown below.

2. When LIN interface (v2.1) detects the first falling edge of Sync Field, the serial timer register (STMR) is initialized to 0.

3. When the fifth falling edge of Sync Field is detected, sync field detection flag (SACSR:SFD) is set to "1". At this time, if the SACSR:SFDE bit is set to "1", the status interrupt is generated. Please set the ESCR:SFDE bit to "0", and prohibit the status interrupt in the LIN assist mode.

4. When the fifth falling edge of Sync Field is detected, it operates according to the value of the serial timer register (STMR) as follows.

   - When the counter value of the STMR is within the defined range of SFLR and SFUR, the baud rate setting flag (SACSR:BST) is set to "1" and the BGR value adopts the STMR counter value..
   - When the value of the serial timer register (STMR) is smaller than the sync field lower limit register (SFLR) or larger than the sync field upper limit register (SFUR), the value of the baud rate generator register (BGR) will not be changed and the baud rate setting flag (SACSR:BST) will be reset to "0".

Figure 7-53 From LIN Break Field Reception to ID Field Reception (STMR is SFUR or smaller, and SFLR or larger).

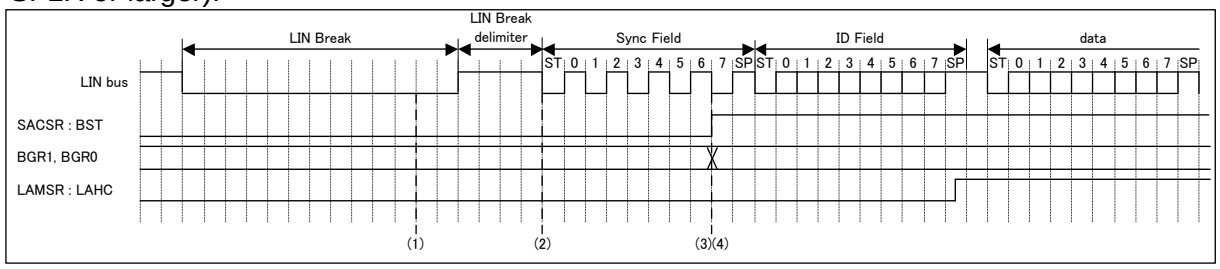
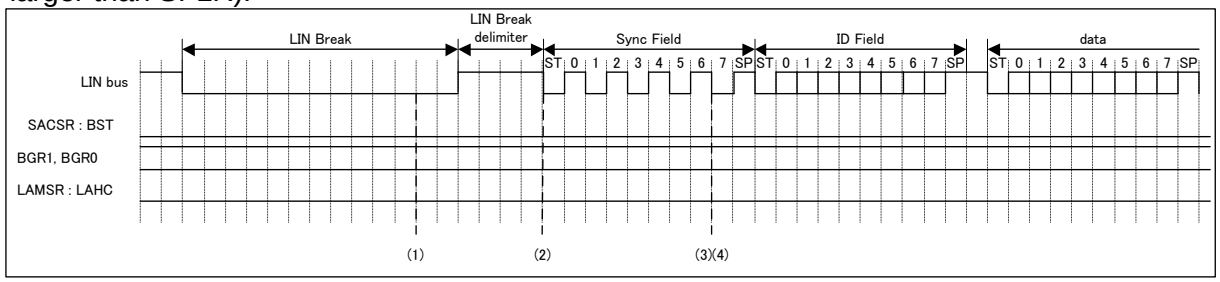


Figure 7-54 From LIN Break Field Reception to ID Field Reception (STMR is smaller than SFUR, and larger than SFLR).



5. When the automatic header reception in the LIN assist mode is completed, the LAMSR:LAHC bit is set to "1". Also when the LIN parity error is generated in ID Field, the LAMSR:LAHC bit becomes "1". Therefore, when the LAMSR:LAHC bit is set to "1", it is necessary to confirm the error has not been detected.

6. Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) when ID Field is normally received.

Figure 7-55 From LIN Break Field Reception to ID Field Reception (Parity Error is Generated).

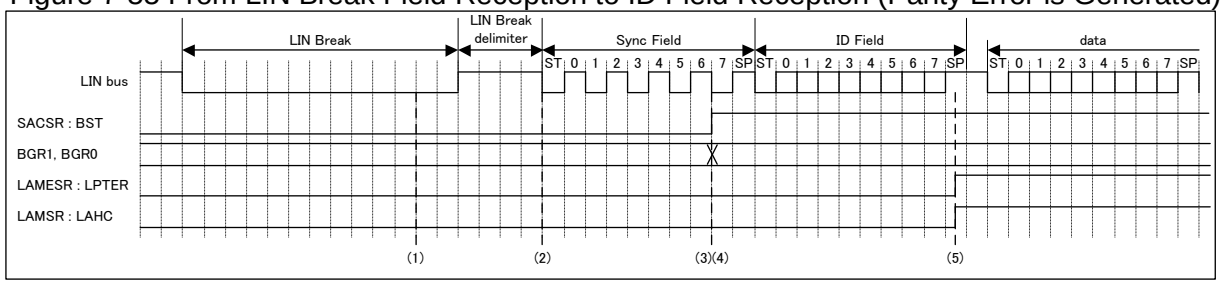
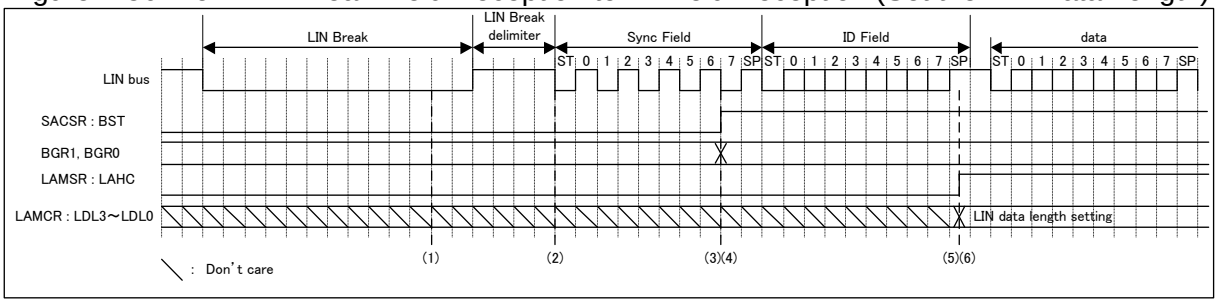


Figure 7-56 From LIN Break Field Reception to ID Field Reception (Set the LIN Data Length).



Notes:

- The setting of reception enable bit (SCR:RXE) and transmission enable bit (SCR:TXE) is disregarded during the header reception of the master in the assist mode.
- However, if the reception enable is set (SCR:RXE=1) when LIN Break Field is received, it is judged that the stop bit is "L" level before LIN Break is detected and detects the framing error. Therefore, please set to reception prohibition setting (SCR:RXE=0) when the header is transmitted.
- The Sync Field value when the assist mode operates cannot be stored in the RDR register.

● From ID Field reception to DATA Field transmission/reception

Whether DATA Field is transmitted to or received in the master can be selected after ID Field reception.

(When DATA Field is transmitted)

- Please write data in transmission data register (TDR) after reception ID Field. At this time, please set to transmission interrupt enable (SCR:TIE=1).
- Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) from the value of reception ID Field.
- Checksum is operated based on LIN data length setting bit (LAMCR:LDL2 to LDL0), and after final data is transmitted, checksum is transmitted automatically.
- The arithmetic operations of checksum can select the arithmetic operations method by LIN checksum type selection bit (LAMCR:LCSTYP).
- When the arithmetic operations of checksum is completed, checksum arithmetic operations completion flag (LAMCR:LCSC) is set. At this time, when the checksum arithmetic operations completion interrupt enable bit is set (LAMIER:LCSCIE=1), the status interrupt is generated.
- After the response transmission is completed (LAMSR:LCSC=1), transmission prohibition setting (SCR:TXE=0) is done.

Notes:

- The response transmission data (Data Field and checksum) when the assist mode operates cannot be stored in the RDR register.
- Please write the dummy value ("don 't care") in the TDR register to operate checksum automatically and transmit it when the LIN data length is set to 0 byte length (LAMCR:LDL3-0="0000") in response transmission. The TDR setting value at this time doesn't influence the checksum arithmetic operations.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes an inverted ID Field.

Figure 7-57 From ID Field Reception to DATA Field Transmission (ID Register Use).

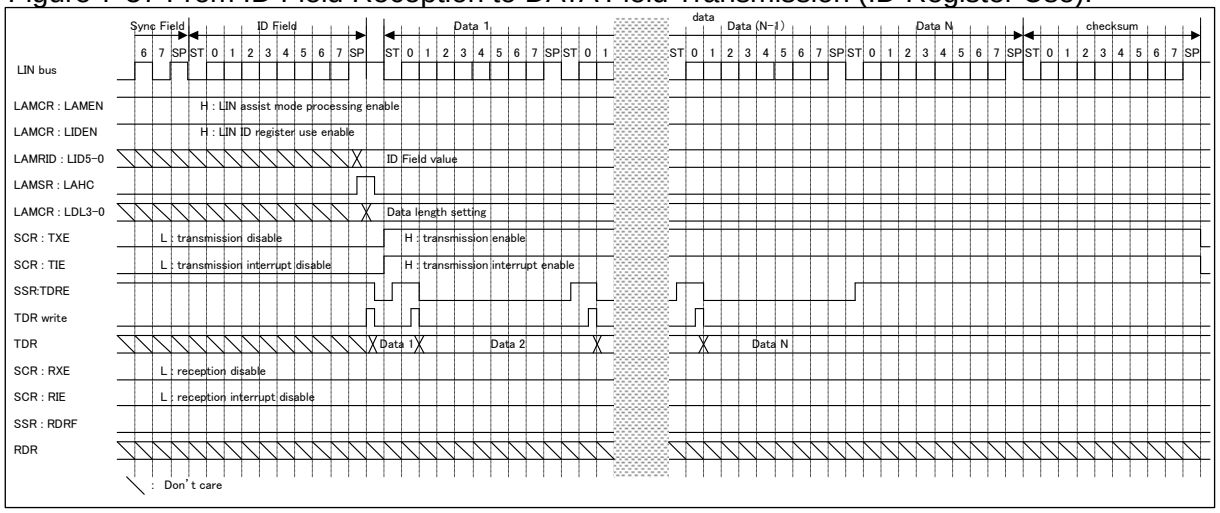
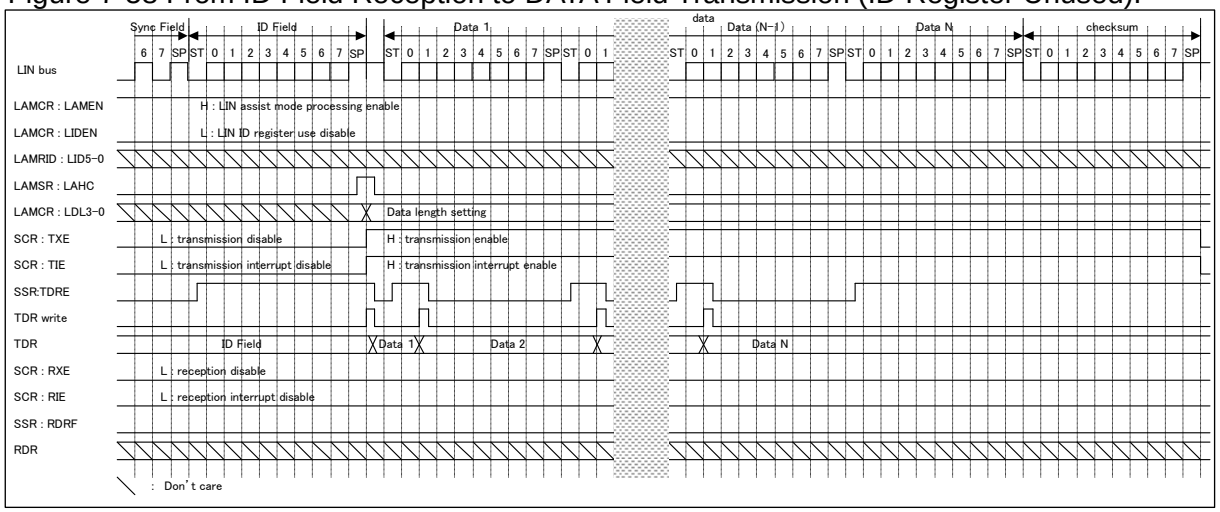


Figure 7-58 From ID Field Reception to DATA Field Transmission (ID Register Unused).



(When receive DATA Field)

- Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) from the value of reception ID Field.
- Please set the reception enable (SCR:RXE=1).
- SSR:RDRF is set to "1" at each DATA Field reception. At this time, if reception interrupt enable (SCR:RDIE=1) is done, the reception interrupt is generated.
- Detection of the start bit is as follows; the falling edge is detected after passing through the noise filter (which samples serial data input in 3 bus clock and decides the value by majority), and the data "L" is detected after the noise filter at the sampling point.
- Checksum is operated based on LIN data length setting bit (LAMCR:LDL3 to LDL0), and the normality of the reception checksum is confirmed automatically. The arithmetic operations result of checksum can be confirmed by LIN checksum error flag bit (LAMESR:LCSER). When LCSER is "1", the checksum error is detected. At this time, when LIN checksum error interrupt enable bit (LAMIER:LCSERIE) is "1", the interrupt is generated.
- When the checksum arithmetic operations is completed, the checksum arithmetic operations completion flag bit (LAMSR:LCSC) becomes "1". At this time, when LIN checksum arithmetic operations completion interrupt enable bit (LAMIER:LCSCIE) is "1", the interrupt is generated.
- After the checksum reception is completed (LAMSR:LCSC=1), reception prohibition setting (SCR:RXE=0) is done.

Figure 7-59 From ID Field Reception to DATA Field Reception (ID Register Use).

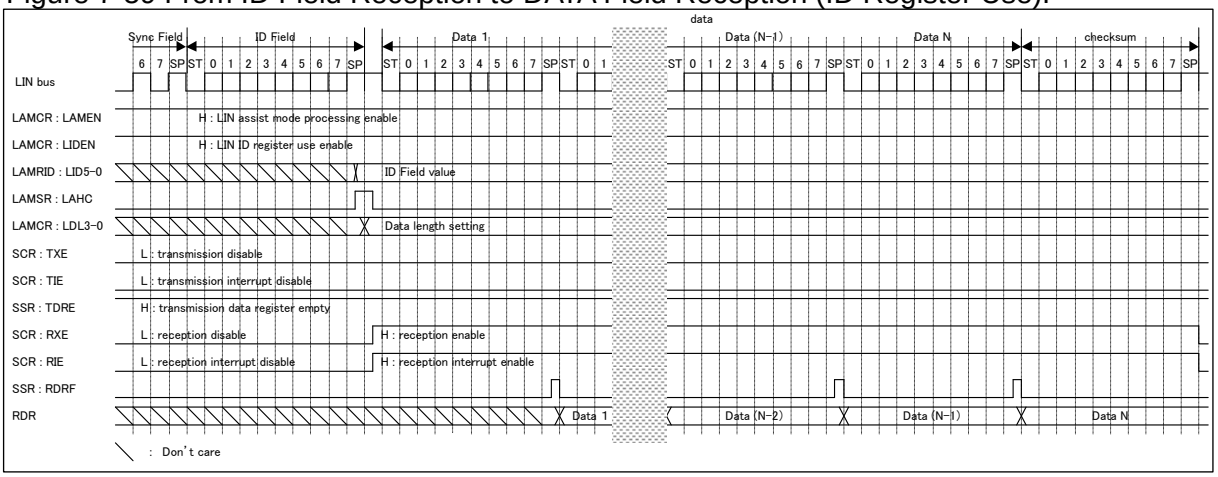
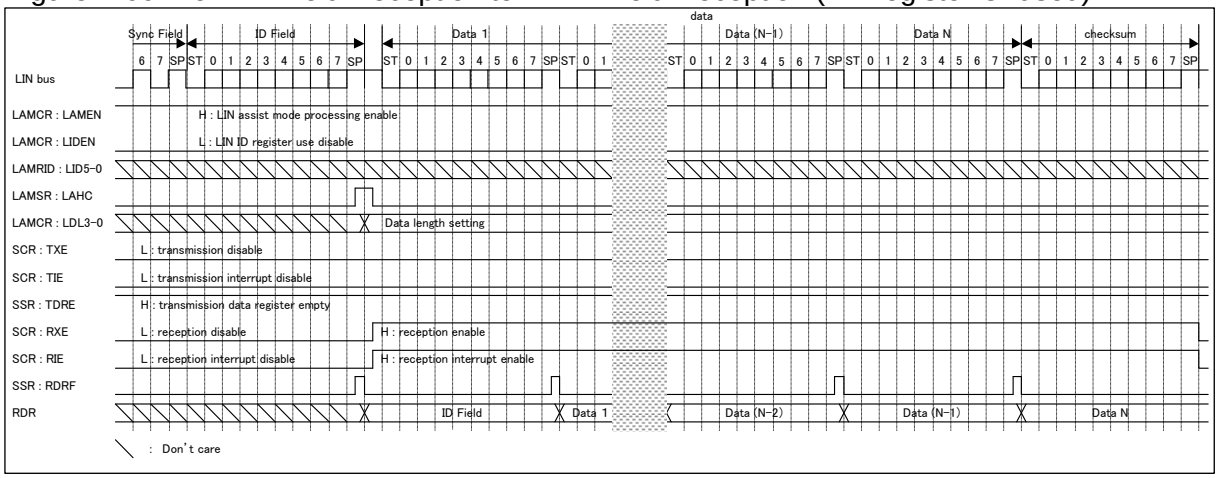


Figure 7-60 From ID Field Reception to DATA Field Reception (ID Register Unused).



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.
- The checksum value of the response reception is not stored in the RDR register when the assist mode operates.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes reversing ID Field.

LIN Break Field reception processing in assist mode

After LIN Break Field is detected (SSR:LBD=1), the following procedures are needed for the LIN frame of retransmitted LIN Break Field when the assist mode is processed (SSR:RDRF=1 or SCR:TXE=1 or SSR:TBI=0).

- First of all, reception prohibition setting (SCR:RXE=0) and transmission prohibition setting (SCR:TXE=0) are done.
- Cancels the reception data before it is received again.
  - When reception FIFO is used, the reception FIFO is reset (FCR0:FCL1=1 or FCR0:FCL2=1) after reception FIFO operation is prohibited (FCR0:FE1=0 or FCR0:FE2=0).
  - Then, to clear the reception data register, the RDR register is read.
- Cancels the transmission data before it is received again.
  - When transmission FIFO is used, the transmission FIFO is reset (FCR0:FCL1=1 or FCR0:FCL2=1) after transmission FIFO operation is prohibited (FCR0:FE1=0 or FCR0:FE2=0).
  - Then, the transmission data register clear is executed (LAMCR:LTDRCL=1), and the state is made the transmission bus idle.
- When the automatic header reception in the LIN assist mode completes, the LAMSR:LAHC bit is set to "1". Moreover, please confirm neither the LIN parity error nor the framing error has been detected with ID Field.
- Please set the LIN data length setting bit (LAMCR:LDL3-0) when ID Field is normally received.
- The following processing is equal to that of the preceding section "From ID Field Reception to Data Field Transmission/Reception".

Notes:

- In LIN Break Field received, the framing error is detected before LIN Break Field is detected when reception enable setting (SCR:RXE=1) is done. However, it operates normally without stopping the header reception.
- In the assist mode (LAMCR:LAMEN), the framing error is detected at "L" level of the tenth bit of new LIN Break Field regardless of reception prohibition setting (SCR:RXE=0) when new LIN Break is transmitted continuously from the master between from the detection of LIN Break Field to the ID Field reception completion. However, it operates normally without stopping the header reception.

- **Slave operation timing chart**

Figure 7-61 LIN Bus Timing (DATA Field Transmission: FIFO Unused, AUTE=1 and ID Register Use).

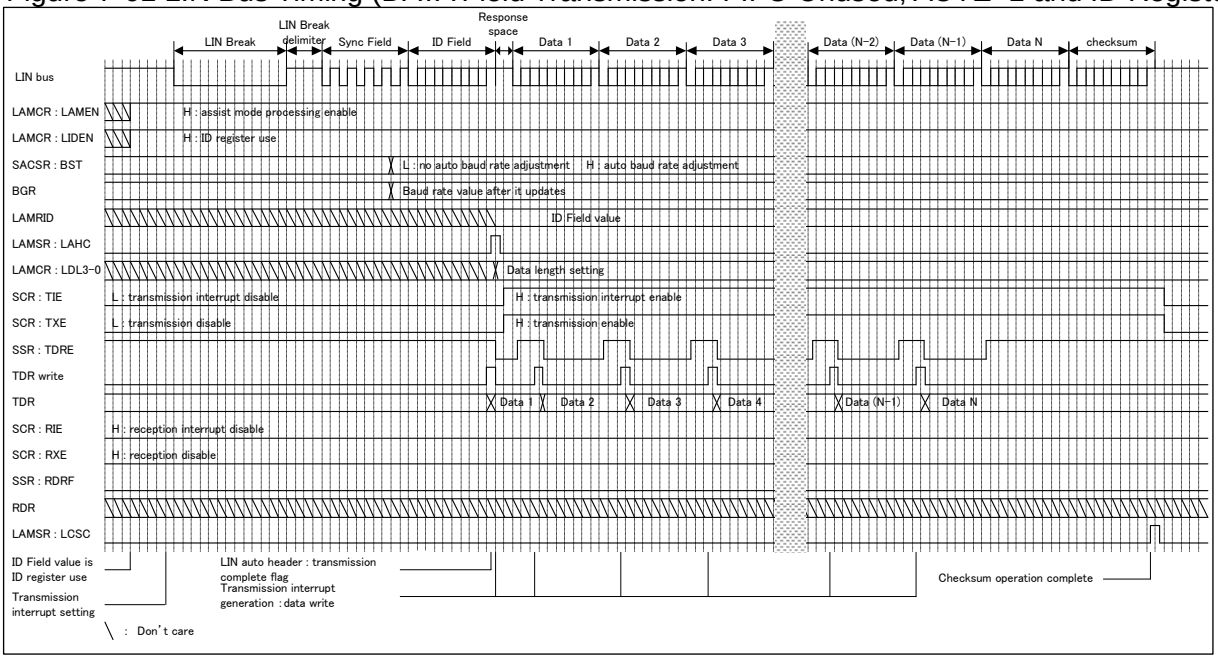


Figure 7-62 LIN Bus Timing (DATA Field Transmission: FIFO Unused, AUTE=1 and ID Register Unused).

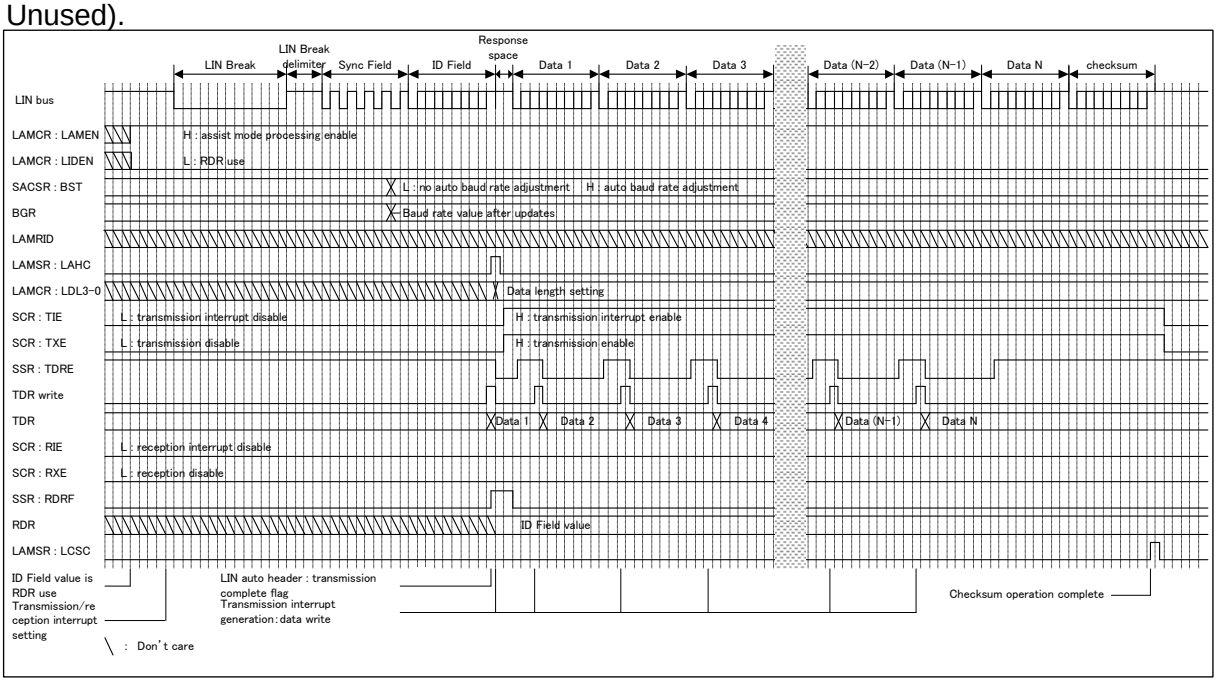


Figure 7-63 LIN Bus Timing (DATA Field Reception: FIFO Unused, AUTE=1 and ID Register Use).

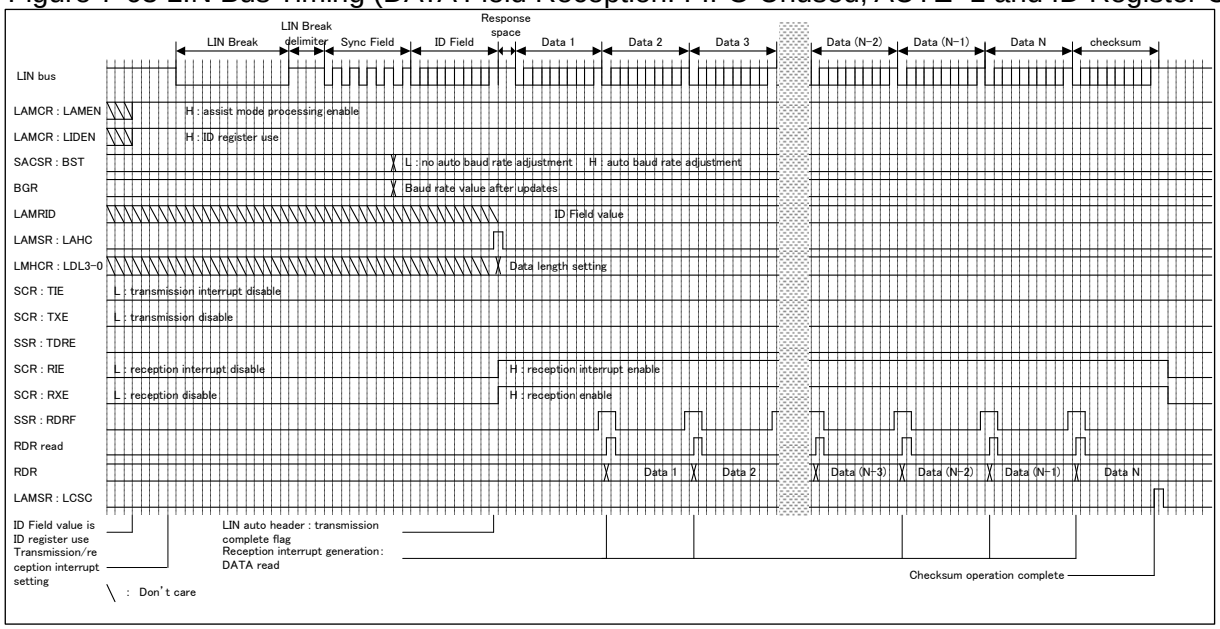


Figure 7-64 LIN Bus Timing (DATA Field Reception: FIFO Unused, AUTE=1 and ID Register Unused).

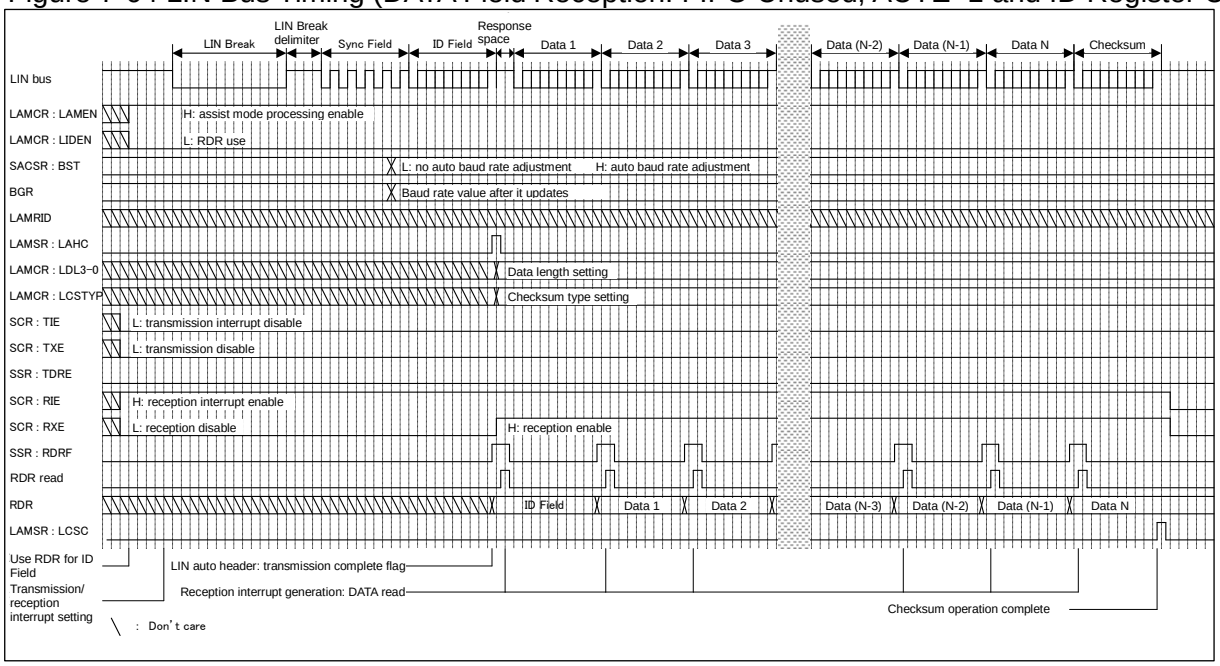


Figure 7-65 LIN Bus Timing (DATA Field Transmission: FIFO Use, AUTE=1 and ID Register Use).

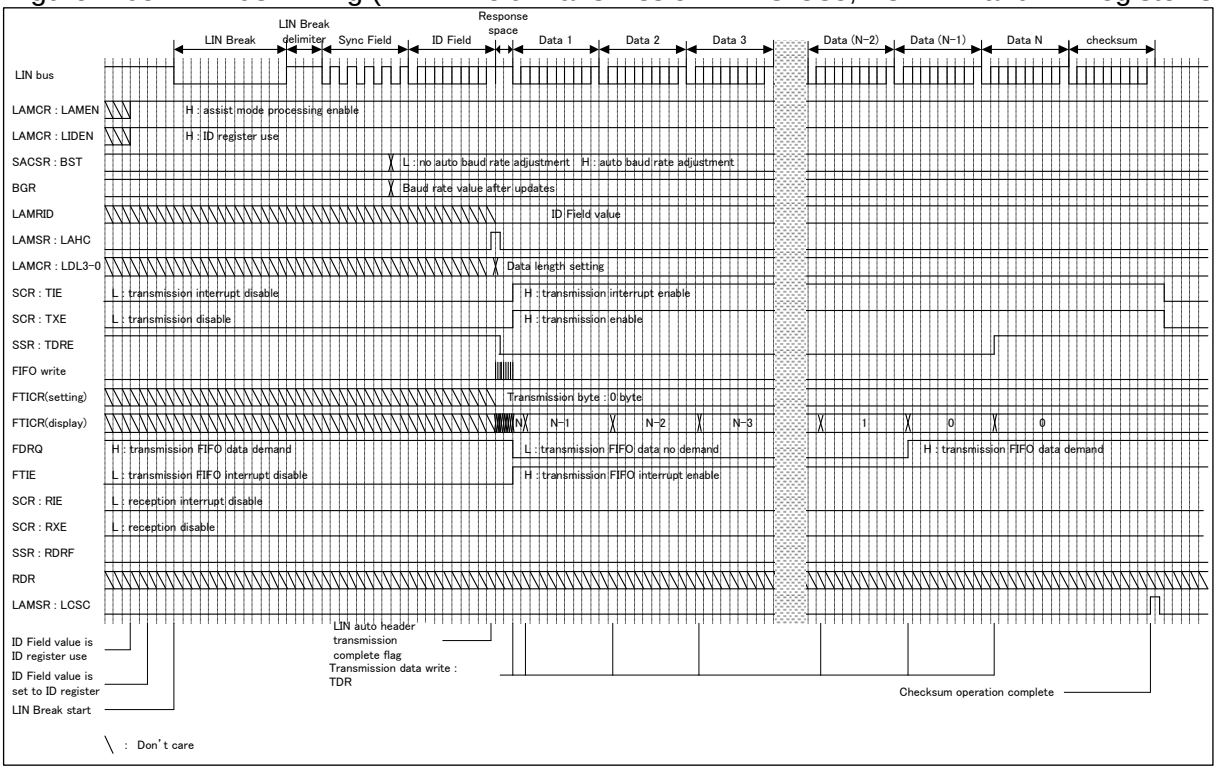


Figure 7-66 LIN Bus Timing (DATA Field Transmission: FIFO Use, AUTE=1 and ID Register Unused).

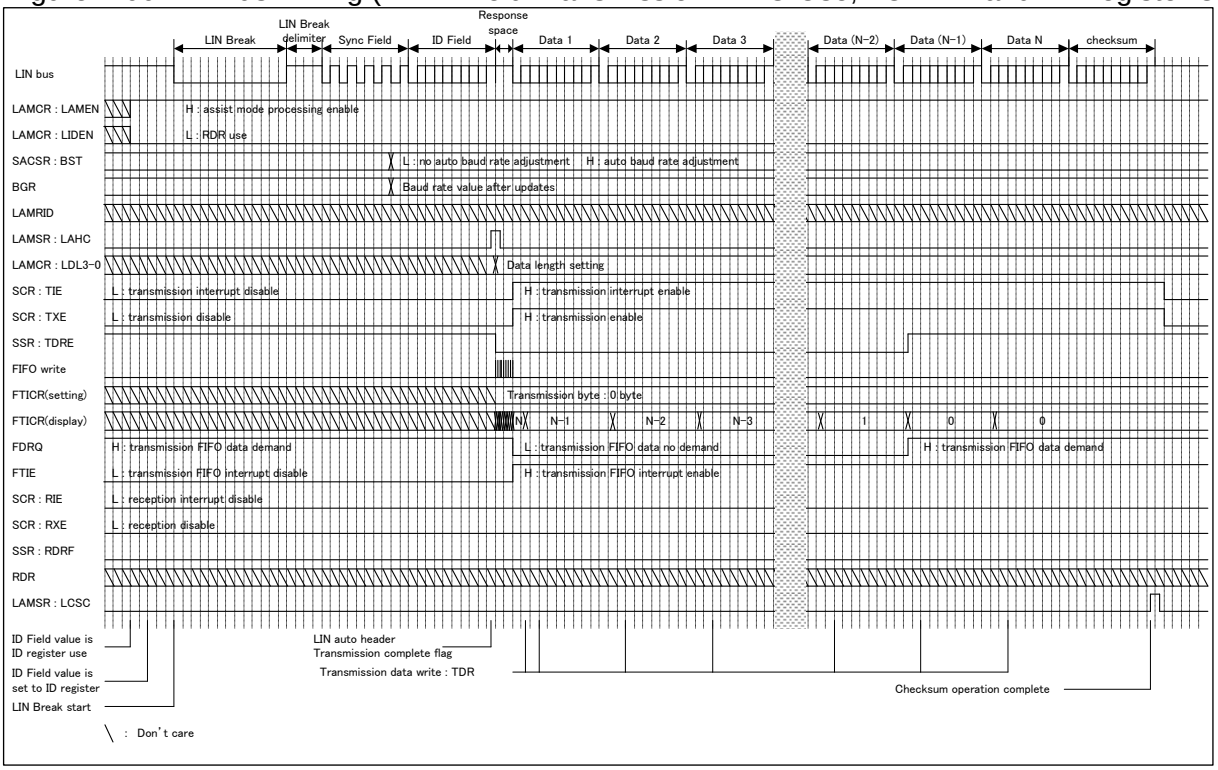


Figure 7-67 LIN Bus Timing (DATA Field Reception: FIFO Use, AUTE=1 and ID Register Use).

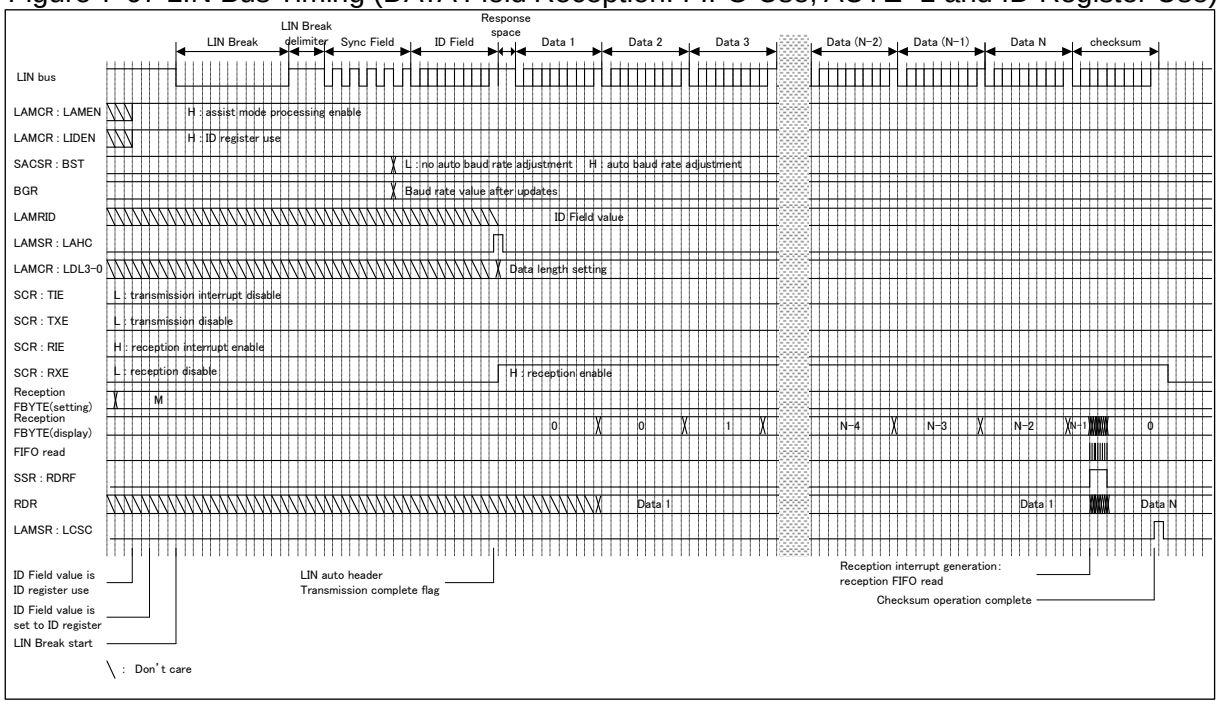
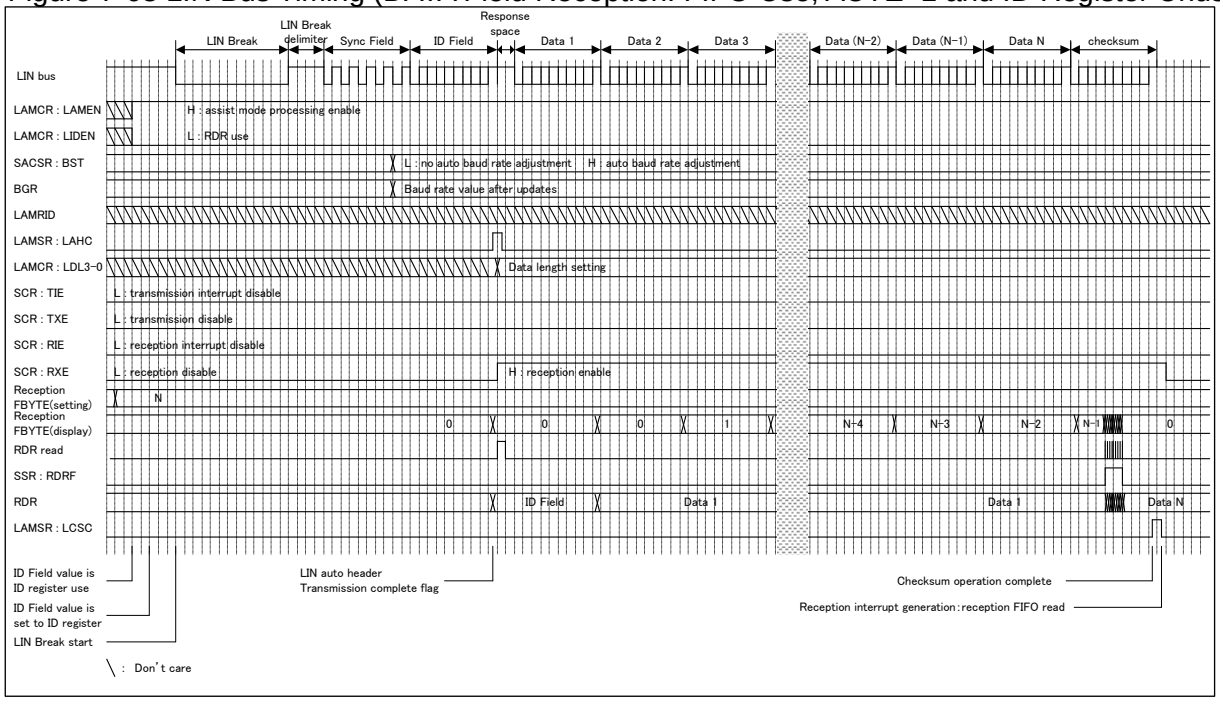


Figure 7-68 LIN Bus Timing (DATA Field Reception: FIFO Use, AUTE=1 and ID Register Unused).



7.5.3. LIN Baud Rate Selection/Setting

This section explains the LIN baud rate selection/setting.

The LIN can use:

- Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the internal clock
- Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the external clock

The setting method is the same as the method used in the case of UART (mode 0/1). See "5.2.12 UART Baud Rate Selection/Setting".

7.6. Setup Procedure and Program Flow

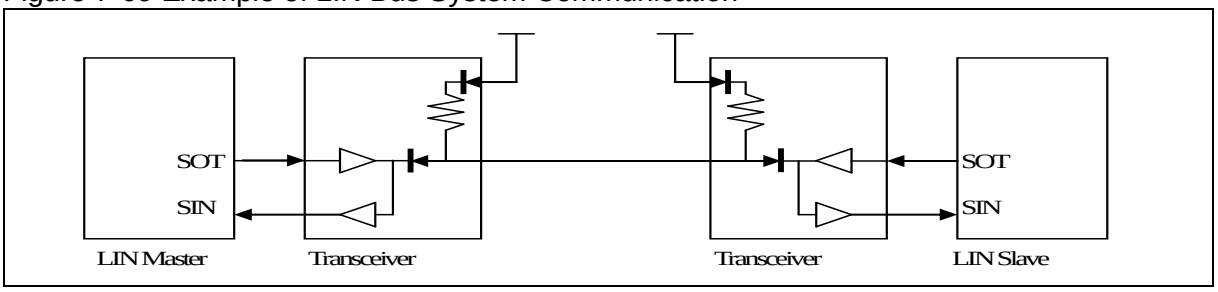
Setup procedure and program flow are shown.

In operation mode 3 (LIN communication mode), the selected baud rate can be used for the LIN master system or LIN slave system.

■ CPU Interconnection

The following Figure shows a communication system that contains one LIN master and one LIN slave. The multi-function serial interface can work as an LIN master or LIN slave.

Figure 7-69 Example of LIN Bus System Communication



7.6.1. Manual mode

This section explains the manual mode.

The example of the flowchart of the master side and the slave side in the manual mode is shown.

■ Flowchart Example

● Master operation

Figure 7-70 Example of a Flowchart in LIN Communication Master Mode (without Using FIFO)

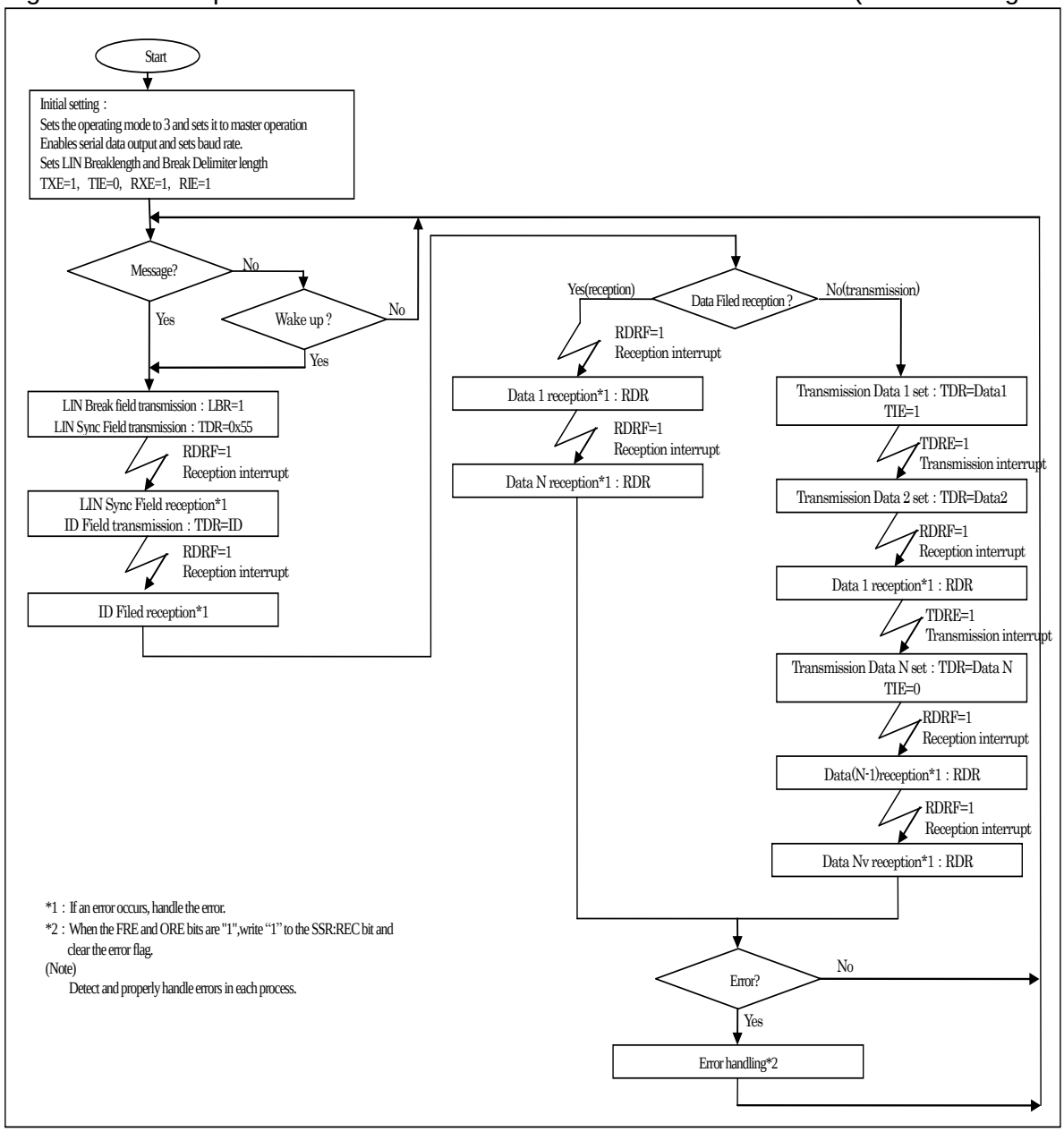
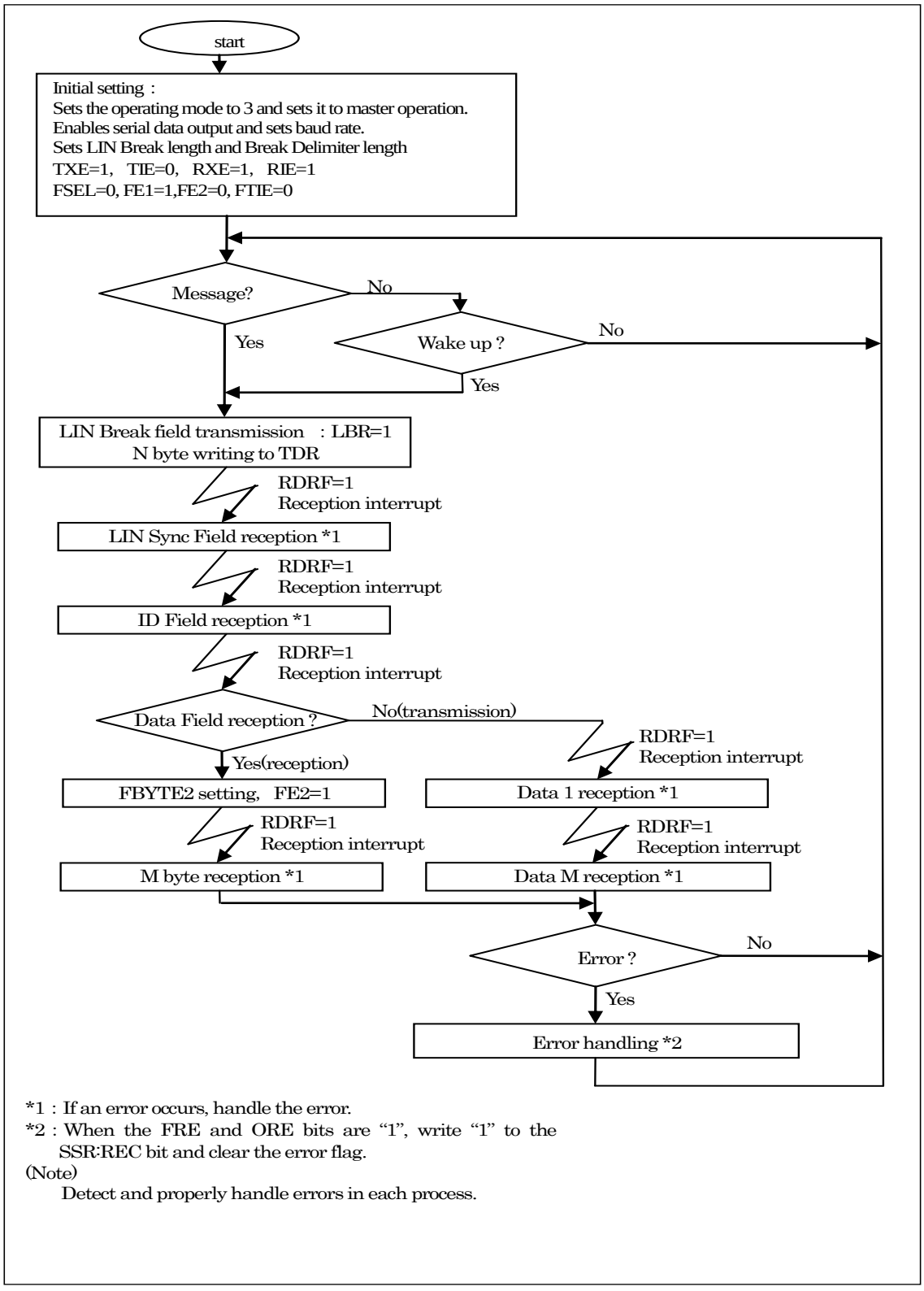


Figure 7-71 Example of a Flowchart in LIN Communication Master Mode (Using FIFO)



*1 : If an error occurs, handle the error.
*2 : When the FRE and ORE bits are "1", write "1" to the SSR:REC bit and clear the error flag.
(Note)
Detect and properly handle errors in each process.

● Slave operation

Figure 7-72 Example of a Flowchart in LIN Communication Slave Mode
(Automatic Baud Rate can be adjusted (SACSR:AUTE=1), without Using FIFO)

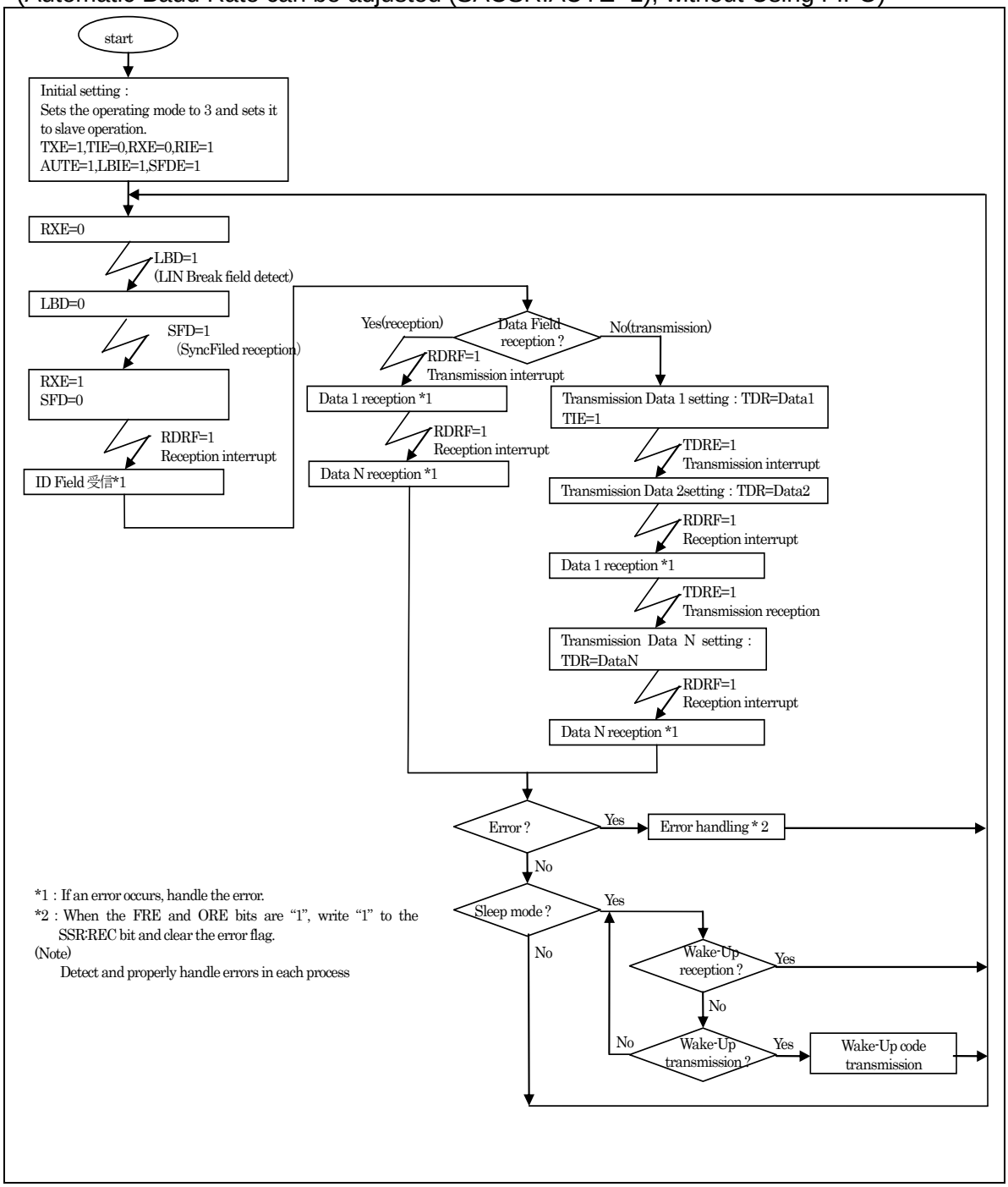
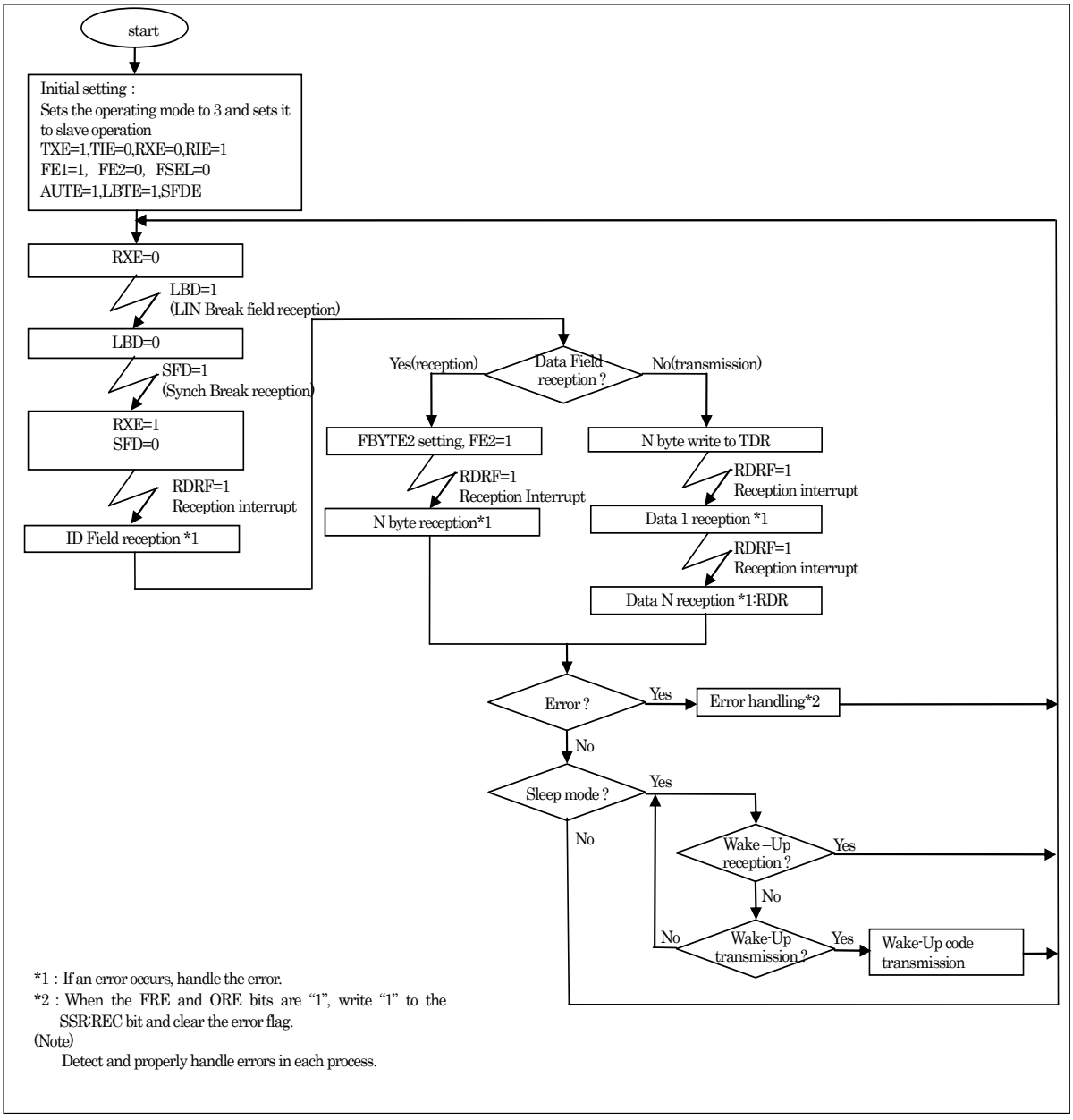


Figure 7-73 Example of a Flowchart in LIN Communication Slave Mode
(Automatic Baud Rate can be adjusted (SACSR:AUTE=1), Using FIFO)



7.6.2. Assist mode

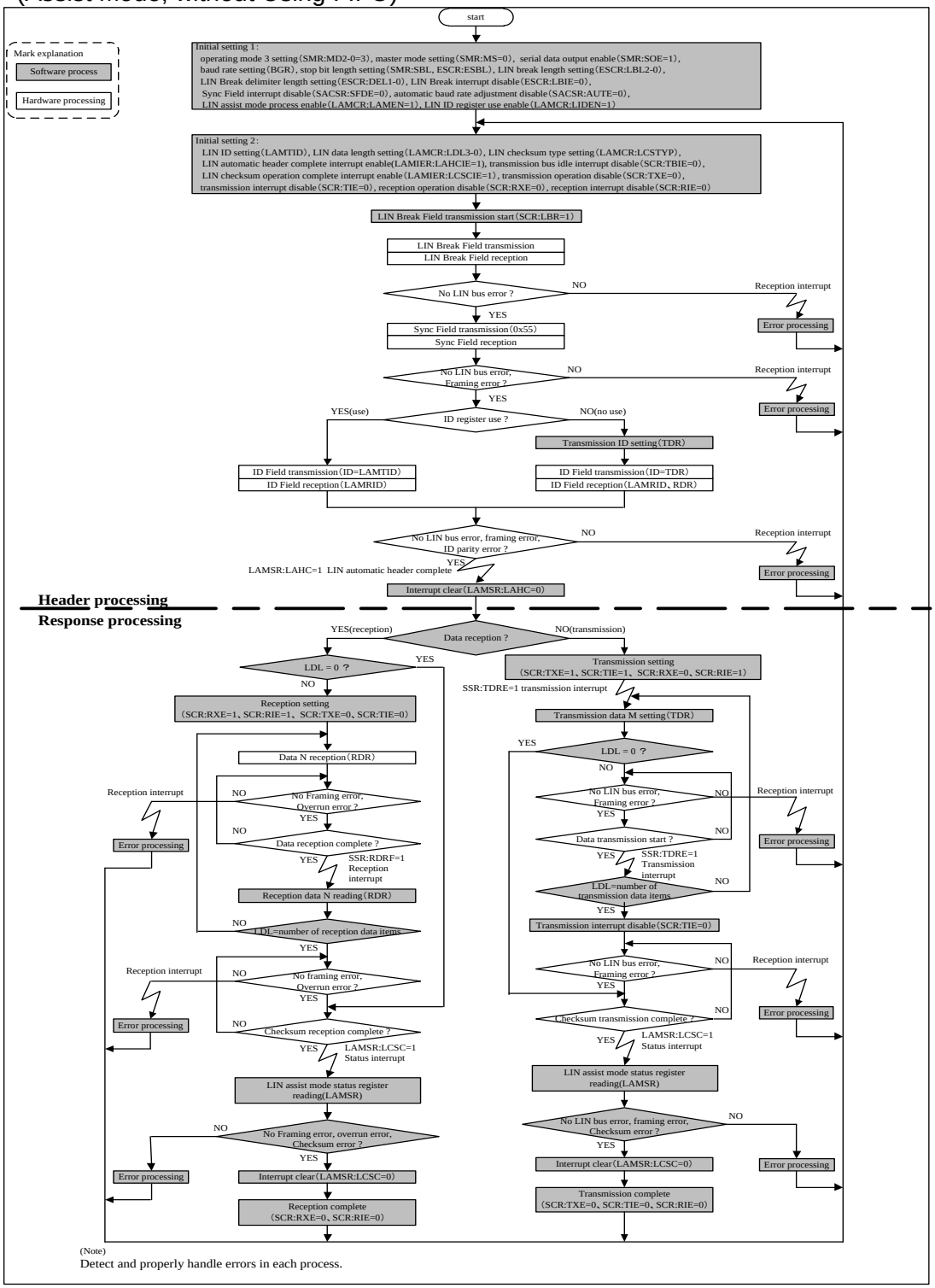
This section explains the assist mode.

The example of the flow chart of the master side and the slave side in the manual mode is shown.

■ Flowchart Example

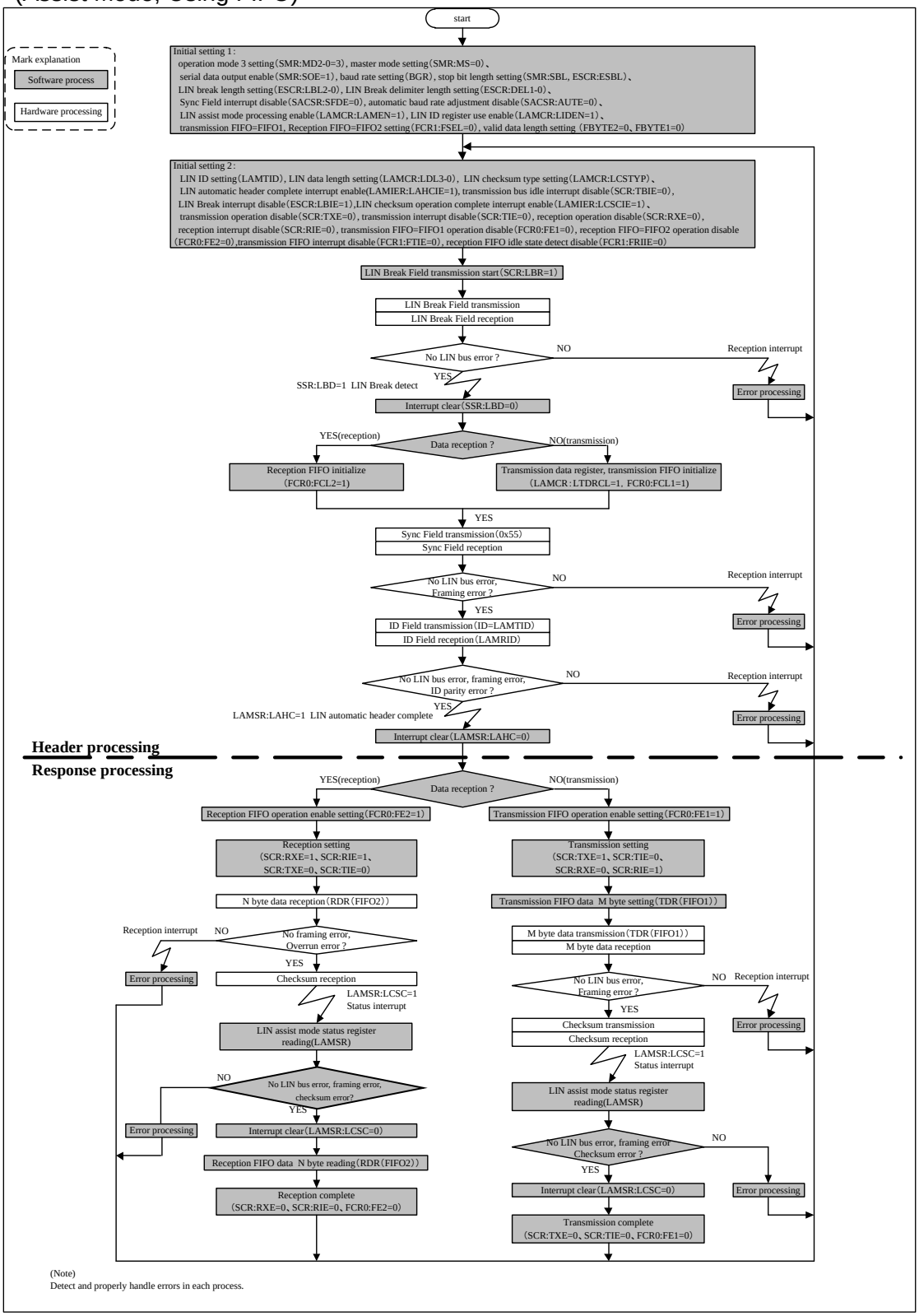
● Master operation

Figure 7-74 Example of a Flowchart in LIN Communication Master Mode (Assist Mode, without Using FIFO)



(Note)
Detect and properly handle errors in each process.

Figure 7-75 Example of a Flowchart in LIN Communication Master Mode (Assist Mode, Using FIFO)



(Note)
Detect and properly handle errors in each process.

● Slave operation

Figure 7-76 Example of a Flowchart in LIN Communication Slave Mode (Assist Mode, without Using FIFO)

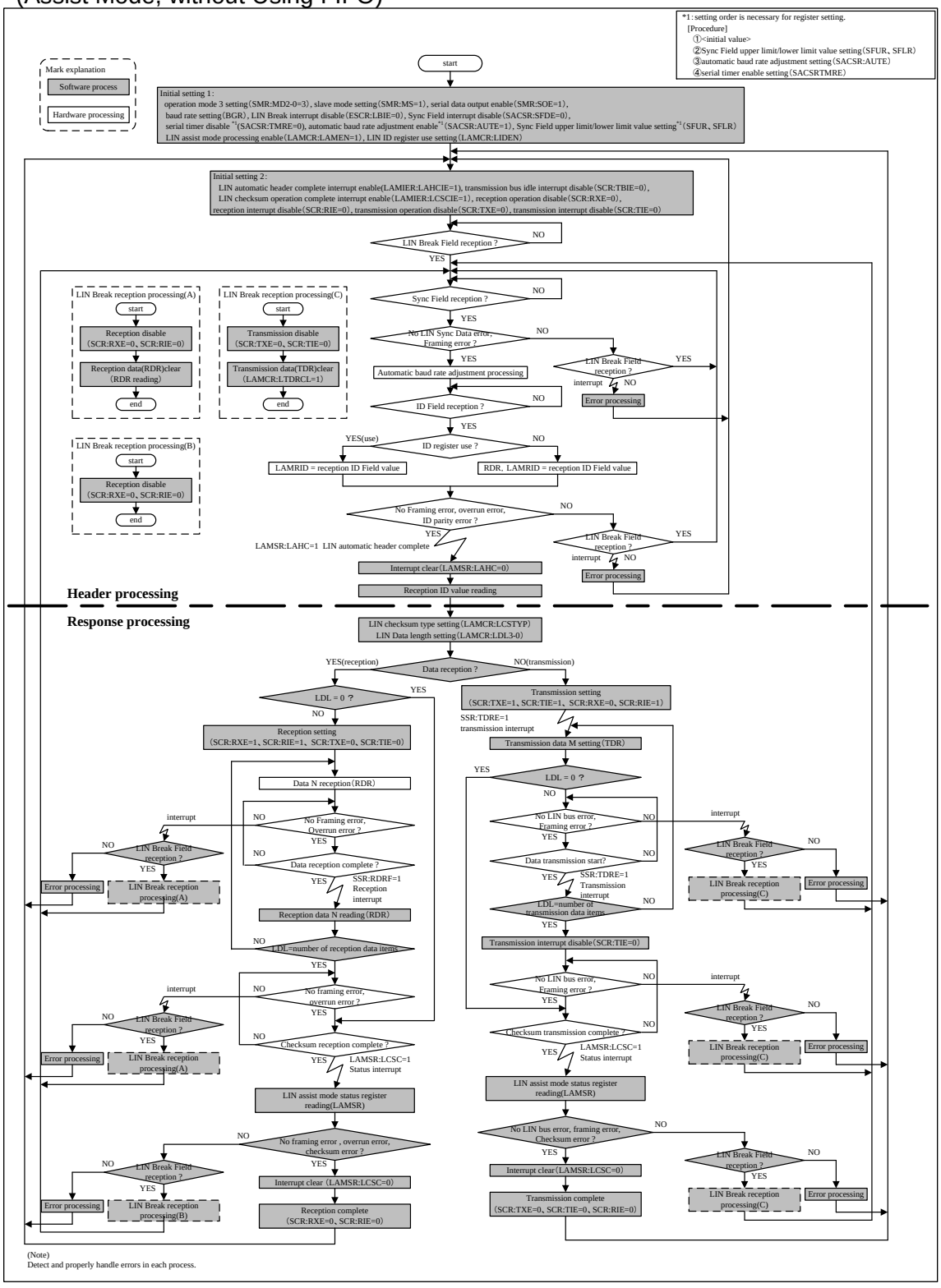
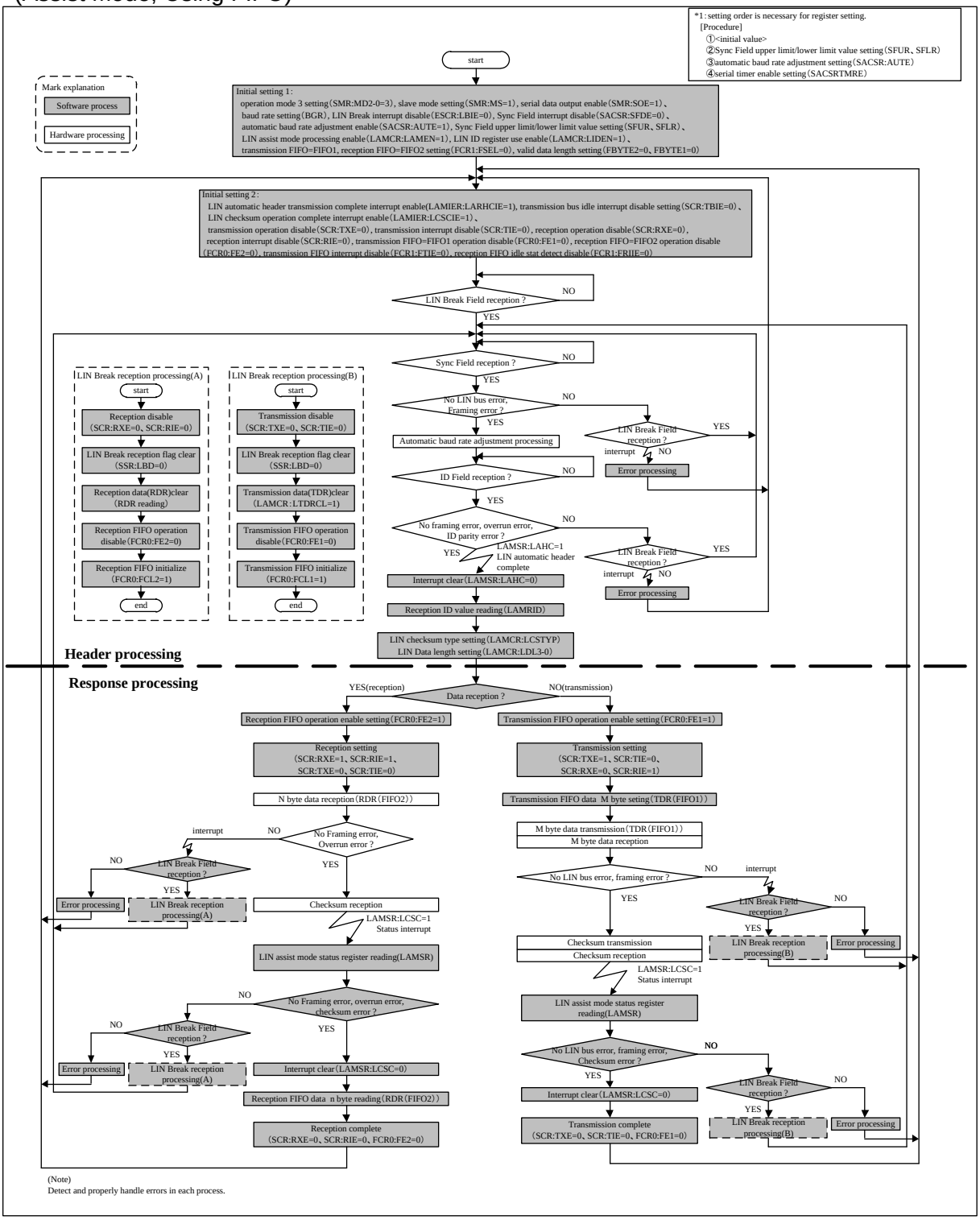


Figure 7-77 Example of a Flowchart in LIN Communication Slave Mode (Assist Mode, Using FIFO)



8. Operation of I²C

This section is explains operation of I²C.

8.1 Interrupts of I²C

8.2 Operation for I²C Interface Communication

8.3 I²C Master Mode

8.4 I²C Slave Mode

8.5 Example of I²C Flowchart

8.1. Interrupts of I²C

Interrupts of I²C is shown.

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

8.1.1. List of Interrupts of I²C Interface

This section explains the list of interrupts of the I²C interface.

The following table indicates how I²C interface interrupt control bits relate to interrupt factors.

Table 8-1 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	IBCR:INTE	Writing "0" to the interrupt flag bit (IBCR:INT)
			After data transmission and reception *1		
			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
	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to timer interrupt flag bit (SACSR:TINT)
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)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Transmis-sion	TDRE	SSR	Transmission register is empty Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)	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 valid data (retransmission) *2
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit or the transmission FIFO is full
	TBI (SSR: DMA=1)	SSR	No transmission operation Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)	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

*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 generate the IBCR:INT flag when data is transmitted or received, the SSR:TDRE bit must be "1" before the IBCR:INT flag is set.

*2: Set the SMR:TIE bit to "1" after the SSR:TDRE bit is cleared to "0".

*3: Set the SSR:TBIE bit to "1" after the SSR:TBI bit is cleared to "0".

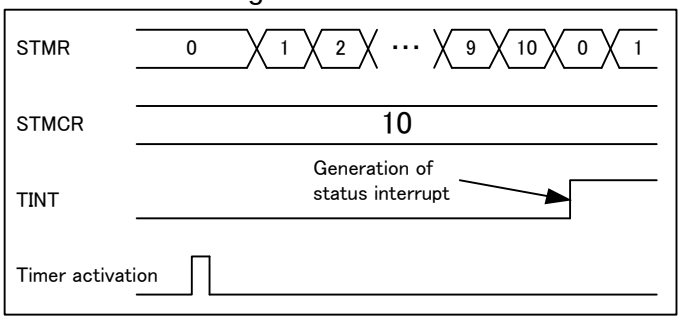
8.1.2. Timing of Timer Interrupt Generation and Flag Setting

This section explains the timing of the timer interrupt generation and flag setting.

Timer interrupt is generated when Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR).

- When Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR), "1" will be set to timer interrupt flag (SACSR:TINT).
  At this time when the timer interrupt is enabled (SACSR:TINTE="1"), a status interrupt will be generated.

Figure 8-1 Timer Interrupt Generation Timing



8.2. Operation for I²C Interface Communication

Operation for I²C interface communication is shown.

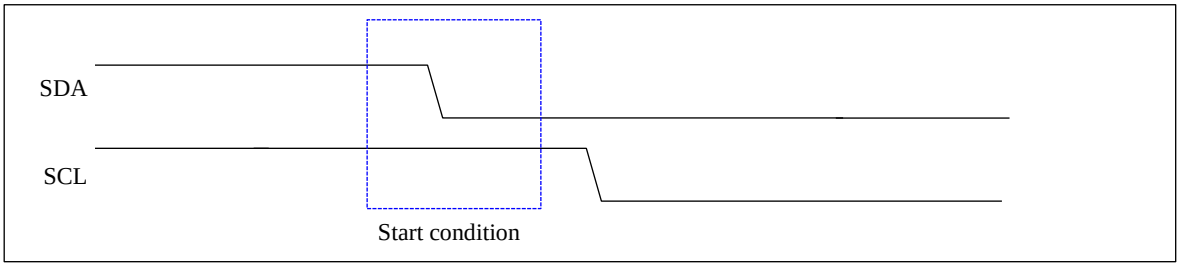
The I²C interface handles communication using two bidirectional bus lines, a serial data line (SDA), and a serial clock line (SCL).

8.2.1. I²C Bus Start Condition

This section explains the I²C bus start condition.

The condition for the I²C bus to be activated is as follows:

Figure 8-2 Start Condition

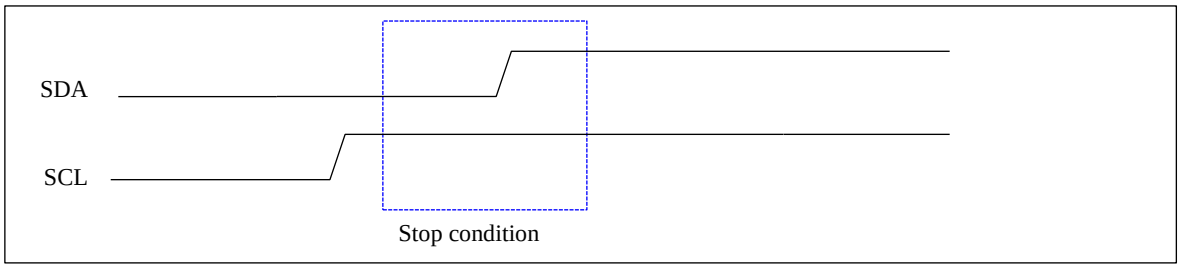


8.2.2. I²C Bus Stop Condition

This section explains the I²C bus stop condition.

The condition for the I²C bus to stop is as follows:

Figure 8-3 Stop Condition

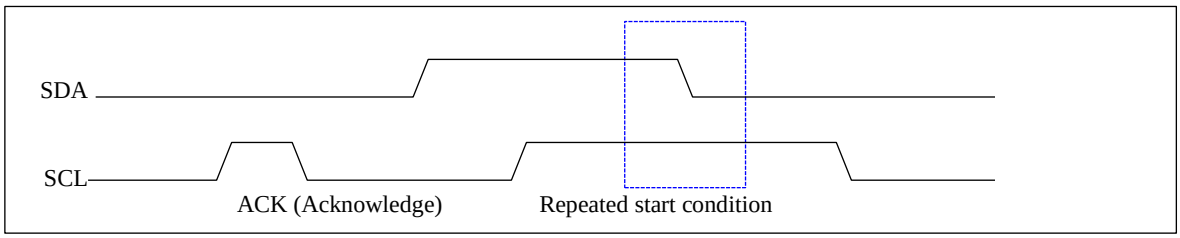


8.2.3. I²C Bus Repeated Start Condition

This section explains the I²C bus repeated start condition.

The condition for the I²C bus to initiate a repeated start is as follows:

Figure 8-4 Repeated Start Condition



8.2.4. I²C Bus Error

This section explains the I²C bus error.

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".

8.2.5. Serial Timer Operations

This section explains the serial timer operations.

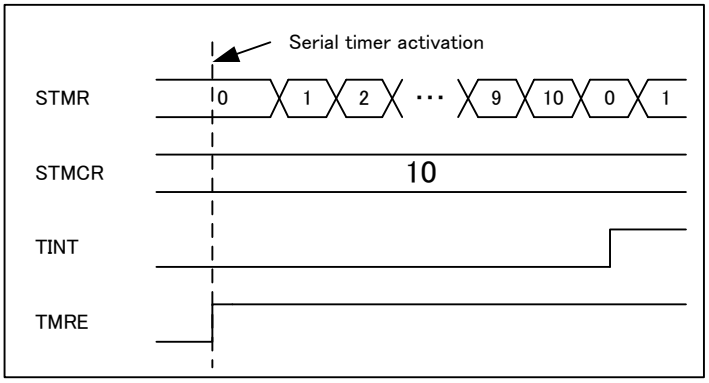
Serial timer can be used as a timer feature.

● Activating Serial Timer

Activating a serial timer: Setting "1" to the serial timer enable bit (SACSR:TMRE).

- Activating with serial timer enable bit (SACSR:TMRE)
  When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer will activate the serial timer and the serial timer register (STMR) will start counting from "0".

Figure 8-5 Activating with Serial Timer Enable Bit (STMCR="10")



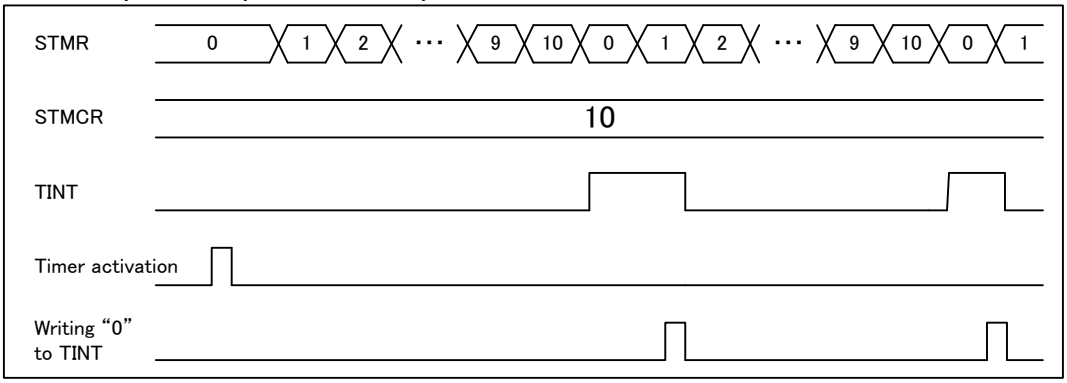
● Stopping Serial timer

When serial timer enable bit (SACSR:TMRE) is set to "0", the serial timer will be stopped. In this case, the values set in the serial timer register (STMR) will be retained.

● Timer Operation

When serial timer register (STMR) matched serial timer comparison register (STMCR), timer interrupt flag (SACSR:TINT) will be set to "1" and serial timer register (STMR) will be reset to "0".

Figure 8-6 Timer Operation (STMCR="10")



8.2.6. Baud Rate Generation

This section explains the baud rate generation.

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:

(1) Reload value

$V = \phi / b - 1$

V: Reload value b: Baud rate φ: Bus clock frequency, external clock 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.

(2) Example of calculation

The reload value is as follows if the bus clock is 16MHz and the baud rate is to be 400kbps:

Reload value:

V = (16 × 1,000,000) / 400,000 - 1 = 39

Therefore, the baud rate is

b = (16 × 1,000,000) / (38+2) = 400kbps

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- Configure the baud rate generator register when the ISMK:EN bit is "0".
- Bus clock should be set with 8MHz or more in operating mode 4 (I^2C mode) and baud rate generator configured in 400kbps 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 8-2 Reload Values Relating to Baud Rates and Internal Clock Frequencies

Baud rate [bps]	Internal clock (peripheral clock (PCLK))					
	8MHz	10MHz	16MHz	20MHz	24MHz	32MHz
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

These numerical values apply when SCL rising of the I^2C bus is 0s. When SCL rising of the I^2C bus is slow, the baud rate becomes lower than the above numerical values.

■ 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.

8.3. I^2C Master Mode

I^2C master mode is shown.

In master mode, a start condition is generated on the I^2C bus, which then receives the clock. If I^2C 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".

8.3.1. Start Condition Generation

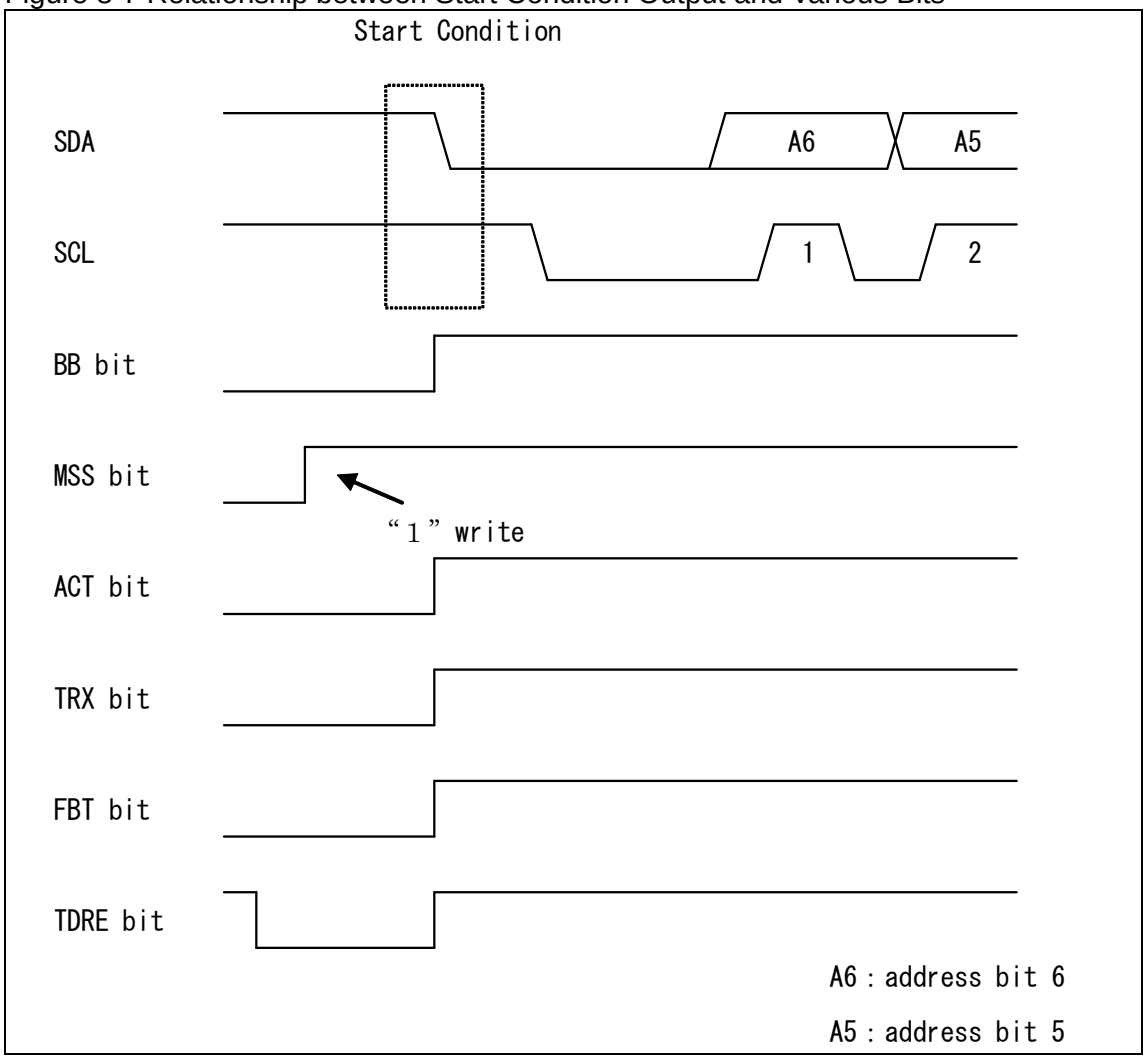
This section explains the start condition generation.

A start condition is output if:

- When SDA="H", SCL="H", EN="1", and BB="0", "1" is written to the IBCR:MSS bit

If a start condition is output to I^2C 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. (See Figure 8-7)

Figure 8-7 Relationship between Start Condition Output and Various Bits



Note:

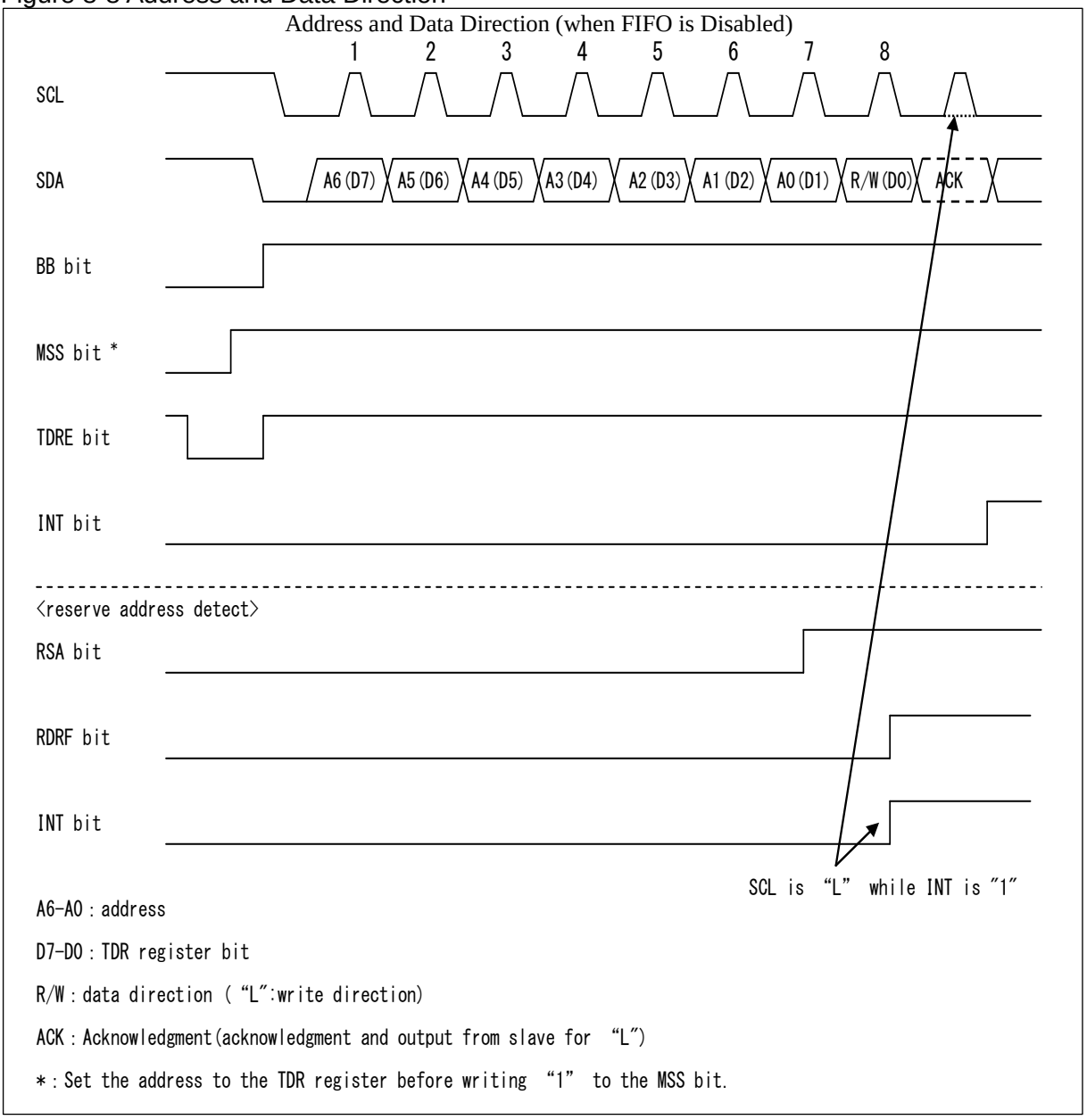
Bus clock should be used at 8MHz or more in operating mode 4 (I^2C mode) and setting a baud rate generator at 400kbps or more is prohibited.

8.3.2. Slave Address Output

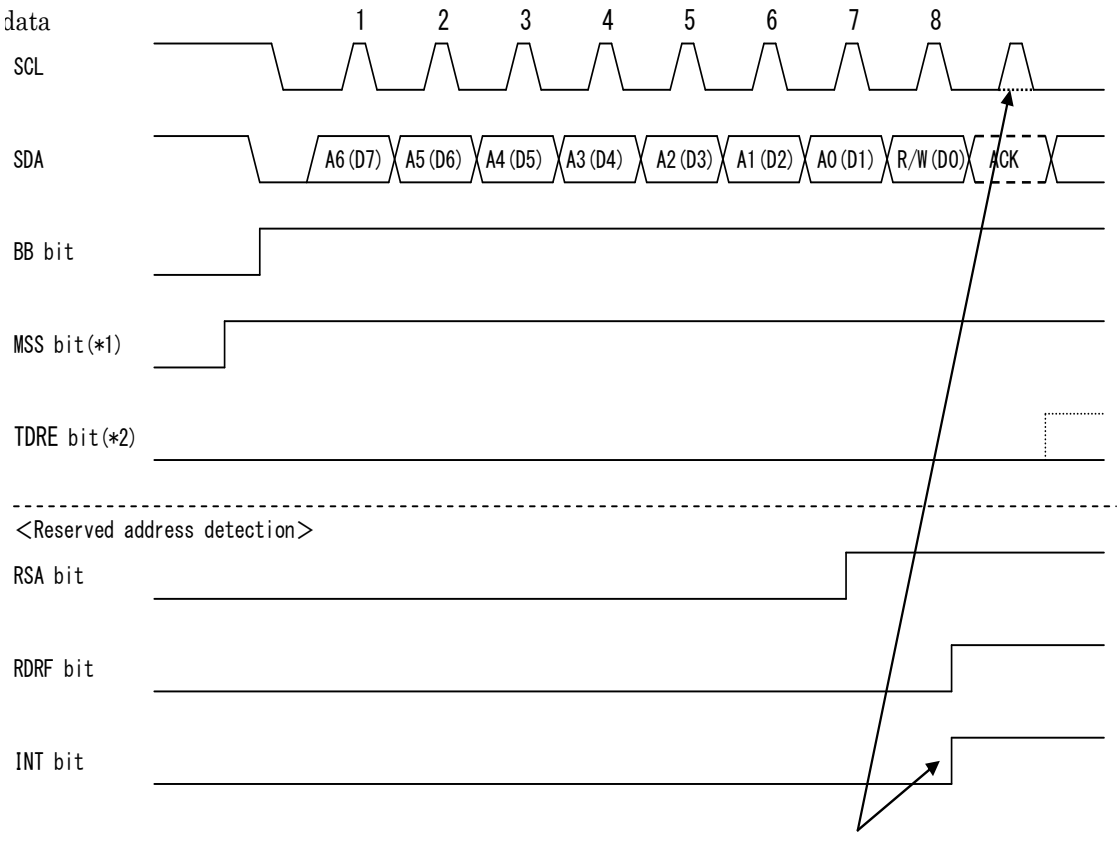
This section explains the slave address output.

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 writing direction (from master to slave). Set the address for the TDR register before "1" is written to IBCR:MSS or IBCR:SCC.

Figure 8-8 Address and Data Direction



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 slave 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".

8.3.3. Acknowledge Reception by Transmitting First Byte

This section explains the acknowledge reception by transmitting first byte.

When the data direction bit (R/W) is output, the I^2C 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 8-3 Operation after Acknowledge Reception (when DMA Mode is Disabled) (IBSR:RSA = "0", SSR:DMA= "0")

Trans-mission FIFO operation	Recep-tion FIFO operation	Trans-mission FIFO status	Recep-tion 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 8-4 Operation after Acknowledge Reception (when DMA Mode is Enabled) (IBSR:RSA="0", SSR:DMA="1")

Trans-mission FIFO operation	Recep-tion FIFO operation	Trans-mission FIFO status	Recep-tion FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge reception	
					Acknowledge is ACK	Acknowledge is ACK
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.	

■ DMA Mode Disabled (SSR:DMA=0)

● FIFO Disabled (Both Transmission and Reception FIFOs Disabled)

- 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 Enabled

- 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"), data including the slave address, etc. is set to the transmission FIFO.
- For data reception from the slave (data direction bit ="1"), the number of bytes to be received is set to the FIFO byte count register, and dummy data is written to the transmission data register for the number of data fields to be received as well as the slave address and data direction bit.
- When the IBSR:RSA bit is "0", after receiving ACK as an acknowledge, the interrupt flag (IBSR:INT) is not set to "1" but data is transmitted/received according to the data direction bit (not waited) if ACK is received. The interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited if NACK is received.
- 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".

■ DMA Mode Enabled (SSR:DMA=1)

● FIFO Disabled (Both Transmission and Reception FIFOs Disabled)

- 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 data transmitted to the TDR register is written, the transmission bus idle flag becomes "0" and waiting 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".

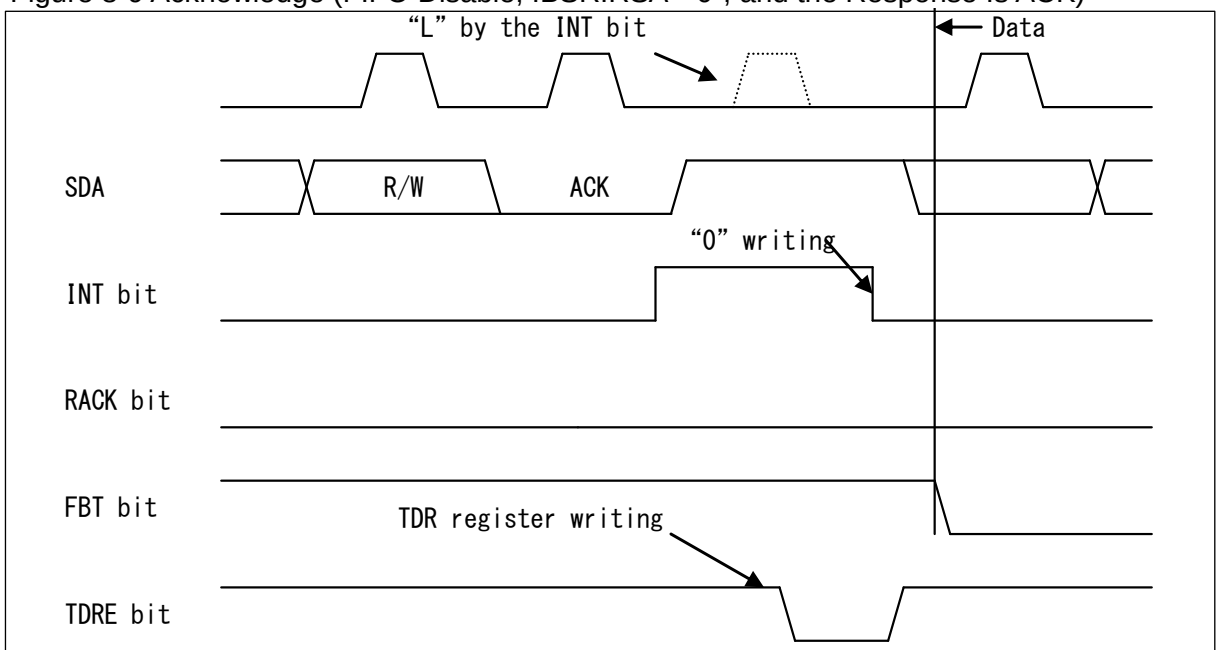
● FIFO Enabling

- 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"), data including the slave address, etc. is set to the transmission FIFO.
- For data reception from the slave (data direction bit ="1"), the number of bytes to be received is set to the FIFO

byte count register, and dummy data is written to the transmission data register for the number of data fields to be received as well as the slave address and data direction bit.

- When the IBCR:RSA bit is "0", after receiving ACK as an acknowledge, the interrupt flag (IBSR:INT) is not set to "1" but data is transmitted/received according to the data direction bit (not waited) if ACK is received. The interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited if NACK is received.
- 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 8-9 Acknowledge (FIFO Disable, IBSR:RSA="0", and the Response Is ACK)



Waiting to the address is as follows.

- After receiving the acknowledge when the RSA bit is "0"
- Before receiving the acknowledge when the RSA bit is "1"

It does not depend on the setting of WSEL.

Figure 8-10 Acknowledge (FIFO Disable, IBSR:RSA="0", and the Response Is NACK)

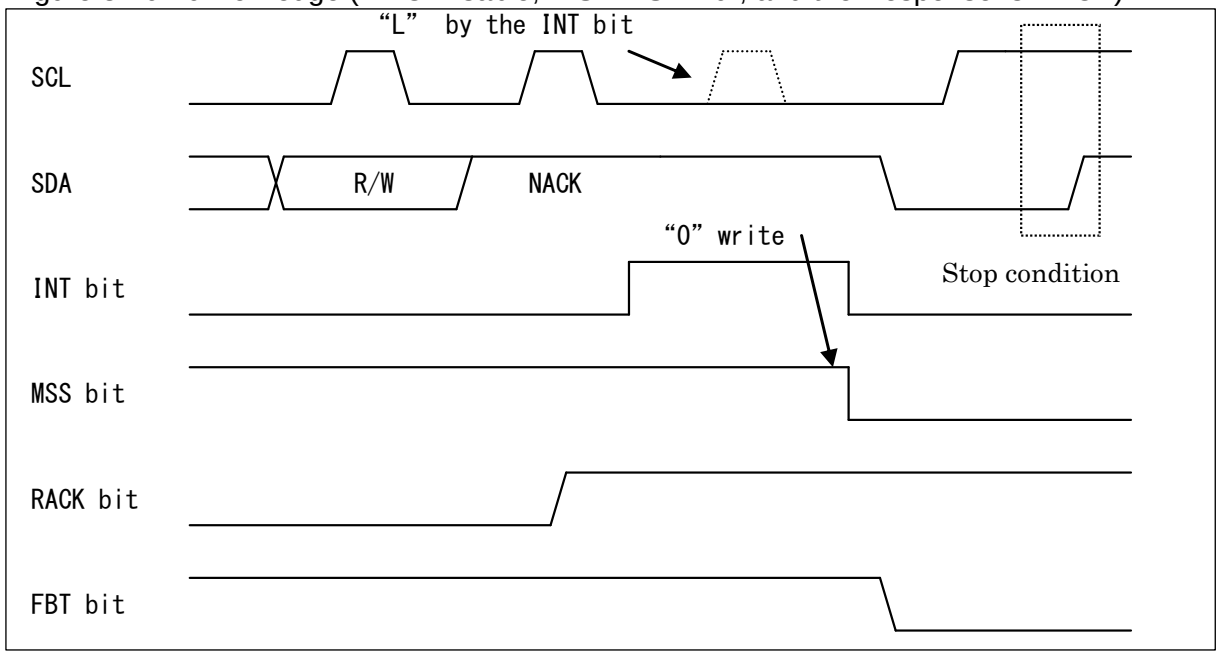


Figure 8-11 Acknowledge (FIFO Disable, IBSR:RSA="1", and the Response Is ACK)

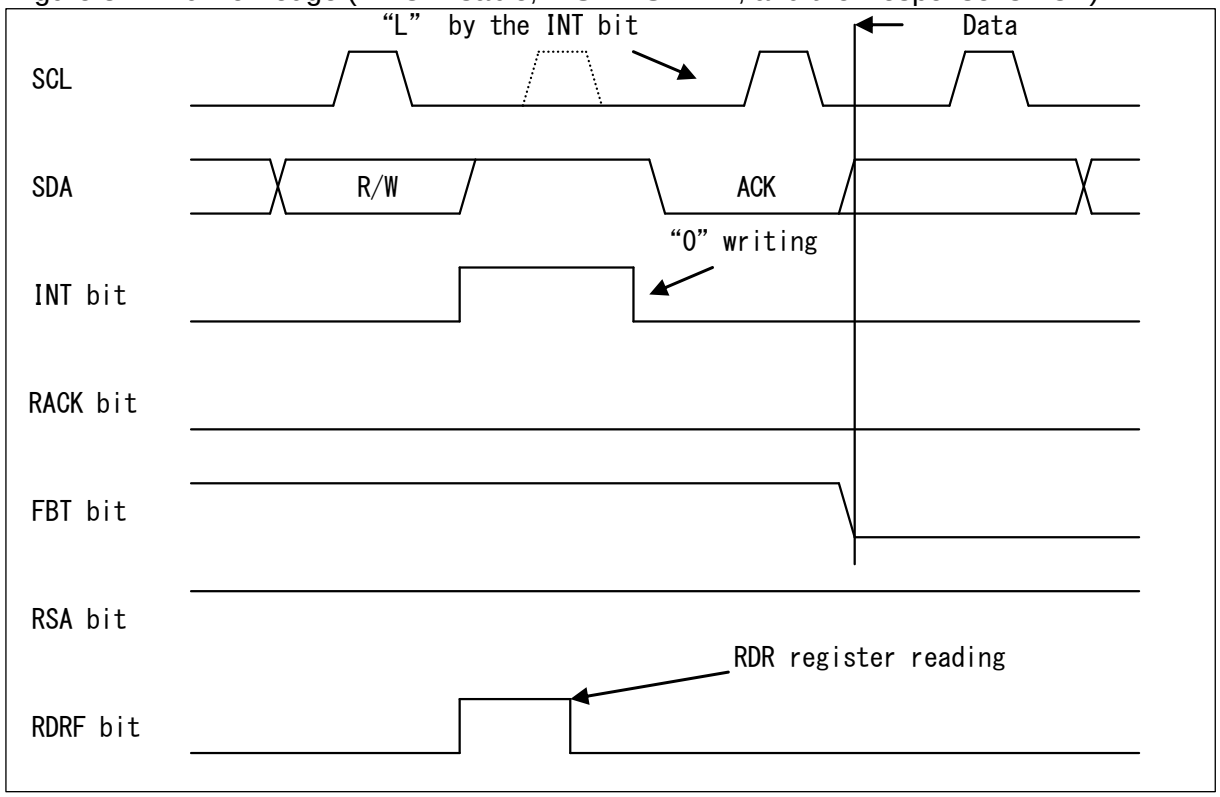


Figure 8-12 Acknowledge (FIFO Disable, IBSR:RSA="1", and the Response Is NACK)

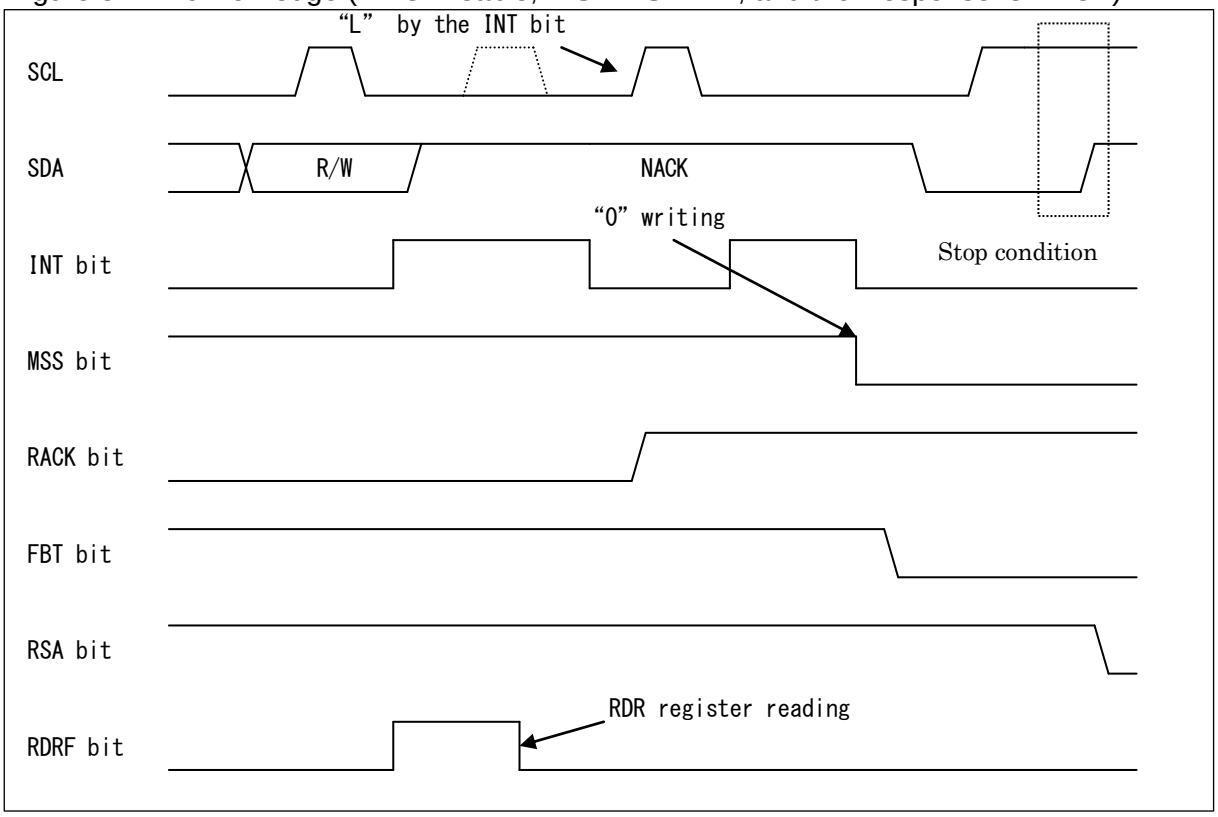
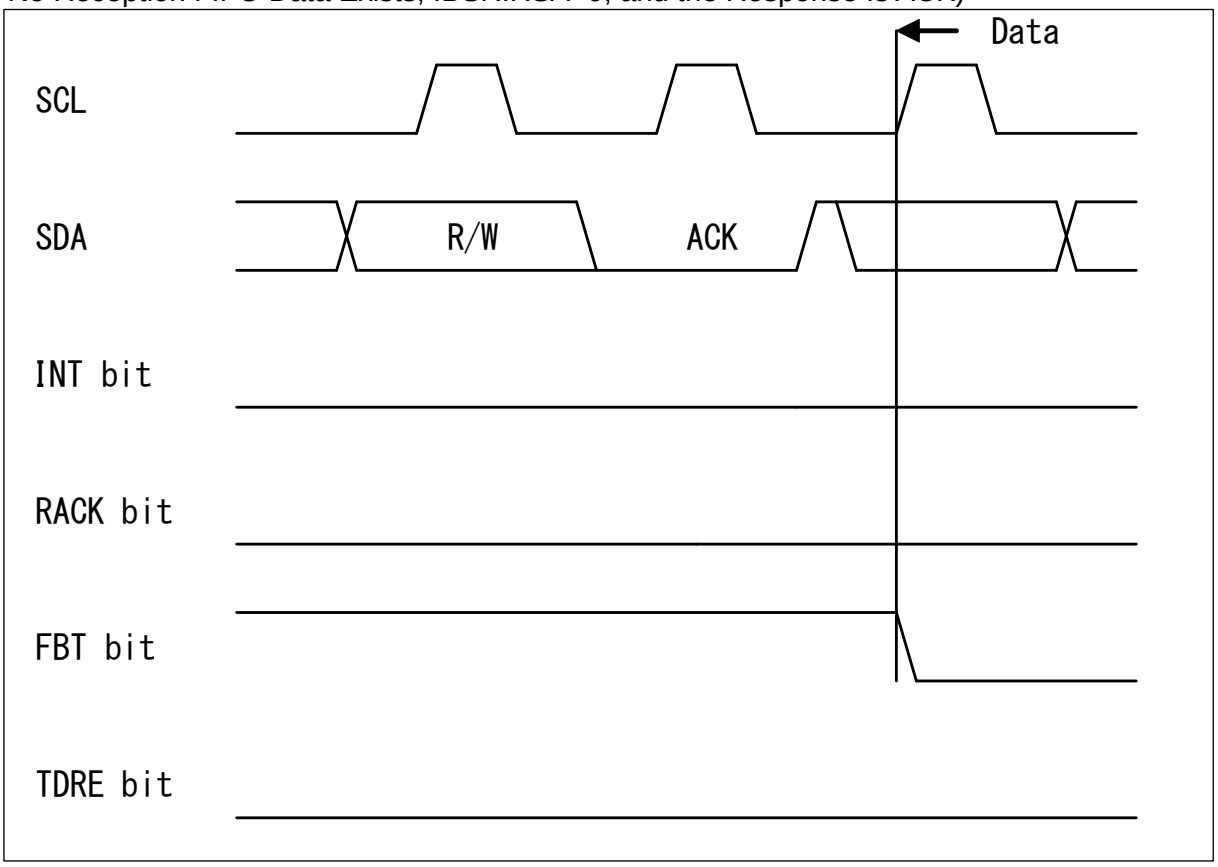


Figure 8-13 Acknowledge (If FIFO Is Enabled, Transmission FIFO Data Exists, No Reception FIFO Data Exists, IBSR:RSA=0, and the Response Is ACK)



8.3.4. Data Transmission by Master

This section explains the data transmission by master.

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 8-5 IBCR:WSEL Bit at the Time of Master Data Transmission (when DMA Mode is Disabled (SSR:DMA=0))

WSEL	Operation
0	<FIFO Unused> 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. <FIFO Used> If FIFO is enabled, the master will set the interrupt flag(IBCR:INT) to "1" after receiving acknowledge and then goes to the wait state, when it detects an arbitration lost or finds no valid data in the transmit data register (SSR:TDRE=1).
1	<FIFO Unused> 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. <FIFO Used> 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 8-6T IBCR:WSEL Bit at the Time of Master Data Transmission (when DMA Mode is Enabled (SSR:DMA=1))

WSEL	Operation
0	<FIFO Unused> In the 2nd or subsequent byte, after acknowledge is received when the SSR:TDRE bit to "1", the transmission bus idle flag (SSR:TBI) is set to "1" and SCL is set to "L" to put the bus into the wait state. <FIFO Used> Moreover, if FIFO is enabled, after acknowledge is received when the transmission data register has lost effective data (SSR:TDRE=1), the transmission bus idle flag (SSR:TBI) is set to "1" to put the bus into the wait state.
1	<FIFO Unused> In the 2nd or subsequent byte, after the master transmits the data of one byte when the SSR:TDRE bit is set to "1", transmission bus idle flag (SSR:TBI) is set to "1" and SCL is set to "L" to put the bus into the wait state. <FIFO Used> Moreover, if the FIFO is permitted, after the master transmits the data of one byte when the transmission data register has lost effective data (SSR:TDRE=1), the transmission bus idle flag (SSR:TBI) is set to "1" to put the bus into the wait state.

However, the master sets the interrupt flag (IBCR:INT) after receiving acknowledge regardless of the IBCR:WSEL setting in the following case:

- 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 when DMA mode is disabled (SSR:DMA=0)

[1] Transmission to a destination other than 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 IBCR: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) Write"0" to the IBCR:MSS bit or "1" to IBCR:SCC bit to generate a stop or repeat start conditions.

[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 I^2C 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 any of following conditions is met, 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.

- Reserved address is a general-call address in a multi-master configuration
- An arbitration lost is detected and the device may work as the slave

■ Data transmission to slave when DMA mode is enabled (SSR:DMA=1)

[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) Write the data to be transmitted to the TDR register, 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 IBCR:WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1".
(5) Write the data to be transmitted to the TDR register, 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 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 (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 are set in the TDR register.
(2) Set the IBCR:WSEL bit and 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 ACK responses are received, 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

[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 I^2C 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".
(7) Write the data to be transmitted to the TDR register, and release the I^2C bus from waiting state.
(8) Put the I^2C bus in a wait by setting the transmission bus idle 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 (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, the interrupt flag (IBCR:INT) is set to "1" 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 I^2C 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 any of following conditions is met, 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.

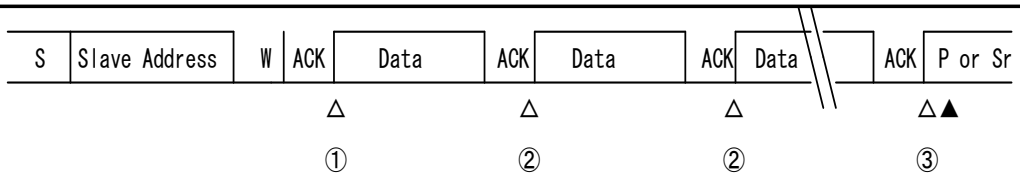
- Reserved address is a general-call address in a multi-master configuration
- An arbitration lost is detected and the device may work as the slave

*2: To issue the repeat start condition when the DMA mode is enabled (SSR:DMA=1), the SSR:TBI bit is set to "1" and the IBCR:INT bit is set to "0", follow the steps blow.

Notes:

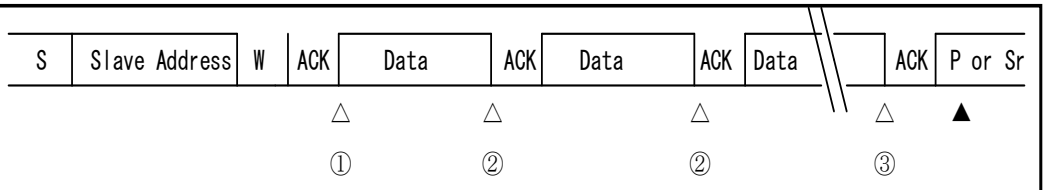
- If the 7-bit slave address detection is enabled (ISBA:SAEN="1"), you cannot specify a 7-bit slave address in the master mode.
- If you need to change the IBCR register during data sending or receiving, change it only when the interrupt flag (IBCR:INT) is "1".
- If the IBCR:WSEL bit is changed, when next data of interrupt flag (IBCR:INT) and the DMA mode is enabled (SSR:DMA=1), it is used for the occurrence condition of transmission bus idle flag (SSR:TBI).
- When transmitting data is written to TDR while SSR:TDRE is "1" during transmitting data, detecting an ACK response will activate following operations.
  - When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) will not become "1" but written data will be transmitted.
  - When DMA mode is enabled (SSR:DMA=1), the transmission bus idle flag (SSR:TBI) will not become "1" but written data will be transmitted.
- When transmitting data is written to TDR while SSR:TDRE is "1" during receiving data, following operations will be activated.
  - When DMA mode is disabled (SSR:DMA=0), the interrupt flag (IBCR:INT) will not become "1" but only SSR:RDRF become "1" (when reception FIFO is enabled or number of data set in the FBYTE register is received).
  - When DMA mode is enabled (SSR:DMA=1), the transmission bus idle flag (SSR:TBI) will not become "1" but only SSR:RDRF become "1" (when reception FIFO is enabled or number of data set in the FBYTE register is received).

Figure 8-14 Master Transmission Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



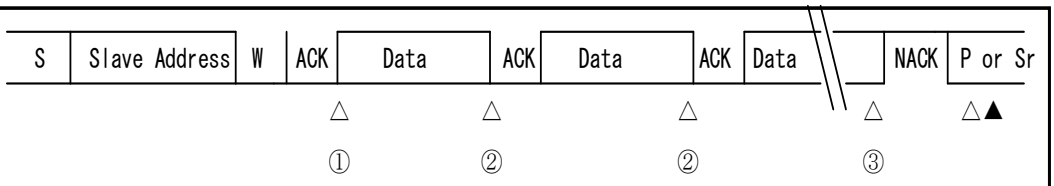
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 8-15 Master Transmission Interrupt (2)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", ACK Response)



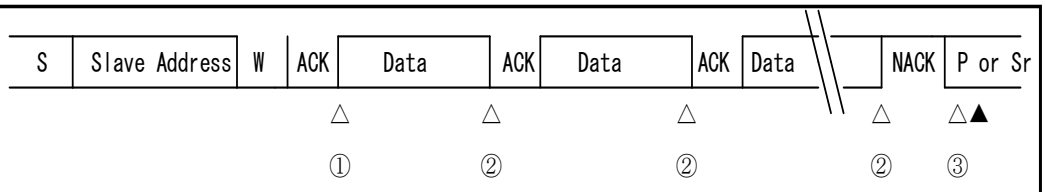
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 8-16 Master Transmission Interrupt (3)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)



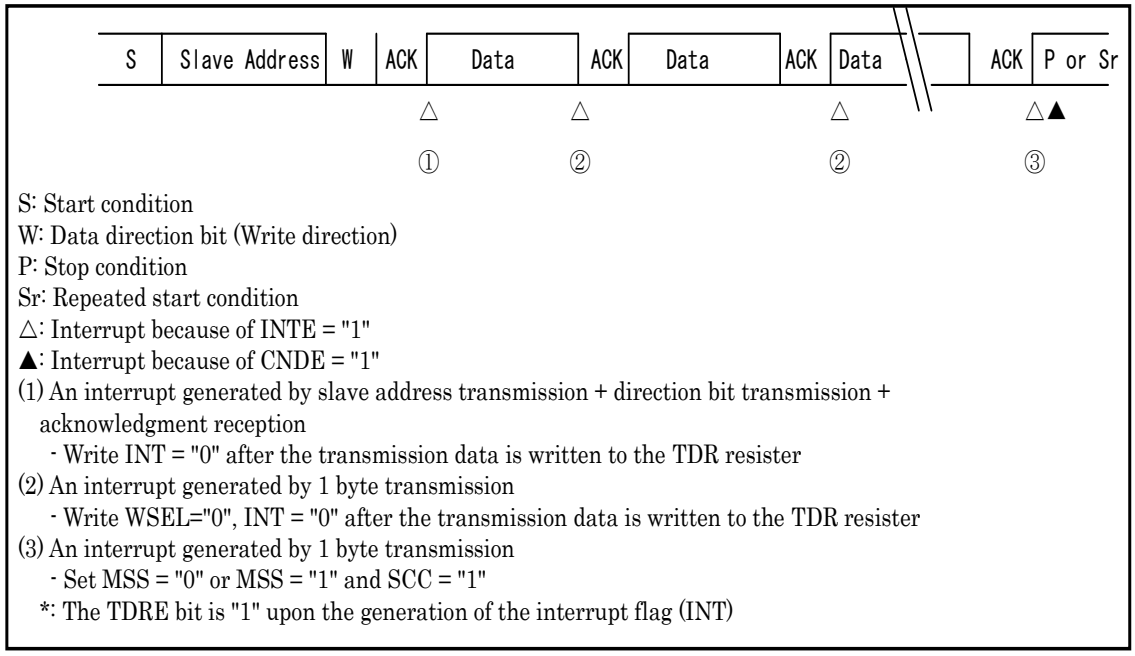
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 8-17 Master Transmission Interrupt (4)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", Intermediate NACK Response)



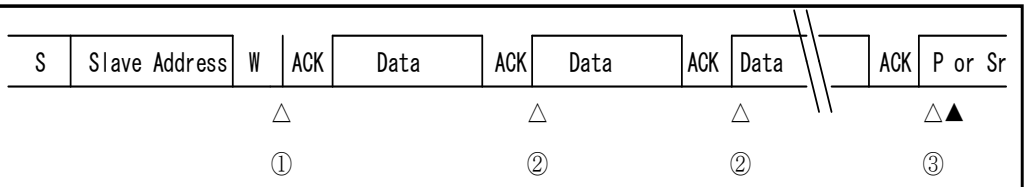
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 8-18 Master Transmission Interrupt (5)-when FIFO is Disabled
(SSR:DMA="0", IBCR:WSEL="1" -> "0", IBSR:RSA="0", ACK Response)



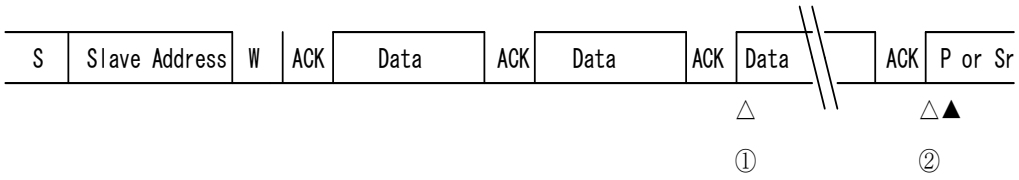
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR resister
(2) An interrupt generated by 1 byte transmission
- Write WSEL="0", INT = "0" after the transmission data is written to the TDR resister
(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 8-19 Master Transmission Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



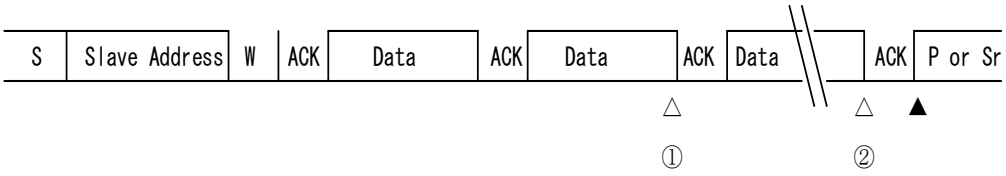
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception
- 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 8-20 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 because of INTE = "1"
▲: Interrupt because of 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 8-21 Master Transmission Interrupt (8)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



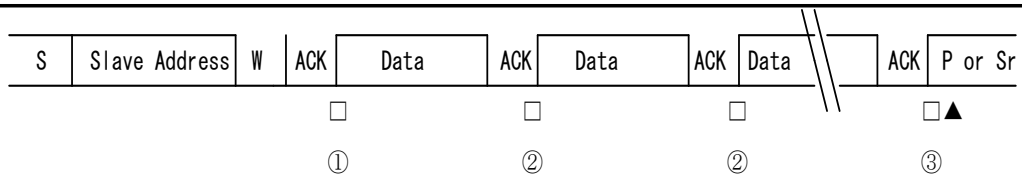
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to 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 8-22 Master Transmission Interrupt (9)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)



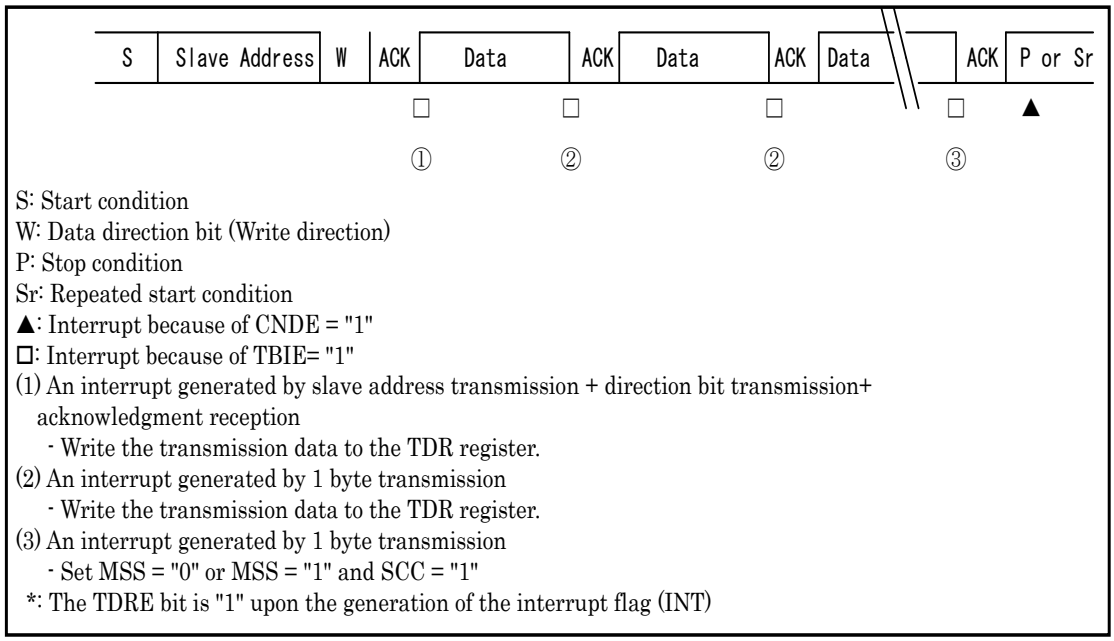
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to the transmission FIFO
(2) An interrupt generated by NACK response
- Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 8-23 Master Transmission Interrupt (10)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



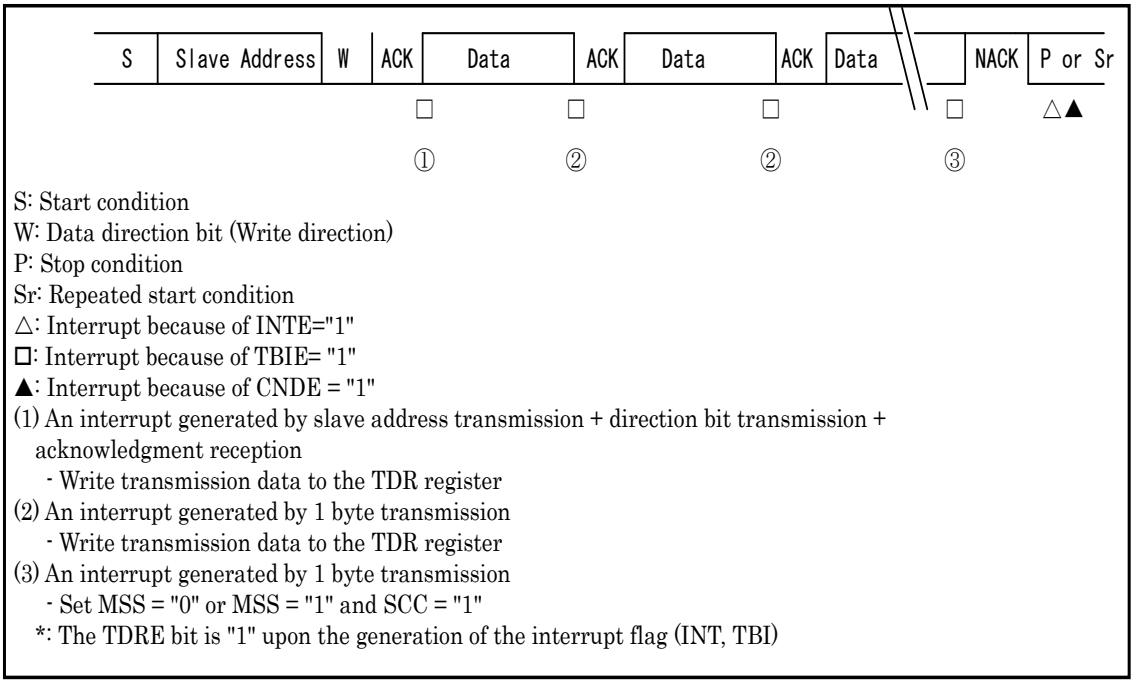
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission+ acknowledgment reception
- Write the transmission data to the TDR register.
(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- Write the transmission data 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 8-24 Master Transmission Interrupt (11)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", ACK Response)



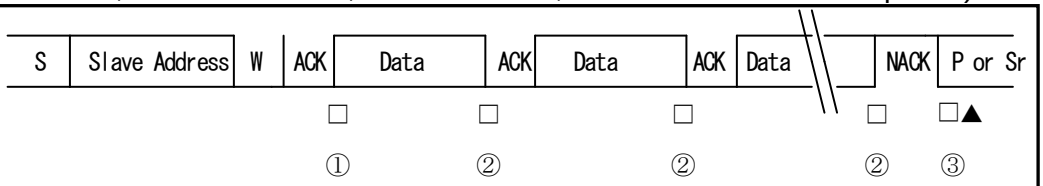
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission+ acknowledgment reception
- Write the transmission data to the TDR register.
(2) An interrupt generated by 1 byte transmission
- Write the transmission data 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 8-25 Master Transmission Interrupt (12)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)



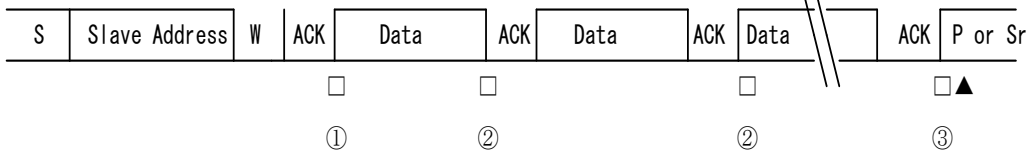
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE="1"
□: Interrupt because of TBIE= "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write transmission data to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write transmission data 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, TBI)

Figure 8-26 Master Transmission Interrupt (13)-when FIFO is Disabled
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", Intermediate NACK Response)



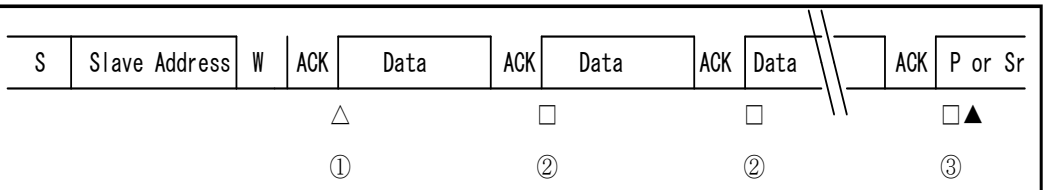
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 8-27 Master Transmission Interrupt (14)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1" –> "0", IBSR:RSA="0", ACK Response)



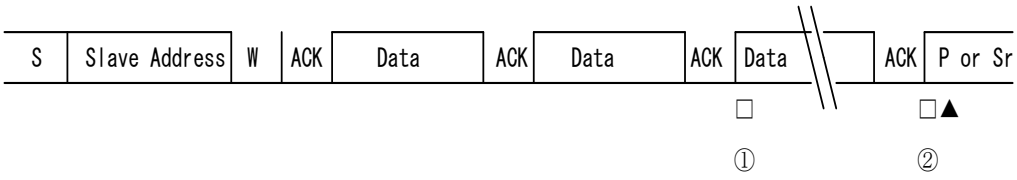
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write transmission data to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write transmission data to the TDR register after WSEL="0" is written
(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 (TBIE)

Figure 8-28 Master Transmission Interrupt (15)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



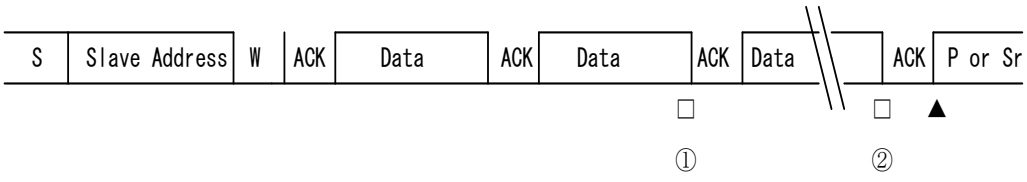
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE="1"
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + 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 transmission data 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, TBI)

Figure 8-29 Master Transmission Interrupt (16)-when FIFO is Enabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



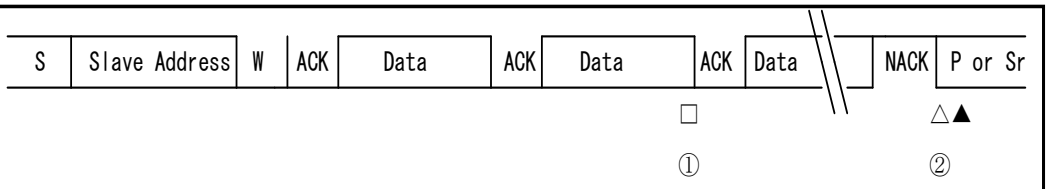
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to 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 8-30 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 because of CNDE = "1"
□: Interrupt because of TBIE= "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 8-31 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 because of INTE = "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "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"

8.3.5. Data Reception by Master

This section explains the data reception by master.

■ When DMA mode is disabled (SSR:DMA=0)

If the data direction bit (R/W) is "1", data sent from the slave device will be received.

When FIFO operation is disabled, the master device operation is as follows:

- If the SSR:TDRE bit is "1", the master device will generate a wait (IBCR:INT="1", SSR:RDRF="1") each time it receives 1 byte of data. At this time, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register before the wait if IBCR:WSEL bit is "1", or after the wait if IBCR:WSEL bit is "0".
- 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 to "1" when the same number of bytes as the received data bytes is received. The interrupt flag is set to "1" if the SSR:TDRE bit is "1", and the I^2C bus is waited. At this time the acknowledgement will operate as follows. The data will be stored in the reception FIFO as received data even if NACK is output.

- When IBCR:WSEL="0" and the SSR:TDRE bit is set to "1", a NACK response will be made if ACKE bit is set to NACK.
- When IBCR:WSEL = "1", the interrupt flag is set to "1" and 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 following explains the waiting by interrupt.

Table 8-7 WSEL Bit when Master Data is Received

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 I^2C bus from waiting state.
(4) Put the I^2C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when 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 items is 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 enabled (SSR:DMA="1")

If the data direction bit (R/W) is "1", data sent from the slave device will be received.

When FIFO operation is disabled, the master device operation is as follows:

· If the SSR:TDRE bit is "1", the master device will generate a wait (SSR:TBI="1", SSR:RDRF="1") each time it receives 1 byte of data. At this time, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register before the wait if IBCR:WSEL bit is "1", or after the wait if IBCR:WSEL bit is "0".
· If the SSR:TDRE bit is "0", the master device will generate a wait (SSR:RDRF="1") each time it receives 1 byte of data. At this time, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register before the wait if IBCR:WSEL bit is "1", or after the wait if IBCR:WSEL bit is "0".

If FIFO operation is enabled, the SSR:RDRF bit is set when the same number of bytes as the received data bytes is received. Set the transmission bus idle flag (SSR:TBI) while the SSR:TDRE bit is "1", to wait the I^2C bus. At this time the ACK will operate as follows. The data will be stored in the reception FIFO as received data even if NACK is output.

· When IBCR:WSEL="0" and the SSR:TDRE bit is set to "1", a NACK response will be made if ACKE bit is set to NACK.
· When IBCR:WSEL = "1", a wait (SSR: TBI="1") is generated after the last byte is received so that 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 following explains the waiting by interrupt.

Table 8-8 WSEL Bit During Master Data Reception

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 not used after the acknowledgment.
1	In the 2nd or subsequent byte, 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 when the reception FIFO not used.

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 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) After the reception of one byte, put the I^2C bus in a wait by setting the transmission bus idle flag (SSR:TBI) and reception data full flag (SSR:RDRF) *2 to "1" with following conditions.
   - After transmitting an acknowledge when IBCR:WSEL="0"
   - Immediately after one byte data is received when IBCR:WSEL="1"
(5) Update the IBCR:WSEL bit and update the IBCR:WSEL bit, read the RDR register, and write dummy data in the TDR register.
(6) After the reception of one byte, put the I^2C bus in a wait by setting the transmission bus idle flag (SSR:TBI) and reception data full flag (SSR:RDRF)*2 to "1" with following conditions.
   - After transmitting an acknowledge when IBCR:WSEL="0"
   - Immediately after one byte data is received when IBCR:WSEL=1

   Repeat 5. to 6. until specified number of data is received.
(7) After reception of the last data, send a NACK response, set the IBCR:MSS bit to "0" or set the IBCR: SCC*1 bit to "1" in order to generate a stop condition or a repeated start condition.

- If the transmission/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) If IBCR:WSEL="0", respond with NACK by the setting of the ACKE bit, and write "1" to 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", set the interrupt flag to "1" after sending a NACK response in order to put the I^2C bus in a wait. If IBCR:WSEL="1", set the transmission bus idle flag (SSR: TBI) to "1" immediately after 1 byte of data reception in order to put the I^2C bus in a wait.
(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*1 bit to "1" in order to generate a stop condition or a repeated start condition.

*1: When you issue the repeat start condition when the DMA mode is permitted (SSR:DMA=1), the SSR:TBI bit is "1" and the IBCR:INT bit is "0", follow the steps below.

1. Write "1" to IBCR:INT bit.
2. Make sure that "1" has been set to the IBCR:INT bit.
3. Write a slave address to the TDR.
4. Set "1" to the IBCR:SCC bit.

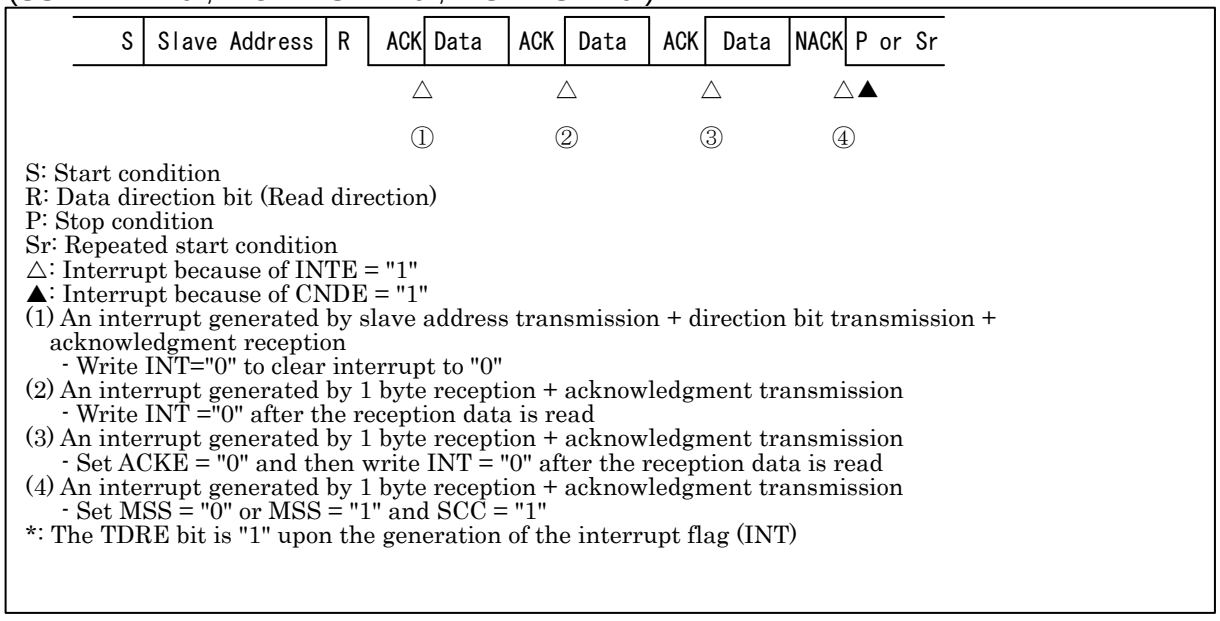
*2: The reception data full flag (SSR:RDRF) is set to "1" after one byte data received independent of the IBCR:WSEL setting. When the reception data full flag (SSR:RDRF) is set to "1" after the second byte, IBCR:WSEL="0" and after an ACK is transmitted while IBCR:WSEL="0", put the I^2C bus in a wait immediately after one byte data is received when IBCR:WSEL=1.

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:TBI="1") is "1" during the DMA mode is enabled (SSR:DMA=1).

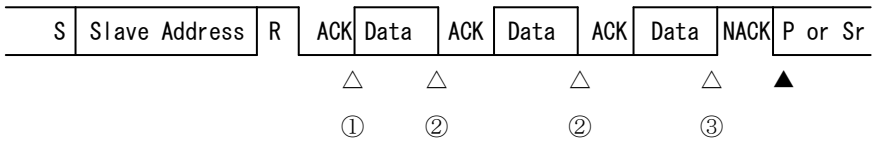
- When the master device is receiving data and the DMA mode is disabled (SSR:DMA=0) and when dummy data is written in the TDR register, the next data will be received with the interrupt flag (IBCR:INT) still "0" when SSR:TDRE bit is "0" at the timing when the interrupt flag (IBCR:INT) becomes "1".
- When the master device is receiving data and the DMA mode is enabled (SSR:DMA=1) and when dummy data is written in the TDR register, the next data will be received with the transmission bus idle flag (SSR:TBI) still "0" when SSR:TDRE bit is "0" at the timing when the transmission bus idle flag (SSR:TBI) becomes "1".
- 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 8-32 Master Reception Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")



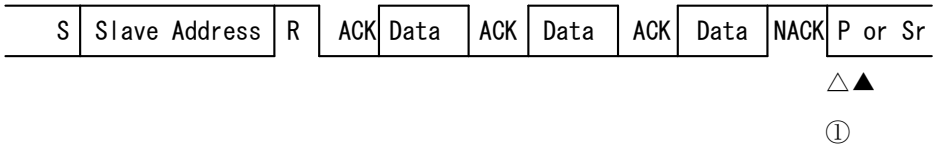
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
  - Write INT="0" to clear interrupt to "0"
(2) An interrupt generated by 1 byte reception + acknowledgment transmission
  - Write INT ="0" after the reception data is read
(3) An interrupt generated by 1 byte reception + acknowledgment transmission
  - Set ACKE = "0" and then write INT = "0" after the reception data is read
(4) An interrupt generated by 1 byte reception + acknowledgment transmission
  - Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 8-33 Master Reception Interrupt (2)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



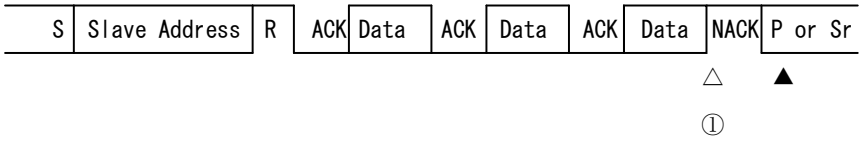
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by slave address transmission, direction bit transmission and acknowledgement reception:
- Clear interrupt to "0" after writing "0" to INT
(2) Generation of interrupt by 1-byte reception:
- Write "0" to INT after reading reception data
(3) Generation of interrupt by 1-byte reception:
- Set "0" to ACKE and sets MSS="0", MSS="1" or SCC="1" after reading reception data
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 8-34 Master Reception Interrupt (3)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



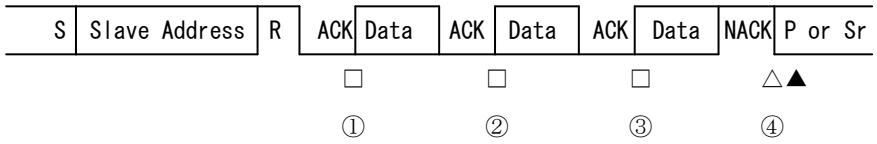
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by TDRE="1":
- Set MSS="0", MSS="1" or SCC="1" after reading all data from reception FIFO

Figure 8-35 Master Reception Interrupt (4)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



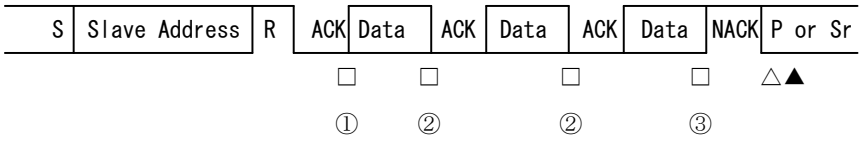
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by TDRE="1":
- Set "0" to ACKE and sets MSS="0" or MSS="1", and SCC="1" after reading all data from reception FIFO

Figure 8-36 Master Reception Interrupt (5)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



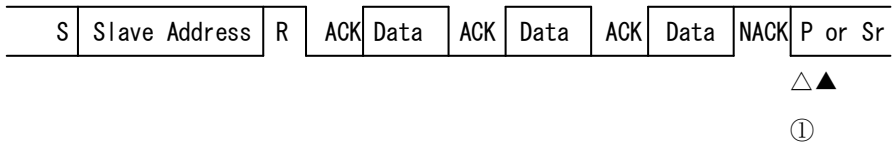
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write dummy data to the TDR register.
(2) An interrupt generated by 1byte reception + acknowledgment transmission
- Read reception data and then write dummy data to the TDR register.
(3) An interrupt generated by 1byte reception + acknowledgment transmission
- Read reception data, set ACKE="0", and then write dummy data to the TDR register.
(4) An interrupt generated by 1byte reception + acknowledgment transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 8-37 Master Reception Interrupt (6)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



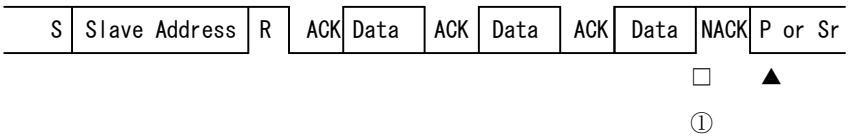
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) Generation of interrupt by slave address transmission + direction bit transmission + acknowledgment reception
- Write dummy data to the TDR register.
(2) An interrupt generated by 1 byte reception
- Read reception data and then write dummy data to the TDR register.
(3) An interrupt generated by 1 byte reception
- Set ACKE= "0", MSS= "0" or MSS= "1", and SCC= "1" after reception data reading
*) Interrupt flag(INT,TBI) generated, TDRE bit is "1"

Figure 8-38 Master Reception Interrupt (7)-when FIFO is Enabled (SSR:DMA="1", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of 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 8-39 Master Reception Interrupt (8)-when FIFO is Enabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt because of TBIE= "1"
(1) Generation of interrupt by TDRE ="1"
- Set ACKE= "0", MSS= "0" or MSS= "1", and SCC= "1" after reading all data from reception FIFO

8.3.6. Arbitration Lost

This section explains the arbitration lost.

If data of the master device is hit by data from another master device and if the data that is different from the send data is received, it will be determined to be an arbitration lost. The IBCR:MSS bit is set to "0" and the IBCR:AL bit is set to "1" so that the device can operate in the slave mode.

The IBCR:AL bit is cleared to "0" if:

- The IBCR:MSS bit is set to "1"
- The IBCR:INT bit is set to "0".
- The IBCR:SPC bit is set to "0" when IBCR:AL="1" and IBCR:SPC="1".
- The I^2C interface operation 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.

8.3.7. Wait of the Master Mode

This section explains the wait of the master mode.

In the case where both of conditions below are met, wait the master mode while IBSR:BB bit is "1" and transmit a start condition after the IBSR:BB became "0".

- When "1" is set to IBCR:MSS bit while IBSR:BB bit is "1"
- The operation is not in slave mode

Whether or not the master mode is in a wait can be determined with IBCR:MSS bit and IBCR:ACT bit (IBCR:MSS="1" and IBCR:ACT="0" mean a wait). Operating in slave mode after IBCR:MSS bit is set to "1", set IBSR:AL bit to "1", IBCR:MSS bit to "0", and IBCR:ACT bit to "1".

8.3.8. Repetition Start Condition Issue when DMA Mode Enabled (SSR:DMA=1)

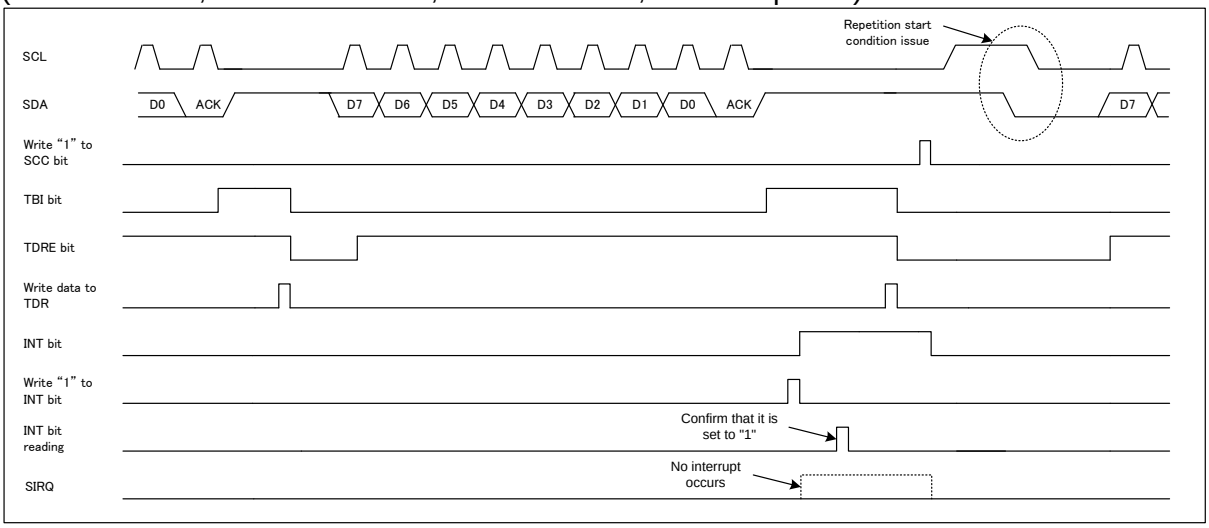
This section explains the repetition start condition issue when DMA mode is enabled (SSR:DMA=1).

When the transmission bus is idle and the interrupt flag (IBCR:INT) is "0", if the slave address is written in the TDR register, the transmission operation begins and the repetition start condition cannot be issued.

Because of this reason, follow the steps blow when you issue the repetition start condition when the transmission bus is idle (SSR:TBI="1") and the interrupt flag (IBCR: INT) is "0".

1. Set "1" to the IBCR:INT bit. At this time, the SIRQ interrupt is not generated.
2. Make sure that "1" has been set to the IBCR:INT bit.
3. Write a slave address to the TDR.
4. Issue a repeat start (IBCR:SCC="1").

Figure 8-40 Repetition Start Condition Issue when DMA Mode is Enabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



8.4. I²C Slave Mode

I²C slave mode is shown.

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.

Notes:

- The next data cannot be received because the bus error is detected (IBCR:BER=1) and the reception is interrupted when the start condition is detected again, while transmitting the address data or bit2-bit9 (acknowledge bit) after

the start condition is detected, at EIBCR:BEC=0.

- In this case, after the interrupt flag (IBCR:INT) is cleared, the transmission processing of the start condition is needed from the master again.

8.4.1. Detection of Slave Address Matching

This section explains the detection of slave address matching.

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 8-9 Operations Immediately After Acknowledgment to Slave Address

Trans-mission FIFO opera-tion	Recep-tion FIFO opera-tion	Trans-missio n FIFO status	Receptio n 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		

Trans-mission FIFO opera-tion	Recep-tion FIFO opera-tion	Trans-missio n FIFO status	Receptio n FIFO status	Data direction bit (R/W)	Operation immediately after acknowledgement	
					Acknowledge is ACK	Acknowledge is NACK
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 transmit or receive FIFO operation. During this time, make following settings after read the received data.

  - If you 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 send data in the TDR register and clear the IBCR:INT bit. Then, the device operates as a slave one.
  - If you don't want to operate the device as a slave one, set the IBCR:ACKE bit to "0" and clear the IBCR:INT bit. The device will not operate as a slave one after an acknowledge is output.

8.4.2. Data Direction Bit

This section explains the data direction bit.

After the address reception, a data direction bit that determines the data transmission or reception is received. If this bit is "0", it shows that the master transmits data, while the slave receives data.

8.4.3. Slave Mode Reception

This section explains the slave mode reception.

If the slave address matches and the data direction bit is "0", it indicates data reception in slave mode.

■ When DMA mode is disabled (SSR:DMA=0)

- If the receive FIFO operation is disabled

(1) After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to place the I^2C bus in the waiting state. An interrupt occurrence due to a slave address match is determined with the IBCR:MSS, IBCR:ACT, and IBSR:FBT

bits. To release the I^{2}C bus from the waiting state, set the IBCR:ACKE bit to "1" and set the interrupt flag to (IBCR:INT) to "0". See Table 8-2.

(2) After receiving one byte of data, set the interrupt flag (IBCR:INT) to "1" based on the IBCR:WSEL setting, and put the I^{2}C bus into the waiting state.

(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 I^{2}C bus from the waiting state.

(4) Repeat Steps (2) and (3) until the stop condition or the repeated start condition is detected.

· If the receive 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 I^{2}C 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 I^{2}C 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, set the interrupt flag to "0" and release the I^{2}C bus from the waiting state.

When the stop condition or the repeated start condition is detected, all of the received data are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".

■ When DMA mode is enabled (SSR:DMA=1)

· If the receive FIFO operation is disabled

(1) After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to place the I2C bus in the waiting state. An interrupt occurrence due to a slave address match is determined with the IBCR:MSS, IBCR:ACT, and IBSR:FBT bits. To release the I2C bus from the waiting state, set the IBCR:ACKE bit to "1" and set the interrupt flag to (IBCR:INT) to "0". See Table 8-2.

(2) Receive data full flag (SSR:RDRF) is set to "1" immediately following the reception of 1 byte after the reception of 1 byte of data. When receive data full flag (SSR:RDRF) is set to "1", the I^{2}C bus is put into the waiting state after the acknowledge is transmitted if IBCR:WSEL="0" or immediately after the reception of 1 byte if IBCR:WSEL=1.

(3) The data received from the RDR register after the IBCR:ACKE bit is set clears receive data full flag (SSR:RDRF) to "0" by reading and releases the I^{2}C bus from the waiting state.

(4) Repeat Steps (2) and (3) until the stop condition or the repeated start condition is detected.

· If the receive FIFO operation is enabled

(1) The interrupt flag (IBCR:INT) becomes "1" and the I^{2}C bus is put into the waiting state by detecting NACK. the I^{2}C bus is put into the waiting state 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 interrupt flag (IBCR:INT) does not become "1" (the I^{2}C bus is not put into the waiting state). 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, set the interrupt flag to "0" and release the I^{2}C bus from the waiting state.

If the data received from the RDR register even once is read when reception FIFO becomes full, the I^2C bus is released from the waiting state. When the stop condition or the repeated start condition is detected, all of the received data are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".

Figure 8-41 Slave Reception Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", 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 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 8-42 Slave Reception Interrupt (2)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



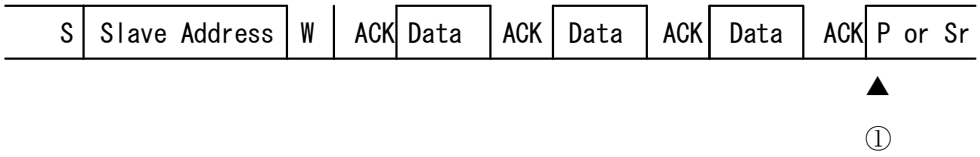
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt 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 8-43 Slave Reception Interrupt (3)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



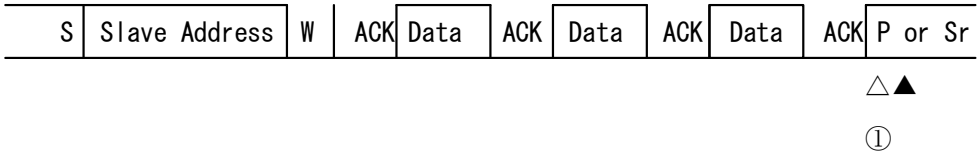
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
 - Write ACKE="1", INT = "0"
(2) An interrupt 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 8-44 Slave Reception Interrupt (4)-when FIFO is Enabled (SSR:DMA="0" IBSR:RSA="0")



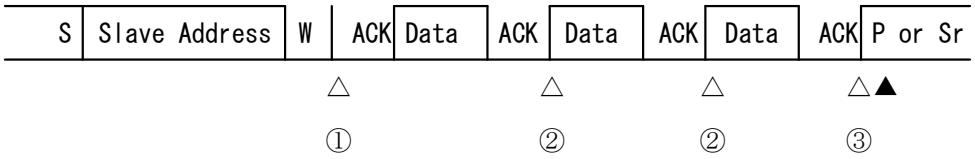
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by the detection of a stop condition or a repeated start conditior
 - Read all data from the reception FIFO

Figure 8-45 Slave Reception Interrupt (5)-when FIFO is Enabled (SSR:DMA="0", IBSR:RSA="0")



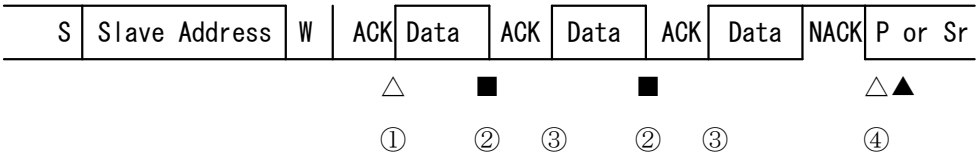
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the reception FIFO gets full
 - Read all data from the reception FIFO, and write INT = "0"

Figure 8-46 Slave Reception Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



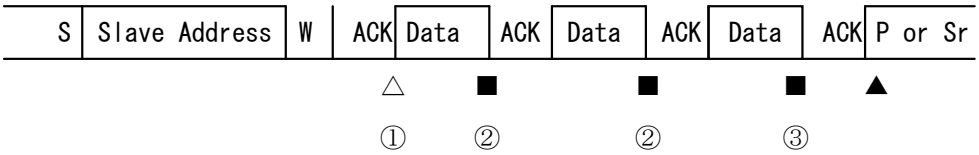
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by matching the reserved address ("0000XXXX" or "1111XXXX")
 - Read the reception data, and write 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 8-47 Slave Reception Interrupt (7)-when FIFO is Disabled
(SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



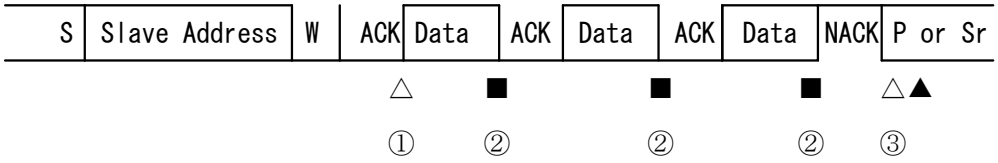
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt because of RIE= "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT="0"
(2) An interrupt generated by 1 byte reception (I^2C bus is no wait)
- Read reception data from the reception buffer
(3) I^2C bus wait by ACK response
- Read reception data from the reception buffer
(4) An interrupt generated by 1 byte reception ＋NACK response
- Read reception data from the reception buffer and write INT="0"

Figure 8-48 Slave Reception Interrupt (8)-when FIFO is Disabled
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



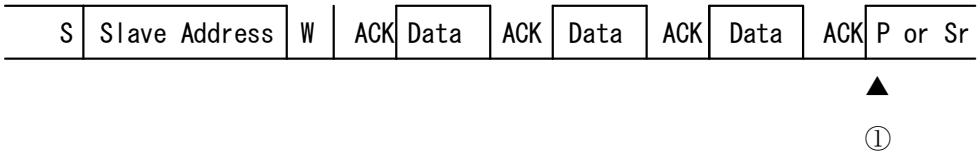
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt because of RIE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT="0"
(2) An interrupt generated by 1 byte reception
- Read reception data from the reception buffer
(3) An interrupt generated by 1 byte reception
- After reading reception data from the reception buffer

Figure 8-49 Slave Reception Interrupt (9)-when FIFO is Disabled
(SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



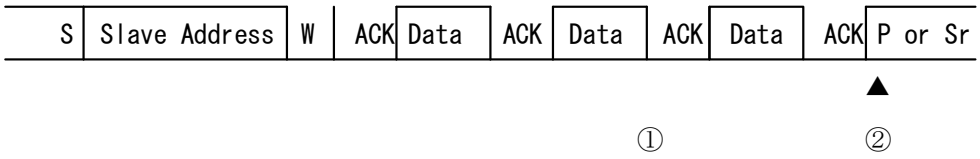
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt because of RIE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT="0"
(2) An interrupt generated by 1 byte reception
- Read reception data from the reception buffer
(3) An interrupt generated by NACK response
- Write INT="0"

Figure 8-50 Slave Reception Interrupt (10)-when FIFO is Enabled
(SSR:DMA="1" IBSR:RSA="0")



S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by the detection of a stop condition or a repeated start conditior
- Read all data from reception FIFO

Figure 8-51 Slave Reception Interrupt (11)-when FIFO is Enabled (SSR:DMA="1", IBSR:RSA="0")



S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of 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 conditior
 - Read all data from reception FIFO

Figure 8-52 Slave Reception Interrupt (12)-when FIFO is Enabled (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 because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt because of RIE = "1"
(1) Interrupt generated because reserved address ("0000xxxx" or "1111xxxx") is match
 - Read reception data, write ACKE="1", INT="0"
(2) An interrupt generated by 1 byte reception + acknowledge output
 - Read reception data
(3) An interrupt generated by 1 byte reception + acknowledge output
 - Read reception data

8.4.4. Slave Mode Transmission

This section explains the slave mode transmission.

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 8-9).

An acknowledgement output from the master device can be checked with the IBCR:RACK bit. 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.

8.5. Example of I²C Flowchart

Example of I²C flowchart is shown.

Figure 8-53 Example of I²C Flowchart (FIFO Memory not Used) (when DMA Mode is Disabled (SSR:DMA=0)) 1/3

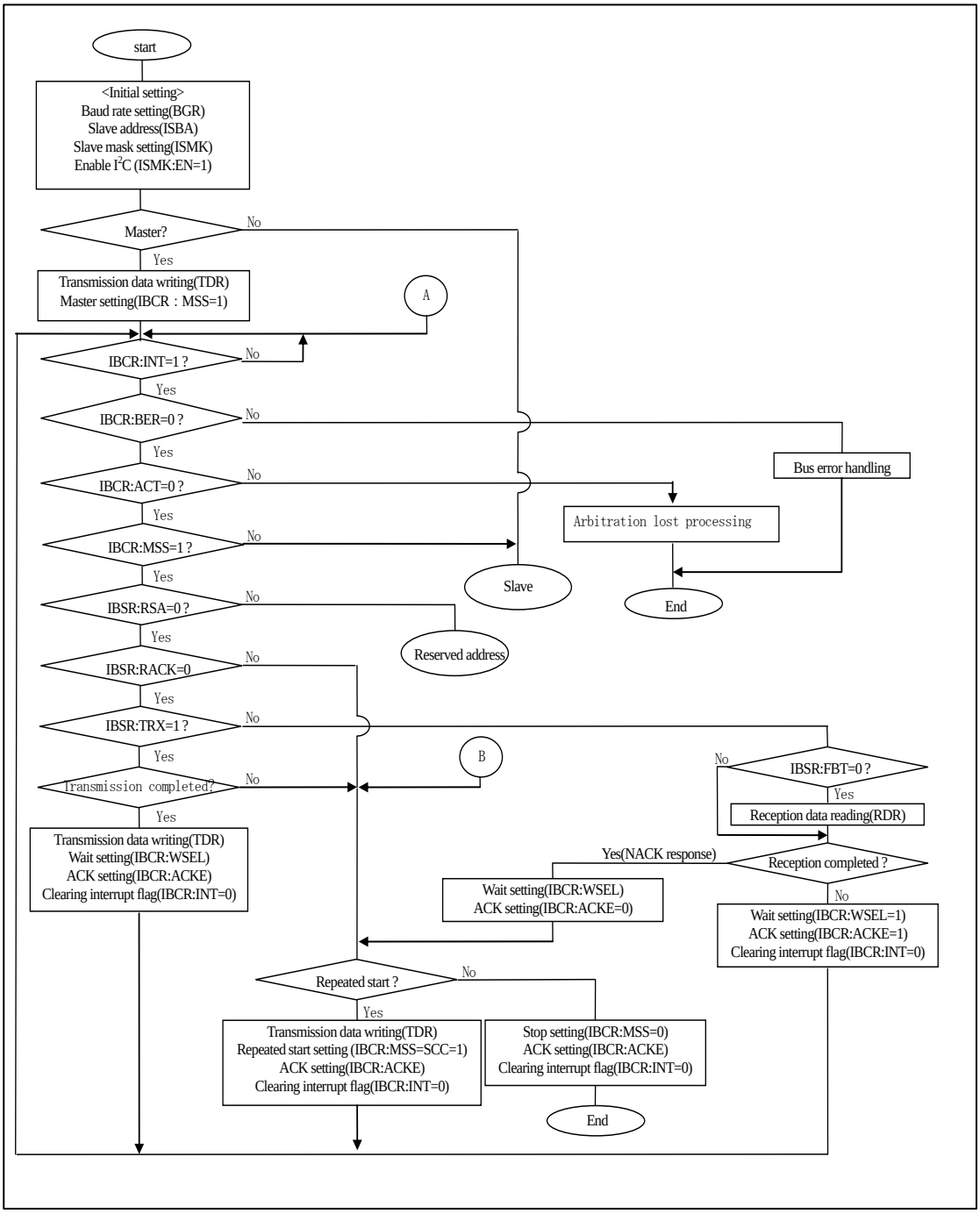


Figure 8-54 Example of I²C Flowchart (FIFO Memory not Used)
(when DMA Mode is Disabled (SSR:DMA=0)) 2/3

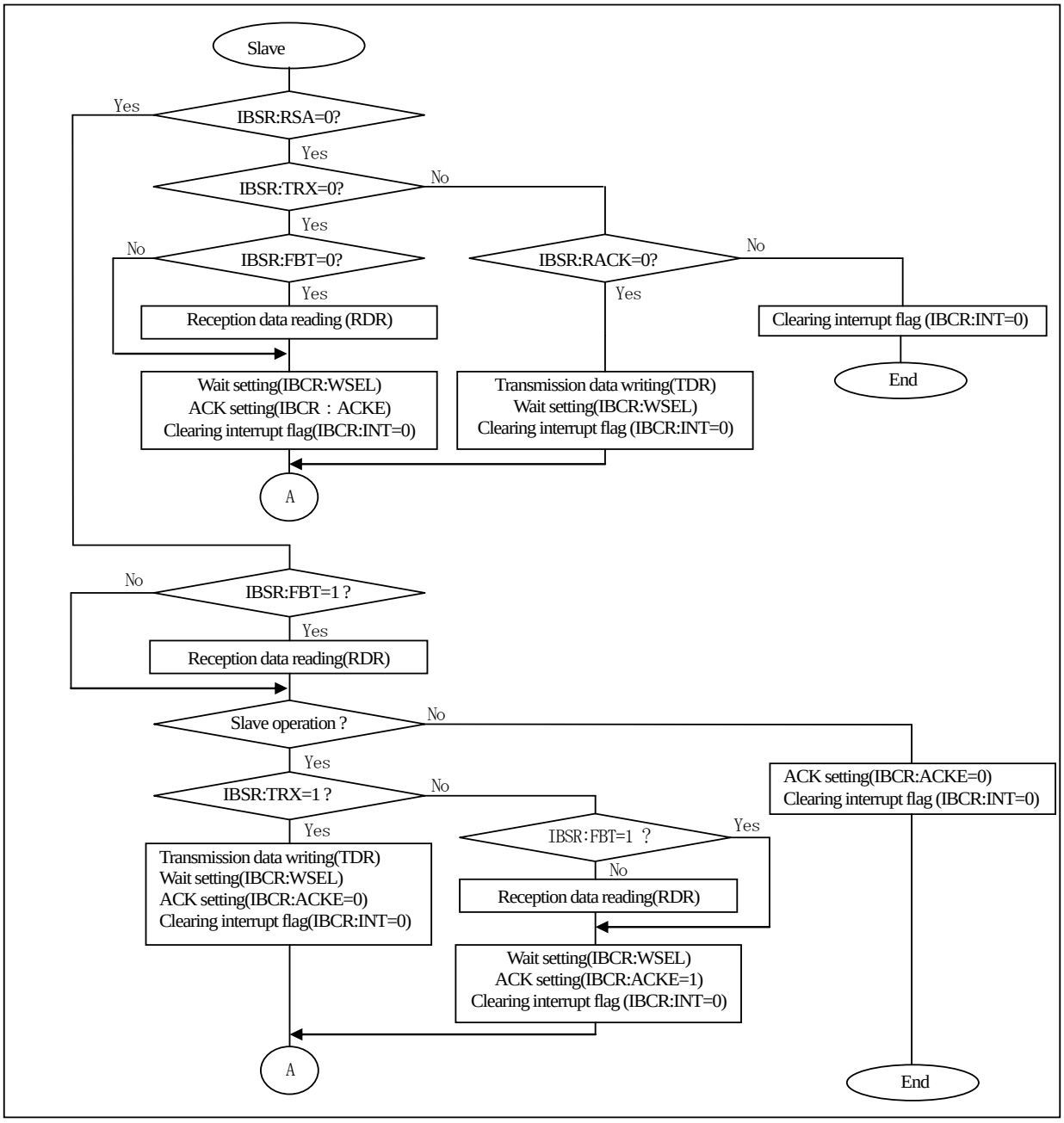


Figure 8-55 Example of I²C Flowchart (FIFO Memory not Used)
(when DMA Mode is Disabled (SSR:DMA=0)) 3/3

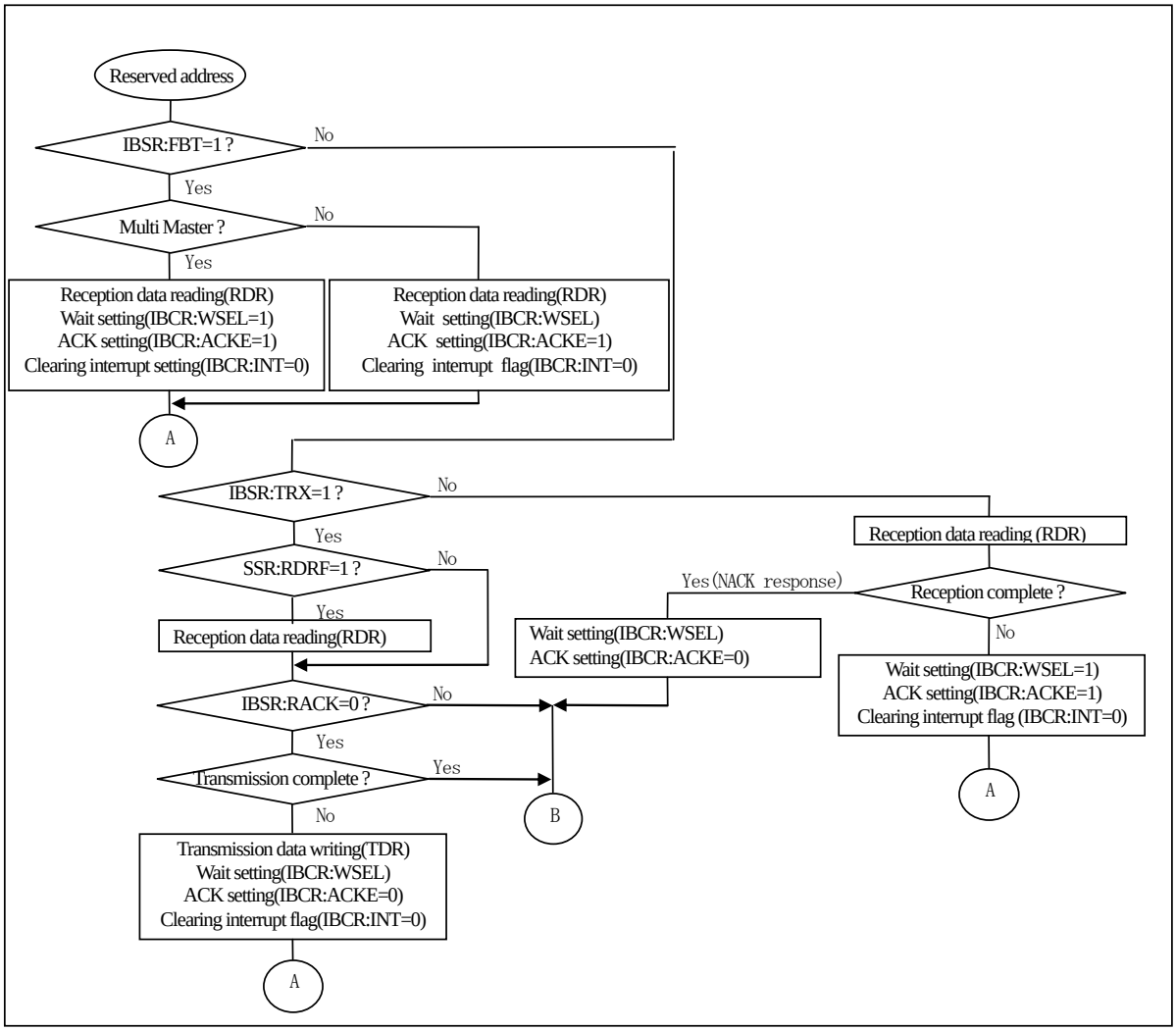


Figure 8-56 Example of I²C Flowchart (FIFO Memory not Used) (when DMA Mode is Enabled (SSR:DMA=1)) 1/4

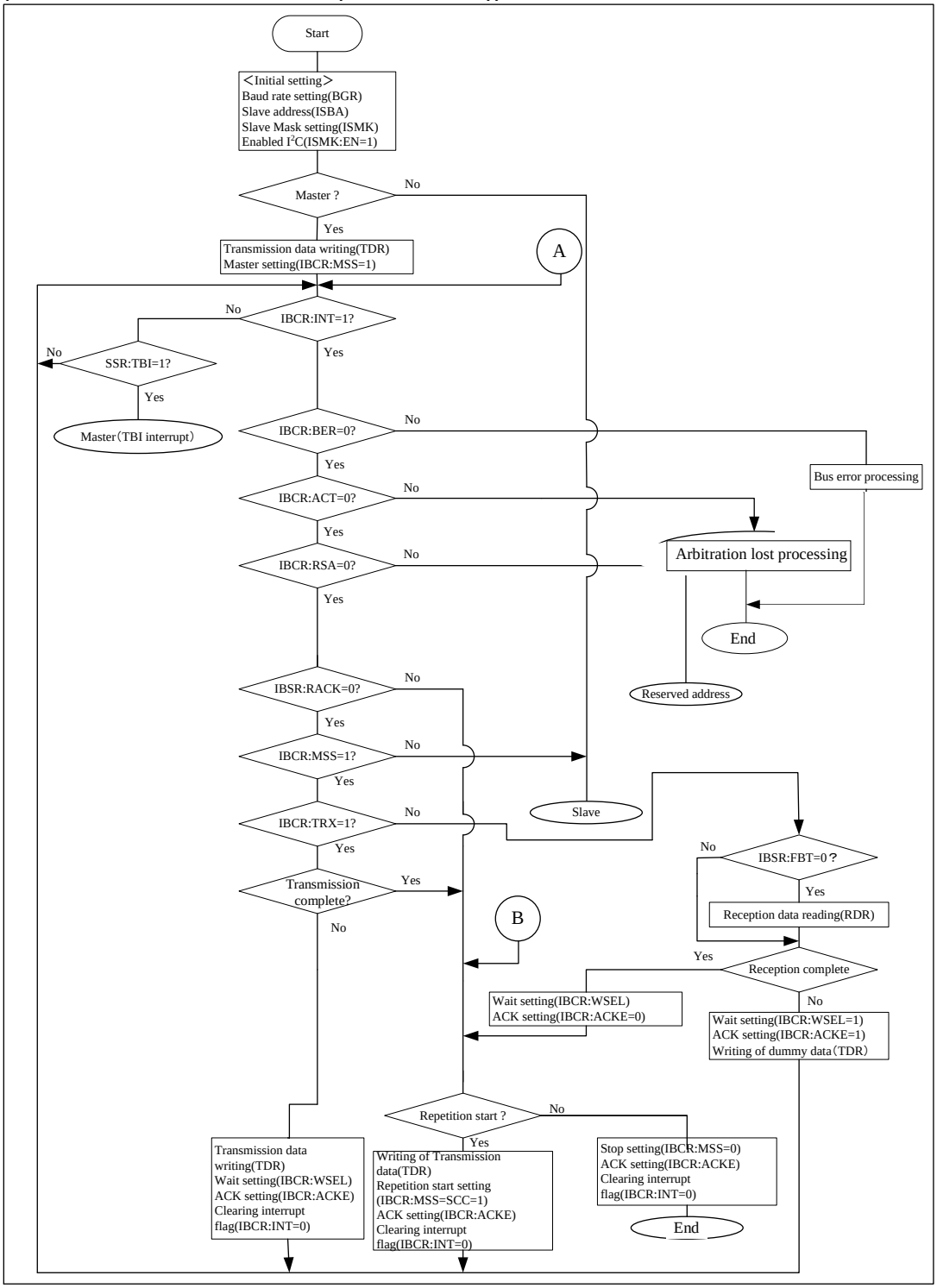


Figure 8-57 Example of I²C Flowchart (FIFO Memory not Used)
(when DMA Mode is Enabled (SSR:DMA=1)) 2/4

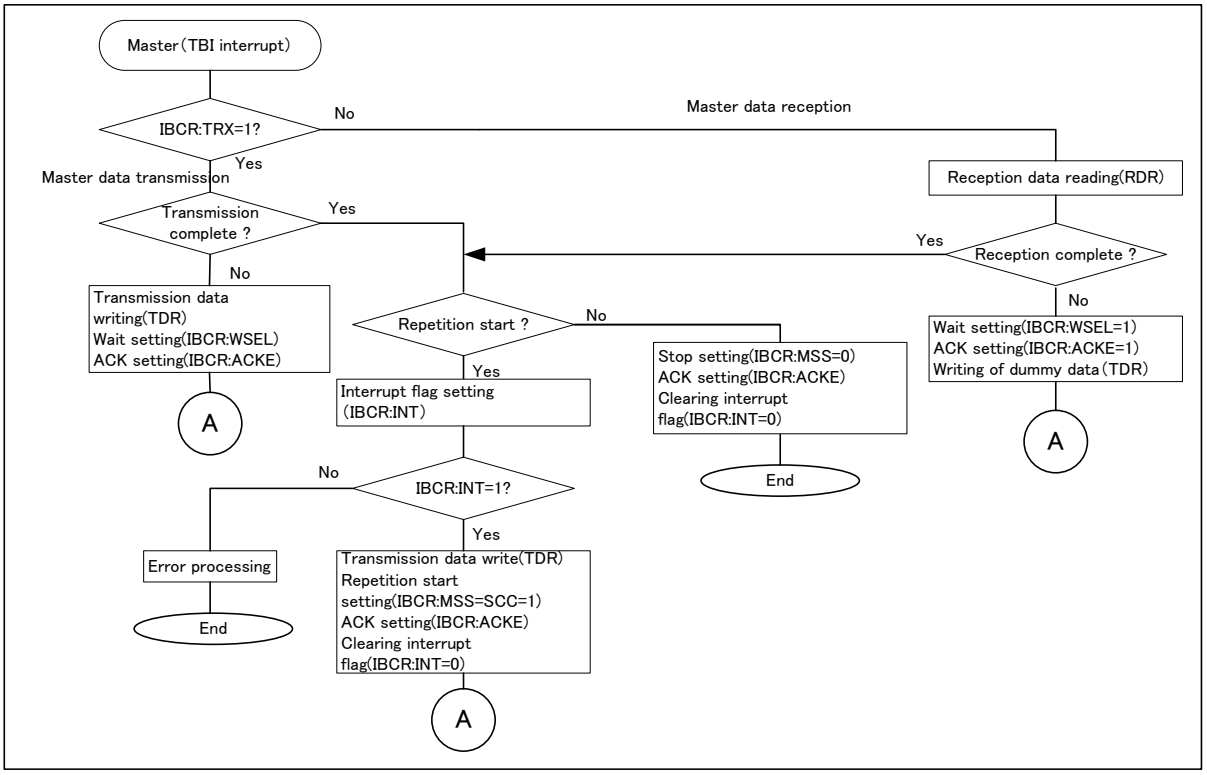


Figure 8-58 Example of I²C Flowchart (FIFO Memory not Used)
(when DMA Mode is Enabled (SSR:DMA=1)) 3/4

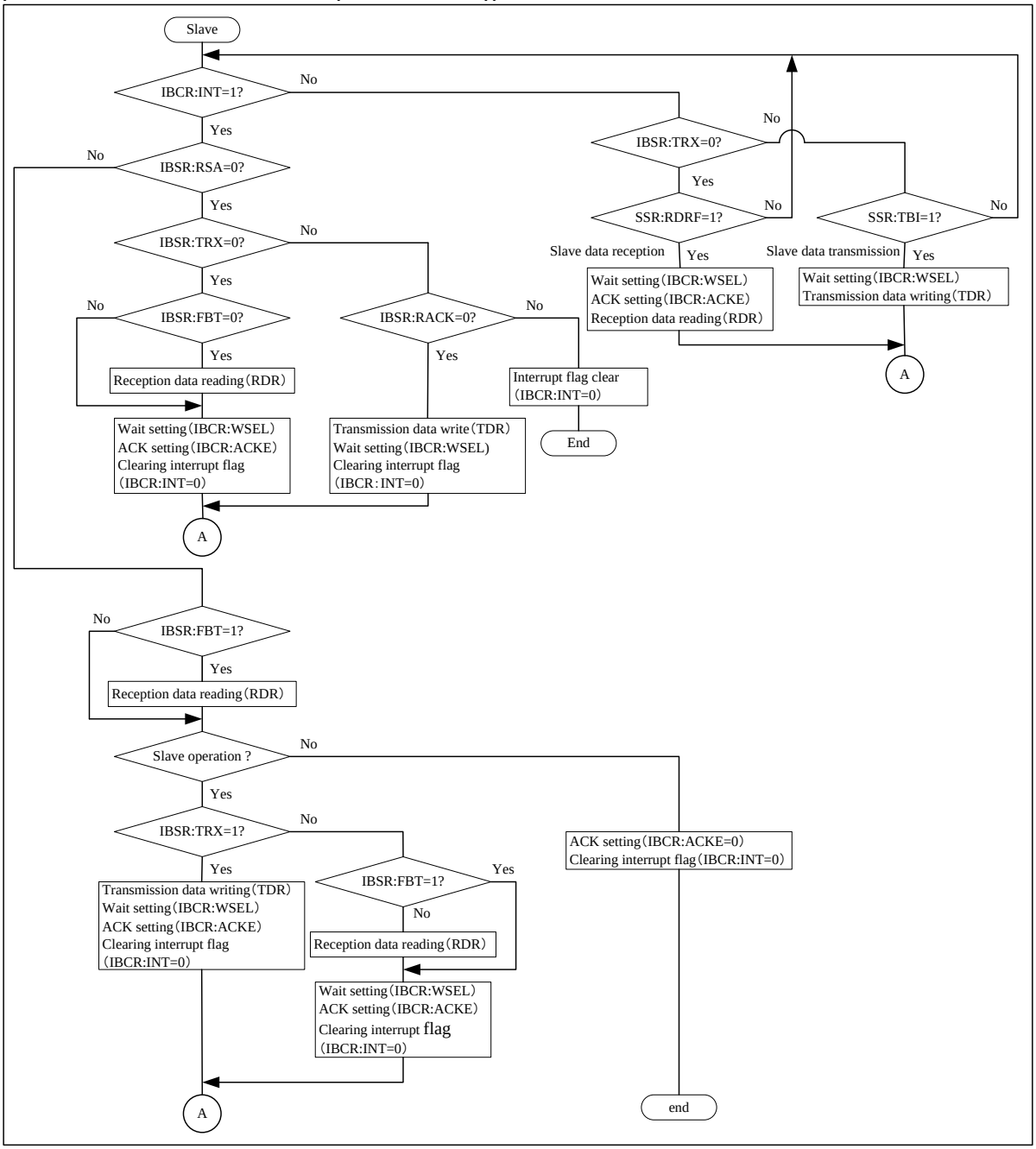
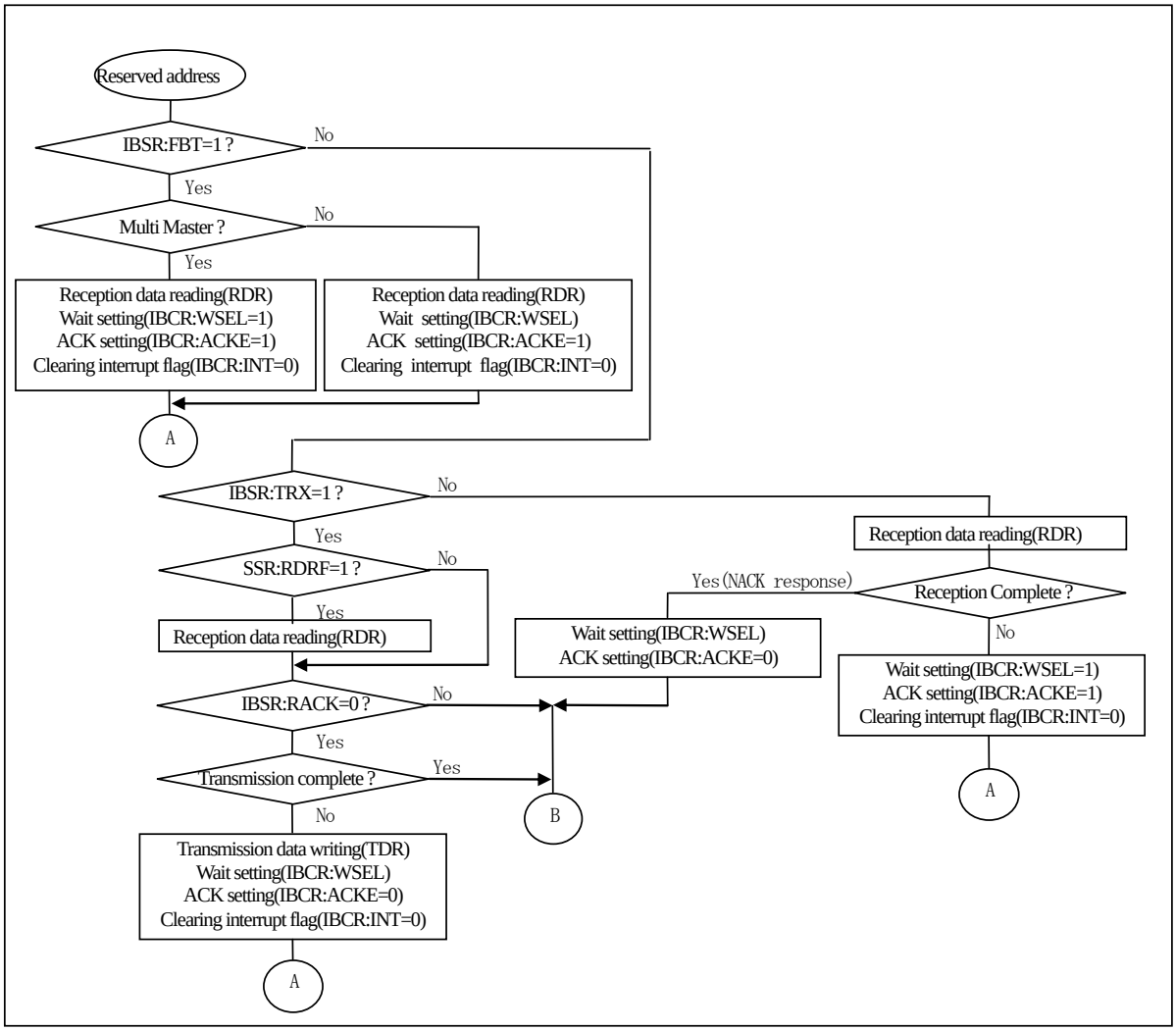


Figure 8-59 Example of I²C Flowchart (FIFO Memory not Used) (when DMA Mode is Enabled (SSR:DMA=1)) 4/4



Note:

This flowchart shows the operation setting outline by the I²C mode. It is necessary to do processing that considers error processing etc. according to the application.

Chapter 41: CAN

This chapter explains the CAN.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Restrictions

Code : FC43-2v3-91520-2-E

1. Overview

This section explains the overview of the CAN.

This series includes three CAN channels.

ch.0 (128 msgb), ch.1, ch.2 (64 msgb)

CAN is based on the CAN protocol ver. 2.0A/B which is a standard protocol for serial communication, and is widely used for automobiles, factory automation, and other industrial fields.

2. Features

This section explains the features of the CAN.

The CAN of this series has the following features:

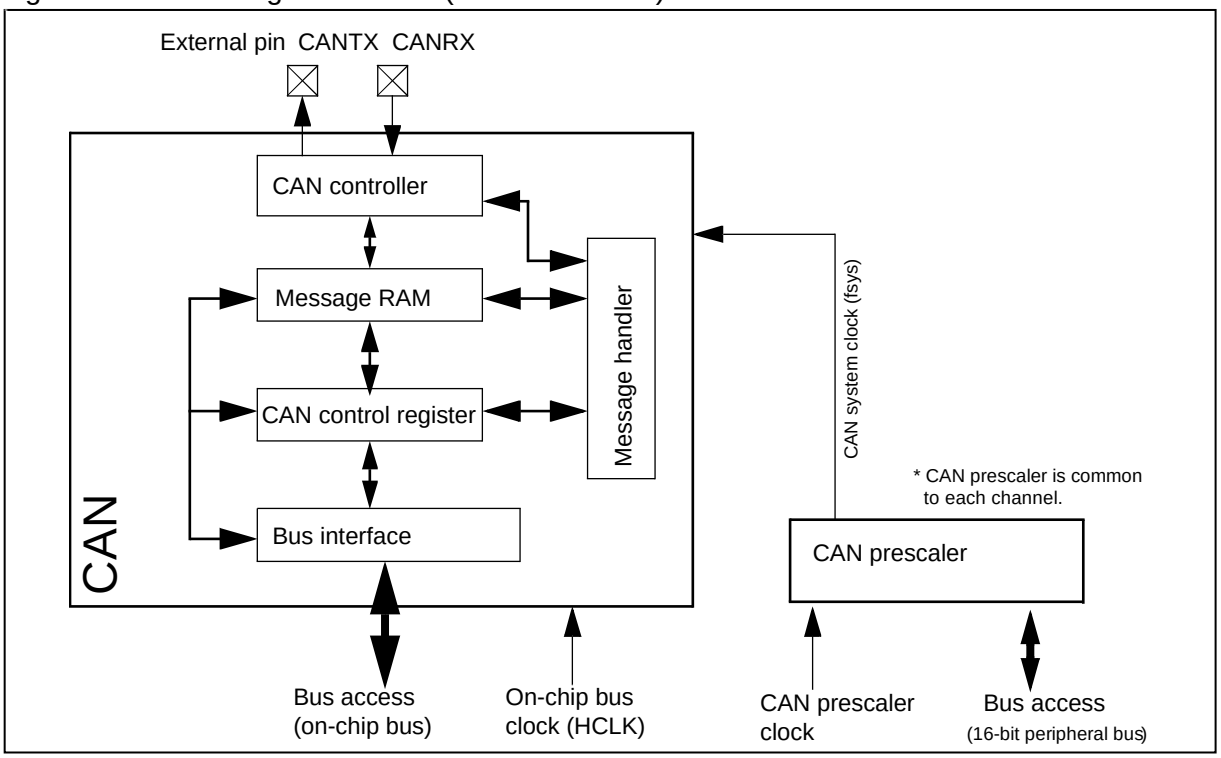
- CAN protocol ver. 2.0A/B is supported.
- Bit rates up to 1 Mbits/s are supported.
- An identification mask is applied to each message object.
- Programmable FIFO mode is supported.
- Maskable interrupts.
- Programmable loopback mode for self-test operation is supported.
- Data can be written to and read from a message buffer using an interface register.
- Support 64/128 message buffers. As the number depends on channels, see "CHAPTER : OVERVIEW".

3. Configuration

This section explains the configuration of the CAN.

A block diagram of the CAN is shown below:

Figure 3-1 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 the interface with the FR internal bus.

CAN prescaler
: Generates CAN system clocks (fsys).

4. Registers

This section explains the registers of the CAN.

4.1 Overview

4.2 Overall Control Registers

4.3 Message Interface Register

4.4 Message Object

4.5 Message Handler Registers

4.1. Overview

This section shows the overview of the registers.

The CAN includes the following registers:

- CAN control register (CTRLR)
- CAN status register (STATR)
- CAN error counter (ERRCNT)
- CAN bit timing register (BTR)
- CAN interrupt register (INTR)
- CAN test register (TESTR)
- CAN prescaler extension register (BRPER)
- IFx command request registers (IFxCREQ)
- IFx command mask registers (IFxCMSK)
- IFx mask registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx arbitration registers 1, 2 (IFxARB1, IFxARB2)
- IFx message control register (IFxMCTR) (IFxMCTR)
- IFx data registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
- CAN transmission request registers 1, 2, 3, 4 (TREQR1, TREQR2, TREQR3, TREQR4)
- CAN New Data registers 1, 2, 3, 4 (NEWDT1, NEWDT2, NEWDT3, NEWDT4)
- CAN interrupt pending registers 1, 2, 3, 4 (INTPND1, INTPND2, INTPND3, INTPND4)
- CAN message valid registers 1, 2, 3, 4 (MSGVAL1, MSGVAL2, MSGVAL3, MSGVAL4)

The CAN register is given an address space of 256 bytes (64 words) and accessible in byte or word mode. The CPU accesses the message RAM via a message interface register.

4.1.1. List of Base-addresses (Base-addr), External Pins and Buffer Size

This section shows the list of base-addresses (Base-addr), external pins and buffer size.

Channel	Base-addr	External pin name				Buffer size
		MB91F52xB, MB91F52xD		MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL		
		CANTX	CANRX	CANTX	CANRX	
0	0x2000	TX0_1	RX0(128)	TX0(128)/ TX0_1	RX0(128)	128 msgb
1	0x2100	TX1(64)	RX1_1	TX1(64)	RX1(64)/ RX1_1	64 msgb
2	0x2200	TX2(64)	RX2(64)	TX2(64)	RX2(64)	64 msgb

4.1.2. List of Overall Control Register

This section shows the list of overall control register.

Table 4-1 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	

Address	Registers				Note
	+0	+1	+2	+3	
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	

4.1.3. List of Message Interface Register

This section shows the list of message interface register.

Table 4-2 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	

Address	Registers				Note
	+0	+1	+2	+3	
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	
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	

Address	Registers				Note
	+0	+1	+2	+3	
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	
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]	
	MXtd, MDir, Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
	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	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 54_H	IF2 data B register 1 (IF2DTB1)		IF2 data B register 2 (IF2DTB2)		Byte order: Big Endian
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 60_H	IF2 data A register 2 (IF2DTA2)		IF2 data A register 1 (IF2DTA1)		Byte order: Little Endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 64_H	IF2 data B register 2 (IF2DTB2)		IF2 data B register 1 (IF2DTB1)		Byte order: Little Endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

4.1.4. List of Message Handler Register

This section shows the list of message handler register

Table 4-3 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	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 84_H	CAN transmission request register 4 (TREQR4)		CAN transmission request register 3 (TREQR3)		INTR3, 4: Read only
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	TxRqst[64:57]	TxRqst[56:49]	TxRqst[48:41]	TxRqst[40:33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 88_H Base-addr + $8C_H$	Reservation area for supporting 128 message buffers (See CAN transmission request registers (TREQR1 to TREQR4)) TREQ5 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	CAN new data register 4 (NEWDT4)		CAN new data register 3 (NEWDT3)		NEWDT3, 4: Read only
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	NewDat[64:57]	NewDat[56:49]	NewData[48:41]	NewData[40:33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 98_H Base-addr + $9C_H$	Reservation area for supporting 128 message buffers (See CAN data update registers (NEWDT1 to NEWDT4)) NEWDT5 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	
Base-addr + $A4_H$	CAN interrupt pending register 4 (INTPND4)		CAN interrupt pending register 3 (INTPND3)		INTPND3, 4: Read only
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	IntPnd[64:57]	IntPnd[56:49]	IntPnd[48:41]	IntPnd[40:33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + $A8_H$ Base-addr + AC_H	Reservation area for supporting 128 message buffers (See CAN interrupt pending registers (INTPND1 to INTPND4)) INTPND5 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 + $B4_H$	CAN message valid register 4 (MSGVAL4)		CAN message valid register 3 (MSGVAL3)		MSGVAL3, 4: Read only
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal[64:57]	MsgVal[56:49]	MsgVal[48:41]	MsgVal[40:33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $B8_H$ Base-addr + BC_H	Reservation area for supporting 128 message buffers (See CAN message valid registers (MSGVAL1 to MSGVAL4)) MSGVAL5 to MSGVAL8: 128 message buffers are supported				

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)

4.2.1. CAN Control Register : CTRLR

The bit configuration of the CAN control register is shown.

Controls the operation mode of the CAN controller.

■ CAN Control Register (upper byte): Address Base + 00_H (Access: Byte, Half-word, Word)

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

■ CAN Control Register (lower byte): Address Base + 01_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Test	CCE	DAR	Reserved	EIE	SIE	IE	Init
Initial value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	R/W

[bit15 to bit8] Reserved bit

The read value is always "0". Be sure to write "0" to these bits.

[bit7]: Test mode enable bit

Test	Function
0	Normal operation [Initial value]
1	Test mode

Note:

Set "1" to the Test bit only when the INIT bit is "1".

[bit6]: 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]: Automatic retransmission disable bit

The CAN controller retransmits the frame automatically when it loses the arbitration or when an error is detected during transfer. To enable automatic retransmission, set "0" to the DAR bit. In order to operate CAN in Time

Triggered CAN environments, "1" needs to be set to the DAR bit.

DAR	Function
0	Enables the automatic retransmission of the message when CAN loses the arbitration or when an error is detected. [Initial value]
1	Disables automatic retransmission.

Note:

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 "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 bit

The read value is always "0". When writing to this bit, set "0".

[bit3]: 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]: 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]: Interrupt enable bit

IE	Function
0	Disables interrupt. [Initial value]
1	Enables interrupt.

[bit0]: 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.
- When the Init bit is set to "1" during transmission, set the Init bit to "1" after the transmission is completed. When the Init bit was set to "1" while the transmission was in progress, execute a transmission cancellation for the message buffer with "1" specified in the transmission request bit (TxRqst) (set the TxRqst bit to "0") and then set the Init bit to "0". After that, set the transmission request (TxRqst="1") after progress more than two bits time of CAN.
- It is necessary to initialize the CAN controller by writing "1" to the Init bit before transiting to the low-power consumption mode (stop mode, clock mode) and before changing the supply clock.
- 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)

4.2.2. CAN Status Register : STATR

The bit configuration of the CAN status register is shown.

Displays the CAN and CAN bus statuses.

■ CAN Status Register (upper byte): Address Base + 02_H (Access: Byte, Half-word, Word)

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

■ CAN Status Register (lower byte): Address Base + 03_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BOff	EWarn	EPass	RxOk	TxOk	LEC[2:0]		
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,W	R,W	R,WX	R,WX	R,WX

[bit15 to bit8]: Reserved bit

The read value is always "0". Be sure to write "0" to these bits.

[bit7]: 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]: 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]: 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]: 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]: 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] : Last error code bits

The LEC bit holds the code that indicates the last error occurred on the CAN bus. This bit is cleared to "0" when a message transfer (reception/transmission) completes without error. The undetected code "111_B" can be used for checking the code update.

LEC[2:0]	State	Function
000	Normal	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)

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).

4.2.3. CAN Error Counter : ERRCNT

The bit configuration of the CAN error counter is shown.

Indicates reception error passive display, reception error counter and transmission error counter.

■ CAN Error Counter Register (upper byte): Address Base + 04_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	RP	REC[6:0]						
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Error Counter Register (lower byte): Address Base + 05_H (Access: Byte, Half-word, Word)

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

[bit15]: Reception error passive display

RP	Function
0	Indicates that it is not the error passive state. [Initial value]
1	Indicates that the error passive state that is defined in the CAN specification has been reached.

[bit14 to bit8] : Reception error counter

Reception error counter value. The range for the reception error counter values is 0 to 127.
When the reception error counter is greater than or equal to 128, "1" is set to the RP bit and the reception error counter is not updated.

Example:

When RP=0 and REC[6:0]=127 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=127.
When RP=0 and REC[6:0]=126 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=126.
When RP=0 and REC[6:0]=119 is incremented by 8 for reception error, the result is RP=0 and REC[6:0]=127.
When RP=1, REC[6:0]=126 and reception has ended successfully, the result is RP=0 and REC[6:0]=125.

[bit7 to bit0] : Transmission error counter

Transmission error counter value. The range for the transmission error counter values is 0 to 255.
When the transmission error counter is greater than or equal to 256, "1" is set to the Init bit of the CAN control register and the transmission error counter is not updated.

Example:

When Init=0 and TEC[7:0]=255 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=255.
When Init=0 and TEC[7:0]=254 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=254.
When Init=0 and TEC[7:0]=247 is incremented by 8 for transmission error, the result is Init=0 and TEC[7:0]=255.

4.2.4. CAN Bit Timing Register : BTR

The bit configuration of the CAN bit timing register is shown.

Sets the prescaler and bit timing.

■ CAN Bit Timing Register (upper byte): Address Base + 06_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	TSeg2			TSeg1			
Initial value	0	0	1	0	0	0	1	1
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ CAN Bit Timing Register (lower byte): Address Base + 07_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SJW		BRP					
Initial value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] : Reserved bit

The read value is always "0". Be sure to write "0" to this bit.

[bit14 to bit12]: Time segment 2 setting bits

Valid setting values are 0 to 7. TSeg2+1 bit value is time segment 2.
Time segment 2 corresponds to the phase buffer segment (PHASE_SEG2) based on the CAN specification.

[bit11 to bit8]: Time segment 1 setting bits

Valid setting values are 1 to 15. 0 cannot be set. TSeg1+1 bit value is time segment 1.
Time segment 1 corresponds to the propagation segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) based on the CAN specification.

[bit7, bit6]: Resynchronization jump width setting bits

Valid setting values are 0 to 3. The SJW+1 bit value is the resynchronization jump width.

[bit5 to bit0]: Baud rate prescaler setting bits

Valid setting values are 0 to 63. The BRP+1 bit value is the baud rate prescaler.

Divides frequency for system clock (fsys) and determines the basic unit time (tq) of the CAN controller.

Note:

When "1" is set to the CCE and Init bits of the CAN control register (CTRLR), set the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER).

4.2.5. CAN Interrupt Register : INTR

The bit configuration of the CAN interrupt register is shown.

Displays the message interrupt and status interrupt codes.

■ CAN Interrupt Register (upper byte): Address Base + 08_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	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 + 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

If more than one interrupt codes are pending, the CAN interrupt register (INTR) will indicate the interrupt code of the highest priority. If a higher-priority interrupt code is generated when an interrupt code is set to the CAN interrupt register (INTR), the CAN interrupt register (INTR) is updated to the higher-order interrupt code.

Higher orders are given to the status interrupt code (8000_H), message interrupt (0001_H, 0002_H, 0003_H, ..., 0040_H) in descending order. (For 64msgb. Same for 128msgb.)

When the IntId bit is other than 0000_H and the IE bit of the CAN control register (CTRLR) is set to "1", the interrupt signal for CPU is active. When the IntId bit is 0000_H (an interrupt factor is reset) or the IE bit of the CAN control register (CTRLR) is reset to "0", the interrupt signal is inactive.

If the IntPnd bit of the target message objects (for message objects, see "4.4 Message Object") is cleared to "0", the message interrupt code will be cleared.

Status interrupt code will be cleared when the CAN status register (STATR) is read.

IntId	Function
0000_H	No interrupt
(For 64msgb) 0001_H to 0040_H (For 128msgb) 0001_H to 0080_H	The message object number is indicated as an interrupt factor. (Message interrupt code)
(For 64msgb) 0041_H to $7FFF_H$ (For 128msgb) 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

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 "5.7 Test Mode".

■ CAN Test Register (upper byte): Address Base + $0A_H$ (Access: Byte, Half-word, Word)

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

■ CAN Test Register (lower byte): Address Base + $0B_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	Reserved
Initial value	r	0	0	0	0	0	0	0
Attribute	R,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.

[bit15 to bit8] : Reserved bit

The read value is always "0". Be sure to write "0" to these bits.

[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 bit

The read value is always "0". Be sure to write "0" to these bits.

Notes:

- After setting "1" to the Test bit of the CAN control register (CTRLR), write into the register. The test mode is valid when the Test bit of the CAN control register (CTRLR) is set to "1". The CAN controller transits from the test mode to the normal mode when the Test bit of the CAN control register (CTRLR) is set to "0".
- Messages cannot be sent when the Tx bit is set to a value other than "00".

4.2.7. CAN Prescaler Extension Register : BRPER

The bit configuration of the CAN prescaler extension register is shown.

Extends the prescaler used in the CAN controller by combining the prescaler set at the CAN bit timing.

■ CAN Prescaler Extension Register (upper byte): Address Base + $0C_H$ (Access: Byte, Half-word, Word)

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

■ CAN Prescaler Extension Register (lower byte): Address Base + $0D_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	BRPE			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit15 to bit4] : Reserved bit

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 bits), BRP (LSB:6 bits)} + 1 value is the prescaler of the CAN controller.

4.3. Message Interface Register

This section shows the message interface register.

Two pairs of message interface registers are provided 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 "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 except 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 4-4 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 4-4 IF1, IF2 Message Interface Registers

Address	IF1 register set	Address	IF2 register set
Base + 10_H	IF1 command request	Base + 40_H	IF2 command request
Base + 12_H	IF1 command mask	Base + 42_H	IF2 command mask
Base + 14_H	IF1 mask 2	Base + 44_H	IF2 mask 2
Base + 16_H	IF1 mask 1	Base + 46_H	IF2 mask 1
Base + 18_H	IF1 arbitration 2	Base + 48_H	IF2 arbitration 2
Base + $1A_H$	IF1 arbitration 1	Base + $4A_H$	IF2 arbitration 1
Base + $1C_H$	IF1 message control	Base + $4C_H$	IF2 message control
Base + 20_H	IF1 data A1	Base + 50_H	IF2 data A1
Base + 22_H	IF1 data A2	Base + 52_H	IF2 data A2
Base + 24_H	IF1 data B1	Base + 54_H	IF2 data B1
Base + 26_H	IF1 data B2	Base + 56_H	IF2 data B2

4.3.1. IFx Command Request Register : IFxCREQ

The bit configuration of the IFx command request register is shown.

This register selects the message number of the message RAM and transfers the message between the message RAM and the message buffer register. In addition, IF1 is used for transmission control and IF2 is used for reception control in the basic mode for tests.

■ IFx Command Request Register (upper byte): Address Base + 10_H & Base + 40_H (Access: Byte, Half-word, Word)

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

■ IFx Command Request Register (lower byte): Address Base + 11_H & Base + 41_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Message Number							
Initial value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Immediately after the message number is written into the IFx command request register (IFxCREQ), the message transfer between the message RAM and the message buffer register (mask, arbitration, message control and data register) is started. This writing operation indicates that "1" is set to the BUSY bit and a message is being transferred. When the transfer is completed, the BUSY bit is reset to "0".

When "1" is set to the BUSY bit, the CPU will be kept waiting until the BUSY bit becomes "0" if the CPU accesses to the message interface register (3 to 6 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

(1) 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.

(2) 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 bit

The read value is always "0". Be sure to write "0" to these bits.

[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 "5.7 Test Mode" for the details of the basic mode.)

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 + 12_H & Base + 42_H (Access: Byte, Half-word, Word)

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

■ IFx Command Mask Register (lower byte): Address Base + 13_H & Base + 43_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WR/RD	Mask	Arb	Control	CIP	TxRqst/ NewDat	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 bit

The read value is always "0". Be sure to write "0" to these bits.

[bit7] WR/RD : Write/read control bit

WR/RD	Function
0	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).

(1) 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*. [Initial value]
1	Indicates updating the mask data (ID mask + MDir + MXtd) of message object*.

[bit5] Arb : Arbitration data update bit

Arb	Function
0	Indicates not updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object*. [Initial value]
1	Indicates updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object*.

[bit4] Control : Control data update bit

Control	Function
0	Indicates not updating the control data (IFx message control register (IFxMCTR)) of message object*. [Initial value]
1	Indicates updating the control data (IFx message control register (IFxMCTR)) of message object*.

[bit3]: CIP: Interrupt clear bit

Operation of CAN controller will not be affected whether "0" or "1" is set.

[bit2] TxRqst/NewDat : Message transmission request bit

TxRqst/ NewDat	Function
0	Indicates not changing the TxRqst bit of message object* and CAN transmission request register (TREQR). [Initial value]
1	Indicates that "1" is set to the TxRqst bit of message object* 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*. [Initial value]
1	Indicates updating Data 0 to 3 of message object*.

[bit0] Data B : Data 4 to 7 update bit

Data B	Function
0	Indicates not updating Data 4 to 7 of message object*. [Initial value]
1	Indicates updating Data 4 to 7 of message object*.

*: See "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.

(2) 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* to IFx mask registers 1, 2 (IFxMSK1, IFxMSK2). [Initial value]
1	Indicates transferring data (ID mask +MDir + MXtd) from message object* 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* to IFx arbitration 1, 2 (IFxARB1, IFxARB2). [Initial value]
1	Indicates transferring data (ID + Dir + Xtd + MsgVal) from message object* to IFx arbitration 1, 2 (IFxARB1, IFxARB2) .

[bit4] Control : Control data update bit

Control	Function
0	Indicates not transferring data from message object* to IFx message control register (IFxMCTR). [Initial value]
1	Indicates transferring data from message object* to IFx message control register (IFxMCTR).

[bit3] CIP : Interrupt clear bit

CIP	Function
0	Indicates holding the IntPnd bit of message object* and CAN interrupt pending register (INTPND). [Initial value]
1	Indicates clearing the IntPnd bit of message object* 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* and CAN data update register. [Initial value]
1	Indicates clearing the NewDat bit of message object* 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* and CAN data registers A1, A2. [Initial value]
1	Indicates updating data of message object* 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* and CAN data registers B1, B2. [Initial value]
1	Indicates updating data of message object* and CAN data registers B1, B2.

*: See "4.4 Message Object".

Notes:

- It is possible to reset the IntPnd and NewDat bits to "0" by reading access to the message object. However, for the IntPnd and NewDat bits of the IFx message control register (IFxMCTR), the IntPnd and NewDat bits are stored before they are reset by reading access.
- It becomes invalid in the test basic mode.

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 "4.4 Message Object" for the functions of each bit.

■ IFx Mask Register 2 (upper byte): Address Base + 14_H & Base + 44_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MXtd	MDir	Reserved	Msk28 to Msk24				
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R1,W1	R/W	R/W	R/W	R/W	R/W

■ IFx Mask Register 2 (lower byte): Address Base + 15_H & Base + 45_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Msk23 to Msk16							
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ IFx Mask Register 1 (upper byte): Address Base + 16_H & Base + 46_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Msk15 to Msk8							
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ **IFx Mask Register 1 (lower byte): Address Base + 17_H & Base + 47_H (Access: Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Msk7 to Msk0							
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

See "4.4 Message Object" for explanation of bits.
For the reserved bit (bit13 of IFx mask register 2), "1" is read out. Be sure to write "0" to these bits.

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 "4.4 Message Object" for the functions of each bit.

■ **IFx Arbitration Register 2 (upper byte): Address Base + 18_H & Base + 48_H (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal	Xtd	Dir	ID28 to ID24				
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ **IFx Arbitration Register 2 (lower byte): Address Base + 19_H & Base + 49_H (Access: Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ID23 to ID16							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ IFx Arbitration Register 1 (upper byte): Address Base + $1A_H$ & Base + $4A_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ID15 to ID8							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ IFx Arbitration Register 1 (lower byte): Address Base + $1B_H$ & Base + $4B_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ID7 to ID0							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

See "4.4 Message Object" for explanation of bits.

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.

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 bits will display the DLC of message received. Other control bits will operate as disabled ("0").

See "4.4 Message Object" for the functions of each bit.

■ IFx Message Control Register (upper byte): Address Base + $1C_H$ & Base + $4C_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

■ IFx Message Control Register (lower byte): Address Base + $1D_H$ & Base + $4D_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EoB	Reserved	Reserved	Reserved	DLC3-0			
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

See "4.4 Message Object" for explanation of bits.

Note:

TxRqst, NewDat, and IntPnd bits operate differently depending on the settings of the WR/RD bit in the IFx command mask register (IFxCMSK).

- If the transfer direction is "write" (IFx command mask register (IFxCMSK): WR/RD="1").
- The TxRqst bit of this register will only be enabled when TxRqst/NewDat in the IFx command mask register (IFxCMSK) is set to "0".
- If the transfer direction is "read" (IFx command mask register (IFxCMSK): WR/RD="0").
- The IntPnd bit before it has been reset will be stored to this register when the message object and the IntPnd bit of the CAN interrupt pending register (INTPND) are reset by a write operation to the IFx command request register (IFxCREQ) after setting the CIP bit of the IFx command mask register (IFxCMSK) to "1".
- The NewDat bit before it has been reset will be stored to this register when the message object and the NewDat bit of the CAN data update register are reset by a write operation to the IFx command request register (IFxCREQ) after setting the TxRqst/NewDat bit of the IFx command mask register (IFxCMSK) to "1".

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.

4.4. Message Object

This section shows the message object.

The message RAM has 64 (128; depending on a 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.

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 4-5 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 of message RAM to "0".

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, When receiving a message, they will be used with Msk28 to Msk0, MXtd and MDir bits in the acceptance filter. ID, IDE, RTR, DLC and DATA for data frame or remote frame passing through the acceptance filter will be stored in the ID28 to ID0, Xtd, Dir, DLC3 to DLC0, Data7 to Data0 of the message objects. Xtd indicates whether the message object is an extension frame or standard frame, a 29-bit ID (extension frame) will be received if Xtd is "1", and an 11-bit ID (standard frame) will be received if Xtd is "0".
If the received data frame or remote frame matches one or more message objects, it will be stored to the lowest matched message number. (See reception message acceptance filter in "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 Msk0, MXtd, and MDir.
1	Uses Msk28 to Msk0, MXtd, and MDir.

Notes:

- Change the UMask bit while the Init bit of the CAN control register (CTRLR) is "1" or while the MsgVal bit is "0".
- When the Dir bit is "1" and the RmtEn bit is "0", it will operate differently depending on the UMask bit setting.
  - If the UMask bit is "1", the TxRqst bit will be reset to "0" when the remote frame is received through the acceptance filter. At this time, the received ID, IDE, RTR and DLC will be stored to the message object, the NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
  - If the UMask bit is "0", the TxRqst bit will remain unchanged by the remote frame reception; and it will ignore the remote frame.

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 of the message object and those for the IDE bit in the received frame. The IDE in the received frame determines whether to compare it as a standard frame ID or an extended frame ID.
1	Compares the values between those set to the Xtd of the message object and those for the IDE in the received frame.

Note:

If an 11-bit ID (standard frame) is set to the message object, ID of the received data frame will be written to ID28 to ID18. Msk28 to Msk18 are used for ID masks.

Dir: Message direction bit

Dir	Function
0	Indicates the reception direction. The remote frame will be transmitted when the TxRqst is set to "1", and the data frame that has passed through the acceptance filter will be received when TxRqst is set to "0".
1	Indicates the transmission direction. Data frame will be transmitted when the TxRqst is set to "1". If the TxRqst is "0" and the RmtEn is set to "1", the CAN controller itself sets 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 "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 128 messages (ch.0) and 2 to 64 messages (ch.1, ch.2).
- Always set the EoB bit to "1" in the case of a single message object (when FIFO is not used).

NewDat: Data update bit

NewDat	Function
0	Valid data does not exist.
1	Valid data exists.

MsgLst: Message lost

MsgLst	Function
0	No message lost occurs.
1	Message lost occurs.

Note:

The MsgLst bit is only enabled when the Dir bit is "0" (reception direction).

RxIE: Reception interrupt flag enable bit

RxIE	Function
0	The IntPnd remains unchanged after successful frame reception.
1	The IntPnd is set to "1" after successful frame reception.

TxIE: Transmission interrupt flag enable bit

TxIE	Function
0	The IntPnd remains unchanged after successful frame transmission.
1	The IntPnd is set to "1" after successful frame transmission.

IntPnd: Interrupt pending bit

IntPnd	Function
0	No interrupt factor exists.
1	Interrupt factor exists. If no other high priority interrupt exists, the IntId bit of the CAN interrupt register (INTR) will indicate this message object.

RmtEn: Remote enable

RmtEn	Function
0	The TxRqst remains unchanged by remote frame reception.
1	The TxRqst will be set to "1" if a remote frame is received while the Dir bit is "1".

Note:

When the Dir bit is "1" and the RmtEn bit is "0", it will operate differently depending on the UMask setting.

- If the UMask is "1", the TxRqst bit will be reset to "0" when the remote frame is received through the acceptance filter. At this time, the received ID, IDE, RTR and DLC will be stored in the message object, the NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
- If the UMask is "0", the TxRqst bit will remain unchanged by the remote frame reception; and it will ignore the remote frame.

TxRqst: Transmission request bits

TxRqst	Function
0	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.

4.5. Message Handler Registers

Message handler registers are shown.

All message handler registers are for reading only. The TxRqst, NewDat, IntPnd, MsgVal, and IntId bits of the message object are used to display a status.

- CAN transmission request registers 1 to 4 (TREQR1 to TREQR4)
- CAN data update registers 1 to 4 (NEWDT1 to NEWDT4)
- CAN interrupt pending registers 1 to 4 (INTPND1 toINTPND4)
- CAN message valid registers 1 to 4 (MSGVAL1 to MSGVAL4)

4.5.1. CAN Transmission Request Registers : TREQR1 to TREQR4

The bit configuration of the CAN transmission request registers is shown.

Displays the TxRqst bit of all message objects. It is possible to check which message objects transmission request is pending by reading the TxRqst bits.

CAN Transmission Request Register 4 (upper byte): Address Base + 84_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst64 to TxRqst57							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 4 (lower byte): Address Base + 85_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst56 to TxRqst49							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 3 (upper byte): Address Base + 86_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst48 to TxRqst41							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 3 (lower byte): Address Base + 87_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst40 to TxRqst33							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 2 (upper byte): Address Base + 80_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst32 to TxRqst25							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Transmission Request Register 2 (lower byte): Address Base + 81_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst24 to TxRqst17							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Transmission Request Register 1 (upper byte): Address Base + 82_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst16 to TxRqst9							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Transmission Request Register 1 (lower byte): Address Base + 83_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst8 to TxRqst1							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

TxRqst64 to TxRqst1: Transmission request bits

TxRqst64 to TxRqst1	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

When the WR/RD is set to "1" and the TxRqst is set to "1" for the IFx command mask register (IFxCMSK), it is possible to set the TxRqst of a specific object by writing to the IFx command request register (IFxCREQ).

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the TxRqst is set to "0", the Control is set to "1", and the TxRqst of the IFx message control register (IFxMCTR) is set to "1", it is possible to set the TxRqst of a specific object by writing data to the IFx command request register (IFxCREQ).

The bit will be set by a reception of remote frame that has passed the acceptance filter when the Dir bit and RmtEn bit are set to "1" respectively.

Reset condition

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the TxRqst is set to "0", the Control is set to "1", and the TxRqst of the IFx message control register (IFxMCTR) is set to "0", it is possible to reset the TxRqst of a specific object by writing data to the IFx command request register (IFxCREQ).

When frame transmission is completed successfully, the bit will be reset.

The bit will be reset by a reception of remote frame that has passed the acceptance filter when the Dir bit is set to "1", the RmtEn bit is set to "0", and the UMask is set to "1".

See the following table to confirm the transmission request bit for CAN macro equipped with 128 message buffers.

Table 4-6 Transmission Request Bit for CAN Macro Equipped with 128 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
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".

4.5.2. CAN Data Update Registers : NEWDT1 to NEWDT4

The bit configuration of the CAN data update registers is shown.

Displays the NewDat bit of all message objects. It is possible to check which message objects data has been updated by reading the NewDat bit.

■ CAN Data Update Register 4 (upper byte): Address Base + 94_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat64 to NewDat57							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 4 (lower byte): Address Base + 95_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat56 to NewDat49							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 3 (upper byte): Address Base + 96_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat48 to NewDat41							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 3 (lower byte): Address Base + 97_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat40 to NewDat33							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 2 (upper byte): Address Base + 90_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat32 to NewDat25							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 2 (lower byte): Address Base + 91_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat24 to NewDat17							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 1 (upper byte): Address Base + 92_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat16 to NewDat9							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Data Update Register 1 (lower byte): Address Base + 93_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat8 to NewDat1							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

NewDat64 to NewDat1: Data update bits

NewDat64 to NewDat1	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 128 message buffers.

Table 4-7 Data Update Bit for CAN Macro Equipped with 128 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
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 to 113	NewDat112 to 105	NewDat104 to 97

4.5.3. CAN Interrupt Pending Registers :INTPND1 to INTPND4

The bit configuration of the CAN interrupt pending registers is shown.

Displays the IntPnd bit of all message objects. It is possible to check which message objects interrupt is pending by reading the IntPnd bit.

■ CAN Interrupt Pending Register 4 (upper byte): Address Base + $A4_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd64 to IntPnd57							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 4 (lower byte): Address Base + $A5_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd56 to IntPnd49							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 3 (upper byte): Address Base + $A6_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd48 to IntPnd41							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 3 (lower byte): Address Base + $A7_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd40 to IntPnd33							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 2 (upper byte): Address Base + $A0_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd32 to IntPnd25							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 2 (lower byte): Address Base + $A1_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd24 to IntPnd17							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 1 (upper byte): Address Base + $A2_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd16 to IntPnd9							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Interrupt Pending Register 1 (lower byte): Address Base + $A3_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd8 to IntPnd1							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

● IntPnd64 to IntPnd1: Interrupt pending bits

IntPnd64 to IntPnd1	Function
0	No interrupt factor exists.
1	Interrupt factor exists.

Set/reset conditions of the IntPnd bits are shown below.

Set condition

If the TxIE is set to "1", the IntPnd bit will be set after the frame transmission has ended successfully.

If the RxIE is set to "1", the bit will be set after the frame reception that has passed the acceptance filter completed successfully.

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the Control is set to "1", and the IntPnd of the IFx message control register is set to "1", it is possible to set the IntPnd 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 128 message buffers.

Table 4-8 Interrupt Pending Bit for CAN Macro Equipped with 128 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
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

4.5.4. CAN Message Valid Registers : MSGVAL1 to MSGVAL4

The bit configuration of the CAN message valid registers is shown.

Displays the MsgVal bit of all message objects. It is possible to check which message object is valid by reading the MsgVal bit.

■ CAN Message Valid Register 4 (upper byte): Address Base + $B4_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal64 to MsgVal57							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 4 (lower byte): Address Base + $B5_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal56 to MsgVal49							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 3 (upper byte): Address Base + $B6_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal48 to MsgVal41							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 3 (lower byte): Address Base + $B7_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal40 to MsgVal33							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 2 (upper byte): Address Base + $B0_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal32 to MsgVal25							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 2 (lower byte): Address Base + $B1_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal24 to MsgVal17							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 1 (upper byte): Address Base + $B2_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal16 to MsgVal9							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

■ CAN Message Valid Register 1 (lower byte): Address Base + $B3_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal8 to MsgVal1							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

● MsgVal64 to MsgVal1: Message valid bits

MsgVal64 to MsgVal1	Function
0	Message object is invalid. Message will not be transmitted/received.
1	Message object is valid. Message transmission/reception is possible.

Set/reset conditions of the MsgVal bits are shown below.

Set condition

When the WR/RD of the IFx command mask register is set to "1", the Arb is set to "1", and the MsgVal bit of the IFx arbitration register 2 is set to "1", it is possible to set the MsgVal bit of a specific object by writing data to the IFx command request register (IFxCREQ).

Reset condition

When the WR/RD of the IFx command mask register is set to "1", the Arb is set to "1", and the MsgVal bit of the IFx arbitration register 2 is set to "0", it is possible to clear the MsgVal bit of a specific object by writing data to the IFx command request register (IFxCREQ).

See the following table to confirm the message valid bit for CAN macro equipped with 128 message buffers.

Table 4-9 Message Valid Bit for CAN Macro Equipped with 128 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
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

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

5.1. Message Object

The message object is shown.

This section explains the message object and interface of message RAM.

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).

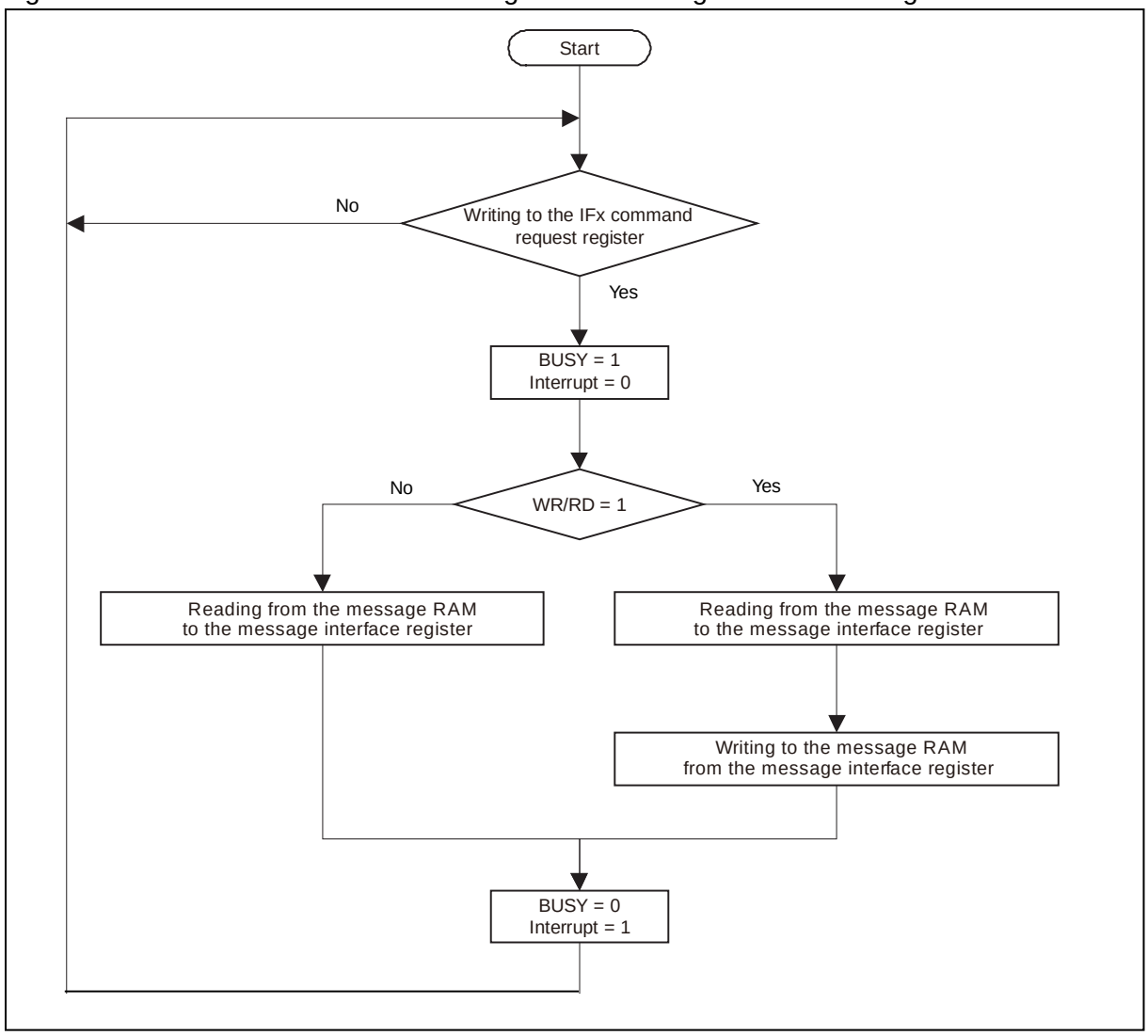
5.1.2. Data Transmission/Reception with Message RAM

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 5-1).

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 5-1 Data Transfer between Message Interface Register and Message RAM



5.2. Message Transmission Operation

Message transmission operation is shown.

This section explains the setting method and transmission operation of the transmission message object.

5.2.1. Message Transmission

Message transmission is shown.

If there is no data transfer between the message interface register and message RAM, the MsgVal bit of the CAN message valid register (MSGVAL) and the TxRqst bit of the CAN transmission request register (TREQR) will be

evaluated. Of all message objects with pending transmission request, a valid message object having the highest priority will be transferred to the transmission shift register. The NewDat bit of the message object will be cleared to "0" at this time.

The TxRqst bit will be reset to "0" if there is no new data in the message object (NewDat=0) when the transmission has ended successfully. If the TxIE is set to "1", the IntPnd bit will be set to "1" after the transmission has ended successfully. If the CAN controller has lost the arbitration on the CAN bus or an error has occurred during the transfer, message will be retransmitted immediately when the CAN bus becomes idle.

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 128 (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".

5.2.3. Transmission Message Object Setting

Transmission message object setting is shown.

The initialization method for the transmission object is shown below:

Table 5-1 Transmission Message Object Initialization

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The IFx arbitration register (ID28 to ID0 and Xtd bit) is provided by the application, and it defines the ID and type of

the transmission message.

ID28 to ID18 will be used and ID17 to ID0 will be disabled if standard frame (11-bit ID) has been set. ID28 to ID0 will be used if extended frame (29-bit ID) has been set.

If the TxIE bit is set to "1", the IntPnd bit will be set to "1" after the transmission of the message object has ended successfully.

If the RmtEn bit is set to "1", the TxRqst bit will be set to "1" and the data frame will be transmitted automatically after receiving the matching remote frame.

Settings for the data registers (DLC3 to 0, Data0 to 7) are provided by the application.

When UMask=1, the IFx mask register (Msk28 to 0, UMask, MXtd and MDir bits) will receive the remote frame having the ID that has been grouped by the mask setting, and then will be used to allow the transmission (sets the TxRqst bit to "1"). See the heading "Remote frame" in "5.3 Message Reception Operation" for details.

Note:

Mask is not allowed for the Dir bit of the IFx mask register.

5.2.4. Update of Transmission Message Object

Update of transmission message object is shown.

CPU can update the data of the transmission message object via the message interface register.

Data of the transmission message object will be written in units of 4 bytes of the corresponding IFx data register (IFxDTx) (in unit of the IFx data register A (IFxDTAx) or IFx data register B (IFxDTBx)). Therefore, it is not possible to change only 1 byte of the transmission message object.

0087_H will be written to the IFx command mask register (IFxCMSK) first when updating 8-byte data. Then, data of the transmission message object (8-byte data) will be updated and "1" will be written to the TxRqst bit at the same time when a message number is written to the IFx command request register (IFxCREQ).

If the NewDat bit and TxRqst bit are both "1", the NewDat bit will be reset to "0" when the transmission starts.

Notes:

- When updating data, perform it in units of 4 bytes of the IFx data register A(IFxDTAx) or IFx data register B(IFxDTBx).
- When the TxRqst bit is "1", do not change the message objects of ID28 to 0, DLC3 to 0, Xtd, and Data7 to 0. Otherwise, message objects before and after the change may be transmitted in a mixed way or message objects after the change may not be transmitted. Change them when the TxRqst bit is "0".

5.3. Message Reception Operation

Message reception operation is shown.

This section explains the setting method and reception operation of the reception message object.

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).

5.3.2. Reception Priority

Reception priority is shown.

Reception priority of a message object is determined by its message number. Message object 1 has the highest priority; and message object 64 (or the maximum equipped message object number) has the lowest priority. If 2 or more message objects match in the acceptance filter, the one having the smaller message number will be the reception message object.

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.

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"
   Even if the received remote frame matches the message object, it will not be processed as being received and will be disabled. (The TxRqst bit of the message object will remain unchanged.)
3. Dir="1" (Transmission direction), RmtEn="0", UMask="1"
   If the received remote frame matches the message object, the TxRqst bit of this message object will be reset to "0", and the remote frame will be processed as a reception data frame. The received arbitration field and control field (ID + IDE + RTR + DLC) will be stored to the message object in message RAM, and the NewDat bit of this message object will be set to "1". Data field of the message object will be unchanged.

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 5-2 Reception Message Object Initialization

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The IFx arbitration register (ID28 to 0 and Xtd bit) is provided by the application; and it defines the ID and type of the reception message to be used in the acceptance filter.

ID28 to ID18 will be used and ID17 to ID0 will be disabled if standard frame (11-bit ID) has been set. ID17 to ID0 will be reset to "0" when a standard frame is received. ID28 to ID0 will be used if extended frame (29-bit ID) has been set.

If the RxIE bit is set to "1", the IntPnd bit will be set to "1" when the reception data frame is stored to the message object.

Data length code (DLC3 to 0) is provided by the application. Reception data length code and an 8-byte data will be stored when the CAN controller stores the reception data frame to the message object. If the data length code is less than 8 bytes, undefined data will be written to the remaining data of the message object.

When UMask="1", the IFx mask register (Msk28 to 0, UMask, MXtd and MDir bits) will be used to allow the reception of the data frame having the ID that has been grouped by the mask setting. See the data frame reception in "5.3 Message Reception Operation" for details.

Note:

The Dir bit of the IFx mask register cannot be set as a mask.

5.3.6. Reception Message Processing

Reception message processing is shown.

CPU can read reception messages at any time via the message interface register.

Generally, "$007F_H$" is written to the IFx command mask register (IFxCMSK). Message number of the message object will then be written to the IFx command request register (IFxCREQ). By using this procedure, reception message of the specified message number will be transferred from message RAM to the message interface register. At this time, the NewDat bit and IntPnd bit of the message object can be cleared to "0" by the setting of the IFx command mask register (IFxCMSK).

The message will be received if it matches the acceptance filter. If the acceptance filter mask is used in the message object, the data that has been set as a mask will be excluded from the acceptance filter, and the decision of whether or not to receive the message will be made.

The NewDat bit indicates whether a new message has been received after the message object was last read.

The MsgLst bit indicates that the next reception data has been received before the previously received data is read from the message object, resulting in the loss of the previous data. The MsgLst bit will not be reset automatically.

The TxRqst bit will be cleared to "0" automatically when a data frame matching the acceptance filter is received while the remote frame transmission is being processed.

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.

5.4.1. Configuration of FIFO Buffer

The configuration of FIFO buffer is shown.

The configuration of the reception message objects in the FIFO buffer is the same as that of other reception message objects, except for the EoB bit (see "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.

5.4.2. Message Reception by FIFO Buffer

Message reception by FIFO buffer is shown.

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 EoB bit is "0", a write operation to the FIFO buffer by the CAN controller will not be performed as the reception message object will be protected until the final reception message object (EoB bit = "1") is reached.

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.

5.4.3. Reading from FIFO Buffer

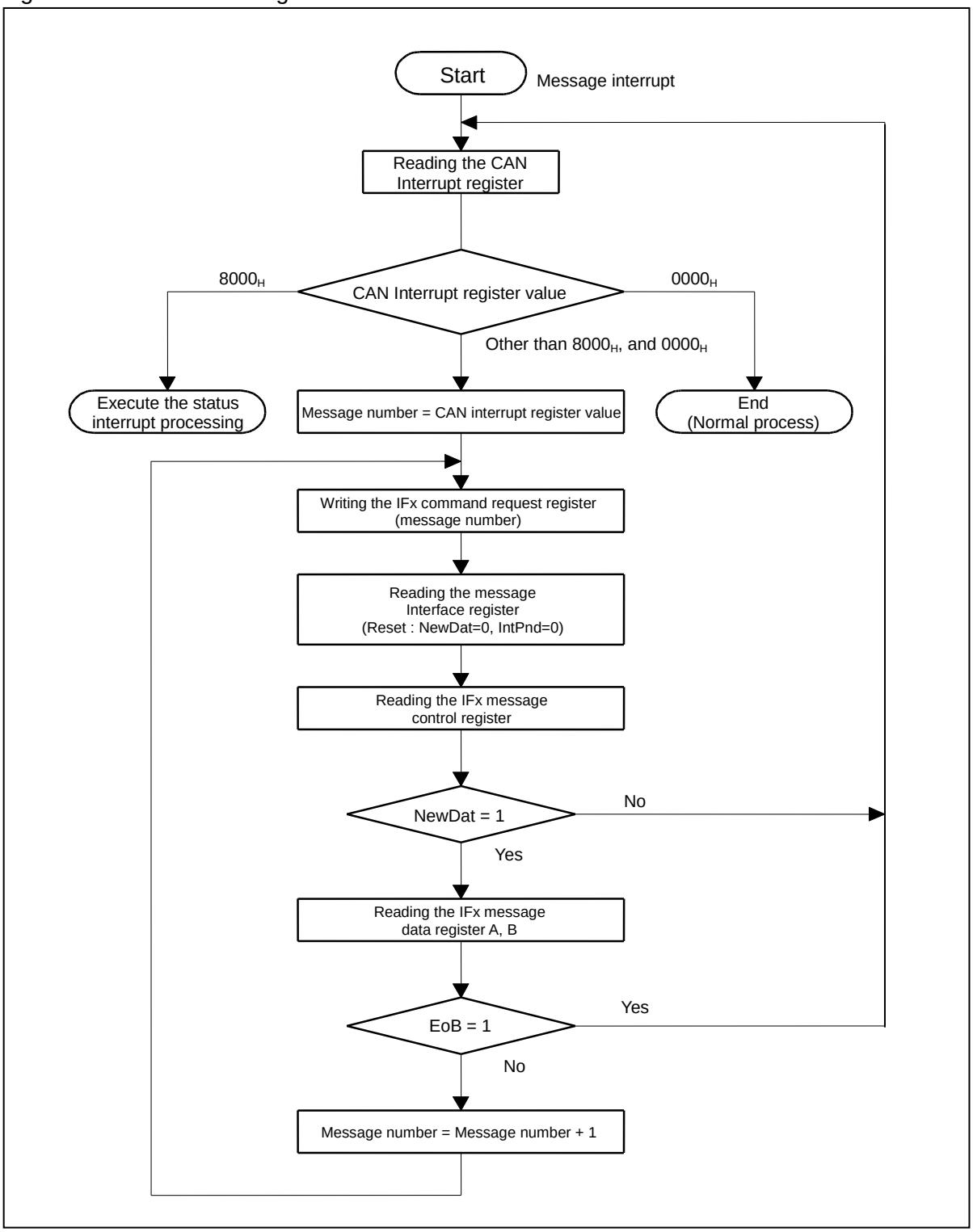
Reading from FIFO buffer is shown.

CPU can read the content of the received message object by writing the reception message number to the IFx command request register (IFxCREQ) that will cause the message object to be transferred to the message interface register. Set the WR/RD to "0" (read), set the TxRqst/NewDat and the IntPnd to "1" and reset NewDat and IntPnd bits to "0" in the IFx command mask register (IFxCMSK) at this time.

In order to guarantee the function of the FIFO buffer, always read the reception message objects in the FIFO buffer starting from the one having the smallest message number.

The figure below shows the CPU processing method for the message objects that are linked in the FIFO buffer.

Figure 5-2 CPU Processing of FIFO Buffer



5.5. Interrupt Function

Interrupt function is shown.

This section explains the processing of status interrupt (IntId=8000_H) and message interrupt (IntId=message number). If 2 or more interrupts are pending, the CAN interrupt register (INTR) will indicate the pending interrupt code of the highest priority interrupt. High priority interrupt codes will always be displayed, ignoring the chronological order in which the interrupt codes were set. Interrupt code will be held until it is cleared by CPU.
Status interrupt (IntId bit = 8000_H) has the highest priority.
Priority of message interrupts becomes higher as the message number gets smaller, and vice versa.

Message interrupt will be cleared when the IntPnd bit of the message object is cleared. Status interrupt will be cleared when the CAN status register (STATR) is read.

The IntPnd bit of the CAN interrupt pending register (INTPND) indicates whether any interrupt exists. The IntPnd bit will indicate "0" if there is no pending interrupt.

The interrupt signal to the CPU will become active when the IntPnd bit becomes "1" while the IE bit of the CAN control register (CTRLR) and TxIE and RxIE bits of the IFx message control register (IFxMCTR) are set to "1". The interrupt signal maintains its active state until the CAN interrupt pending register (INTPND) is cleared to "0" (interrupt factor reset) or until IE bit of the CAN control register (CTRLR) is reset to "0".

The CAN interrupt register (INTR) being set to "8000_H" indicates an update of the CAN status register (STATR) by the CAN controller; and this interrupt will have the highest priority. The interrupt generated by updating the CAN status register (STATR) can allow or prohibit the setting of the CAN interrupt register (INTR) by using EIE and SIE bits of the CAN control register (CTRLR). Interrupt signal to the CPU can be controlled by the IE bit of the CAN control register (CTRLR).

The RxOk bit, TxOk bit and LEC bit of the CAN status register (STATR) can be updated (reset) by a write from the CPU. However, interrupt cannot be set or reset by the write operation.
The CAN interrupt register (INTR) set to other than "8000_H" and "0000_H" indicates that the message interrupt is currently pending and that it has a high priority.
The CAN interrupt register (INTR) will be updated even when IE has been reset.

Message interrupt factor to the CPU can be confirmed in the CAN interrupt register (INTR) or CAN interrupt pending register (INTPND).(See "4.5 Message Handler Registers".) When clearing a message interrupt, it is possible to read the message data at the same time. When the message interrupt specified by the CAN interrupt register (INTR) is cleared, the next priority interrupt will be set to the CAN interrupt register (INTR), waiting for the next interrupt process. The CAN interrupt register (INTR) will indicate "0000_H" if there is no interrupt.

Notes:

- Status interrupt (IntId=8000_H) will be cleared by a read access from the CAN status register (STATR).
- Status interrupt (IntId=8000_H) by a write access to the CAN status register (STATR) will not be generated.

5.6. Bit Timing and CAN System Clock (fsys) Generation

Bit timing and CAN system clock (fsys) generation is shown.

This section explains the overview of bit timing and its role in the CAN controller.

Each CAN node of the CAN network has a clock oscillator (normally a crystal oscillator). Time parameter of bit time can be configured individually for each CAN node. A common bit rate can be produced even if the oscillation cycle (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 controller 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 5-3 Schematic Diagram of CAN System Clock (fsys) Generation

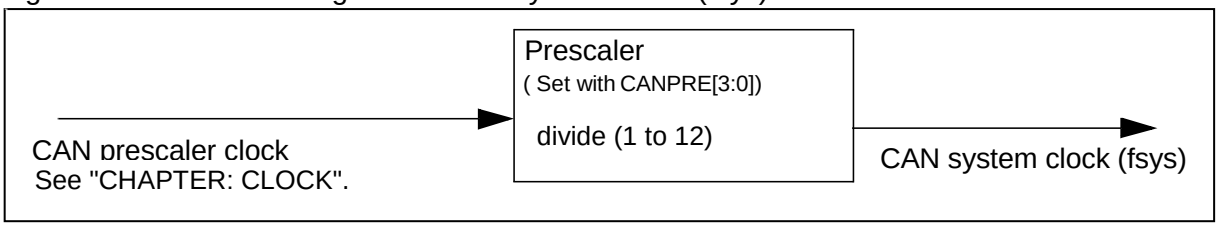


Figure 5-4 Bit Timing

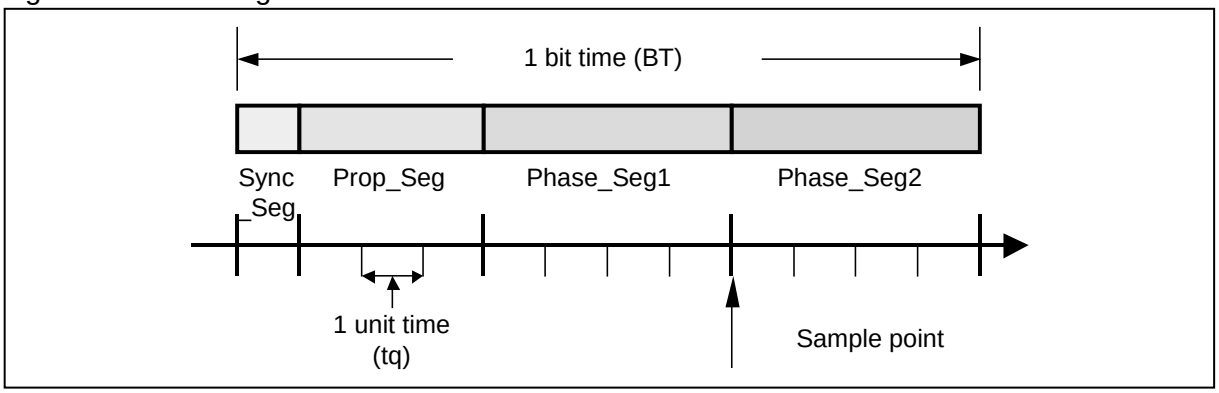


Table 5-3 CAN Bit Time Parameters

Parameter	Range	Function
BRP	[1 to 32]	Defines the time quantity tq.
Sync_Seg	1 tq	Fixed length. Synchronizes the bit time with the system clock.
Prop_Seg	[1 to 8] tq	Compensates for physical delay time.
Phase_Seg1	[1 to 8] tq	Guarantees identification of edge-phase errors prior to the sample point. The bit time may be temporarily prolonged due to synchronization.
Phase_Seg2	[1 to 8] tq	Guarantees identification of edge-phase errors subsequent to the sample point. The bit time may be temporarily shortened due to synchronization.
SJW	[1 to 4] tq	Defines the resynchronization jump width. It will not be greater than either of the phase buffer segments.

The bit timing effected by the CAN controller is shown in the following.

Figure 5-5 Bit Timing Effected by the CAN Controller

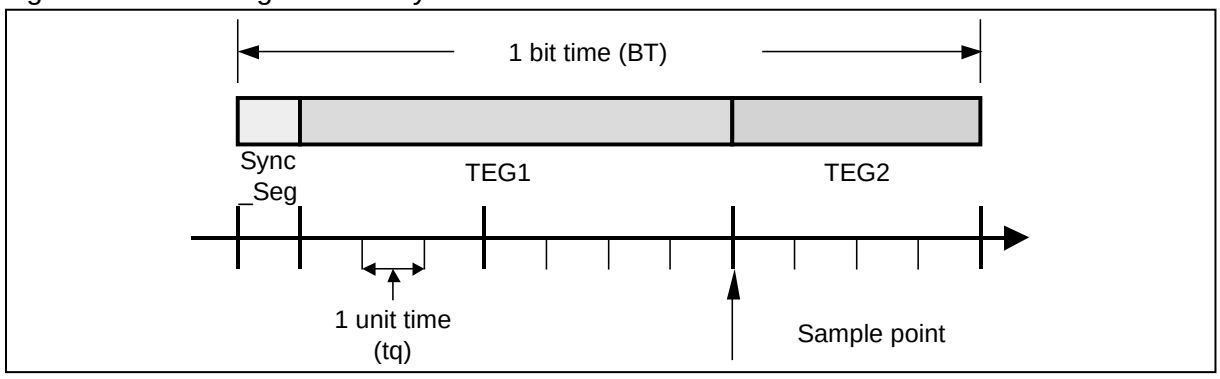


Table 5-4 CAN Controller Parameters

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

5.7. Test Mode

Test mode is shown.

This section explains the test mode setting method and operation.

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".

5.7.2. Silent Mode

Silent mode is shown.

The CAN controller enters silent mode when the Silent bit of the CAN test register (TESTR) is set to "1".

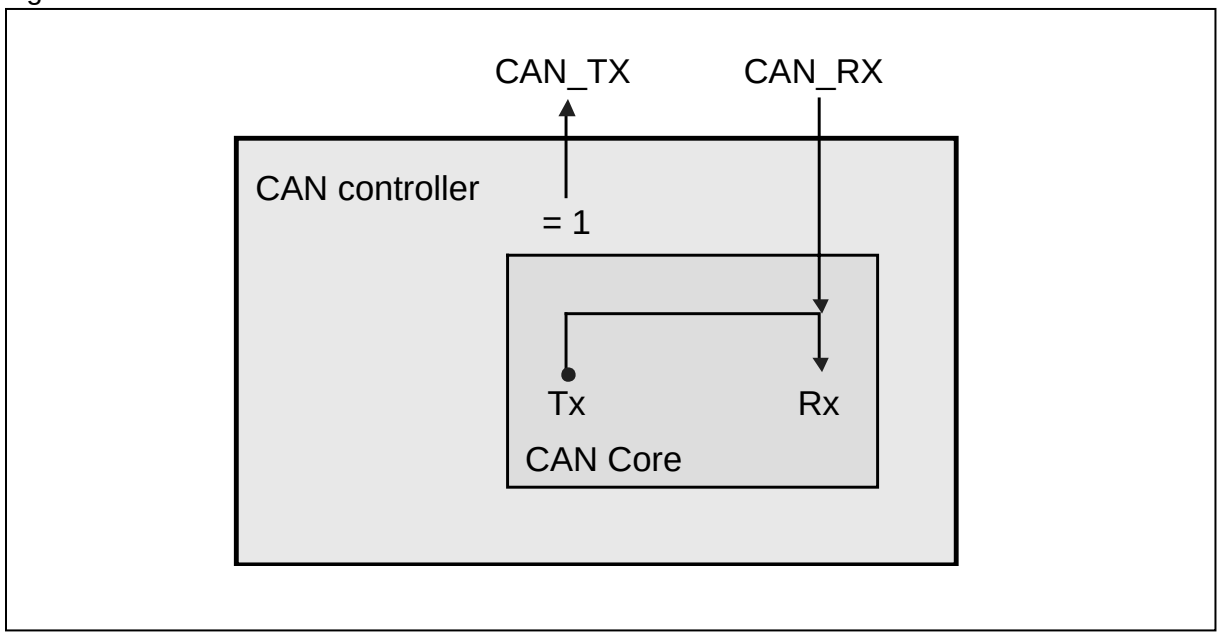
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs a recessive level to the CAN bus and does not send messages or ACKs.

When the CAN controller is requested to send a dominant bit (the ACK bit, overload flag, or active error flag), it sends the dominant bit to the RX end through a loopback circuit within the CAN controller. During this operation, the receiving end can receive the dominant bit that is sent through the loopback circuit within the CAN controller even if the CAN bus is in the recessive-level state.

In silent mode, traffic over the CAN bus can be analyzed without influence from the transmission of dominant bits (ACK bits and error flags).

The figure below shows how signals CAN_TX and CAN_RX are connected to the CAN controller in silent mode:

Figure 5-6 CAN Controller in Silent Mode



5.7.3. Loopback Mode

Loopback mode is shown.

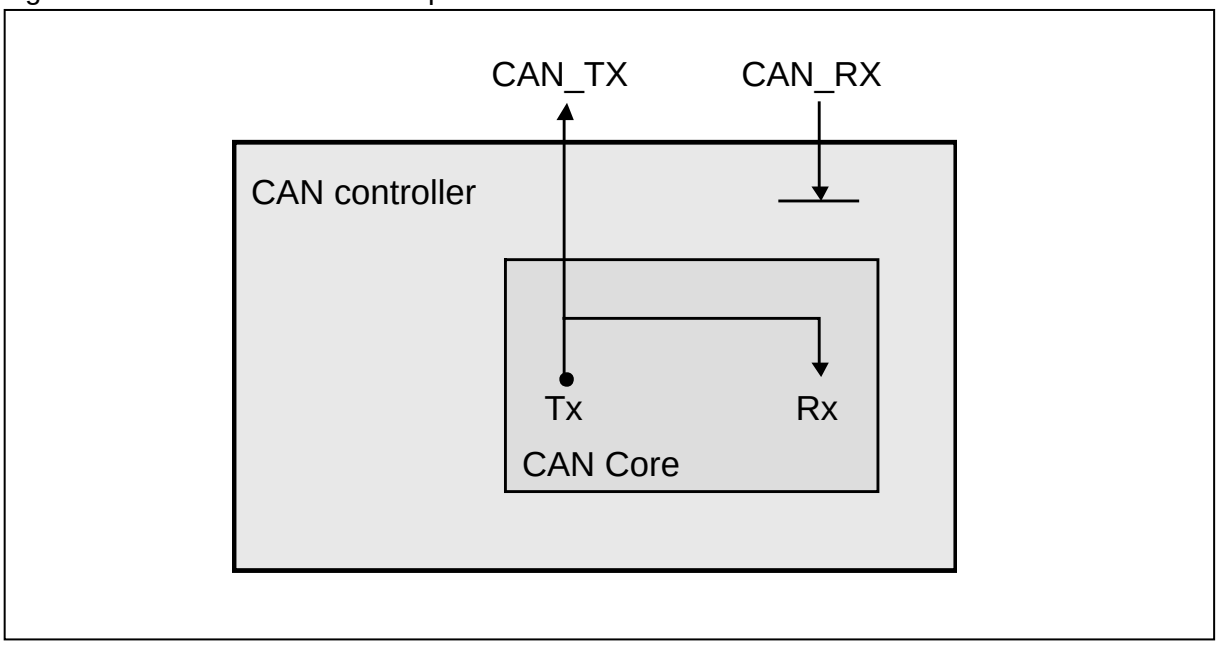
The CAN controller enters loopback mode when the LBack bit of the CAN test register (TESTR) is set to "1".

Loopback mode can be used for self-diagnostics.

In loopback mode, the TX end and the RX end are connected within the CAN controller, messages sent by the CAN controller are handled as messages received by the RX end, and messages that have passed through the acceptance filter are stored in the receive buffer.

The figure below shows how signals CAN_TX and CAN_RX are connected to the CAN controller in loopback mode:

Figure 5-7 CAN Controller in Loopback Mode



Note:

Dominant bits from the acknowledge slot of data/remote frames are not sampled to ensure that they are left independent of external signals. Therefore, the CAN controller will not generate acknowledge errors in test mode although it may generate these errors in other mode.

5.7.4. Combination of Silent and Loopback Modes

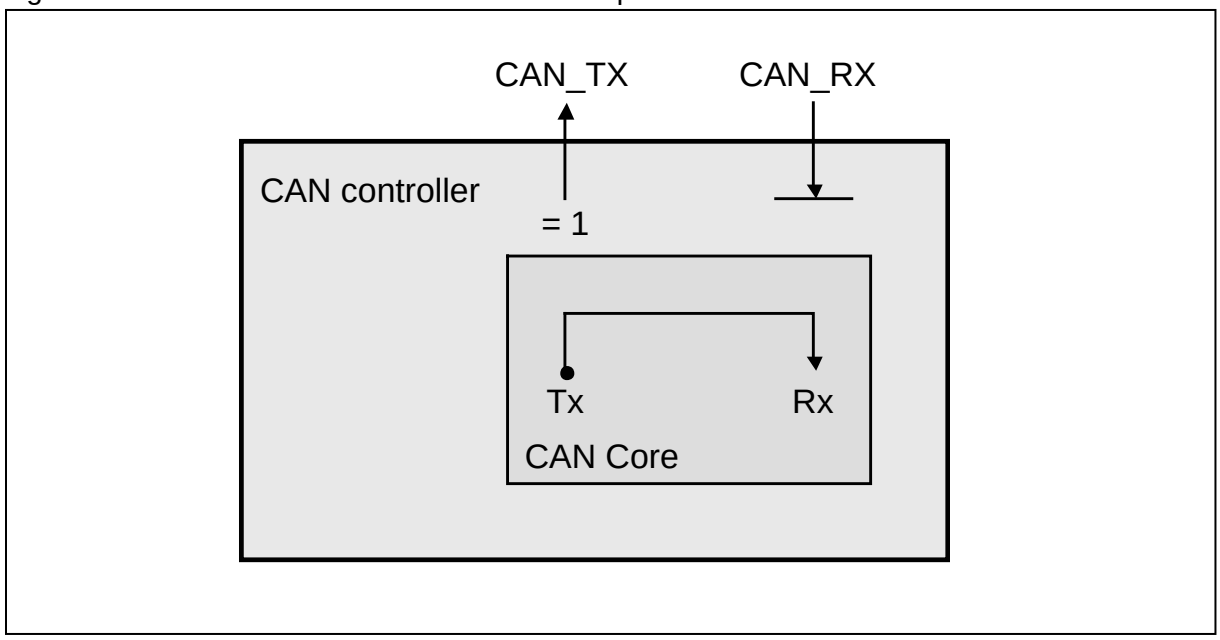
Combination of silent and loopback modes is shown.

The CAN controller can work in a mode that combines loopback and silent modes, when the LBack and Silent bits of the CAN test register (TESTR) are set to "1" simultaneously.

This combined mode can be used for hot self-tests. "Hot self-test" means that when the CAN controller is in process of tests in loopback mode, CAN system operation receives no influence from these tests because a fixed recessive-level output is at the CAN_TX 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 5-8 CAN Controller in the Silent and Loopback Combined Mode



5.7.5. Basic Mode

Basic mode is shown.

The CAN controller enters basic mode when the Basic bit of the CAN test register (TESTR) is set to "1".

In basic mode, the CAN controller works without using the message RAM.

The IF1 message interface register is used for transmission control.

The message transmission procedure begins with the setting of the send data in the IF1 message interface register. The next step is to set the BUSY bit of the IF1 command request register to "1" to issue a transmission request. While the BUSY bit is set to "1", the IF1 message interface register is locked or transmission is held.

When the BUSY bit is set to "1", the CAN controller performs the following operation:

As soon as the CAN bus becomes idling, the CAN controller begins transmission by loading the content of the IF1 message interface register to the transmission shift register. When transmission ends normally, the BUSY bit is reset to "0", and the locked IF1 message interface register is released.

While transmission is held, it can be suspended anytime by resetting the BUSY bit of the IF1 command request register to "0". When the BUSY bit is reset to "0" during transmission, retransmission that would be initiated after an arbitration loss or error will not be initiated.

The IF2 message interface register is used for reception control.

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".

5.7.6. Software Control of the CAN_TX Pin

Software control of the CAN_TX pin is shown.

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.

5.8. Software Initialization

Software initialization is shown.

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.

5.9. CAN Wake Up Function

CAN Wake Up function is shown.

It can be wake up in the reception operation of the CAN by connecting the RX pin of the CAN with the external interrupt pin.

■ About the pin used by the CAN wake up function.

Because the RX0 pin and the INT0 pin, the RX1 pin and the INT1 pin or the RX2 pin and the INT7 pin are shared, the wake up function can be used.

Table 5-5 shows the relation among the CAN wake up function, the RX pin, and the INT pin.

Table 5-5 Pins of CAN Wake Up Function

	RX pin	Interrupt function
CAN0	RX0(128)	INT0_0
CAN1	RX1(64)	INT1_0
	RX1_1	INT1_1
CAN2	RX2(64)	INT7_0

■ About CAN wake up function

It is possible to return from the sleep mode or the standby mode by the reception data of the CAN.

Note:

It is necessary to set an external interrupt before it shifts to the sleep mode or the standby mode when the wake up function is used.

6. Restrictions

This section explains the restrictions on CAN.

6.1 INIT bit

6.1. INIT bit

This section provides information on INIT bit.

6.1.1 Restrictions

6.1.2 Bypass method

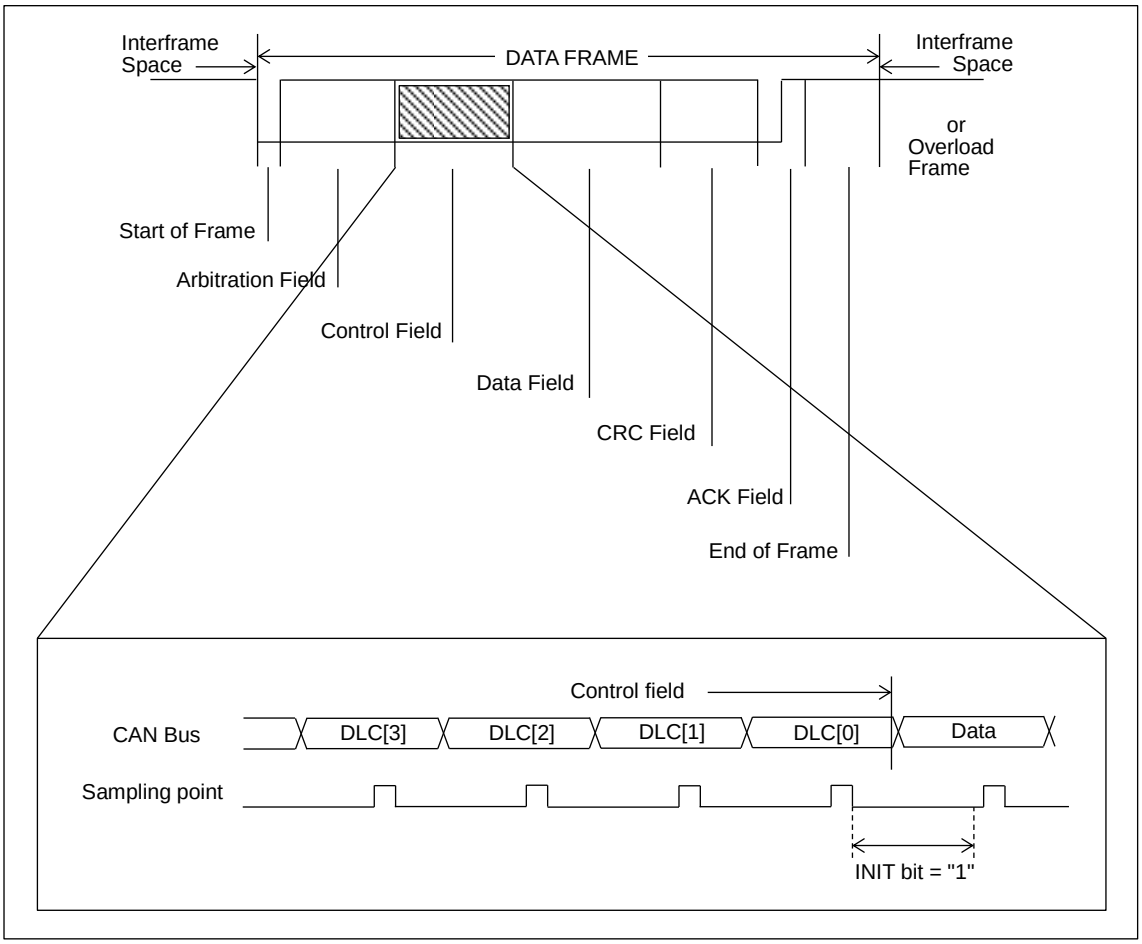
6.1.1. Restrictions

This section explains the restrictions.

If the INIT bit of the CAN control register (CTRLR) is set to "1" (Figure 6-1) during the transmission of the last bit in the control field, the INIT bit is cleared and then the data field of the frame transmitted first is shifted to left by one bit.

Note that subsequent messages are transmitted correctly.

As for remote frames or data frames with the data length of zero, the INIT bit setting at this timing does not have any influence on them.



6.1.2. Bypass method

This section explains the bypass method.

Apply either of the following methods to bypass this restriction.

1. When "1" is set in the INIT bit of CAN control register (CTRLR), set the INIT bit of CAN control register (CTRLR) to "1" immediately after the transmission completed.
2. When "1" is set in the INIT bit of CAN control register (CTRLR) during transmission and then "0" is set in the INIT bit to transmit, first, set the INIT bit to "1" and execute the transmission cancel for the message buffer with "1" set in the transmission request bit (TxRqst) (set TxRqst bit to "0"). Then set the INIT bit to "0". Next, after 2 bits interval of time of CAN has passed, set the transmission request bit (TxRqst) of the transmitted message buffer to "1".

Chapter 42: CAN Clock Prescaler

This chapter explains the CAN clock prescaler.

1. Overview
2. Features
3. Configuration
4. Registers

Code : BZCANPRE-1v1-91520-2-E

1. Overview

This section gives an overview of the CAN clock prescaler.

This module generates clocks (fsys) supplied from each clock source to the CAN macro. Figure 3-1 indicates CAN and CAN interface, and CAN clock prescaler and clock source selector circuit.

2. Features

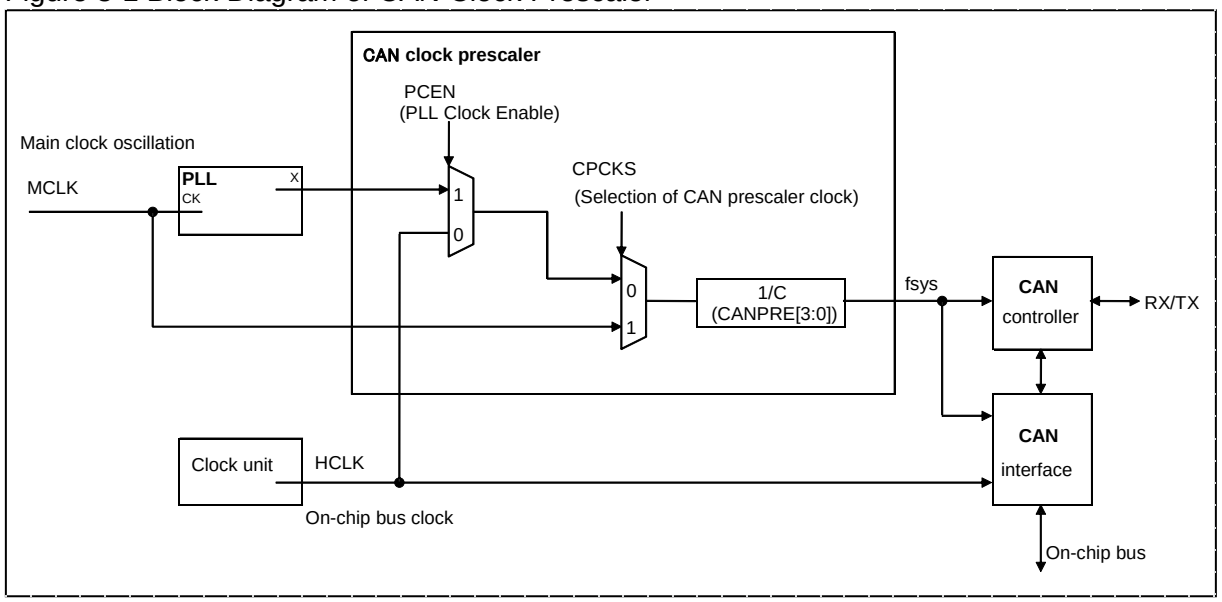
This section explains features of the CAN clock prescaler.

- As a source clock of the CAN clock prescaler, PLL clock/HCLK or main oscillation can be selected.
- PLL clock/HCLK is changed by PLL oscillation enable (PCEN).
- The counter which can divide CAN system clock frequency (fsys) by C (C=1to12) is installed.

3. Configuration

This section shows configuration of the CAN clock prescaler.

Figure 3-1 Block Diagram of CAN Clock Prescaler



4. Registers

This section shows registers of the CAN clock prescaler.

Table 4-1 Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x04A4	CANPRE	Reserved	Reserved	Reserved	CAN clock prescaler cotrol register

4.1. CAN Prescaler Register : CANPRE

The bit configuration of CAN prescaler register is shown.

This register sets the CAN system clock (fsys) generation prescaler. For details, see "CHAPTER: CAN 5.6. Bit Timing and CAN System Clock (fsys) Generation ". When changing the value of this register, set the initialization bit (Init) in the CAN control register (CTRLR) to "1" in order to stop all the bus operations.

■ CAN Prescaler Register: Address $04A4_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved		CPCKS	CANPRE3	CANPRE2	CANPRE1	CANPRE0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

[bit7] Reserved

Be sure to write "0" to this bit.

[bit6 to bit5] Reserved

The value read is always "0". Be sure to write "0" to these bits.

[bit4] CPCKS :CAN prescaler clock select bits

CPCKS	CAN prescaler source clock
0	PLL clock/ HCLK (on-chip bus clock)
1	Main oscillation (MCLK)

[bit3 to bit0] CANPRE[3:0] :CAN prescaler setting bits

CANPRE [3:0]	Function	Input CAN prescaler clock: 80MHz	Input CAN prescaler clock: 64MHz	Input CAN prescaler clock: 48MHz
0000	Selects 1/1 period of the system clock as the CAN clock(Initial value: CANPRE[3:0]=0000).	80MHz	64MHz	48MHz
0001	Selects 1/2 period of the system clock as the CAN clock.	40MHz	32MHz	24MHz
001x	Selects 1/4 period of the system clock as the CAN clock.	20MHz	16MHz	12MHz
01xx	Selects 1/8 period of the system clock as the CAN clock.	10MHz	8MHz	6MHz
1000	Selects 2/3 period of the system clock as the CAN clock. The duty of the clock is 67%.	53.3MHz	42.7MHz	32MHz
1001	Selects 1/3 period of the system clock as the CAN clock.	26.7MHz	21.4MHz	16MHz
1010	Selects 1/6 period of the system clock as the CAN clock.	13.3MHz	10.7MHz	8MHz
1011	Selects 1/12 period of the system clock as the CAN clock.	6.7MHz	5.4MHz	4MHz
110x	Selects 1/5 period of the system clock as the CAN clock.	16.0MHz	12.8MHz	9.6MHz
111x	Selects 1/10 period of the system clock as the CAN clock.	8.0MHz	6.4MHz	4.8MHz

Notes:

- When changing the CAN prescaler setting bit set the initialization bit (Init) in the CAN control register (CTRLR) to "1" in order to stop all the bus operations.
- By setting the register, be sure to keep the frequency of a clock, supplied to the CAN interface, 16MHz or less.

Chapter 43: D/A Converter

This chapter explains the D/A converter.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Note

Code: FW11-1v0-91520-2-E

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. The device incorporates 2 channels of the 8-bit D/A converter.

In MB91F52xB (64 pin) and MB91F52xD (80 pin), only ch.0 is available.

2. Features

This section explains the features of the D/A converter.

- **Power Down Function:**
  The D/A converter has the power down function built-in which turns the power off when the outputs from the D/A converter are disabled.
- **Key Code Function:**
  This function protects registers against miss-writing. If the key code register (KEYCDR) is not written in a prescribed way, the writing to the target registers becomes invalid. Moreover, word access to the target registers is disabled.

  For details of the key code function, see "CHAPTER: I/O PORTS".

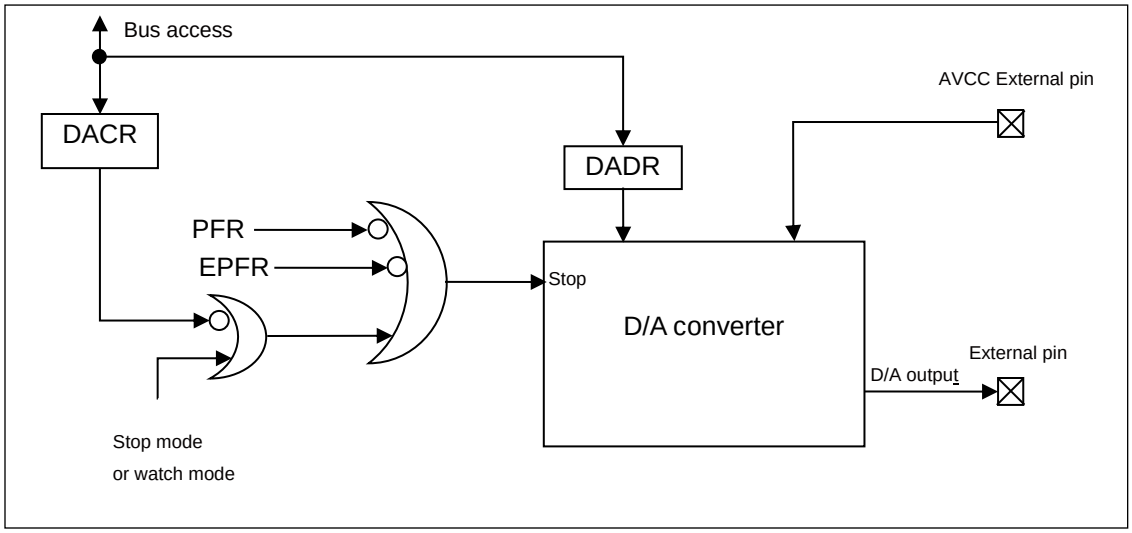
  The register that is the target of the key code function is as follows:

  · D/A control register

3. Configuration

This section explains the configuration of the D/A converter.

Figure 3-1 Block Diagram



4. Registers

This section explains the registers of the D/A converter.

■ Base Address (Base_addr) and External Pin Table

Channel	Base_addr	External pin	
		MB91F52xB, MB91F52xD	MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL
0	0x023C	DAO0	DAO0
1	0x023E	none	DAO1

■ Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x023C	DACR0	DADR0	DACR1	DADR1	D/A control register 0 D/A data register 0 D/A control register 1 D/A data register 1

4.1. D/A Control Register : DACR

The bit configuration of the D/A control register is shown below.

This register enables the output from the DAO pin.

This register is the target of the key code function.

■ 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 (D/A Enable) : D/A output enable bit

0: Output disabled

1: Output enabled

Note:

When the D/A converter has been selected as a port function, if "0" is set in this bit, the D/A converter as a port function becomes disabled, regardless of the PFR/EPFR setting.

When the D/A converter enters the stop mode or the watch mode, the pin function of the D/A converter becomes forcibly disabled.

As written in "5.2. EPFR Setting Priority" in "I/O Ports", the D/A converter output function has the first priority.

Therefore, note that if any functions other than the D/A converter, such as A/D, PPG, OCU, and SG, are enabled when the D/A converter function becomes disabled, such functions become enabled.

If any resource functions other than the D/A converter are not selected in PFR/EPFR/ADCH, the port becomes a general-purpose port function. So, PDR/DDR can select the pin status.

4.2. D/A Data Register : DADR

The bit configuration of the D/A data register is shown below.

This register sets 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 + 01H (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) : D/A 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.

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.

If the DAE bit of the D/A control register (DACR) is set to "0" when the D/A converter is selected as a port function, the D/A converter as a port function becomes disabled, regardless of PFR/EPFR setting. When the D/A converter enters the stop mode or the watch mode, the pin function of the D/A converter becomes forcibly disabled.

As written in "5.2 EPFR Setting Priority" in "I/O Ports", the D/A converter output function has the first priority.

Therefore, note that if any functions other than the D/A converter, such as A/D, PPG, OCU, and SG, are enabled when the D/A converter function becomes disabled, such functions become enabled.

When any resource functions other than the D/A converter are not selected in PFR/EPFR/ADCH, the port becomes a general-purpose port function. So, PDR/DDR can select the pin status.

For output of the conversion result of the D/A converter to the external pin, a pin must be set to D/A output by the DAE bit in the D/A control register (DACR), and PFR register and EPFR register.

6. Note

The section explains the notes about the D/A converter.

The D/A output value cannot be read by the PDDR register.
The DACR is the register that is the target of the key code function.
When writing the register that is the target of the key code function, the key code register (KEYCDR) needs to be set as follows:

- KEY1 + KEY0 + access size (SIZE) + access address (RADR[12:0]) is set to the key code register by half word.
- The writing to the (KEY1, KEY0) must be done continuously in the order of (0, 0), (0, 1), (1, 0), and (1, 1). The address and access size when (KEY1, KEY0) is written four times must be the same values.

For details of the key code function, see "5.8 Settings of the Key Code Register Function" in "CHAPTER: I/O PORTS".

Chapter 44: 12-Bit A/D Converter

This chapter explains the 12-bit A/D converter.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

Code : FIP004FS24-2v2-91520-2-E

1. Overview

This section explains the overview of the 12-bit A/D converter.

The 12-bit A/D converter can convert analog input voltages to 12-bit digital values using the RC successive approximation conversion method. It begins A/D conversion when an A/D activation trigger is entered. When an A/D activation trigger is entered again during A/D conversion, another sequence of A/D conversion starts. The converter also supports forced stop due to an A/D conversion cancel input signal.

2. Features

This section explains features of the 12-bit A/D converter.

2.1 Function of A/D Activation Compare

2.2 Function of A/D Activation Arbitration

2.3 Functions of 12-bit A/D Converter Control

2.1. Function of A/D Activation Compare

The function of A/D activation compare is explained.

■ Analog input control

This function can enable/disable Max 48 channels of analog inputs with the A/D converter of 2 units.

■ Activation channel

- It performs operation for A/D activation request control and A/D conversion data storage with each activation channel.
- The A/D activation channel corresponds to each unit of 12-bit A/D converter. The correspondence is as follows.

Table 2-1 Channel Allocation of Each Unit

	Unit 0	Unit 1
MB91F52xB	ch.0, ch.1, ch.7, ch.10, ch.11, ch.14 to ch.17, ch.22, ch.27, ch.28, ch.31	ch.32, ch.34, ch.35, ch.37, ch.38, ch.40 to ch.47
MB91F52xD	ch.0, ch.1, ch.7, ch.10, ch.11, ch.12, ch.14 to ch.17, ch.19, ch.22, ch.26, ch.27, ch.28, ch.31	ch.32 to ch.47
MB91F52xF	ch.0, ch.1, ch.7, ch.9 to ch.19, ch.22, ch.23, ch.26 to ch.29, ch.31	ch.32 to ch.47
MB91F52xJ	ch.0, ch.1, ch.7, ch.9, ch.10 to ch.31	ch.32 to ch.47
MB91F52xK	ch.0 to ch.31	ch.32 to ch.47
MB91F52xL	ch.0 to ch.31	ch.32 to ch.47

- Each activation channels is composed of the following register.
  - Compare buffer register / Compare register
  - A/D activation trigger control status register
  - A/D data register
  - A/D activation trigger extend control register
  - Range compare control status register
  - Range compare threshold over flag register
  - Range compare flag register
  - Activation channel conversion count setting register
  - Activation channel conversion count completion flag register

■ A/D activation request

- Each activation channel issues an A/D activation request by one of the following methods: software, external trigger (falling), reload timer (rising), compare match, and PPG. A given activation channel cannot reissue an A/D activation request during A/D conversion that was initiated by that channel.
- For software activation, external trigger, reload timer, and PPG activation, any activation channel can be selected.
- Compare match activation is such that an A/D activation request is issued when the 16-bit free-run timer value matches the value of the compare register for an activation channel. The free-run timer value to be used is selected when a free-run timer is selected and is supplied to each activation channel. This can be set at the free-run timer select register (FRS2 to FRS7). Please refer to "Free-run timer select register: FRS" of "CHAPTER: 16-BIT FREE-RUN TIMER" for details.
- If compare match activation is in effect, an A/D activation request is issued when the free-run timer value matches the compare register value provided that the 16-bit free-run timer only counts up, only counts down, or counts up and down.
- For each activation channel, either single mode or repeat mode can be specified as the activation request method.

- In single mode, one activation request is issued when one activation factor is encountered. One sequence of A/D conversion is performed and the activation request is reset when the A/D conversion completes.
- In repeat mode, activation requests are issued in succession as triggered by one activation factor. A/D conversion is performed repeatedly and the activation request continues its effect as long as repeat mode prevails.

■ A/D conversion data

- When A/D conversion completes, conversion-result data is stored in the A/D data register. One A/D data register is assigned to each activation channel.
- Each A/D data register contains error flag and error status bits, whose values indicate the status of the A/D conversion data.

■ Scan conversion with A/D conversion specified

- The scan conversion that specifies the A/D conversion count of each activation channel can be executed.
- The scan conversion that specifies the A/D conversion count can set 1 type of each 12-bit A/D converter unit.
- The conversion count specification can select 1 to 4 times.
- The scan conversion that specifies the A/D conversion count can select the continuous scan conversion mode and the stop scan conversion mode.
- As for the continuous scan conversion mode, the specified activation channel is activated one by one. When the conversion of the final channel of the scan conversion is completed, the scan conversion is executed repeatedly from the top.
- As for the stop scan conversion mode, the specified activation channel is activated one by one. When the conversion of the final activation channel of the scan conversion is completed, the conversion stops. When the next activation factor is input, the scan conversion is executed from the top. However, the activation factor input converting the scan is disregarded.

■ Range compare function

- Range compare in each activation channel.
- Settings of the upper and lower bound threshold can set Max 4 types. Each activation channel selects one combination from among 4 types of upper and lower bound threshold settings and executes the range comparison.
- The range comparison can select the confirmation of inside or outside the range of the upper and lower bound threshold.
- The range comparison result can remove the noise according to a continuous detecting function. The continuous detecting function sets the range comparison flag by continuously detecting the range comparison result.
- A continuous detection count between 1 and 7 times can be selected.
- The state of the continuous detection count of the range comparison result can be confirmed.
- When the confirmation outside the range of the range comparison, the detection of larger than upper bound threshold or less than lower bound threshold can be confirmed.

■ Interrupt request

Each activation channel can generate an interrupt request when A/D conversion completes.

■ Data protection function

- Each A/D data register can be configured to set the data protection function. Note that the data protection function works when the activation factor is not a compare match.
- If the data protection function is enabled, A/D activation requests are masked until data is read from the A/D data register and the interrupt flag is cleared. Data reading and interrupt flag clearing may be in any sequence. In addition, whether clearing of the interrupt flag is included in the condition can be configured.
- Whether an A/D activation request or a conversion operation is in progress can be reported using register bits. Moreover, the current A/D conversion request or conversion operation can be forced to terminate by resetting the A/D activation request bit to "0".

2.2. Function of A/D Activation Arbitration

The function of A/D activation arbitration is explained.

- A/D activation arbitration is provided for each 12-bit A/D converter unit. Please refer to "Channel allocation of each unit" for the allocation to A/D activation arbitration 0 and 1 of each A/D activation channel.
- The A/D activation arbitration consists of arbitration circuit, an A/D activation trigger generation section, and an analog channel number select section.
- Activation requests from A/D activation compare is arbitrated and the activation triggers, A/D conversion cancel signals, and analog channel numbers are generated.
- An activation trigger is generated for one selected activation request from an A/D activation compare channel. If a contention occurs between activation requests from A/D activation compare channels, A/D activation arbitration uses priority control. The priority order is as follows: "Lower activation channel numbers are assigned higher priorities (priority control based on channel numbers)" or "Compare match > external trigger/reload timer/PPG activation by activation request > software activation (priority control based on activation factors)". Activation requests that are not selected are made to wait. When the current A/D conversion completes, arbitration restarts. Priority control based on activation factors is also performed during A/D conversion. In this case, the current conversion is suspended and the activation factor assigned a higher priority is serviced. The suspended activation factor will be activated again after the higher-priority conversion is processed if the re-arbitration process does not find a lower channel number of a higher-priority activation factor.
  - If an activation factor with the same priority is encountered during A/D conversion suspension:
    The request from the activation channel with the lower number is processed first.

  - If an activation factor with a different priority is encountered during A/D conversion suspension:
    The request based on the higher-priority activation factor is processed first.
  - If an activation factor with a higher priority is encountered during A/D conversion:
    The current conversion is suspended, and the higher-priority activation factor is processed. After this processing completion, arbitration is performed again. The suspended activation factor is then processed.
  - If an activation factor with a lower priority is encountered during A/D conversion:
    After the current conversion completions, arbitration is performed again. The activation factor with the lower priority is then processed.
  - If an activation factor with the same priority is encountered during A/D conversion:
    After the current conversion completes, arbitration is performed again. The activation factor with the same priority is then processed.
- A conversion cancel signal is generated to force the current conversion processing to termination when the activation factor that is in process of conversion becomes inactive and there is no other active activation factor.
- For the analog channel number, the activation request analog number entered from the activation channel of the activation request arbitration result is selected.

2.3. Functions of 12-bit A/D Converter Control

The function of 12-bit A/D converter control is explained.

- There are 12-bit A/D converters 0 to 1, to which analog input pins are assigned. Please refer to "Table 2-1 Channel Allocation of Each Unit" for the correspondence of each input pin and the unit.

A function used to A/D-convert analog voltages (input voltages) input to the analog input pins to digital values is provided. Its features are as follows:

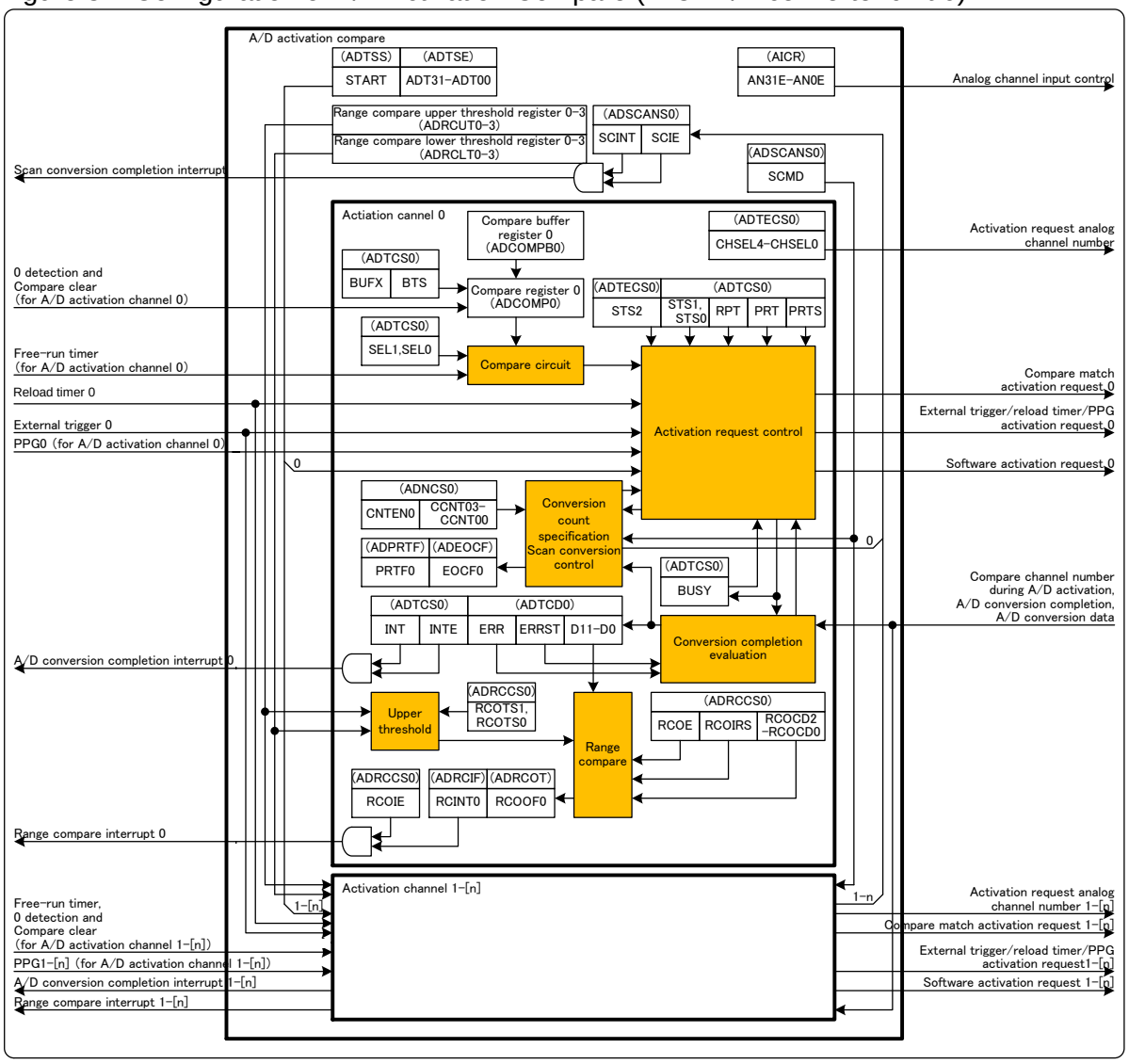
- The conversion time is at least 1.4 µs (including sampling time).
- The conversion method is the RC successive approximation conversion method with a sample hold circuit.
- The analog input pins can be selected under program control. (This program is configured in the A/D activation compare section)
- Activation signals are entered as pulse signals.
- One sequence of A/D conversion is performed for one activation factor.
- If an activation signal is entered again during A/D conversion, reactivation is performed. (Reactivation function)
- If an A/D conversion cancel signal is received during A/D conversion, the current processing is stopped and initialization takes place. (Forced stop function)
- As for the setting of the sampling time, a common sampling time setting to all channels and the sampling time settings of each channel can be selected.

3. Configuration

This section explains the configuration of the 12-bit A/D converter.

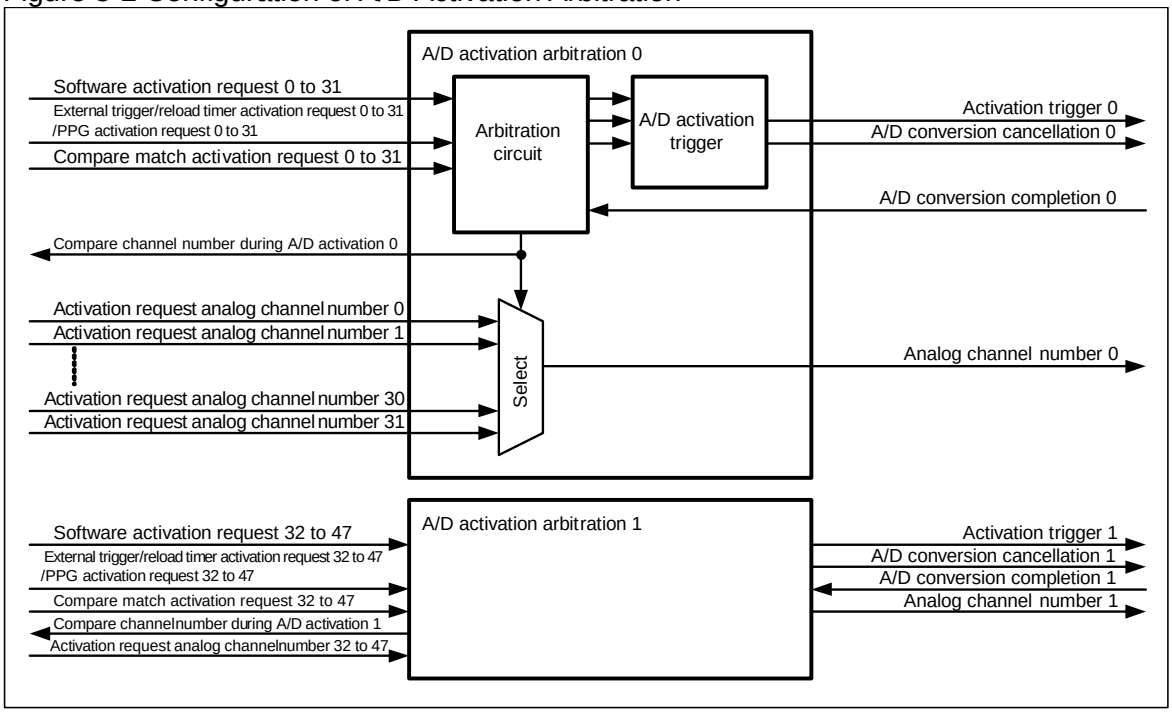
■ Configuration of A/D Activation Compare

Figure 3-1 Configuration of A/D Activation Compare (n=31 A/D converter unit 0)



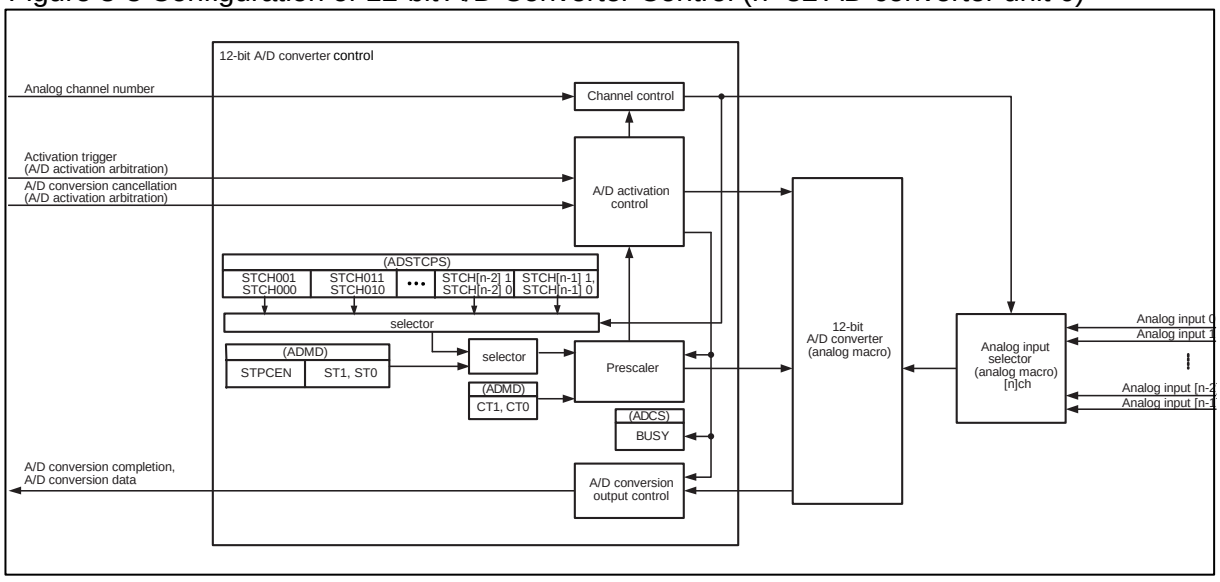
■ Configuration of A/D Activation Arbitration

Figure 3-2 Configuration of A/D Activation Arbitration



■ Configuration of 12-bit A/D Converter Control

Figure 3-3 Configuration of 12-bit A/D Converter Control (n=32 AD converter unit 0)



4. Registers

This section explains registers of the 12-bit A/D converter.

■ List of Analog Input Control Registers

Table 4-1 List of Analog Input Control Registers

Address	+0	+1	+2	+3
0x04AC	Analog input enable register upper (ADERH0) (key code target register)		Analog input enable register lower (ADERL0) (key code target register)	
0x04B0	Reserved		Analog input enable register lower (ADERL1) (key code target register)	

■ List of A/D Activation Compare Registers

Table 4-2 List of A/D Activation Compare Registers

Address	+0	+1	+2	+3
0x1304	A/D software activation register 0 (ADTSS0)	Reserved	Reserved	Reserved
0x1308	A/D software activation channel select register 0 (ADTSE0)			
0x130C	Compare buffer register 0 (ADCOMPB0) Compare register 0 (ADCOMP0)		Compare buffer register 1 (ADCOMPB1) Compare register 1 (ADCOMP1)	
0x1310	Compare buffer register 2 (ADCOMPB2) Compare register 2 (ADCOMP2)		Compare buffer register 3 (ADCOMPB3) Compare register 3 (ADCOMP3)	
0x1314	Compare buffer register 4 (ADCOMPB4) Compare register 4 (ADCOMP4)		Compare buffer register 5 (ADCOMPB5) Compare register 5 (ADCOMP5)	
0x1318	Compare buffer register 6 (ADCOMPB6) Compare register 6 (ADCOMP6)		Compare buffer register 7 (ADCOMPB7) Compare register 7 (ADCOMP7)	
0x131C	Compare buffer register 8 (ADCOMPB8) Compare register 8 (ADCOMP8)		Compare buffer register 9 (ADCOMPB9) Compare register 9 (ADCOMP9)	
0x1320	Compare buffer register 10 (ADCOMPB10) Compare register 10 (ADCOMP10)		Compare buffer register 11 (ADCOMPB11) Compare register 11 (ADCOMP11)	
0x1324	Compare buffer register 12 (ADCOMPB12) Compare register 12 (ADCOMP12)		Compare buffer register 13 (ADCOMPB13) Compare register 13 (ADCOMP13)	
0x1328	Compare buffer register 14 (ADCOMPB14) Compare register 14 (ADCOMP14)		Compare buffer register 15 (ADCOMPB15) Compare register 15 (ADCOMP15)	

Address	+0	+1	+2	+3
0x132C	Compare buffer register 16 (ADCOMPB16) Compare register 16 (ADCOMP16)		Compare buffer register 17 (ADCOMPB17) Compare register 17 (ADCOMP17)	
0x1330	Compare buffer register 18 (ADCOMPB18) Compare register 18 (ADCOMP18)		Compare buffer register 19 (ADCOMPB19) Compare register 19 (ADCOMP19)	
0x1334	Compare buffer register 20 (ADCOMPB20) Compare register 20 (ADCOMP20)		Compare buffer register 21 (ADCOMPB21) Compare register 21 (ADCOMP21)	
0x1338	Compare buffer register 22 (ADCOMPB22) Compare register 22 (ADCOMP22)		Compare buffer register 23 (ADCOMPB23) Compare register 23 (ADCOMP23)	
0x133C	Compare buffer register 24 (ADCOMPB24) Compare register 24 (ADCOMP24)		Compare buffer register 25 (ADCOMPB25) Compare register 25 (ADCOMP25)	
0x1340	Compare buffer register 26 (ADCOMPB26) Compare register 26 (ADCOMP26)		Compare buffer register 27 (ADCOMPB27) Compare register 27 (ADCOMP27)	
0x1344	Compare buffer register 28 (ADCOMPB28) Compare register 28 (ADCOMP28)		Compare buffer register 29 (ADCOMPB29) Compare register 29 (ADCOMP29)	
0x1348	Compare buffer register 30 (ADCOMPB30) Compare register 30 (ADCOMP30)		Compare buffer register 31 (ADCOMPB31) Compare register 31 (ADCOMP31)	
0x134C	A/D activation trigger control status register 0 (ADTCS0)		A/D activation trigger control status register 1 (ADTCS1)	
0x1350	A/D activation trigger control status register 2 (ADTCS2)		A/D activation trigger control status register 3 (ADTCS3)	
0x1354	A/D activation trigger control status register 4 (ADTCS4)		A/D activation trigger control status register 5 (ADTCS5)	
0x1358	A/D activation trigger control status register 6 (ADTCS6)		A/D activation trigger control status register 7 (ADTCS7)	
0x135C	A/D activation trigger control status register 8 (ADTCS8)		A/D activation trigger control status register 9 (ADTCS9)	
0x1360	A/D activation trigger control status register 10 (ADTCS10)		A/D activation trigger control status register 11 (ADTCS11)	
0x1364	A/D activation trigger control status register 12 (ADTCS12)		A/D activation trigger control status register 13 (ADTCS13)	
0x1368	A/D activation trigger control status register 14 (ADTCS14)		A/D activation trigger control status register 15 (ADTCS15)	
0x136C	A/D activation trigger control status register 16 (ADTCS16)		A/D activation trigger control status register 17 (ADTCS17)	

Address	+0	+1	+2	+3
0x1370	A/D activation trigger control status register 18 (ADTCS18)		A/D activation trigger control status register 19 (ADTCS19)	
0x1374	A/D activation trigger control status register 20 (ADTCS20)		A/D activation trigger control status register 21 (ADTCS21)	
0x1378	A/D activation trigger control status register 22 (ADTCS22)		A/D activation trigger control status register 23 (ADTCS23)	
0x137C	A/D activation trigger control status register 24 (ADTCS24)		A/D activation trigger control status register 25 (ADTCS25)	
0x1380	A/D activation trigger control status register 26 (ADTCS26)		A/D activation trigger control status register 27 (ADTCS27)	
0x1384	A/D activation trigger control status register 28 (ADTCS28)		A/D activation trigger control status register 29 (ADTCS29)	
0x1388	A/D activation trigger control status register 30 (ADTCS30)		A/D activation trigger control status register 31 (ADTCS31)	
0x138C	A/D data register 0 (ADTCD0)		A/D data register 1 (ADTCD1)	
0x1390	A/D data register 2 (ADTCD2)		A/D data register 3 (ADTCD3)	
0x1394	A/D data register 4 (ADTCD4)		A/D data register 5 (ADTCD5)	
0x1398	A/D data register 6 (ADTCD6)		A/D data register 7 (ADTCD7)	
0x139C	A/D data register 8 (ADTCD8)		A/D data register 9 (ADTCD9)	
0x13A0	A/D data register 10 (ADTCD10)		A/D data register 11 (ADTCD11)	
0x13A4	A/D data register 12 (ADTCD12)		A/D data register 13 (ADTCD13)	
0x13A8	A/D data register 14 (ADTCD14)		A/D data register 15 (ADTCD15)	
0x13AC	A/D data register 16 (ADTCD16)		A/D data register 17 (ADTCD17)	
0x13B0	A/D data register 18 (ADTCD18)		A/D data register 19 (ADTCD19)	
0x13B4	A/D data register 20 (ADTCD20)		A/D data register 21 (ADTCD21)	
0x13B8	A/D data register 22 (ADTCD22)		A/D data register 23 (ADTCD23)	
0x13BC	A/D data register 24 (ADTCD24)		A/D data register 25 (ADTCD25)	
0x13C0	A/D data register 26 (ADTCD26)		A/D data register 27 (ADTCD27)	
0x13C4	A/D data register 28 (ADTCD28)		A/D data register 29 (ADTCD29)	
0x13C8	A/D data register 30 (ADTCD30)		A/D data register 31 (ADTCD31)	

Address	+0	+1	+2	+3
0x13CC	A/D activation trigger extend control register 0 (ADTECS0)		A/D activation trigger extend control register 1 (ADTECS1)	
0x13D0	A/D activation trigger extend control register 2 (ADTECS2)		A/D activation trigger extend control register 3 (ADTECS3)	
0x13D4	A/D activation trigger extend control register 4 (ADTECS4)		A/D activation trigger extend control register 5 (ADTECS5)	
0x13D8	A/D activation trigger extend control register 6 (ADTECS6)		A/D activation trigger extend control register 7 (ADTECS7)	
0x13DC	A/D activation trigger extend control register 8 (ADTECS8)		A/D activation trigger extend control register 9 (ADTECS9)	
0x13E0	A/D activation trigger extend control register 10 (ADTECS10)		A/D activation trigger extend control register 11 (ADTECS11)	
0x13E4	A/D activation trigger extend control register 12 (ADTECS12)		A/D activation trigger extend control register 13 (ADTECS13)	
0x13E8	A/D activation trigger extend control register 14 (ADTECS14)		A/D activation trigger extend control register 15 (ADTECS15)	
0x13EC	A/D activation trigger extend control register 16 (ADTECS16)		A/D activation trigger extend control register 17 (ADTECS17)	
0x13F0	A/D activation trigger extend control register 18 (ADTECS18)		A/D activation trigger extend control register 19 (ADTECS19)	
0x13F4	A/D activation trigger extend control register 20 (ADTECS20)		A/D activation trigger extend control register 21 (ADTECS21)	
0x13F8	A/D activation trigger extend control register 22 (ADTECS22)		A/D activation trigger extend control register 23 (ADTECS23)	
0x13FC	A/D activation trigger extend control register 24 (ADTECS24)		A/D activation trigger extend control register 25 (ADTECS25)	
0x1400	A/D activation trigger extend control register 26 (ADTECS26)		A/D activation trigger extend control register 27 (ADTECS27)	
0x1404	A/D activation trigger extend control register 28 (ADTECS28)		A/D activation trigger extend control register 29 (ADTECS29)	
0x1408	A/D activation trigger extend control register 30 (ADTECS30)		A/D activation trigger extend control register 31 (ADTECS31)	
0x140C	Upper bound threshold setting register 0 (ADRCUT0)		Lower bound threshold setting register 0 (ADRCLT0)	

Address	+0	+1	+2	+3
0x1410	Upper bound threshold setting register 1 (ADRCUT1)		Lower bound threshold setting register 1 (ADRCLT1)	
0x1414	Upper bound threshold setting register 2 (ADRCUT2)		Lower bound threshold setting register 2 (ADRCLT2)	
0x1418	Upper bound threshold setting register 3 (ADRCUT3)		Lower bound threshold setting register 3 (ADRCLT3)	
0x141C	Range compare control status register 0 (ADRCCS0)	Range compare control status register 1 (ADRCCS1)	Range compare control status register 2 (ADRCCS2)	Range compare control status register 3 (ADRCCS3)
0x1420	Range compare control status register 4 (ADRCCS4)	Range compare control status register 5 (ADRCCS5)	Range compare control status register 6 (ADRCCS6)	Range compare control status register 7 (ADRCCS7)
0x1424	Range compare control status register 8 (ADRCCS8)	Range compare control status register 9 (ADRCCS9)	Range compare control status register 10 (ADRCCS10)	Range compare control status register 11 (ADRCCS11)
0x1428	Range compare control status register 12 (ADRCCS12)	Range compare control status register 13 (ADRCCS13)	Range compare control status register 14 (ADRCCS14)	Range compare control status register 15 (ADRCCS15)
0x142C	Range compare control status register 16 (ADRCCS16)	Range compare control status register 17 (ADRCCS17)	Range compare control status register 18 (ADRCCS18)	Range compare control status register 19 (ADRCCS19)
0x1430	Range compare control status register 20 (ADRCCS20)	Range compare control status register 21 (ADRCCS21)	Range compare control status register 22 (ADRCCS22)	Range compare control status register 23 (ADRCCS23)
0x1434	Range compare control status register 24 (ADRCCS24)	Range compare control status register 25 (ADRCCS25)	Range compare control status register 26 (ADRCCS26)	Range compare control status register 27 (ADRCCS27)
0x1438	Range compare control status register 28 (ADRCCS28)	Range compare control status register 29 (ADRCCS29)	Range compare control status register 30 (ADRCCS30)	Range compare control status register 31 (ADRCCS31)
0x143C	Range compare Threshold over flag register 0 (ADRCOT0)			
0x1440	Range compare flag register 0 (ADRCIF0)			
0x1444	Scan conversion control status register 0 (ADSCANS0)	Reserved	Reserved	Reserved

Address	+0	+1	+2	+3
0x1448	Activation channel conversion count setting register 0 (ADNCS0)	Activation channel conversion count setting register 1 (ADNCS1)	Activation channel conversion count setting register 2 (ADNCS2)	Activation channel conversion count setting register 3 (ADNCS3)
0x144C	Activation channel conversion count setting register 4 (ADNCS4)	Activation channel conversion count setting register 5 (ADNCS5)	Activation channel conversion count setting register 6 (ADNCS6)	Activation channel conversion count setting register 7 (ADNCS7)
0x1450	Activation channel conversion count setting register 8 (ADNCS8)	Activation channel conversion count setting register 9 (ADNCS9)	Activation channel conversion count setting register 10 (ADNCS10)	Activation channel conversion count setting register 11 (ADNCS11)
0x1454	Activation channel conversion count setting register 12 (ADNCS12)	Activation channel conversion count setting register 13 (ADNCS13)	Activation channel conversion count setting register 14 (ADNCS14)	Activation channel conversion count setting register 15 (ADNCS15)
0x1458	Data protection status flag register 0 (ADPRTF0)			
0x145C	Activation channel conversion completion flag register 0 (ADEOCF0)			
0x1470	A/D software activation register 1 (ADTSS1)	Reserved	Reserved	Reserved
0x1474	A/D software activation channel select register 1 (ADTSE1)			
0x1478	Compare buffer register 32 (ADCOMPB32) Compare register 32 (ADCOMP32)		Compare buffer register 33 (ADCOMPB33) Compare register 33 (ADCOMP33)	
0x147C	Compare buffer register 34 (ADCOMPB34) Compare register 34 (ADCOMP34)		Compare buffer register 35 (ADCOMPB35) Compare register 35 (ADCOMP35)	
0x1480	Compare buffer register 36 (ADCOMPB36) Compare register 36 (ADCOMP36)		Compare buffer register 37 (ADCOMPB37) Compare register 37 (ADCOMP37)	
0x1484	Compare buffer register 38 (ADCOMPB38) Compare register 38 (ADCOMP38)		Compare buffer register 39 (ADCOMPB39) Compare register 39 (ADCOMP39)	
0x1488	Compare buffer register 40 (ADCOMPB40) Compare register 40 (ADCOMP40)		Compare buffer register 41 (ADCOMPB41) Compare register 41 (ADCOMP41)	
0x148C	Compare buffer register 42 (ADCOMPB42) Compare register 42 (ADCOMP42)		Compare buffer register 43 (ADCOMPB43) Compare register 43 (ADCOMP43)	
0x1490	Compare buffer register 44 (ADCOMPB44) Compare register 44 (ADCOMP44)		Compare buffer register 45 (ADCOMPB45) Compare register 45 (ADCOMP45)	

Address	+0	+1	+2	+3
0x1494	Compare buffer register 46 (ADCOMPB46) Compare register 46 (ADCOMP46)		Compare buffer register 47 (ADCOMPB47) Compare register 47 (ADCOMP47)	
0x1498 to 0x14B4	Reserved		Reserved	
0x14B8	A/D activation trigger control status register 32 (ADTCS32)		A/D activation trigger control status register 33 (ADTCS33)	
0x14BC	A/D activation trigger control status register 34 (ADTCS34)		A/D activation trigger control status register 35 (ADTCS35)	
0x14C0	A/D activation trigger control status register 36 (ADTCS36)		A/D activation trigger control status register 37 (ADTCS37)	
0x14C4	A/D activation trigger control status register 38 (ADTCS38)		A/D activation trigger control status register 39 (ADTCS39)	
0x14C8	A/D activation trigger control status register 40 (ADTCS40)		A/D activation trigger control status register 41 (ADTCS41)	
0x14CC	A/D activation trigger control status register 42 (ADTCS42)		A/D activation trigger control status register 43 (ADTCS43)	
0x14D0	A/D activation trigger control status register 44 (ADTCS44)		A/D activation trigger control status register 45 (ADTCS45)	
0x14D4	A/D activation trigger control status register 46 (ADTCS46)		A/D activation trigger control status register 47 (ADTCS47)	
0x14D8 to 0x14F4	Reserved		Reserved	
0x14F8	A/D data register 32 (ADTCD32)		A/D data register 33 (ADTCD33)	
0x14FC	A/D data register 34 (ADTCD34)		A/D data register 35 (ADTCD35)	
0x1500	A/D data register 36 (ADTCD36)		A/D data register 37 (ADTCD37)	
0x1504	A/D data register 38 (ADTCD38)		A/D data register 39 (ADTCD39)	
0x1508	A/D data register 40 (ADTCD40)		A/D data register 41 (ADTCD41)	
0x150C	A/D data register 42 (ADTCD42)		A/D data register 43 (ADTCD43)	
0x1510	A/D data register 44 (ADTCD44)		A/D data register 45 (ADTCD45)	
0x1514	A/D data register 46 (ADTCD46)		A/D data register 47 (ADTCD47)	

Address	+0	+1	+2	+3
0x1518 to 0x1534	Reserved		Reserved	
0x1538	A/D activation trigger extend control register 32 (ADTECS32)		A/D activation trigger extend control register 33 (ADTECS33)	
0x153C	A/D activation trigger extend control register 34 (ADTECS34)		A/D activation trigger extend control register 35 (ADTECS35)	
0x1540	A/D activation trigger extend control register 36 (ADTECS36)		A/D activation trigger extend control register 37 (ADTECS37)	
0x1544	A/D activation trigger extend control register 38 (ADTECS38)		A/D activation trigger extend control register 39 (ADTECS39)	
0x1548	A/D activation trigger extend control register 40 (ADTECS40)		A/D activation trigger extend control register 41 (ADTECS41)	
0x154C	A/D activation trigger extend control register 42 (ADTECS42)		A/D activation trigger extend control register 43 (ADTECS43)	
0x1550	A/D activation trigger extend control register 44 (ADTECS44)		A/D activation trigger extend control register 45 (ADTECS45)	
0x1554	A/D activation trigger extend control register 46 (ADTECS46)		A/D activation trigger extend control register 47 (ADTECS47)	
0x1558 to 0x1574	Reserved		Reserved	
0x1578	Upper bound threshold setting register 4 (ADRCUT4)		Lower bound threshold setting register 4 (ADRCLT4)	
0x157C	Upper bound threshold setting register 5 (ADRCUT5)		Lower bound threshold setting register 5 (ADRCLT5)	
0x1580	Upper bound threshold setting register 6 (ADRCUT6)		Lower bound threshold setting register 6 (ADRCLT6)	
0x1584	Upper bound threshold setting register 7 (ADRCUT7)		Lower bound threshold setting register 7 (ADRCLT7)	
0x1588	Range compare control status register 32 (ADRCCS32)	Range compare control status register 33 (ADRCCS33)	Range compare control status register 34 (ADRCCS34)	Range compare control status register 35 (ADRCCS35)
0x158C	Range compare control status register 36 (ADRCCS36)	Range compare control status register 37 (ADRCCS37)	Range compare control status register 38 (ADRCCS38)	Range compare control status register 39 (ADRCCS39)

Address	+0	+1	+2	+3
0x1590	Range compare control status register 40 (ADRCCS40)	Range compare control status register 41 (ADRCCS41)	Range compare control status register 42 (ADRCCS42)	Range compare control status register 43 (ADRCCS43)
0x1594	Range compare control status register 44 (ADRCCS44)	Range compare control status register 45 (ADRCCS45)	Range compare control status register 46 (ADRCCS46)	Range compare control status register 47 (ADRCCS47)
0x1598 to 0x15A4	Reserved		Reserved	
0x15A8	Range compare threshold over flag register 1 (ADRCOT1)			
0x15AC	Range compare flag register 1 (ADRCIF1)			
0x15B0	Scan conversion control status register 1 (ADSCANS1)	Reserved	Reserved	Reserved
0x15B4	Activation channel conversion count setting register 16 (ADNCS16)	Activation channel conversion count setting register 17 (ADNCS17)	Activation channel conversion count setting register 18 (ADNCS18)	Activation channel conversion count setting register 19 (ADNCS19)
0x15B8	Activation channel conversion count setting register 20 (ADNCS20)	Activation channel conversion count setting register 21 (ADNCS21)	Activation channel conversion count setting register 22 (ADNCS22)	Activation channel conversion count setting register 23 (ADNCS23)
0x15BC	Reserved	Reserved	Reserved	Reserved
0x15C0	Reserved	Reserved	Reserved	Reserved
0x15C4	Data protection status flag register 1 (ADPRTF1)			
0x15C8	Activation channel conversion completion flag register 1 (ADEOCF1)			

■ List of 12-bit A/D Converter Control Registers

Table 4-3 List of 12-bit A/D Converter Control Registers

Address	+0	+1	+2	+3
0x1460	A/D control status register 0 (ADCS0)		A/D channel status register 0 (ADCH0)	A/D mode setting register 0 (ADMD0)
0x1464	Sampling time setting per A/D channel register 0 (ADSTPCS0)	Sampling time setting per A/D channel register 1 (ADSTPCS1)	Sampling time setting per A/D channel register 2 (ADSTPCS2)	Sampling time setting per A/D channel register 3 (ADSTPCS3)
0x1468	Sampling time setting per A/D channel register 4 (ADSTPCS4)	Sampling time setting per A/D channel register 5 (ADSTPCS5)	Sampling time setting per A/D channel register 6 (ADSTPCS6)	Sampling time setting per A/D channel register 7 (ADSTPCS7)
0x15CC	A/D control status register 1 (ADCS1)		A/D channel status Register 1 (ADCH1)	A/D mode setting register 1 (ADMD1)
0x15D0	Sampling time setting per A/D channel register 8 (ADSTPCS8)	Sampling time setting per A/D channel register 9 (ADSTPCS9)	Sampling time setting per A/D channel register 10 (ADSTPCS10)	Sampling time setting per A/D channel register 11 (ADSTPCS11)

4.1. Register of Analog Input Control

The register of the analog input control is explained.

The analog input control register is used to control the analog input.

4.1.1. Analog Input Enable Register : ADER

The bit configuration of the analog input enable register is shown.

The analog input enable register (ADERH0, ADERL0, ADERL1) controls the analog input.

■ ADERH0: Address $04AC_H$ (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADE31	ADE30	ADE29	ADE28	ADE27	ADE26	ADE25	ADE24
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
	ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16
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

■ ADERL0: Address 04AE$_H$ (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADE15	ADE14	ADE13	ADE12	ADE11	ADE10	ADE9	ADE08
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
	ADE07	ADE06	ADE05	ADE04	ADE03	ADE02	ADE01	ADE00
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] ADE31 to ADE0 : Analog input enable bits

ADE31 to ADE0	Function
0	Analog input disabled
1	Analog input enabled

- This bit controls analog input pin.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.
-

Note:

- This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled.
- For pins with the A/D converter function, see "5.6. Notes on Pins with the A/D pins converter Function" in "CHAPTER: I/O PORTS".

■ ADERL1: Address $04B2_H$ (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADE47	ADE46	ADE45	ADE44	ADE43	ADE42	ADE41	ADE40
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
	ADE39	ADE38	ADE37	ADE36	ADE35	ADE34	ADE33	ADE32
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] ADE47 to ADE32 : Analog input enable bits

ADE47 to ADE32	Function
0	Analog input disabled
1	Analog input enabled

- This bit controls analog input pin.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

Note:

- This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled.
- For pins with the A/D converter function, see "5.6. Notes on Pins with the A/D pins converter Function" in "CHAPTER: I/O PORTS".

4.2. Register of A/D Activation Compare

The registers of the A/D activation compare is explained.

4.2.1. A/D Software Activation Register: ADTSS0, ADTSS1

The bit configuration of the A/D software activation register is shown.

The A/D software activation register (ADTSS) issues a 12-bit A/D convertor A/D activation request. The activation channel is specified by the A/D software activation channel select register (ADTSE).

■ ADTSS0: Address 1304_H (Access: Byte, Half-word, Word)

■ ADTSS1: Address 1470_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							START
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W

[bit7 to bit1] Reserved

Always write "0" to these bits.

[bit0] START: A/D conversion activation bit (software)

START	Function
0	Does not activate the A/D conversion function.
1	Activates the A/D conversion function.

- This bit activates the A/D conversion operation under software control.
- The A/D conversion activates when this bit is set to "1". The activation channel is specified by the A/D software activation channel select register(ADTSE).
- The A/D conversion cannot be reactivated by changing this bit.

4.2.2. A/D Software Activation Channel Select Register : ADTSE0, ADTSE1

The bit configuration of the A/D software activation channel select register is shown.

The A/D software activation channel select register (ADTSE) is a register to select the activation channel that issues an A/D activation request.

■ ADTSE0: Address 1308_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ADT31	ADT30	ADT29	ADT28	ADT27	ADT26	ADT25	ADT24
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ADT23	ADT22	ADT21	ADT20	ADT19	ADT18	ADT17	ADT16
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADT15	ADT14	ADT13	ADT12	ADT11	ADT10	ADT09	ADT08
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
	ADT07	ADT06	ADT05	ADT04	ADT03	ADT02	ADT01	ADT00
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit0] ADT31 to ADT00 : Software activation channel select bits

ADT31 to ADT00	Function
0	Software activation disabled
1	Software activation enabled

- These bits control the software activation from the activation channel.
- If these bits are "0", the software activation is disabled.
- If these bits are "1", the software activation is enabled.

■ ADTSE1: Address 1474_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADT47	ADT46	ADT45	ADT44	ADT43	ADT42	ADT41	ADT40
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
	ADT39	ADT38	ADT37	ADT36	ADT35	ADT34	ADT33	ADT32
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] Undefined

"1" is always read. Writing has no influence on an operation.

[bit15 to bit0] ADT47 to ADT32 : Software activation channel select bits

ADT47 to ADT32	Function
0	Software activation disabled
1	Software activation enabled

- These bits control the software activation from the activation channel.
- If these bits are "0", the software activation is disabled.
- If these bits are "1", the software activation is enabled.

4.2.3. Compare Buffer Register / Compare Register : ADCOMPB0 to ADCOMPB47 / ADCOMP0 to ADCOMP47

The bit configuration of the compare buffer register / compare register is shown.

The compare buffer register (ADCOMPB) is a 16-bit buffer register for the compare register (ADCOMP). When the value of compare register (ADCOMP) and the free-run timer is matched, the A/D converter is activated. Both the ADCOMPB register and the ADCOMP register exist at the same address.

■ **ADCOMPB0 to ADCOMPB31: Address $130C_H$ to $134A_H$ (Access: Half-word, Word)**

■ **ADCOMPB32 to ADCOMPB47: Address 1478_H to 1496_H (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00
Initial value	0	0	0	0	0	0	0	0
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] CMP15 to CMP00 : Compare value buffer bits

CMP15 to CMP00	Function
	Compare value buffer

- The compare buffer register is the buffer register for the A/D activation compare register (ADCOMP). If the buffer function is disabled (BUFX=1 in the A/D activation trigger control status register (ADTCS)) or when the free-run timer is stopped, the value in the compare buffer is immediately transferred to the compare register.
- If the buffer function is enabled (BUFX=0 in the A/D activation trigger control status register (ADTCS), the compare buffer value will be transferred to the compare register when it matches the compare clear register of the 16-bit free-run timer or when 0 is detected.

Note:

When accessing this register, use a half-word or word access instruction.

■ **ADCOMP0 to ADCOMP31: Address $130C_H$ to $134A_H$ (Access: Half-word, Word)**

■ **ADCOMP32 to ADCOMP47: Address 1478_H to 1496_H (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00
Initial value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] CMP15 to CMP00 : Compare value bits

CMP15 to CMP00	Function
	Compare value buffer

- The compare value is updated via the compare buffer register.
- The compare register holds the compare value to be compared with the 16-bit free-run timer count value, so that A/D conversion can be activated when the free-run timer value matches the compare value.
- When a value is stored in the compare register, it is immediately compared.
- Compare-match operation is not performed if SEL1, SEL0=11b holds true in the A/D activation trigger control status register (ADTCS).

Note:

When reading the compare register, use a word or half-word access.
Do not use a read-modify-write instruction when accessing the compare register.

4.2.4. A/D Activation Trigger Control Status Register : ADTCS0 to ADTCS47

The bit configuration of the A/D activation trigger control status register is shown.

The A/D activation trigger control status register (ADTCS) verifies A/D activation requests, enable/ disable interrupt requests, verify an interrupt request status, select an activation factor, select a conversion mode, control the protection function, select a compare value to be used for a comparison operation, control the compare value buffer, and select an analog input channel.

■ ADTCS0 to ADTCS31: Address $134C_H$ to $138A_H$ (Access: Byte, Half-word, Word)

■ ADTCS32 to ADTCS47: Address $14B8_H$ to $14D6_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	INT	INTE	STS1	STS0	RPT	PRT	PRTS
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R(RM1),W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SEL1	SEL0	BUFX	BTS	Reserved			
Initial value	0	0	1	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R0/W0	R0/W0	R0/W0	R0/W0

[bit15] BUSY : A/D conversion request bit

BUSY	Function	
	Read	Write
0	A/D conversion is not requested	A/D activation request forced stop
1	A/D conversion is requested or A/D conversion is operating	This bit value does not change and there is no influence on an operation by writing.

- This bit shows the operation of the A/D activation request or conversion.
- When the reading value of this bit is "0", it is shown that A/D conversion is not requested. When the reading value of this bit is "1", it is shown that A/D conversion is requested or A/D conversion is operating.
- The A/D activation request or conversion forced stops by writing "0" in this bit. This bit value does not change and there is no influence on an operation by writing "1".

Note:

If the read-modify-write (RMW) instruction is executed, "1" will be read out.

[bit14] INT : Interrupt request flag bit

INT	Function	
	Read	Write
0	A/D conversion is not completed.	Clear of bit
1	A/D conversion was completed.	This bit value does not change and there is no influence on an operation by writing.

- When data is set in A/D data register (ADTCD) by the A/D conversion, this bit is set to "1".
- When this bit and interrupt request enable bit (ADTCS:INTE) are "1", an interrupt request is generated.
- This bit is cleared by writing "0". This bit value does not change and there is no influence on an operation by writing "1".
- When the A/D conversion completion interrupt clear signal is "H", this bit is cleared.

Note:

If the read-modify-write (RMW) instruction is executed, "1" will be read out.
If the software clear (INT = "0" writing) or the clear due to an interrupt clear signal ("H") occurs at the same time with the hardware set, the hardware set takes priority.

[bit13] INTE : Interrupt request enable bit

INTE	Function
0	Interrupt request output disabled
1	Interrupt request output enabled

- This bit enables/disables the interrupt output to CPU.
- When this bit and interrupt request flag bit (ADTCS:INT) are "1", the interrupt request is generated.

[bit12, bit11] STS1, STS0 : A/D activation factor select bit

ADTECSn: STS2	STS1	STS0	Function
0	0	0	Software trigger
0	0	1	External trigger activation (falling edge)
0	1	0	Reload timer activation (rising edge)
0	1	1	Compare match activation
1	0	0	PPG activation (rising edge)
1	0	1	Setting disable
1	1	0	
1	1	1	

- The activation factor of the A/D conversion is selected by a combination of the STS1, STS0 bit, and bit8 (STS2) of the A/D activation trigger extend control register (ADTECS).

Notes:

- Since the A/D activation factor select bit changes immediately when the bits are rewritten, change these bits while activation factors of the current target and of the target to be changed are inactive and the A/D conversion is not being requested (ADTCS:BUSY=1).
- Please set these bits including ADTECS.STS2 as software activation ("000_B"), and set a corresponding bit of

ADTSE (activation channel) to the software activation disable (ADT bit =0) when A/D conversion is not being requested.

- Please confirm the 16-bit free-run timer has stopped before setting the A/D activation factor select bit.

[bit10] RPT : Repeat conversion select bit

RPT	Function
0	Single conversion
1	Repeat conversion

- This bit specifies an A/D conversion mode.
- To select single conversion mode, set this bit to "0". In this mode, one activation factor leads to the issuance of one A/D conversion request. The A/D conversion is performed once.
- To select repeat conversion mode, set this bit to "1". In this mode, one activation factor leads to a sequence of A/D conversion requests. The A/D conversion is performed repeatedly until single conversion mode is selected.

[bit9] PRT : A/D data register protection enable bit

PRT	Function
0	Protection disabled
1	Protection enabled

- If this bit is set to "1", the A/D data register is protected against being overwritten. The protection function works if the activation factor is not compare-match activation (STS1, STS0 = 11).
- After conversion data is stored in the A/D data register, the next activation request will be masked to protect the A/D data register against being overwritten until the factor specified by the A/D data register protection clear select bit (PRTS) occurs.

Note:

Please set the A/D data register protection enable bit before operating the A/D conversion.
Please do not change the A/D data register protection enable bit with the A/D conversion requested or the A/D data register protected.

[bit8] PRTS : A/D data register protection clear select bit

PRTS	Function
0	Data read and interrupt flag clear
1	Data read

- This bit selects a condition for clearing the activation request mask if the A/D data register protection function is enabled (PRT=1).
- If this bit is set to "0", the A/D data register (ADTCD) reading and the interrupt request flag bit (INT) clearing become the protection release conditions (random order).
- If this bit is set to "1", the A/D data register (ADTCD) reading becomes the protection release condition.

Note:

Please set the A/D data register protection clear select bit before operating the A/D conversion.
Please do not change the A/D data register protection clear select bit with the A/D conversion requested or the A/D data register protected.

[bit7,bit6] SEL1, SEL0 : Counting direction select bits

SEL1	SEL0	Function
0	0	Both counting up and down
0	1	Only counting up
1	0	Only counting down
1	1	Compare disabled

- "If these bits are set to "00_B", the compare-match operation is performed regardless of whether the free-run timer is counting up or down.
- "If these bits are set to "01_B", the compare-match operation is performed only when the free-run timer is counting up.
- "If these bits are set to "10_B", the compare-match operation is performed only when the free-run timer is counting down.
- "If these bits are set to "11_B", the compare-match operation is not performed.
- The compare-match operation is not performed while the selected free-run timer is inactive.

Note:

These bits must not be set to "10_B" when the 16-bit free-run timer is in the up count mode.

[bit5] BUFX : Compare register buffer function control bit

BUFX	Function
0	Enabled
1	Disabled

- If this bit is set to "1", the buffer function is disabled.
- If this bit is set to "0", the buffer function is enabled.

[bit4] BTS : Compare register buffer transfer control bit

BTS	Function
0	When 0 detection
1	When compare clear

- This bit sets the transfer condition when the buffer function of the compare register is enabled (ADTCS.BUFX="0").
- The BTS setting is enabled when the buffer function of the compare register is enabled (ADTCS.BUFX="0").
- If this bit is set to "0", the compare buffer register value will be transferred to the compare register when 0 is detected as the free-run timer value.
- If this bit is set to "1", the compare buffer register value will be transferred to the compare register when it matches the compare clear register of the free-run timer.

Note:

Please confirm the 16-bit free-run timer has stopped whenever the compare register buffer transfer control bit is set.

[bit3 to bit0] Reserved

Be sure to write "0" to these bits.

4.2.5. A/D Data Register : ADTCD0 to ADTCD47

The bit configuration of the A/D data register is shown.

The A/D data register (ADTCD) stores A/D conversion results.

■ ADTCD0 to ADTCD31: Address $138C_H$ to $13CA_H$ (Access: Byte, Half-word, Word)

■ ADTCD32 to ADTCD47: Address $14F8_H$ to 1516_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ERR	ERRST	Reserved		D11	D10	D9	D8
Initial value	1	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] ERR : Conversion data error flag bit

ERR	Function
0	The conversion data is normal
1	The conversion data is not normal.

- This bit indicates the presence of an error contained in the A/D conversion data. If this bit is "1", the value of the ERRST bit indicates the content of the error.
- When this bit is read, it is set to "1".
- When new conversion result are written in this register, this bit is cleared to "0".
- If the A/D data register protection function is enabled (ADTCS.PRT=1) and the activation factor is not compare-match activation (ADTECS.STS2="0", ADTCS.STS1, STS0=11), this bit is read as "0".

[bit14] ERRST : Conversion data error status bit (only when ERR = 1)

ERRST	Function
0	The conversion data is the old result.
1	The conversion data is overwritten by the new data.

- If the ERR bit is "1", this flag indicates the content of the error in the A/D conversion data.
- If the ERR bit is "1" and this bit is "0", the conversion result read by the CPU are old data.
- If the ERR bit and this bit are "1", the previous conversion results have been overwritten by new conversion results and are lost before they have not been completely read by CPU.
- This bit is set to "1" if the previous conversion results have been overwritten by new conversion results and lost before they have not been completely read by the CPU.
- When this bit is read, it is cleared to "0".
- If the A/D data register protection function is enabled (ADTCS.PRT=1) and the conversion factor is not compare-match activation (ADTECS.STS2="0", ADTCS.STS1, STS0=11), this bit is read as "0".

[bit13,bit12] Reserved

Be sure to write "0" to these bits.

[bit11, bit0] D11 to D0 : A/D data bits

D11 to D0	Function
	Conversion data

- A/D conversion results are stored in this register, when one sequence of conversion is completed, the register is rewritten.
- In general, the register holds the last conversion value.

Note:

Be sure to avoid writing data to these bits.

4.2.6. A/D Activation Trigger Extend Control Register : ADTECS0 to ADTECS47

The bit configuration of the A/D activation trigger extend control register is shown.

The A/D activation trigger extend control register (ADTECS) selects the activation factor select and the analog input channel.

■ ADTECS0 to ADTECS31: Address $13CC_H$ to $140A_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							STS2
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

[bit15 to bit9] Reserved

Be sure to write "0" to these bits.

[bit8] STS2 : A/D activation factor select bit

The activation factor of the A/D conversion is selected by a combination of this bit and bit12, bit11 (STS1,STS0) of the A/D activation trigger control status register (ADTCS). Please refer to "4.2.4 A/D Activation Trigger Control Status Register : ADTCS0 to ADTCS47" for details.

Notes:

- Since the A/D activation factor select bit changes immediately when the bits are rewritten, change this bit while the activation factors of the current target and of the target to be changed are inactive and the A/D conversion is not being requested (ADTCS:BUSY=1).
- Please set these bits including ADTCS.STS1 and ADTCS.STS0 as software activation ("000_B"), and set a corresponding bit of ADTSE (activation channel) to the software activation disable (ADT bit =0) when A/D conversion is not being requested.
- Please confirm the 16-bit free-run timer has stopped before setting the A/D activation factor select bit.

[bit7 to bit5] Reserved

Be sure to write "0" to these bits.

[bit4 to bit0] CHSEL4 to CHSEL0 : Analog channel select bit

CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	Explanation
0	0	0	0	0	Channel 0
0	0	0	0	1	Channel 1
			:		:
0	0	1	1	0	Channel 6
0	0	1	1	1	Channel 7
0	1	0	0	0	Channel 8
			:		:
0	1	1	1	0	Channel 14
0	1	1	1	1	Channel 15
1	0	0	0	0	Channel 16
			:		:
1	1	1	1	0	Channel 30
1	1	1	1	1	Channel 31

These bits select the analog channel with the specified value.

Do not change the analog channel select bit when A/D conversion is being requested.

■ ADTECS32 to ADTECS47: Address 1538_H to 1556_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							STS2
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	CHSEL3	CHSEL2	CHSEL1	CHSEL0
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 bit9] Reserved

Be sure to write "0" to these bits.

[bit8] STS2 : A/D activation factor select bit

The activation factor of the A/D conversion is selected by a combination of this bit and bit12, bit11 (STS1,STS0) of the A/D activation trigger control status register (ADTCS). Please refer to "4.2.4 A/D Activation Trigger Control Status Register : ADTCS0 to ADTCS47" for details.

Notes:

- Since the A/D activation factor select bit changes immediately when the bits are rewritten, change this bit while the activation factors of the current target and of the target to be changed are inactive and the A/D conversion is not being requested (ADTCS:BUSY=1).
- Please set these bits including ADTCS.STS1 and ADTCS.STS0 as software activation ("000_B"), and set a corresponding bit of ADTSE (activation channel) to the software activation disable (ADT bit =0) when A/D conversion is not being requested.
- Please confirm the 16-bit free-run timer has stopped before setting the A/D activation factor select bit.

[bit7 to bit4] Reserved

Be sure to write "0" to these bits.

[bit3 to bit0] CHSEL3 to CHSEL0 : Analog channel select bit

CHSEL3	CHSEL2	CHSEL1	CHSEL0	Explanation
0	0	0	0	Channel 32
0	0	0	1	Channel 33
0	0	1	0	Channel 34
:				:
1	1	1	0	Channel 46
1	1	1	1	Channel 47

4.2.7. Upper Bound Threshold Setting Register : ADRCUT0 to ADRCUT7

The bit configuration of the upper bound threshold setting register is shown.

The upper bound threshold setting register (ADRCUT) sets the upper bound threshold used by the range comparison. Moreover, the upper bound threshold can set 4 types.

- **ADRCUT0: Address $140C_H$ (Access: Byte, Half-word, Word)**
- **ADRCUT1: Address 1410_H (Access: Byte, Half-word, Word)**
- **ADRCUT2: Address 1414_H (Access: Byte, Half-word, Word)**
- **ADRCUT3: Address 1418_H (Access: Byte, Half-word, Word)**
- **ADRCUT4: Address 1578_H (Access: Byte, Half-word, Word)**
- **ADRCUT5: Address $157C_H$ (Access: Byte, Half-word, Word)**
- **ADRCUT6: Address 1580_H (Access: Byte, Half-word, Word)**
- **ADRCUT7: Address 1584_H (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				C11	C10	C9	C8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	C7	C6	C5	C4	C3	C2	C1	C0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit12] Reserved

Be sure to write "0" to these bits.

[bit11 to bit0] C11 to C0 : Upper bound threshold bit

C11 to C0	Function
	Upper bound threshold

These bits set the upper bound threshold used by the range comparison.

Note:

Please do not change the upper bound threshold bit while requesting the A/D conversion.

4.2.8. Lower Bound Threshold Setting Register : ADRCLT0 to ADRCLT7

The bit configuration of the lower threshold setting register is shown.

The lower bound threshold setting register (ADRCLT) sets the lower bound threshold used by the range comparison. Moreover, the lower bound threshold can set 4 types.

- **ADRCLT0: Address $140E_H$ (Access: Byte, Half-word, Word)**
- **ADRCLT1: Address 1412_H (Access: Byte, Half-word, Word)**
- **ADRCLT2: Address 1416_H (Access: Byte, Half-word, Word)**
- **ADRCLT3: Address $141A_H$ (Access: Byte, Half-word, Word)**
- **ADRCLT4: Address $157A_H$ (Access: Byte, Half-word, Word)**
- **ADRCLT5: Address $157E_H$ (Access: Byte, Half-word, Word)**
- **ADRCLT6: Address 1582_H (Access: Byte, Half-word, Word)**
- **ADRCLT7: Address 1586_H (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				C11	C10	C9	C8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	C7	C6	C5	C4	C3	C2	C1	C0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit12] Reserved

Be sure to write "0" to these bits.

[bit11 to bit0] C11 to C0 : Upper bound threshold bit

C11 to C0	Function
	Lower bound threshold

These bits set the lower bound threshold used by the range comparison.

Note:

Please do not change the lower bound threshold bit while requesting the A/D conversion.

4.2.9. Range Compare Control Status Register: ADRCCS0 to ADRCCS47

The bit configuration of the range compare control status register is shown.

The range compare control status register (ADRCCS) confirms the instruction of the continuous detection count and the state of the continuous detection count. And, this register selects inside/outside the range confirmation, range comparison interrupt request enable/disable, range comparison execution enable/disable, and upper and lower bound threshold.

■ ADRCCS0 to ADRCCS31: Address $141C_H$ to $143B_H$ (Access: Byte, Half-word, Word)

■ ADRCCS32 to ADRCCS47: Address 1588_H to 1597_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RCOCD2	RCOCD1	RCOCD0	RCOIRS	RCOIE	RCOE	RCOTS1	RCOTS0
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] RCOCD2 to RCOCD0 : Continuous detection count specification and state indicate bit

RCOCD2 to RCOCD0	Explanation	
	Read by an instruction other than read-modify-write (RMW) instruction	Read by a read-modify-write (RMW) instruction or writing
"000_B"	State of continuous detection : 0 time	Setting disable
"001_B"	State of continuous detection : 1 time	Continuous detection is specified 1 time
"010_B"	State of continuous detection : 2 times	Continuous detection is specified 2 times
"011_B"	State of continuous detection : 3 times	Continuous detection is specified 3 times
"100_B"	State of continuous detection : 4 times	Continuous detection is specified 4 times
"101_B"	State of continuous detection : 5 times	Continuous detection is specified 5 times
"110_B"	State of continuous detection : 6 times	Continuous detection is specified 6 times
"111_B"	State of continuous detection : 7 times	Continuous detection is specified 7 times

· These bits show the specification of the continuous detection count of the range comparison result and the state of

the continuous detection count.

- When the range comparison result reaches the continuous count specification value, "1" is set to the range comparison interrupt factor flag bit of the corresponding activation channel. Moreover, continuous detection stops by the continuous count specification value.

- When an instruction other than the read-modify-write (RMW) instruction is used, state of continuous detection is read.
- When the read-modify-write (RMW) instruction is used, the writing value (continuous count instruction value) is read.

Notes:

- Please do not set "000_B" to the continuous detection count specification and state indicate bit.
- Please do not change the continuous detection count specification and state indicate bit while requesting the A/D conversion.
- Please do not change the continuous detection count specification and state indicate bit (ADRCCS.RCOE="1") while the range comparison execution enable.

[bit4] RCOIRS : Confirmation select bit inside/outside range

RCOIRS	Explanation
0	Confirmation outside the range
1	Confirmation inside the range

- The A/D data bit selects the range comparison condition of inside/outside the range for the upper bound threshold bit and the lower bound threshold bit selected by the upper/lower bound threshold select bit.
- When confirming outside the range (RCOIRS="0"), the range comparison condition is the following.
  A/D data bit (ADTCD.D11 to D0) > upper bound threshold bit (ADRCUT.C11 to C0)
  or
  A/D data bit (ADTCD.D11 to D0) < lower threshold bit (ADRCLT.C11 to C0).
- When confirming inside the range (RCOIRS="1"), the range comparison condition is the following.
  A/D data bit (ADTCD.D11 to D0) ≤ upper bound threshold bit (ADRCUT.C11 to C0)
  or
  A/D data bit (ADTCD.D11 to D0) ≥ lower threshold bit (ADRCLT.C11 to C0)
- When the range comparisons of confirmations outside the range are detected, it is possible to confirm that this bit is larger than the upper bound threshold or smaller than the lower bound threshold by threshold over flag bit.

Note:

Please do not change the confirmation select bit inside/outside range while requesting the A/D conversion.

[bit3] RCOIE : Range comparison interrupt request enable bit

RCOIE	Explanation
0	Range comparison interrupt disabled
1	Range comparison interrupt enabled

When the range comparison interrupt factor flag bit of the corresponding activation channel is setting in "1" and this bit is set to the range comparison interrupt request enable, the interrupt request is generated.

[bit2] RCOE : Range comparison execution enable bit

RCOE	Explanation
0	Range comparison execution disabled
1	Range comparison execution enabled

- This bit selects range comparison execution enable/disable.
- When range comparison execution enable bit is "0", the range comparison execution is disabled. Moreover, the state of the continuous detection count is initialized by "000_B".
- When range comparison execution enable bit is "1", the range comparison execution is enabled.

[bit1,bit0] RCOTS1, RCOTS0 : Upper and lower bound threshold select bit

- ADRCCS0 to ADRCCS31 (A/D converter unit 0)

RCOTS1	RCOTS0	Explanation
0	0	Upper bound threshold setting register 0 / lower bound threshold setting register 0 is selected.
0	1	Upper bound threshold setting register 1 / lower bound threshold setting register 1 is selected.
1	0	Upper bound threshold setting register 2 / lower bound threshold setting register 2 is selected.
1	1	Upper bound threshold setting register 3 / lower bound threshold setting register 3 is selected.

.

· ADRCCS32 to ADRCCS47 (A/D converter unit 1)

RCOTS1	RCOTS0	Explanation
0	0	Upper bound threshold setting register 4 / lower bound threshold setting register 4 is selected.
0	1	Upper bound threshold setting register 5 / lower bound threshold setting register 5 is selected.
1	0	Upper bound threshold setting register 6 / lower bound threshold setting register 6 is selected.
1	1	Upper bound threshold setting register 7 / lower bound threshold setting register 7 is selected.

One combination is selected from 4 types of the upper bound threshold setting register 0 to 3/4 to 7 and the lower bound threshold setting register 0 to 3/4 to 7.

Note:

Please do not change the upper and lower bound threshold select bit while requesting the A/D conversion.

4.2.10. Range Compare Threshold Over Flag Register : ADRCOT0, ADRCOT1

The bit configuration of the range compare threshold over flag register is shown.

The range compare threshold over flag register (ADRCOT) indicates a bit larger than the upper bound threshold or smaller than the lower bound threshold as a result of comparing the range in the setting of the confirmation outside the range.

■ ADRCOT0: Address $143C_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	RCOOF31	RCOOF30	RCOOF29	RCOOF28	RCOOF27	RCOOF26	RCOOF25	RCOOF24
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	RCOOF23	RCOOF22	RCOOF21	RCOOF20	RCOOF19	RCOOF18	RCOOF17	RCOOF16
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
	RCOOF15	RCOOF14	RCOOF13	RCOOF12	RCOOF11	RCOOF10	RCOOF9	RCOOF8
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
	RCOOF7	RCOOF6	RCOOF5	RCOOF4	RCOOF3	RCOOF2	RCOOF1	RCOOF0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit0] RCOOF31 to RCOOF0 : Conversion data error flag bits

RCOOF[n]	Function
0	Smaller than the lower bound threshold (A/D data < lower bound threshold bit)
1	Larger than the upper bound threshold (A/D data > upper bound threshold bit)

(n=0 to 31)

- When outside the range is confirmed, these bits indicate the state that the range comparison result is larger than the upper bound threshold setting register or it is smaller than the lower bound threshold setting register.
- When outside the range is confirmed, the threshold over flag bit maintains the last value when the range comparison result is inside the range.
- When outside the range is confirmed, the threshold over flag bit is not updated and the last value is maintained, even if the range comparison result detects outside the range if the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1".
- When inside the range is confirmed, the threshold over flag bit does not have the meaning, and maintains the last value.

■ ADRCOT1: Address $15A8_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	RCOOF47	RCOOF46	RCOOF45	RCOOF44	RCOOF43	RCOOF42	RCOOF41	RCOOF40
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
	RCOOF39	RCOOF38	RCOOF37	RCOOF36	RCOOF35	RCOOF34	RCOOF33	RCOOF32
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit16] Reserved

Be sure to write "0" to these bits.

[bit15 to bit0] RCOOF47 to RCOOF32 : Conversion data error flag bits

RCOOF[n]	Function
0	Smaller than the lower bound threshold (A/D data < lower bound threshold bit)
1	Larger than the upper bound threshold (A/D data > upper bound threshold bit)

(n=32 to 47)

- When outside the range is confirmed, these bits indicate the state that the range comparison result is larger than the upper bound threshold setting register or it is smaller than the lower bound threshold setting register.
- When outside the range is confirmed, the threshold over flag bit maintains the last value when the range comparison result is inside the range.
- When outside the range is confirmed, the threshold over flag bit is not updated and the last value is maintained, even if the range comparison result detects outside the range if the range comparison interrupt factor flag bit of the corresponding activation channel is set to "1".
- When inside the range is confirmed, the threshold over flag bit does not have the meaning, and maintain the last value.

4.2.11. Range Compare Flag Register : ADRCIF0, ADRCIF1

The bit configuration of the range compare flag register is shown.

The range compare flag register (ADRCIF) indicates the interrupt factor by the continuous detection of the range comparison result.

■ ADRCIF0: Address 1440_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	RCINT31	RCINT30	RCINT29	RCINT28	RCINT27	RCINT26	RCINT25	RCINT24
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	RCINT23	RCINT22	RCINT21	RCINT20	RCINT19	RCINT18	RCINT17	RCINT16
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
	RCINT15	RCINT14	RCINT13	RCINT12	RCINT11	RCINT10	RCINT9	RCINT8
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
	RCINT7	RCINT6	RCINT5	RCINT4	RCINT3	RCINT2	RCINT1	RCINT0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit0] RCINT31 to RCINT 0 : Conversion data error flag bits

RCINT[n]	Explanation	
	Read	Write
0	Range comparison interrupt factor clear state	Clear of bit
1	State of interrupt factor generation by continuous detection of range comparison result	This bit value does not change and there is no influence on an operation by writing.

(n=0 to 31)

- RCINT bit is set to "1" by the continuous detection of the range comparison result of the corresponding activation channel.
- When RCINT bit and the range comparison interrupt request enable bit of the corresponding activation channel are "1", the range comparison interrupt request is generated.

Notes:

- If the read-modify-write (RMW) instruction is executed, "1" will be read out.
- The hardware setting is given priority when software clearing (RCINT ="0" writing) and the hardware setting are generated at the same time.

■ ADRCIF1: Address $15AC_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	RCINT47	RCINT46	RCINT45	RCINT44	RCINT43	RCINT42	RCINT41	RCINT40
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
	RCINT39	RCINT38	RCINT37	RCINT36	RCINT35	RCINT34	RCINT33	RCINT32
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit16] Reserved

Be sure to write "0" to these bits.

[bit15 to bit0] RCINT47 to RCINT32 : Conversion data error flag bits

RCINT[n]	Explanation	
	Read	Write
0	Range comparison interrupt factor clear state	Clear of bit
1	State of interrupt factor generation by continuous detection of range comparison result	This bit value does not change and there is no influence on an operation by writing.

(n=32 to 47)

- RCINT bit is set to "1" by the continuous detection of the range comparison result of the corresponding activation channel.
- When RCINT bit and the range comparison interrupt request enable bit of the corresponding activation channel are "1", the range comparison interrupt request is generated.

Notes:

- If the read-modify-write (RMW) instruction is executed, "1" will be read out.
- The hardware setting is given priority when software clearing (RCINT ="0" writing) and the hardware setting are generated at the same time.

4.2.12. Scan Conversion Control Status Register : ADSCANS0, ADSCANS1

The bit configuration of the scan conversion control status register is shown.

The scan conversion control status register (ADSCANS) selects continuous or stop scan mode when the conversion count is specified, selects scan conversion completion interrupt request enable/disable, and indicates the state of the scan conversion completion interrupt request.

■ ADSCANS0: Address 1444_H (Access: Byte, Half-word, Word)

■ ADSCANS1: Address $15B0_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SCINT	SCIE	SCMD	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R/W	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit7] SCINT : Scan conversion completion interrupt factor flag bit

SCINT	Explanation	
	Read	Write
0	Scan conversion completion interrupt factor clear state when conversion count of each channel is specified	Clear of bit
1	State of interrupt factor generation by scan conversion completion when conversion count of each channel is specified	This bit value does not change and there is no influence on an operation by writing.

- The SCINT bit is set to "1" by the scan conversion completion when the conversion count of each channel is specified.
- When SCINT bit and scan conversion completion interrupt request enable bit are "1", the scan conversion completion interrupt request when the conversion count is specified is generated.

Notes:

- If the read-modify-write (RMW) instruction is executed, "1" will be read out.
- The hardware setting is given priority when software clearing (SCINT ="0" writing) and the hardware setting are generated at the same time.

[bit6] SCIE : Scan conversion completion interrupt request enable bit

SCIE	Explanation
0	Scan conversion completion interrupt disabled
1	Scan conversion completion interrupt enabled

When the scan conversion completion interrupt factor flag bit and the scan conversion completion interrupt request enable bit are set in "1", the interrupt request is generated.

[bit5] SCMD : Continuous and stop scan conversion mode select bit

SCMD	Explanation
0	Continuous scan conversion mode
1	Stop scan conversion mode

The scan conversion mode is selected when the conversion count of each channel is specified.

[bit4 to bit0] Reserved

Be sure to write "0" to these bits.

4.2.13. Activation Channel Conversion Count Setting Register : ADNCS0 to ADNCS23

The bit configuration of the activation channel conversion count setting register is shown.

The activation channel conversion count setting register (ADNCS) sets conversion count specification scan conversion execution enable/disable and specifies the conversion count for each activation channel.

■ **ADNCSm(m = 0 to 15): Address 1448_H to 1457_H (Access: Byte, Half-word, Word)**

■ **ADNCSn(n = 16 to 23): Address $15B4_H$ to $15BB_H$ (Access: Byte, Half-word, Word)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CNTEN (m*2+1)	Reserved	CCNT (m*2+1)1	CCNT (m*2+1)0	CNTEN (m*2)	Reserved	CCNT (m*2)1	CCNT (m*2)0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W

[bit7, bit3] CNTEN : Conversion count specification scan conversion execution enable bit

CNTEN[n]	Explanation
0	Conversion count specification scan conversion execution disabled
1	Conversion count specification scan conversion execution enabled

(N= 0 to 47)

- These bits can set conversion count specification scan conversion execution enable/disable for each activation channel.
- Only one scan conversion group can be controlled by the scan conversion by conversion count specification for each channel, in units of 12-bit A/D converter unit.

[bit6, bit2] Reserved

Be sure to write "0" to these bits.

[bit5, bit4, bit1, bit0] CCNT : Conversion count specification bit

CCNT[n]1	CCNT[n]0	Explanation
0	0	1 time of conversion count
0	1	2 times of conversion count
1	0	3 times of conversion count
1	1	4 times of conversion count

(n= 0 to 47)

When the conversion count specification scan conversion execution of the activation channel is enabled, the scan conversion count is controlled.

Note:

Please do not change the conversion count specification bit while the corresponding activation channel is requesting the A/D conversion.

4.2.14. Data Protection Status Flag Register : ADPRTF0, ADPRTF1

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

The data protection state flag register (ADPRTF) indicates the protection state of A/D data register of each activation channel.

■ ADPRTF0: Address 1458_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	PRTF31	PRTF30	PRTF29	PRTF28	PRTF27	PRTF26	PRTF25	PRTF24
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	PRTF23	PRTF22	PRTF21	PRTF20	PRTF19	PRTF18	PRTF17	PRTF16
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
	PRTF15	PRTF14	PRTF13	PRTF12	PRTF11	PRTF10	PRTF9	PRTF8
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
	PRTF7	PRTF6	PRTF5	PRTF4	PRTF3	PRTF2	PRTF1	PRTF0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit0] PRTF31 to PRTF0 : Data protection status flag bits

PRTF	Explanation
0	Not in data protection state
1	In data protection state

- These bits indicate the data protection state for the A/D data registers of each activation channel.
- The writing operation doesn't influence the data protection state.

■ ADPRTF1: Address 15C4$_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	PRTF47	PRTF46	PRTF45	PRTF44	PRTF43	PRTF42	PRTF41	PRTF40
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
	PRTF39	PRTF38	PRTF37	PRTF36	PRTF35	PRTF34	PRTF33	PRTF32
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit16] Reserved

Be sure to write "0" to these bits.

[bit15 to bit0] PRTF47 to PRTF32 : Data protection status flag bits

PRTF	Explanation
0	Not in data protection state
1	In data protection state

- These bits indicate the data protection state for the A/D data registers of each activation channel.

- The writing operation doesn't influence the data protection state.

4.2.15. Activation Channel Conversion Completion Flag Register : ADEOCF0, ADEOCF1

The bit configuration of the activation channel conversion completion flag register is shown.

The activation channel conversion completion flag register (ADEOCF) indicates the conversion count completion for each activation channel when the conversion count specification scan is converted.

■ ADEOCF0: Address $145C_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	EOCF31	EOCF30	EOCF29	EOCF28	EOCF27	EOCF26	EOCF25	EOCF24
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	EOCF23	EOCF22	EOCF21	EOCF20	EOCF19	EOCF18	EOCF17	EOCF16
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

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EOCF15	EOCF14	EOCF13	EOCF12	EOCF11	EOCF10	EOCF9	EOCF8
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
	EOCF7	EOCF6	EOCF5	EOCF4	EOCF3	EOCF2	EOCF1	EOCF0
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

[bit31 to bit0] EOCF31 to EOCF0 : Conversion count completion flag bits

EOCF	Explanation
0	The count of the conversion count specification is not completed.
1	The count of the conversion count specification is completed.

- When the conversion count specification scan conversion execution is enabled (ADNCS.CNTEN[n]="1") for corresponding activation channel, the indicated value of the conversion count completion flag bit (EOCF) is valid.
- The state of completion for scan conversion process of specified count (ADNCS.CCNT1, CCNT0) is indicated.
- When EOCF="0 ", scan conversion process of specified count has not been completed yet.
- When EOCF="1 ", scan conversion process of specified count has been completed.
- When A/D conversion is not requested (ADTCS.BUSY="0") or scan conversion is resumed from the beginning, conversion count completion flag bit (EOCF) is cleared to "0".
- The write operation has no effect on data protection status.

Notes:

- When the conversion count specification scan conversion execution is disabled (ADNCS.CNTEN[n]="0") for corresponding activation channel, the read data of the conversion count completion flag bit (EOCF[n]) has no meaning.
- During the continuous scan conversion mode (ADSCANS.SCMD="0"), conversion count completion flag bit (EOCF) which is set for the last channel of scan conversion is resumed from the beginning soon after completing scan conversion. Therefore, if it is set to "1", it is cleared immediately to "0".

■ ADEOCF1: Address $15C8_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
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

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EOCF47	EOCF46	EOCF45	EOCF44	EOCF43	EOCF42	EOCF41	EOCF40
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
	EOCF39	EOCF38	EOCF37	EOCF36	EOCF35	EOCF34	EOCF33	EOCF32
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

[bit31 to bit16] Reserved

Be sure to write "0" to these bits.

[bit15 to bit0] EOCF47 to EOCF32 : Conversion count completion flag bits

EOCF	Explanation
0	The count of the conversion count specification is not completed.
1	The count of the conversion count specification is completed.

- When the conversion count specification scan conversion execution is enabled (ADNCS.CNTEN[n]="1") for corresponding activation channel, the indicated value of the conversion count completion flag bit (EOCF) is valid.
- The state of completion for scan conversion process of specified count (ADNCS.CCNT1, CCNT0) is indicated.
- When EOCF="0 ", scan conversion process of specified count has not been completed yet.
- When EOCF="1 ", scan conversion process of specified count has been completed.
- When A/D conversion is not requested (ADTCS.BUSY="0") or scan conversion is resumed from the beginning, conversion count completion flag bit (EOCF) is cleared to "0".
- The write operation has no effect on data protection status.

Notes:

- When the conversion count specification scan conversion execution is disabled (ADNCS.CNTEN[n]="0") for corresponding activation channel, the read data of the conversion count completion flag bit (EOCF[n]) has no meaning.
- During the continuous scan conversion mode (ADSCANS.SCMD="0"), conversion count completion flag bit (EOCF) which is set for the last channel of scan conversion is resumed from the beginning soon after completing scan conversion. Therefore, if it is set to "1", it is cleared immediately to "0".

4.3. Register of 12-BIT A/D Converter Control

The registers of the 12-bit A/D converter control is explained.

The 12-bit A/D converter control uses A/D control status register, A/D channel status register, A/D mode setting register, and A/D sampling time setting register.

4.3.1. A/D Control Status Register: ADCS0, ADCS1

The bit configuration of the A/D control status register is shown.

The A/D control status register (ADCS) provides the function to confirm conversion.

■ **ADCS0: Address 1460_H (Access: Byte, Half-word, Word)**

■ **ADCS1: Address $15CC_H$ (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit15] BUSY : A/D conversion operating bit

BUSY	Function
0	The A/D conversion is stopping.
1	The A/D conversion is operating.

- This bit indicates the A/D converter operation state.
- When the reading value of this bit is "0", it is shown that A/D conversion is stopping. When the reading value of this bit is "1", it is shown that A/D conversion is operating.
- This bit value does not change and there is no influence on an operation by writing.

[bit14 to bit0] Reserved

Be sure to write "0" to these bits.

4.3.2. A/D Channel Status Register : ADCH

The bit configuration of the A/D channel status register is shown.

The A/D channel status register (ADCH) shows the analog channel number of conversion target while the A/D conversion is operating.

■ ADCH0: Address 1462_H (Access: Byte, Half-word, Word)

■ ADCH1: Address $15CE_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			CH4	CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7 to bit5] Reserved

Be sure to write "0" to these bits.

[bit4 to bit0] CH4 to CH0 : Analog channel bits

CH4	CH3	CH2	CH1	CH0	Function	
					ADCH0	ADCH1
0	0	0	0	0	Ch.0	Ch.32
0	0	0	0	1	Ch.1	Ch.33
:					:	:
0	0	1	1	0	Ch.6	Ch.38
0	0	1	1	1	Ch.7	Ch.39
0	1	0	0	0	Ch.8	Ch.40
:					:	:
0	1	1	1	0	Ch.14	Ch.46
0	1	1	1	1	Ch.15	Ch.47
1	0	0	0	0	Ch.16	Setting disable
1	0	0	0	1	Ch.17	
:					:	
1	1	1	1	0	Ch.30	
1	1	1	1	1	Ch.31	

These bits show the analog channel number of the conversion target while the A/D conversion is operating.

4.3.3. A/D Mode Setting Register : ADMD

The bit configuration of the A/D mode setting register is shown.

The A/D mode setting register (ADMD) sets the compare time and the sampling time of the A/D conversion.

■ ADMD0: Address 1463_H (Access: Byte, Half-word, Word)

■ ADMD1: Address $15CF_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STPCEN	Reserved			CT1	CT0	ST1	ST0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7] STPCEN : Sampling time setting per channel enable bit

STPCEN	Explanation
0	Sampling time setting common to all channels
1	Sampling time setting of each channel

- This bit selects setting of each channel or setting common to all channels as the sampling time setting in the A/D conversion.
- When STPCEN="0", the sampling time common to all channels is enabled. The sampling time is set by the sampling time set bit.
- When STPCEN="1", the sampling time of each channel is enabled. The sampling time of each channel is set by sampling time setting bit per A/D channel.

[bit6 to bit4] Reserved

Be sure to write "0" to these bits.

[bit3, bit2] CT1 , CT0 : Compare time setting bits

CT1	CT0	Function
0	0	28 Peripheral clock (A/D clock output : Peripheral clock /2)
0	1	42 Peripheral clock (A/D clock output : Peripheral clock /3)
1	0	56 Peripheral clock (A/D clock output : Peripheral clock /4)
1	1	112 Peripheral clock (A/D clock output : Peripheral clock /8)

- These bits select the compare time in the A/D conversion.

- After the analog input is taken (sampling time passage), the data of the conversion result is fixed when the time set to these bits has elapsed.

Example of setting

Peripheral clock (MHz)	CT1	CT0	Compare time (ns)
40	0	0	700
32	0	0	875
24	0	0	1166.7
16	0	0	1750

Note:

Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

[bit1,bit0] ST1, ST0 : Sampling time setting bits

ST1	ST0	Function
0	0	12 Peripheral clock (A/D clock output : Peripheral clock /2)
0	1	18 Peripheral clock (A/D clock output : Peripheral clock /3)
1	0	24 Peripheral clock (A/D clock output : Peripheral clock /4)
1	1	48 Peripheral clock (A/D clock output : Peripheral clock /8)

- These bits select the sampling time in the A/D conversion.
- The analog input is taken during the period set in these bits after A/D is activated.

Example of setting (Use conditions: AV_{CC}=2.7V to 5.5V)

Peripheral clock (MHz)	ST1	ST0	Sampling time (ns)
40	1	1	1200
32	1	1	1500
24	1	0	1000
16	0	1	1125

Example of setting (Use conditions: AV_{CC}=4.5V to 5.5V)

Peripheral clock (MHz)	ST1	ST0	Sampling time (ns)
40	1	1	1200
32	1	0	750
24	0	1	750
16	0	0	750

Note:

Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

4.3.4. A/D Sampling Time Setting Per Channel Register : ADSTPCS

The bit configuration of the A/D sampling time setting per channel register is shown.

The A/D sampling time setting per channel register (ADSTPCS) sets the sampling time of the A/D conversion for each channel.

■ ADSTPCSm (m=0 to 7): Address 1464_H to $146B_H$ (Access: Byte, Half-word, Word)

■ ADSTPCSm (m=8 to 11): Address $15D0_H$ to $15D3_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STCH (m*4+3)1	STCH (m*4+3)0	STCH (m*4+2)1	STCH (m*4+2)0	STCH (m*4+1)1	STCH (m*4+1)0	STCH (m*4)1	STCH (m*4)0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] STCHn1, STCHn0 (n=0 to 47) : Sampling time setting bits

STCHn1	STCHn0	Explanation
0	0	12 Peripheral clock (A/D clock output : Peripheral clock /2)
0	1	18 Peripheral clock (A/D clock output : Peripheral clock /3)

STCHn1	STCHn0	Explanation
1	0	24 Peripheral clock (A/D clock output : Peripheral clock /4)
1	1	48 Peripheral clock (A/D clock output : Peripheral clock /8)

- These bits select the sampling time in the A/D conversion for each channel.
- If the sampling time setting permission bit (ADMD.STPCEN) is set to "1" for each channel, the sampling time setting (STCHn1, STCHn0) for each channel is enabled.
- The correspondence of a set channel and an analog channel is shown below.

Note:

Rewrite these bits while the A/D conversion operations are still stopped.

Table 4-4 Correspondence between Sampling Time Setting for Each Channel and Analog Channel (A/D converter unit 0)

Sampling time setting per channel enable bit	Analog channel number
	ADSTPCS0 to 31
STCH001, STCH000	Channel 0
STCH011, STCH010	Channel 1
:	:
STCH061, STCH060	Channel 6
STCH071, STCH070	Channel 7
STCH081, STCH080	Channel 8
:	:
STCH141, STCH140	Channel 14
STCH151, STCH150	Channel 15
:	:
STCH301, STCH300	Channel 30
STCH311, STCH310	Channel 31

Table 4-5 Correspondence between Sampling Time Setting for Each Channel and Analog Channel (A/D converter unit 1)

Sampling time setting per channel enable bit	Analog channel number
	ADSTPCS32 to 47
STCH321, STCH320	Channel 32
STCH331, STCH330	Channel 33
:	:
STCH461, STCH460	Channel 46
STCH471, STCH470	Channel 47

5. Operation

This section explains operation.

5.1 Interrupt of A/D activation compare

5.2 A/D activation compare operation

5.3 A/D Activation Arbitration Operation

5.4 12-bit A/D Converter Operation

5.1. Interrupt of A/D activation compare

This section explains the interrupt control bit and the interrupt factor of A/D activation compare.

The interrupt control bit and the interrupt factor of A/D activation compare are shown.

5.1.1. A/D conversion completion interrupt

The A/D conversion completion interrupt is explained.

Table 5-1 Interrupt Control Bit and Interrupt Factor by A/D Conversion Completion Interrupt

	A/D conversion completion interrupt
Interrupt request flag bit	INT:bit14 of A/D activation trigger control status register (ADTCS)
Interrupt request enable bit	INTE:bit13 of A/D activation trigger control status register (ADTCS)
Interrupt factor	Writing of A/D conversion result in A/D data register

The A/D conversion completion interrupt request can be generated when the A/D conversion of compare channel that activates the A/D converter completes. Moreover, the A/D conversion completion interrupt can be controlled in units of activation channel.
When the A/D conversion result is set in A/D data register (ADTCD), the INT bit of A/D activation trigger control status register (ADTCS) is set to "1". At this time, when the interrupt request enable bit has been enabled (ADTCS.INTE="1"), the interrupt request is output to the interrupt controller.

5.1.2. Scan conversion completion interrupt by conversion count specification

The scan conversion completion interrupt by conversion count specification is explained.

Table 5-2 Interrupt Control Bit and Interrupt Factor by Scan Conversion Completion by Conversion Count Specification

	Scan conversion completion interrupt by conversion count specification
Interrupt request flag bit	SCINT:bit7 of scan conversion control status register (ADSCANS)
Interrupt request enable bit	SCIE:bit6 of scan conversion control status register (ADSCANS)
Interrupt factor	After completing the A/D conversion of a specified count of the final activation channel in the scan conversion of the conversion count specification

A/D activation compare can generate the interrupt request at scan conversion by the conversion count specification, when the A/D conversion of a specified count of the final activation channel is completed. Moreover, the scan conversion completion interrupt by the conversion count specification is controlled in units of 12-bit A/D converter unit.
When the A/D conversion of a specified count of the final activation channel of the scan conversion by the conversion count specification is completed, scan conversion completion interrupt factor flag bit (ADSCANS.SCINT) is set to "1". At this time, when the scan conversion completion interrupt request enable bit has been enabled (ADSCANS.SCIE="1"), the interrupt request is output to the interrupt controller. Moreover, the scan conversion by the conversion count specification is controlled by conversion count specification scan conversion execution enable bit (ADNCS.CNTEN) and conversion count specification bit (ADNCS.CCNT1, CCNT0).

5.1.3. Range comparison interrupt

The range comparison interrupt is explained.

Table 5-3 Interrupt Control Bit and Interrupt Factor by Range Comparison Interrupt

	Range comparison interrupt
Interrupt request flag bit	RCINT[n]:bit[n] of range compare flag register (ADRCIF)
Interrupt request enable bit	RCOIE:bit3 of range compare control status register (ADRCCS[n])
Interrupt factor	After judging a continuous detecting function by the range comparison execution

(n=Corresponding activation channel)

When the range comparison execution has been enabled, the A/D activation comparison performs a range comparison of the conversion result stored in the A/D data register (ADTCD) with one of the combinations within the upper bound threshold setting register 0 to 7 (ADRCUT0 to 7)/lower bound threshold setting register 0 to 7 (ADRCLT0 to 7) of the range comparison. When continuity can be confirmed as a result of comparing ranges, the range comparison interrupt request can be generated. Moreover, the range comparison interrupt can be controlled with each activation channel.
When the A/D conversion result is set in A/D data register (ADTCD) when the range comparison execution is enabled (ADRCCS.RCOE="1"), the range comparison is executed. The range comparison execution can select the range comparison condition by the inside/outside the range confirmation select bit (ADRCCS.RCOIRS).

- When confirming outside the range (ADRCCS.RCOIRS="0"), the range comparison condition is the following.
  Lower bound threshold setting register > A/D conversion result
  or
  Upper bound threshold setting register < A/D conversion result
- When confirming inside the range (ADRCCS.RCOIRS ="1"), the range comparison condition is the following.
  Lower bound threshold setting register ≤ A/D conversion result
  and
  Upper bound threshold setting register ≥ A/D conversion result

When the range comparison condition is continuously detected, the range comparison interrupt factor flag bit (ADRCIF.RCINT) is set to "1". At this time, when the range comparison interrupt request enable bit has been enabled (ADRCCS.RCOIE="1"), the interrupt request is output to the interrupt controller. The timing of output by the range comparison interrupt request is delayed by two cycles with the peripheral clock than the A/D conversion completion interrupt. Moreover, the continuous detection count can be selected from 1 to 7 times by the continuous detection count specification bit (ADRCCS.RCOCD2 to RCOCD0).

5.2. A/D activation compare operation

This section explains the A/D activation compare operation.

The A/D activation can be requested by either the software, the external trigger, the reload timer, the compare match

(When the value of 16-bit free-run timer reaches the instruction value) or PPG.

5.2.1. A/D activation

The A/D activation is explained.

The activation is requested to either of two A/D converters. The activation request can be generated at each analog channel (Max 48 channels).
The activation request signal is generated to the A/D activation arbitration with either the software, the external trigger (falling), the reload timer (rising), the compare match (When the free-run timer is corresponding to the compare register value) or PPG. There are three A/D activation request signals, either "Software activation request ", "External trigger/reload timer/PPG activation request " or "Compare match activation request " becomes active in each activation channel.
The activation request is cleared on completion of the A/D conversion of the corresponding channel and the conversion data is stored in the A/D data register. In that case, the interrupt can be generated.
Even if the activation factor is received in the activation request, the activation request is not reactivated in the activation channel.

- The activation channels are assigned to either of two A/D converters. The assignment of each unit is as shown in the table below.

Table 5-4 Channel Allocation of Each Unit

	Unit 0	Unit 1
MB91F52xB	ch.0, ch.1, ch.7, ch.10, ch.11, ch.14 to ch.17, ch.22, ch.27, ch.28, ch.31	ch.32, ch.34, ch.35, ch.37, ch.38, ch.40 to ch.47
MB91F52xD	ch.0, ch.1, ch.7, ch.10 to ch.12, ch.14 to ch.17, ch.22, ch.26, ch.27, ch.28, ch.31	ch.32 to ch.47
MB91F52xF	ch.0, ch.1, ch.7, ch.9 to ch.19, ch.22, ch.23, ch.26 to ch.29, ch.31	ch.32 to ch.47
MB91F52xJ	ch.0, ch.1, ch.7, ch.9, ch.10 to ch.31	ch.32 to ch.47
MB91F52xK	ch.0 to ch.31	ch.32 to ch.47
MB91F52xL	ch.0 to ch.31	ch.32 to ch.47

5.2.2. A/D activation enable

The A/D activation enable is explained.

The A/D activation factor is selected by A/D activation factor select bit (ADTECS.STS2, ADTCS.STS1, STS0). Either the software, the external trigger, the reload timer, the compare match or PPG is selected. When the selected activation factor is generated, the A/D activation request signal is generated for the A/D activation arbitration.
For the activation channel that does not activate A/D, it is possible to disable the A/D activation request by selecting software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B") and disabling the software activation of a corresponding channel of A/D software activation channel select register (ADTSE).

5.2.3. Free-run timer input

The free-run timer input is explained.

The free-run timer input used for the compare match is input independently in each activation channel.
When the same free-run timer is used with two or more activation channels, the timer is set with the free-run timer selector.

5.2.4. Analog channel select

The analog channel select is explained.

An analog channel that does the A/D conversion can be selected by the CHSEL bit of A/D activation trigger extend control register (ADTECS).

5.2.5. Software activation

The software activation is explained.

The A/D activation trigger control factor select bit is set to software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B").
The channel that wants to activate software is set to the activation enable according to the A/D software activation channel select register (ADTSE). Two or more channels set to the ADTSE register can generate the activation request at the same time.

And, the software activation request signal is set by writing "1" in the START bit of the A/D software activation register (ADTSS).

5.2.6. External trigger activation

The external trigger activation is explained.

The A/D activation trigger control factor select bit is set to external trigger activation (ADTECS.STS2="0", ADTCS.STS1, STS0="01_B").
The external trigger is input every 12-bit A/D converter unit one by one. The external trigger 0 corresponds to activation channel 0 to 31, and the external trigger 1 corresponds to activation channel 32 to 47. Please refer to "Table 5-4 Channel Allocation of Each Unit" for the correspondence of the activation channel of each A/D converter unit in this series.
When the falling edge of the external trigger is detected, the activation request signal of external trigger/reload timer/PPG is set.

5.2.7. Reload timer activation

The reload timer activation is explained.

The A/D activation trigger control factor select bit is set to reload timer activation (ADTECS.STS2="0", ADTCS.STS1, STS0="10_B").
The reload timer is input every 12-bit A/D converter unit one by one. The reload timer 0 corresponds to activation channel 0 to 31, and the reload timer 1 corresponds to activation channel 32 to 47. Please refer to "Table 5-4 Channel Allocation of Each Unit" for the correspondence of the activation channel of each A/D converter unit in this series.
When the rising edge of the reload timer is detected, the activation request signal of external trigger/reload timer/PPG is set.

5.2.8. Compare match activation

The compare match activation is explained.

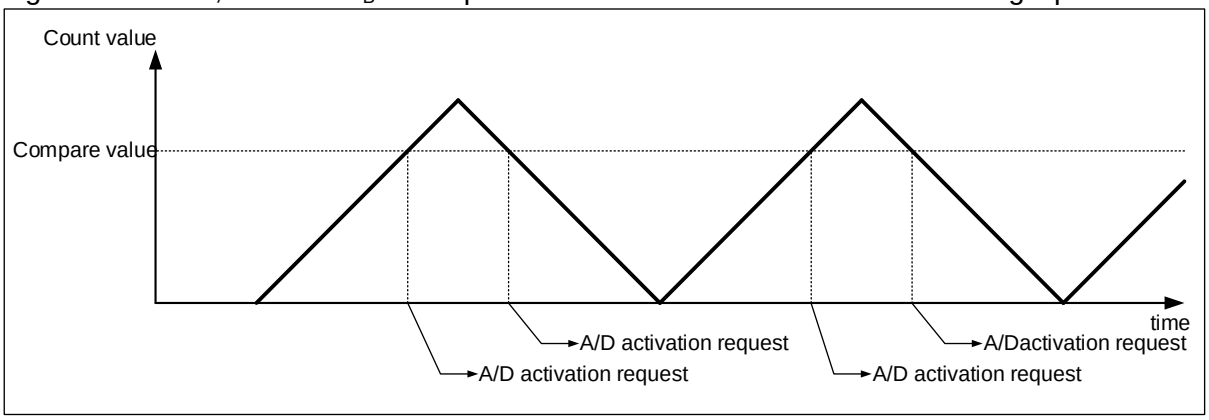
The A/D activation trigger control factor select bit is set to compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").
The compare register is in each activation channel. When 16-bit free-run timer matches the compare register value, the compare match activation request is set.
Set the timer value to be activated to the compare register. To perform the A/D activation request with the same timing as that of the free-run timer at detection of "0", set "0000_H". Moreover, to perform A/D activation at the same timing as that matching the compare clear of the free-run timer, set the same value as that of the compare clear of the free-run timer.

■ Operation mode of compare match activation

The operation mode of the A/D compare activation mode can be set by the SEL1, SEL0 bits of A/D activation trigger control status register (ADTCS).
SEL1, SEL0 can be set so that, when performing a comparison between the compare register value and the 16-bit free-run timer, you can set count up/count down times, count up time only, or count down time only.
Moreover, even if the free-run timer is corresponding to compare register value, the activation request signal is not generated for the A/D activation arbitration if the SEL1, SEL0 bits is set to "11_B".

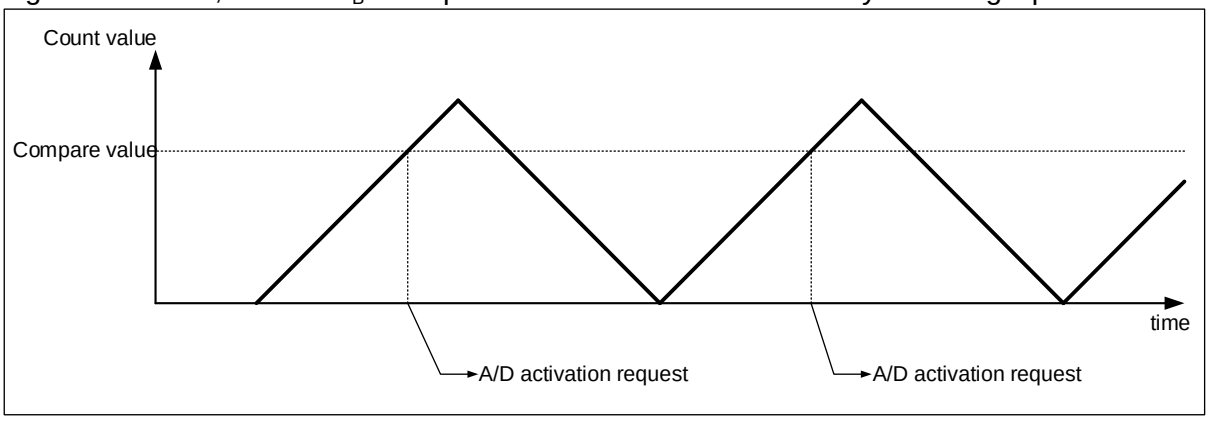
- **SEL1, SEL0=00_B: Compare match activation**

Figure 5-1 SEL1, SEL0="00_B": Compare Match Activation When Both Counting Up and Down



- **SEL1, SEL0=01_B: Compare match activation when only counting up**

Figure 5-2 SEL1, SEL0=01_B: Compare Match Activation When Only Counting Up



- **SEL1, SEL0=10_B: Compare match activation when only counting down**

Figure 5-3 SEL1, SEL0=10_B: Compare Match Activation When Only Counting Down

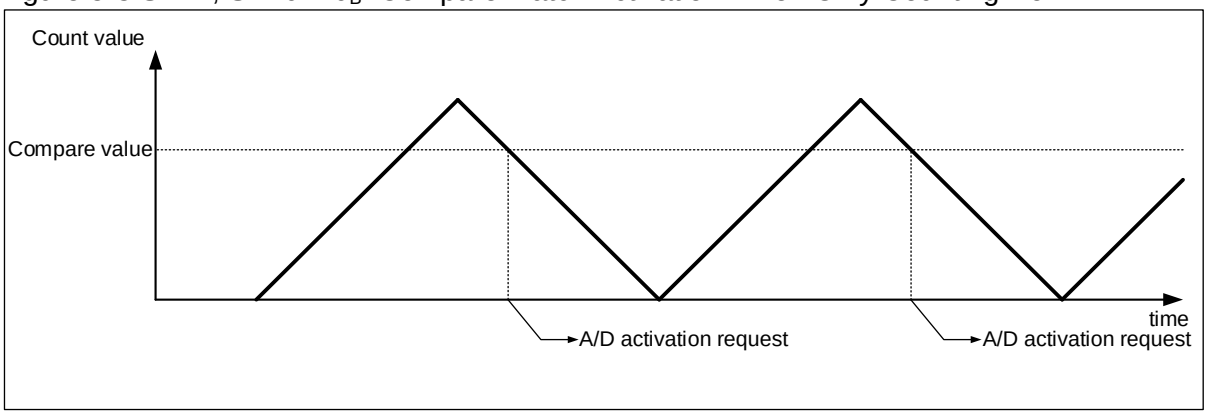
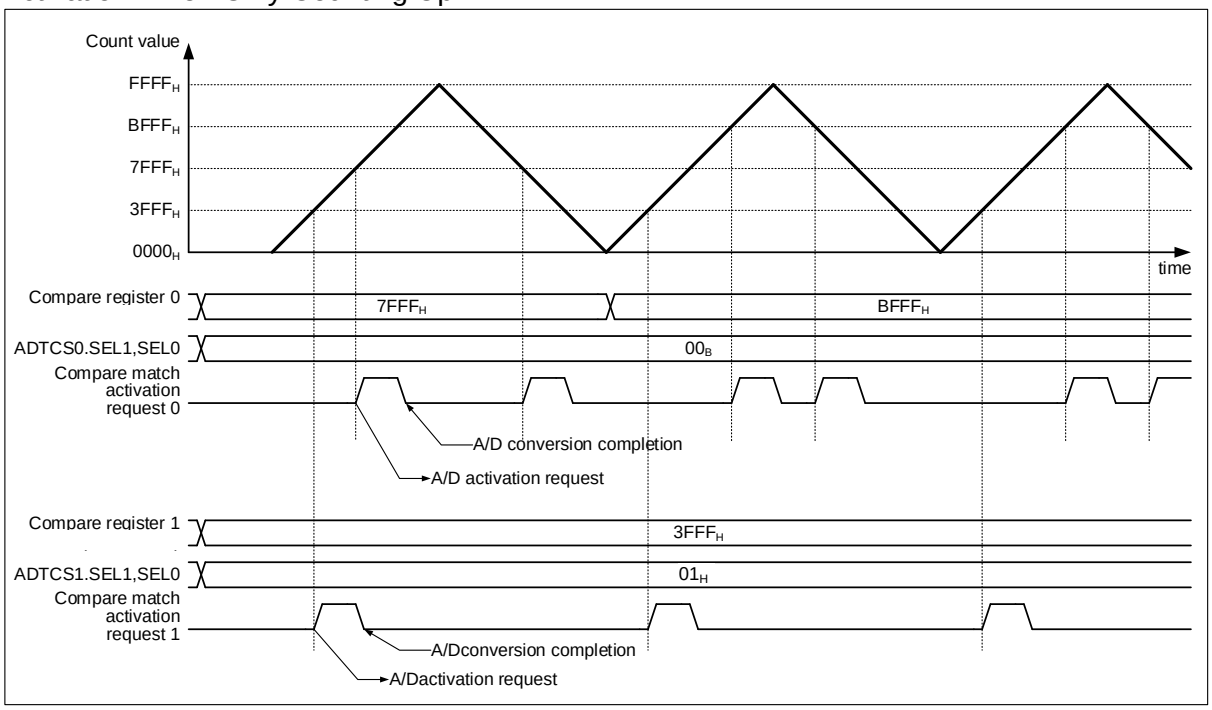


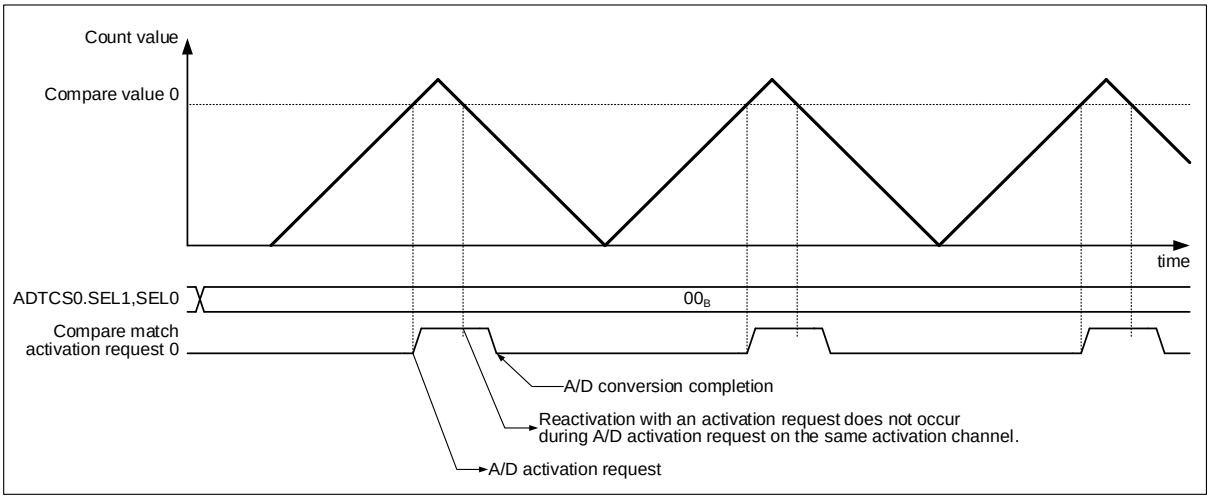
Figure 5-4 Activation Channel 0: When Both Counting Up and Down, Activation Channel 1: A/D Activation When Only Counting Up



■ About the setting of the compare value of the compare match activation

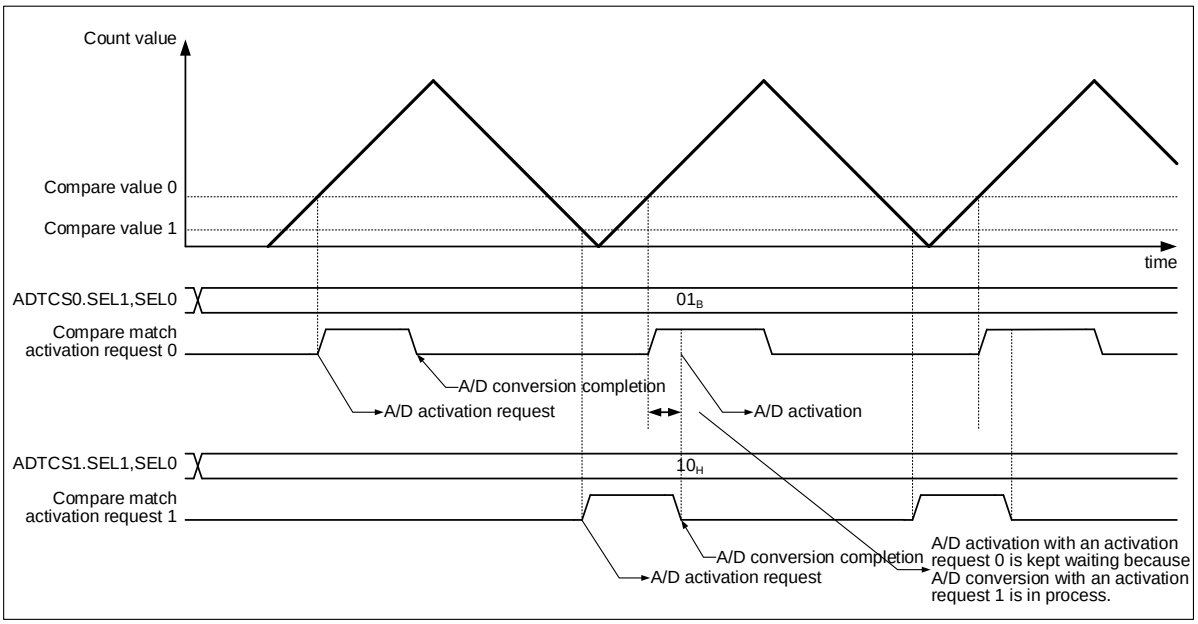
When the generation interval of the compare match is shorter than the conversion time of A/D in the same activation channel that does the compare match activation, the compare match generated while the A/D conversion is operating is disregarded.

Figure 5-5 When the Generation I the Compare Match is Shorter Than the Conversion Time of A/D in the Same Activation Channel



Moreover, when the generation interval of the compare match is shorter than the conversion time of A/D in between the activation channels that activate the same A/D converter, the A/D conversion start by the activation request generated while the A/D conversion is operating is kept waiting. In this case, The A/D conversion does not start according to the intended timing. The start of the A/D conversion is delayed more than the intended timing.

Figure 5-6 When the Generation Interval of the Compare Match is the Shorter Than the Conversion Time of A/D in Between the Activation Channels



■ Compare register buffer function of compare match activation

If "0" is written in the BUFX bit of A/D activation trigger control status register (ADTCS), the buffer function of compare register becomes effective.

The buffer timing can be selected by the BTS bit of the ADTCS register. The value written in the compare buffer register is transferred to the compare register when compare is cleared at BTS="1" or 0 detection at BTS="0".

Table 5-5 Transferring Condition from Compare Buffer Register to Compare Register

BUFX	BTS	Transferring condition from compare buffer register (ADCOMPB) to compare register (ADCOMP)
0	0	When 0 detection of the 16-bit free-run timer or The 16-bit free-run timer is stopping
0	1	When the compare of the 16-bit free-run timer is cleared or The 16-bit free-run timer is stopping
1	X	Buffer function unused (Transferring immediately)

Figure 5-7 Compare register 0: Buffer Function is Valid, Compare Register 1: Buffer Function is Invalid

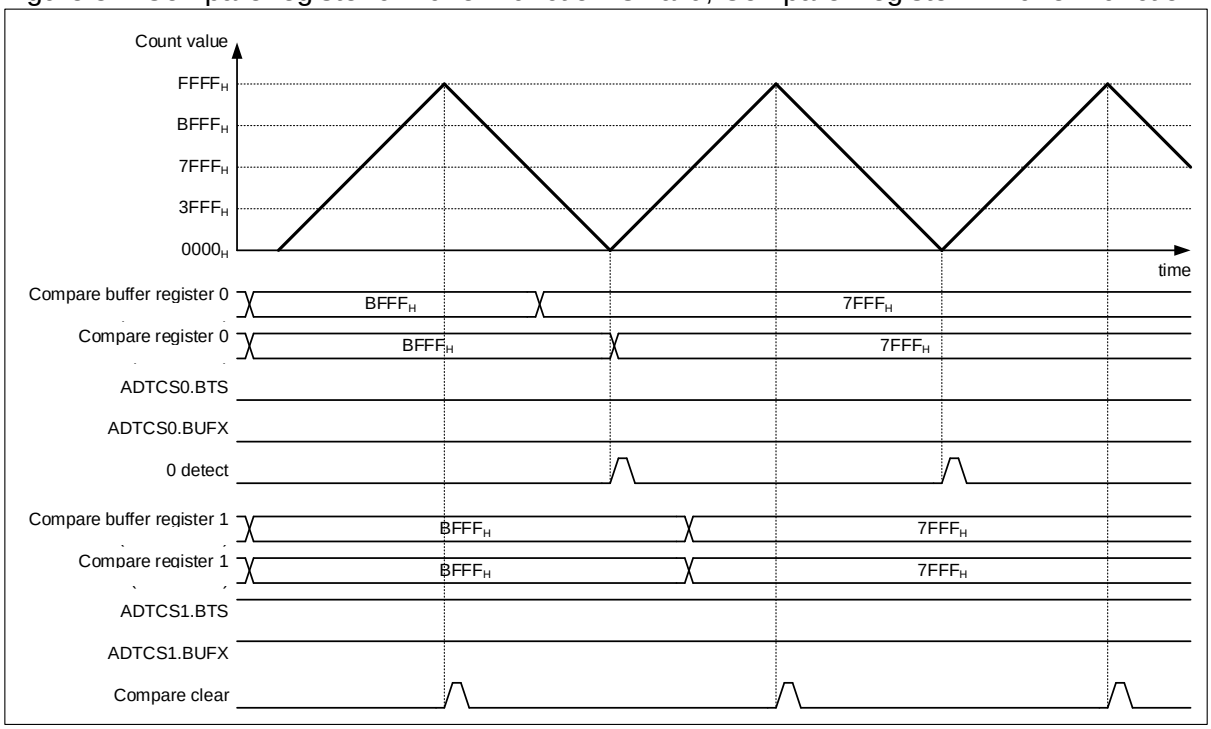


Figure 5-8 When 16-bit Free-run Timer Counting Up, the Compare Register Data Transfer Timing at Compare Match

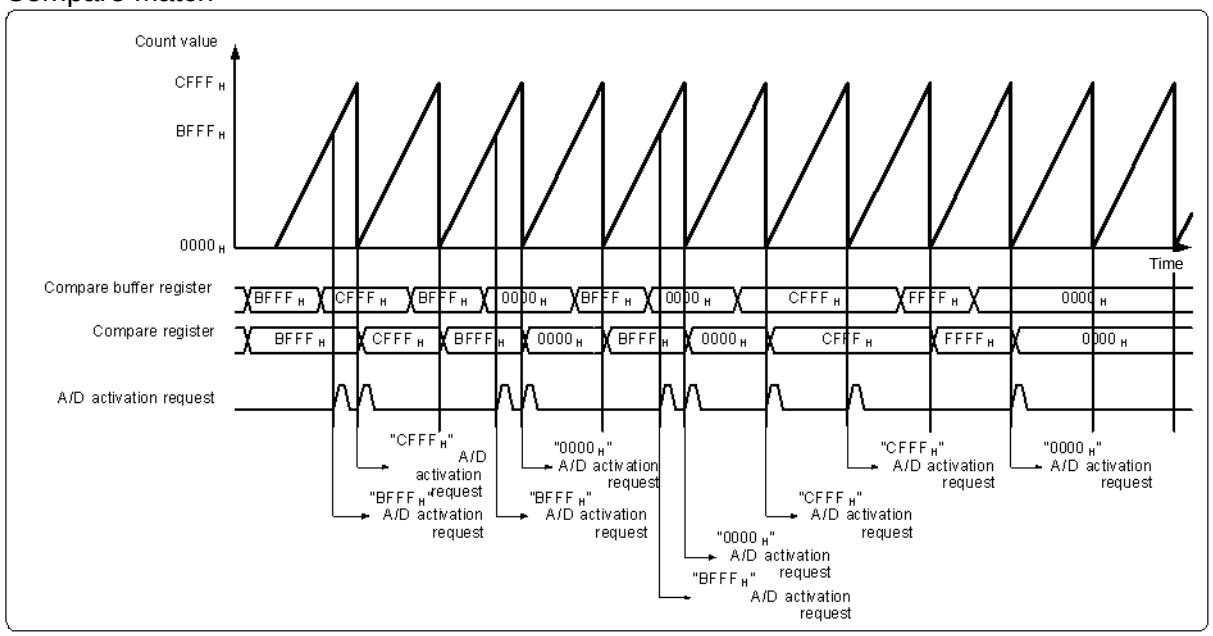


Figure 5-9 When 16-bit Free-run Timer Counting Up, the Compare Register Data Transfer Timing at 0 Detection

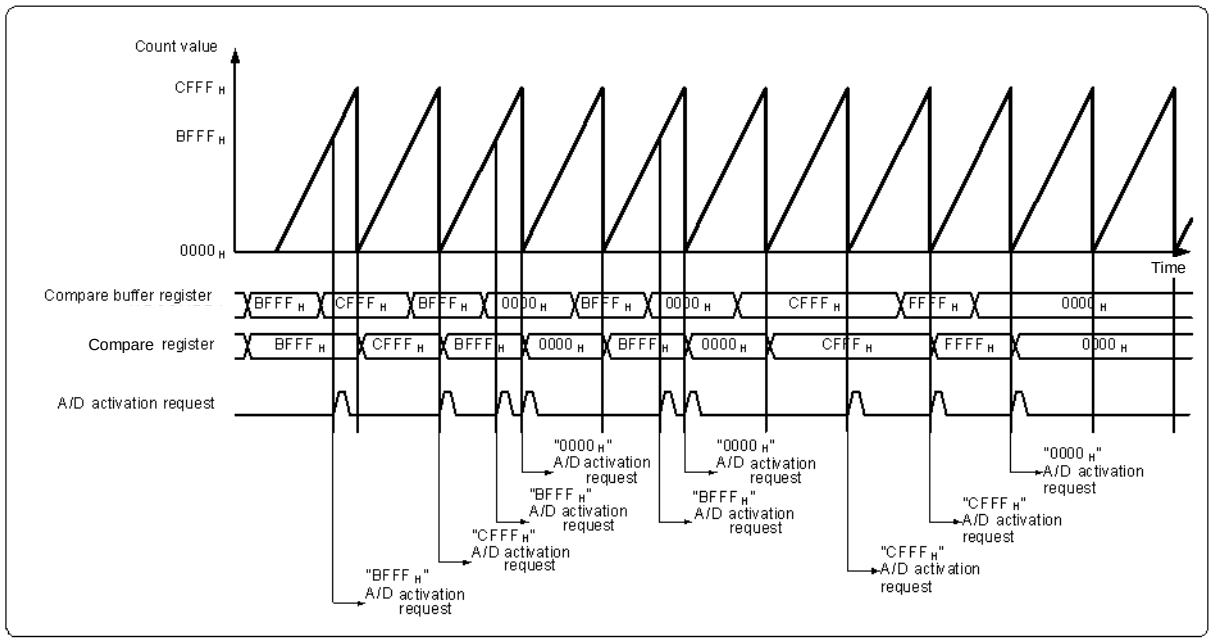


Figure 5-10 When 16-bit Free-run Timer Counting Up and Down, the Compare Register Data Transfer Timing at Compare Match

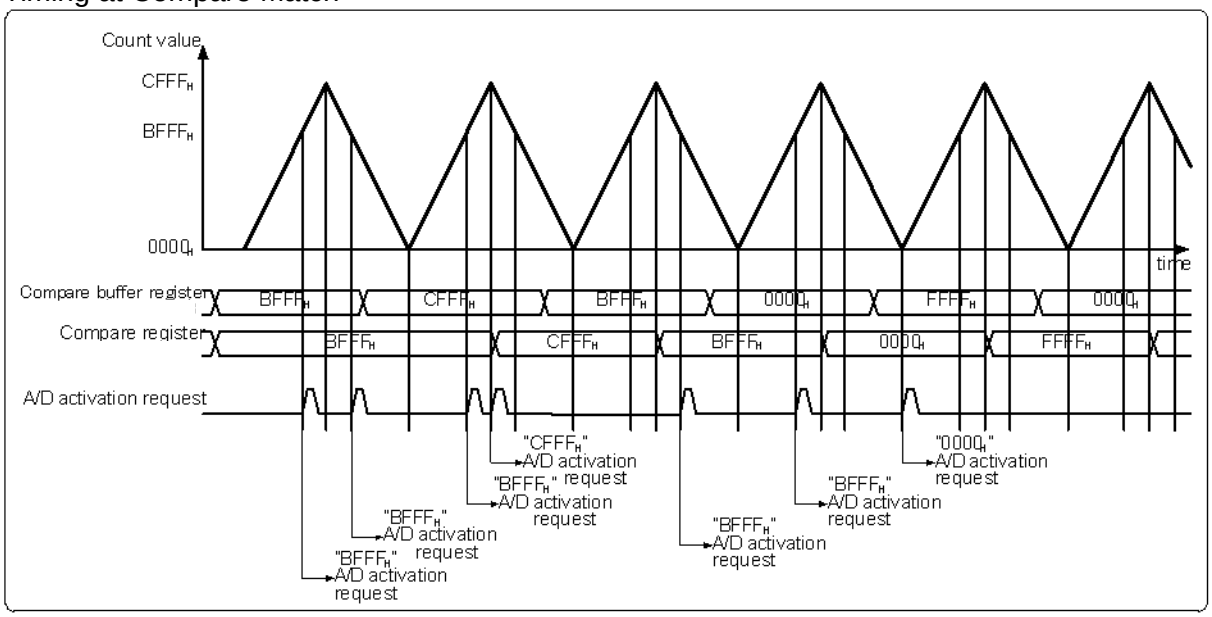
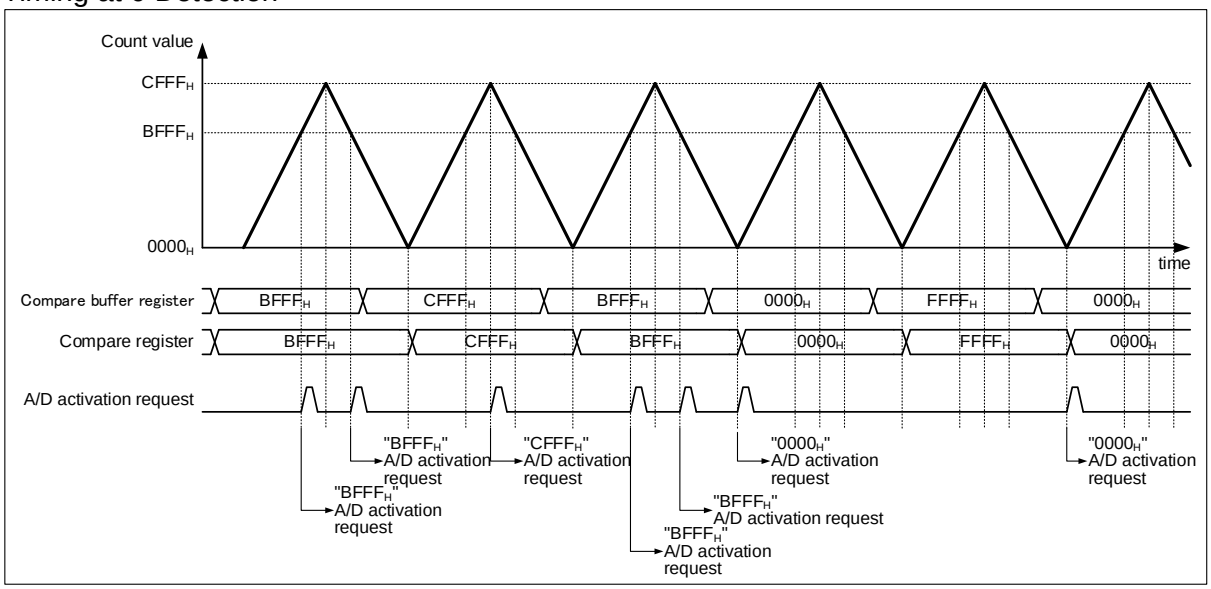


Figure 5-11 When 16-bit Free-run Timer Counting Up and Down, the Compare Register Data Transfer Timing at 0 Detection



5.2.9. PPG activation

The PPG activation is explained.

The A/D activation trigger control factor select bit is set to PPG activation (ADTECS.STS2="1", ADTCS.STS1, STS0="00_B").

The PPG activation is input independently in each activation channel.
When rising of the PPG is detected, the activation request signal of external trigger/reload timer/PPG is set.

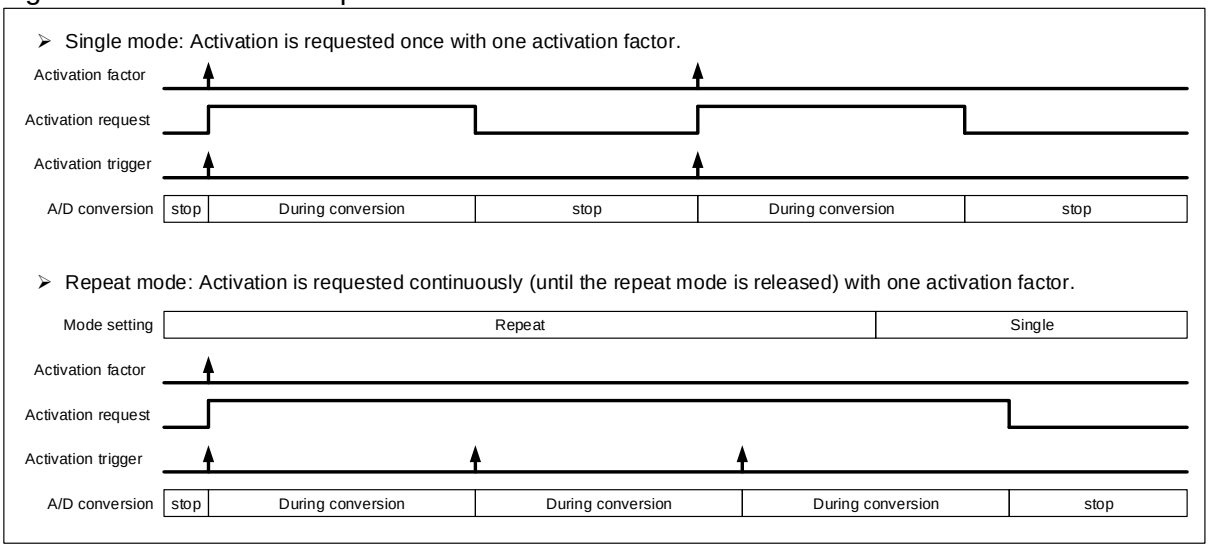
5.2.10. Activation request mode

The activation request mode is explained.

The activation request mode of each activation channel can be set. There are two activation request modes (the single mode and the repeat mode). The mode is set by the RPT bit of A/D activation trigger control status register (ADTCS).

- In single mode, activation is requested once with one activation factor. The A/D conversion is done once, and the activation request is released by the A/D conversion completion.
- In repeat mode, activation is requested continuously with one activation factor. The A/D conversion is repeatedly executed, and the activation request continues until the repeat mode is released.

Figure 5-12 Activation Request Mode



5.2.11. A/D conversion data

The A/D conversion data is explained.

The A/D conversion result data is stored in A/D data bit (ADTCD.D11 to D0) at each activation channel.
Moreover, when the data protection function is disabled (ADTCS.PRT="0"), the state of the A/D conversion data stored in A/D data bit (ADTCD.D11 to D0) can be confirmed by the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST).
When the data protection function is enabled (ADTCS.PRT="1"), the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST) are fixed to "0".

Table 5-6 State Confirmation of Conversion Data (When the Data Protection Function is Disabled)

ADTCD:ERR	ADTCD:ERRST	State of A/D conversion data
0	0	The latest data (read out not yet)
0	1	- (It does not have the meaning)
1	0	The old data (already read out) (Note) initial value
1	1	The latest data/ The over-writing is generated (read out not yet) (Note) There is lost data

5.2.12. Protection function

The protection function is explained.

Each A/D data register can set the data protection function. The protection function is set by the PRT bit of A/D activation trigger control status register (ADTCS). The protection function works at factors other than compare match activation.

When the protection function is effective, the data becomes the protection state if the conversion result is stored in the A/D data register. The condition to release the data protection state can be selected by the PRTS bit of the ADTCS register.

- When the PRTS bit is "0", the activation request is masked until the data of the A/D data register is read and the interrupt flag is cleared. The data reading and the interrupt flag clear are in random order.
- When the PRTS bit is "1", the activation request is masked until the data of the A/D data register is read.
- When the state of data protection, even if the next activation factor is generated, the activation request signal is masked (not active). Therefore, the data (read out not yet) of the A/D data register is not over-written by the next A/D conversion data.

5.2.13. Scan conversion mode

The scan conversion mode is explained.

The scan conversion is operation that does the A/D conversion from the activation channel number with a small activation channel one by one. The scan conversion can set 1 type of each 12-bit A/D converter unit.
The scan conversion can be operated as follows.

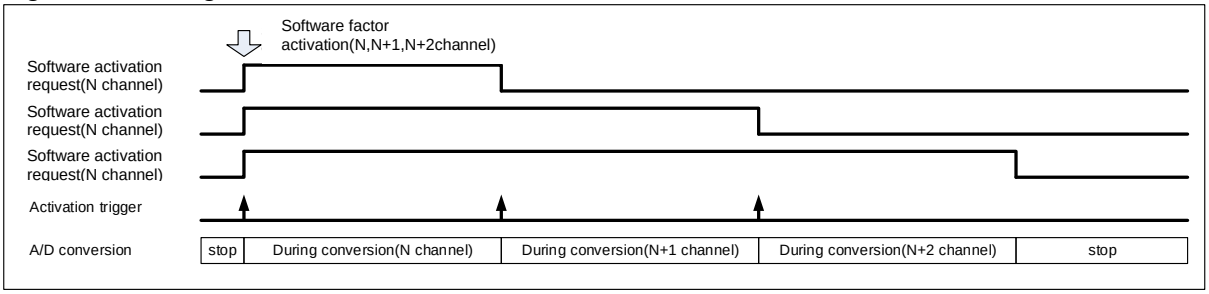
- Single scan conversion
- Continuous scan conversion
- Continuous scan conversion when conversion count of each channel is specified
- Stop scan conversion when conversion count of each channel is specified

■ Single scan conversion

Single scan conversion is executed by setting the repeat conversion selection bit to single mode (ADTCS.RPT="0") for each activation channel of the scan conversion target, and executing A/D activation with the same activation factor

at the same time for these channels.
The scan conversion starts A/D conversion from a small activation channel number (activation channel from which A/D is activated) one by one by the A/D activation arbitration of latter part. When the A/D conversion of the final activation channel is completed, the scan conversion is stopped.

Figure 5-13 Single Scan Conversion



■ Continuous scan conversion

Continuous scan conversion is executed by setting the repeat conversion select bit to repeat mode (ADTCS.RPT="1") and setting the data protection function to enable (ADTCS.PRT="1") for each activation channel of the scan conversion target, and executing the A/D activation with the same activation factor at the same time for these channels.
When two or more repeat modes (ADTCS.RPT="1") are set between the activation channels corresponding to the same A/D converter, only a certain channel (channel with the highest priority level) will be processed in the A/D activation arbitration of latter part.
In this case, please make the protection function effective about those activation channels.

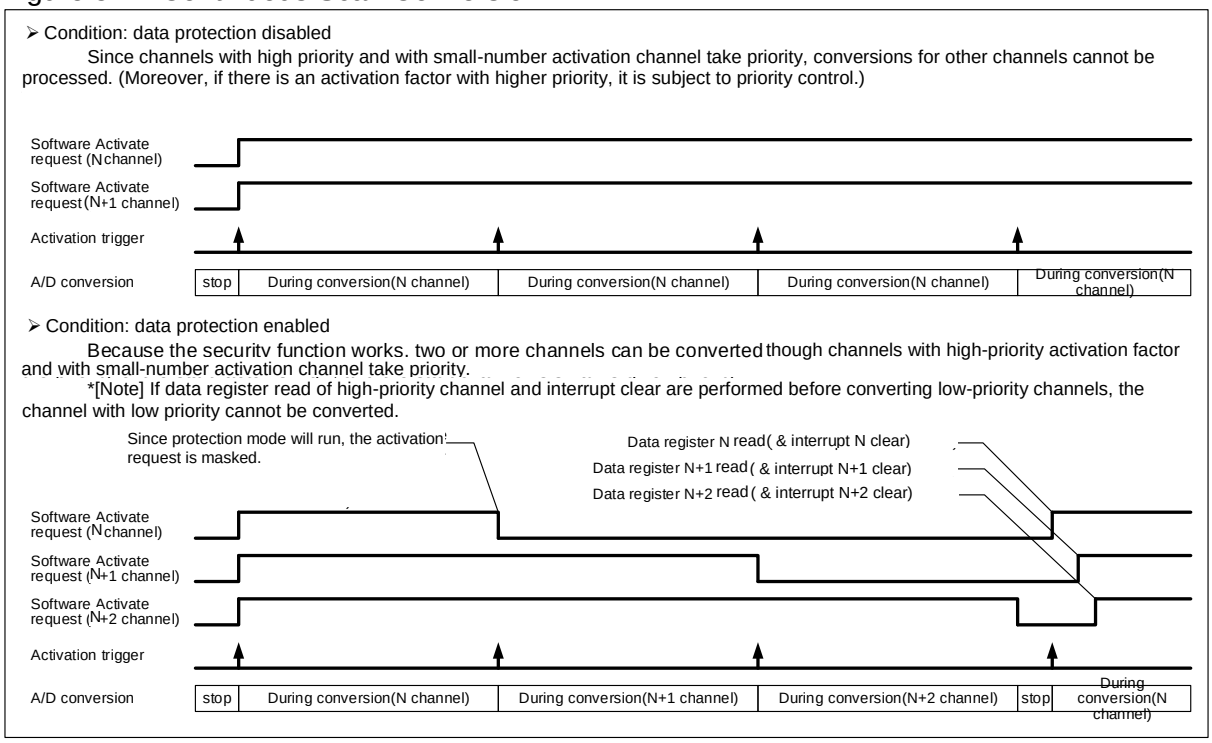
- In case of data protection function invalid (ADTCS.PRT="0")
  In A/D activation arbitration, channels with high-priority activation factors and with small activation channel numbers take priority. When the repeat mode is set (ADTCS.RPT="1"), the activation request is continuously output. Therefore, the conversion of the activation channel with low priority is not executed.

- In case of data protection function effective (ADTCS.PRT="1")
  In A/D activation arbitration, channels with high-priority activation factors and with small activation channel numbers take priority. When the repeat mode is set (ADTCS.RPT="1"), the activation request is continuously output. However, it enters the data protection state if the A/D conversion result is stored in A/D data register (ADTCD), and the activation request is masked. As a result, the conversion of the next activation channel can be executed.

Note:

When the data protection function is effective (ADTCS.PRT="1") and two or more repeat modes are set (ADTCS.RPT="1"), the conversion of the activation channel with low priority is not executed if the data protection state of the activation channel with high priority is released before converting the activation channel with low priority.

Figure 5-14 Continuous Scan Conversion



■ Continuous scan conversion when conversion count of each channel is specified

Continuous scan conversion (when the conversion count is specified for each continuous scan conversion channel when the conversion count is specified for each channel) is executed as follows: set continuous scan conversion mode (ADSCANS.SCMD="0"); for each activation channel of the scan conversion target, set repeat conversion selection bit to repeat mode (ADTCS.RPT="1"), set conversion count specification scan conversion execution enable (ADNCS.CNTEN="1"), and set conversion count specification (ADNCS.CCNT1, CCNT0="count"); then, perform A/D activation with the same activation factor and at the same time for each channel.

The continuous scan conversion when conversion count of each channel is specified advances to the A/D conversion of the next activation channel when the A/D conversion of conversion count specification (ADNCS.CCNT1, CCNT0) is executed from activation channel (activation channel of the scan conversion target) with small number, and conversion is completed. Moreover, the activation channel that completes the A/D conversion of the conversion count specification can be confirmed by conversion count completion flag bit (ADEOCF.EOCF).
Then, when the A/D conversion of the conversion count specification of the final activation channel is completed, the scan conversion completion interrupt factor flag (ADSCANS.SCINT) is set to "1" and the scan conversion is automatically executed repeatedly from the top.
Moreover, when the scan conversion is executed repeatedly from the top, the conversion count completion flag bit (ADEOCF.EOCF) of each activation channel is automatically cleared to "0".
The conversion count specification (ADNCS.CCNT1, CCNT0) can select 1 to 4 times for each activation channel.
Disable and enable can be selected for the data protection function for each activation channel. Moreover, when the data protection function is effective (ADTCS.PRT="1"), the data protection state of each activation channel can be confirmed by data protection state flag bit (ADPRTF.PRTF).

- In case of data protection function invalid (ADTCS.PRT="0")
  The A/D conversion is continuously executed without stop. The A/D conversion data is not protected, and. the last A/D conversion data converted in the activation channel is stored.
- In case of data protection function effective (ADTCS.PRT="1")
  The A/D conversion data is protected. Therefore, the A/D conversion in the same activation channel stops the A/D conversion for the data protection period. However, when the scan conversion shifts to a different activation channel, the A/D conversion is continuously executed without stopping because A/D data register (ADTCD) is also different.

Notes:

- In continuous scan conversion, the conversion count completion flag (ADEOCF.EOCF) of the last activation channel of the scan conversion target is set to "1", but is soon automatically cleared to "0".
- When an activation factor that is the same as the scan conversion target is set to an activation channel with a number larger than that of the final channel of continuous scan conversion, the activation channel with the number larger than that of the final channel is evaluated once by the A/D activation arbitration of the latter part after completion of the final channel of the continuous scan conversion.
- The data protection function works at factors other than compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").
- In the data protection function is effective (ADTCS.PRT="1"), please release the data protection state before executing the next scan conversion repeatedly. The activation channel in the data protection state does not execute the A/D conversion.

Figure 5-15 Continuous Scan Conversion When Conversion Count of Each Channel is Specified (In Case of Data Protection Function Invalid)

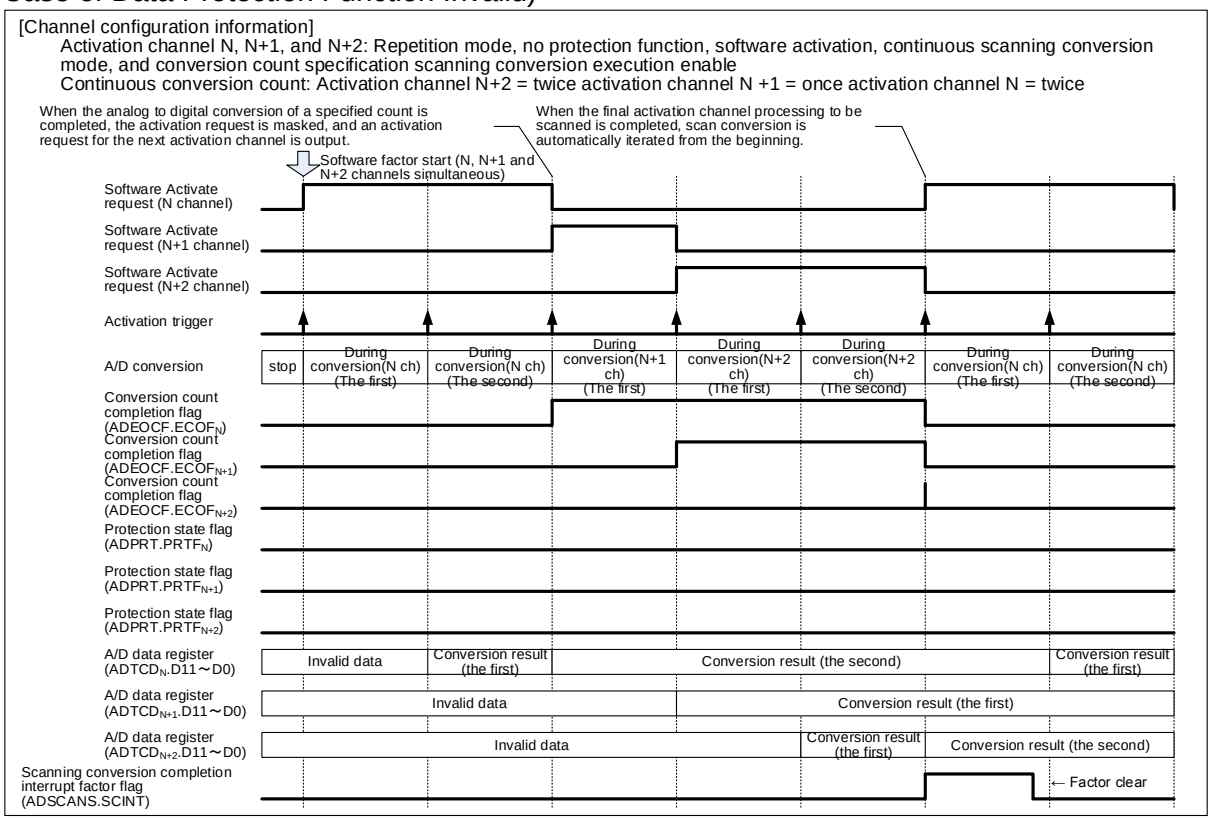
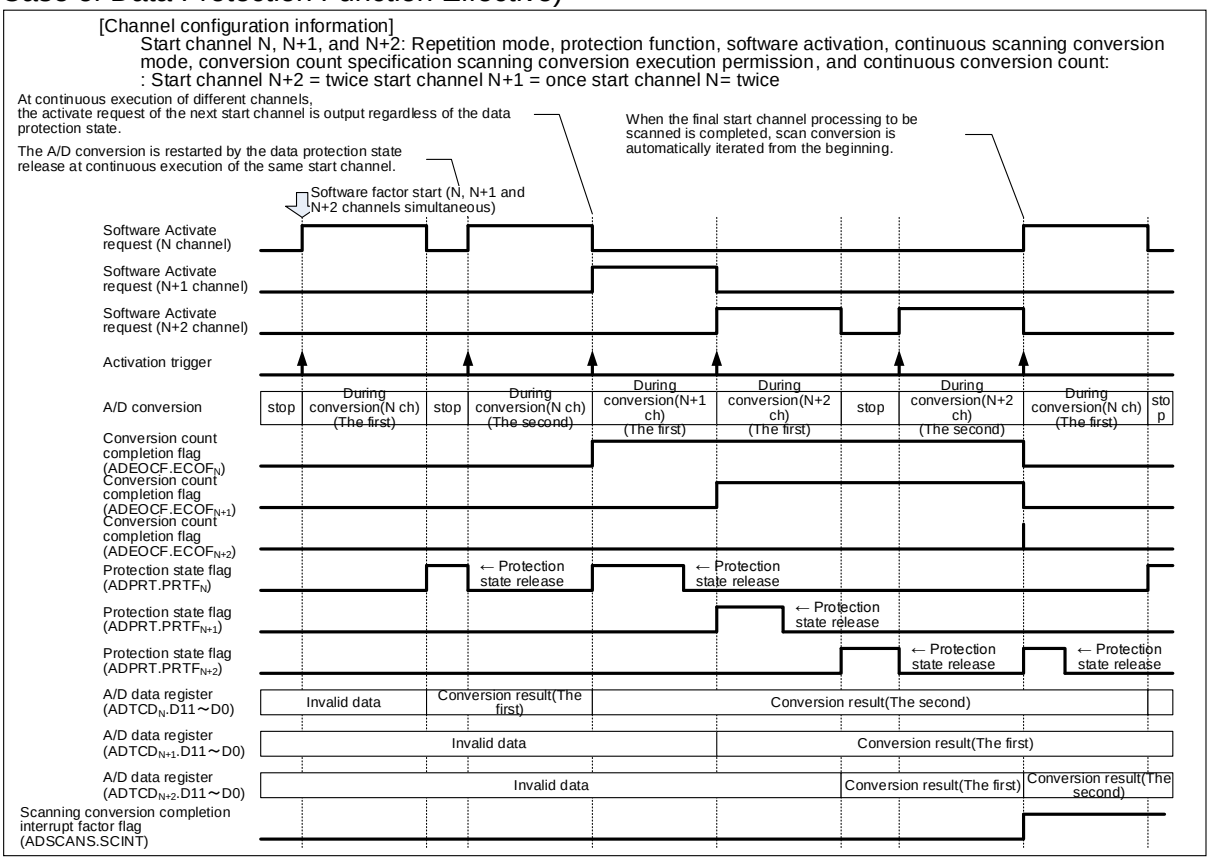


Figure 5-16 Continuous Scan Conversion When Conversion Count of Each Channel is Specified (In Case of Data Protection Function Effective)



■ Stop scan conversion when conversion count of each channel is specified

Stop scan conversion (when the conversion count is specified for each channel) is executed as follows: set stop scan conversion mode (ADSCANS.SCMD="1"); for each activation channel of the scan conversion target, set repeat conversion selection bit to repeat mode (ADTCS.RPT="1"), set conversion count specification scan conversion execution enable (ADNCS.CNTEN="1"), and set conversion count specification (ADNCS.CCNT1, CCNT0="count"); then, perform A/D activation with the same activation factor and at the same time for each channel.
The stop scan conversion when conversion count of each channel is specified advances to the A/D conversion of the next activation channel when the A/D conversion of conversion count specification (ADNCS.CCNT1, CCNT0) is executed from activation channel (activation channel of the scan conversion target) small number, and conversion is completed. Moreover, the activation channel that completes the A/D conversion of the conversion count specification can be confirmed by conversion count completion flag bit (ADEOCF.EOCF).
Then, when the A/D conversion of the conversion count specification of the final activation channel is completed, the scan conversion completion interrupt factor flag (ADSCANS.SCINT) is set in "1" and the scan conversion becomes the state of the stop. The scan conversion restarts (return from the state of the stop) when the A/D activation factor is generated even by one of the scan conversion targets.
Moreover, the conversion count completion flag bit (ADEOCF.EOCF) of each activation channel is automatically cleared to "0" by restarting (return from the state of the stop) the scan conversion.
The conversion count specification (ADNCS.CCNT1, CCNT0) can select 1 to 4 times of each activation channel.
The data protection function can select invalidity and effective of each activation channel. Moreover, when the data protection function is effective (ADTCS.PRT="1"), the data protection state of each activation channel can be confirmed by data protection state flag bit (ADPRTF.PRTF).

- In case of data protection function invalid (ADTCS.PRT="0")
  The scan conversion continuously executes A/D conversions other than the state of the stop without stopping. The data of the A/D conversion data is not protected, and. the last A/D conversion data converted in the activation channel is stored.
- In case of data protection function effective (ADTCS.PRT="1")
  The data of the A/D conversion data is protected. Therefore, the A/D conversion in the same activation channel stops the A/D conversion for the data protection period. However, when the scan conversion shifts to a different activation channel, the A/D conversion is continuously executed without stopping because A/D data register (ADTCD) is also different.

Notes:

- When an activation factor that is the same as the scan conversion target is set to an activation channel with a number larger than that of the final channel of stop scan conversion, A/D conversion can be executed only during the stop period, for the activation channel with the number larger than the final channel.
- The data protection function works at factors other than compare match activation (ADTECS.STS2="0", ADTCS.STS1, STS0="11_B").
- In the data protection function is effective (ADTCS.PRT="1"), please release the data protection state before starting the next scan conversion. The activation channel in the data protection state does not execute the A/D conversion.

Figure 5-17 Stop Scan Conversion When Conversion Count of Each Channel is Specified (In Case of Data Protection Function Invalid)

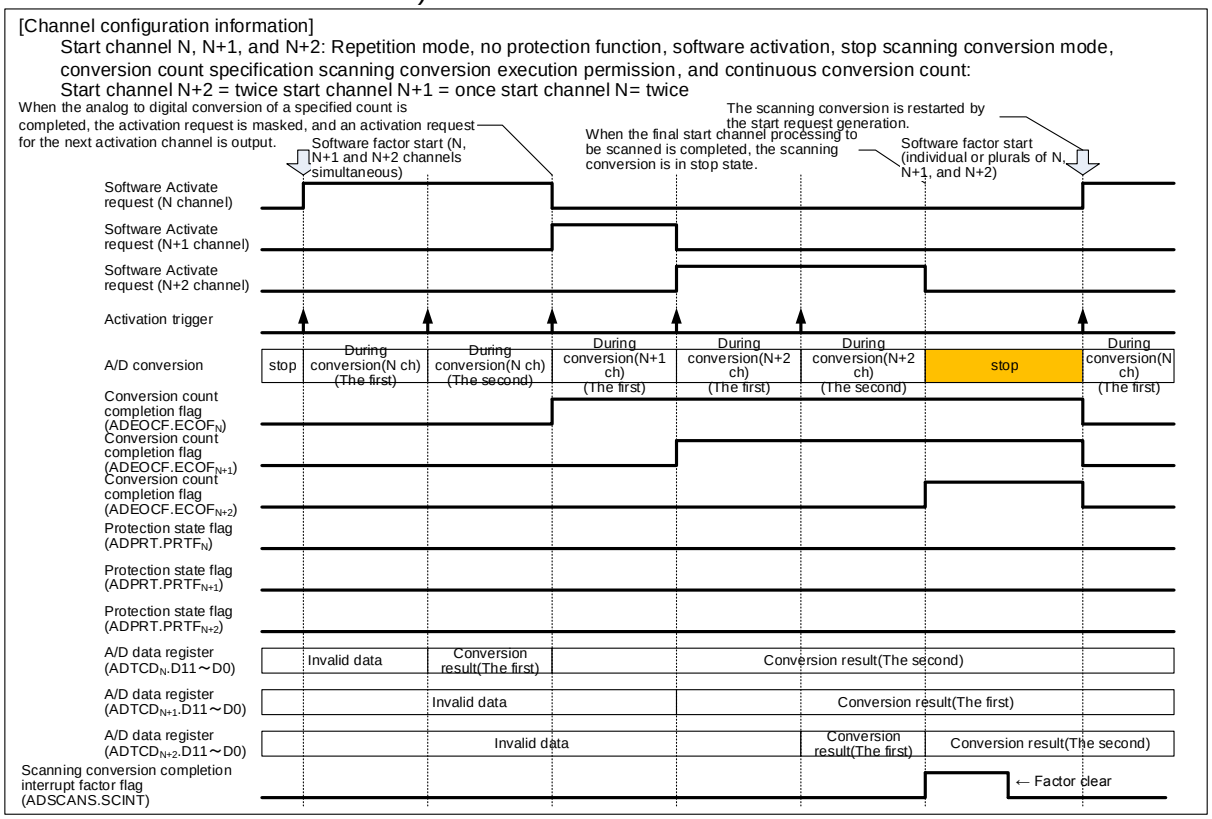
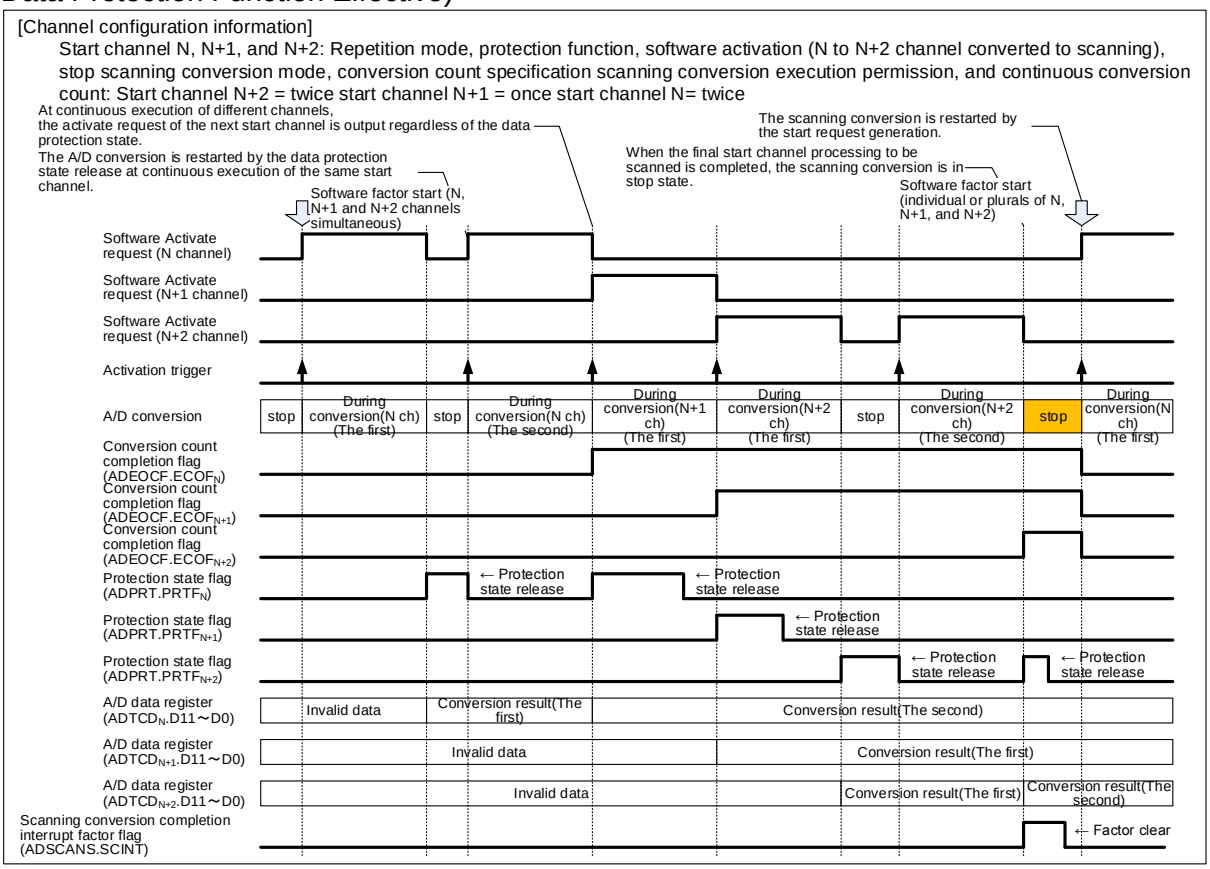


Figure 5-18 Stop Scan Conversion When Conversion Count of Each Channel is Specified (In Case of Data Protection Function Effective)



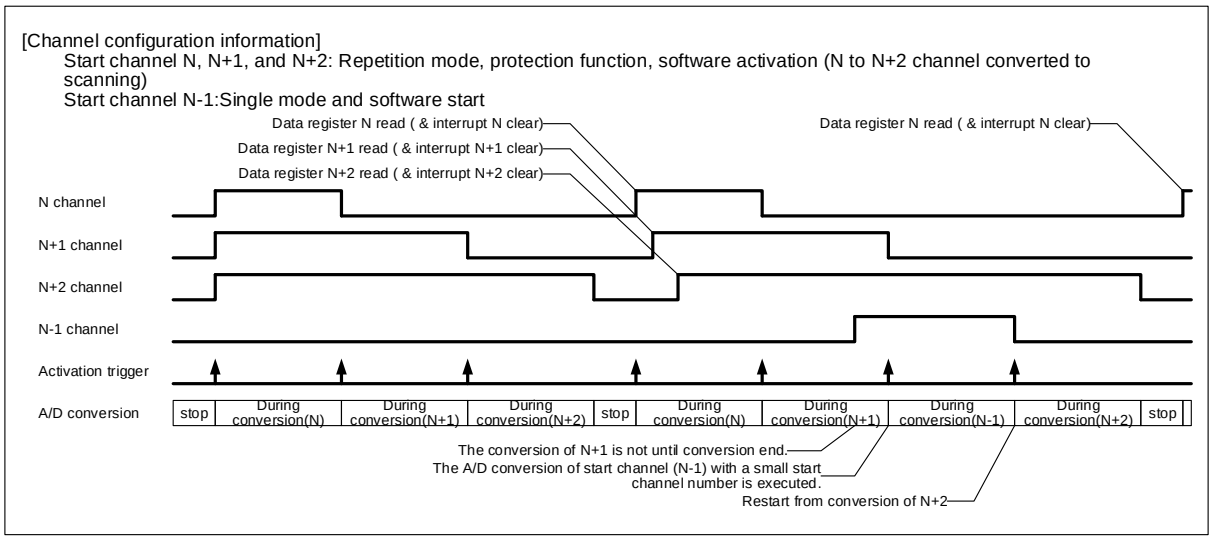
5.2.14. High priority activation request operation of other activation channel during the scan conversion

The high priority activation request operation of other activation channel during the scan conversion is explained.

When the scan conversion is executed between the activation channels corresponding to the same A/D converter unit, the priority of the activation channels are controlled based on the activation factor and activation channel information. The priority control is done by the A/D activation arbitration of latter part.

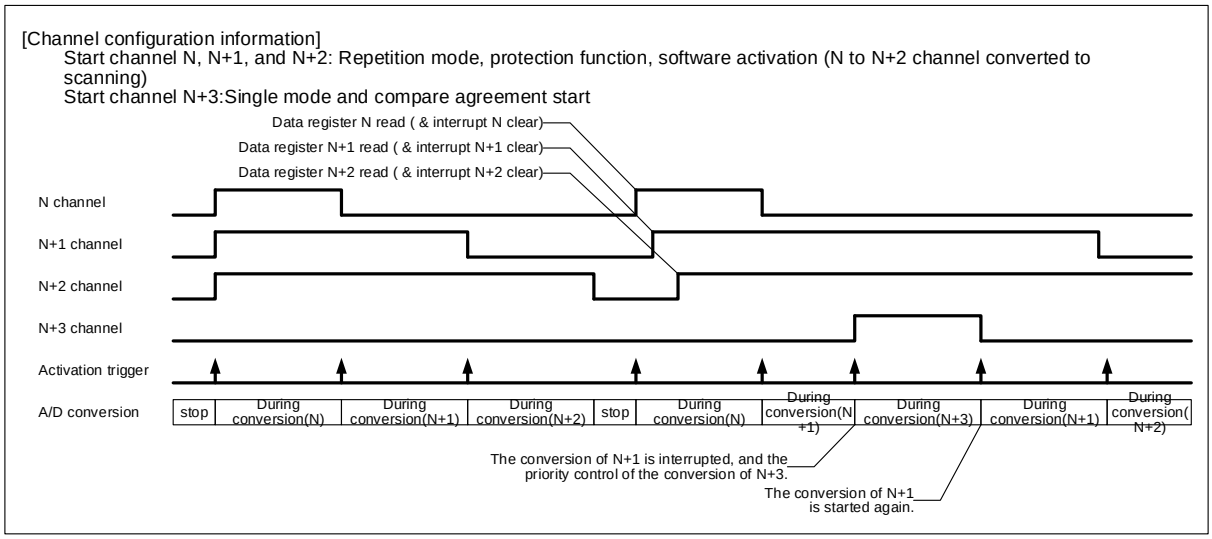
When an activation channel with the same activation factor is generated from an activation channel with an activation channel number that is smaller than that of an activation channel performing scan conversion, the scan conversion stops temporarily after completion of A/D conversion. Then, the scan conversion that stops temporarily is restarted after completing the A/D conversion of the activation channel from a small activation channel number.

Figure 5-19 Operation of the Same Activation Factor Request from Small Activation Channel Number at Continuous Scan Conversion



When the activation factor whose priority is higher than the activation channel that does the scan conversion is generated from other activation channels, the scan conversion is interrupted. Then, the A/D conversion is restarted from the activation channel of the interrupted scan conversion after the A/D conversion of the high priority is completed.

Figure 5-20 Operation of High Priority Activation Factor Request by Other Activation Channel at Continuous Scan Conversion



5.2.15. Forced termination of activation request

The forced termination of activation request is explained.

Whether an A/D activation request or a conversion operation is in progress can be reported using the A/D conversion request bit (ADTCS0-47.BUSY) of the A/D activation trigger control status register. The current A/D activation

request or conversion operation can be terminated forcibly by resetting this bit to "0".

5.2.16. Range comparison function

The range comparison function is explained.

■ Range comparison upper and lower bound threshold setting

The upper bound threshold setting register (ADRCUT) and the lower bound threshold setting register (ADRCLT) can set 4 types of each unit. The setting selects one from the combination of the upper and lower bound threshold setting register of 4 types by the upper and lower bound threshold select bit (ADRCCS.RCOTS1, RCOTS0) of each activation channel.

Table 5-7 Range Comparison Upper and Lower Bound Threshold Setting

Upper and lower bound threshold select bit (ADRCCS: RCOTS1, RCOTS0)	Select result	
	A/D convertor unit 0	A/D convertor unit 1
"00_B"	Upper bound threshold setting register 0 (ADRCUT0)/ Lower bound threshold setting register 0 (ADRCLT0)	Upper bound threshold setting register 4 (ADRCUT4)/ Lower bound threshold setting register 4 (ADRCLT4)
"01_B"	Upper bound threshold setting register 1 (ADRCUT1)/ Lower bound threshold setting register 1 (ADRCLT1)	Upper bound threshold setting register 5 (ADRCUT5)/ Lower bound threshold setting register 5 (ADRCLT5)
"10_B"	Upper bound threshold setting register 2 (ADRCUT2)/ Lower bound threshold setting register 2 (ADRCLT2)	Upper bound threshold setting register 6 (ADRCUT6)/ Lower bound threshold setting register 6 (ADRCLT6)
"11_B"	Upper bound threshold setting register 3 (ADRCUT3)/ Lower bound threshold setting register 3 (ADRCLT3)	Upper bound threshold setting register 7 (ADRCUT7)/ Lower bound threshold setting register 7 (ADRCLT7)

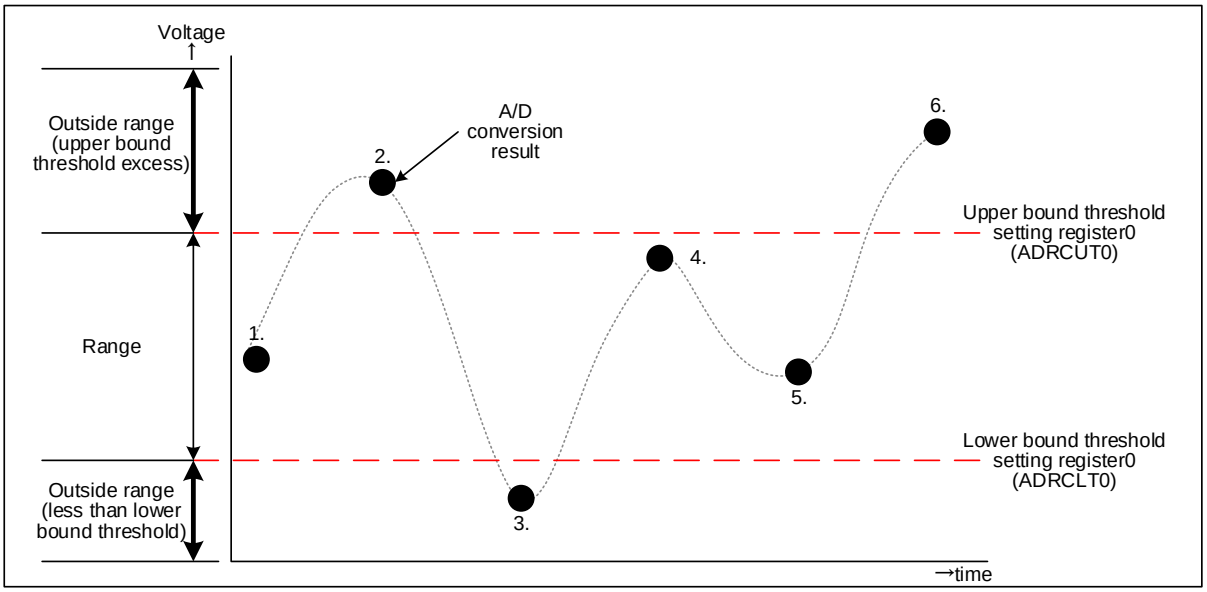
■ Range comparison operation

When the range comparison enable setting (ADRCCS.RCOE="1"), the range comparison is executed when the A/D conversion is completed and data is stored in A/D data bit (ADTCD.D11 to D0). The range comparison compares the upper and lower bound threshold setting register (ADRCUT/ADRCLT) selected by the range comparison upper and lower bound threshold selection bit (ADRCCS.RCOTS1, RCOTS0) with the A/D data bit (ADTCD.D11 to D0). The range comparison result is input to a continuous detecting function.

Table 5-8 Range Comparison Upper and Lower Bound Threshold Select

Range comparison result	Confirmation outside the range (ADRCCS: RCOIRS="0")	Confirmation inside the range (ADRCCS: RCOIRS="1")
Outside the range (larger than upper bound threshold) A/D data bit > Upper bound threshold setting register	Detected	Undetected
Inside the range A/D data bit ≤ Upper bound threshold setting register and A/D data bit ≥ Lower bound threshold setting register	Undetected	Detected
Outside the range (Smaller than lower bound threshold) A/D data bit < Lower bound threshold setting register	Detected	Undetected

Figure 5-21 Range Comparison Condition



■ Continuous detecting function of range comparison result

A continuous detecting function continuously detects the range comparison result, and removes the noise etc.
When the count in which the detection of the range comparison result is set by range comparison continuous detection count specification setting (ADRCCS.RCOCD2 to RCOCD0) is continuously detected, "1" is set to range comparison interrupt factor flag bit (ADRCIF.RCINT). When it becomes undetected even one time in the continuous detection by the range comparison result, the continuous detection measurement is cleared to 0 times and the measurement is restarted.

Table 5-9 Continuous Detecting Function Operation Condition

Continuous detection measurement operation	· Each activation channel is controlled. · It operates whenever range comparison execution enable setting (ADRCCS:RCOE="1").
Continuous detection count	· The count can select from 1 to 7 times by continuous detection count specification (ADRCCS:RCOCD2 to RCOCD0). · The state of the detection count can be confirmed by continuous detection count state indicate (ADRCCS:RCOCD2 to RCOCD0).
Clear condition	· When range comparison execution disable setting (ADRCCS:RCOE="0") · When undetected by range comparison result
Increment condition	· When detected by range comparison result However, it stops by the continuous detection count specification value when reaching continuous detection count specification (ADRCCS:RCOCD2 to RCOCD0).

Notes:

- When confirming outside the range, the continuous detection measurement does not clear to 0 times and continue the continuous detection even if the state of the range comparison result changes from larger than the upper bound threshold into smaller than lower bound threshold.
- When you want to initialize the continuous detection count of the range comparison result, please change to enable setting (ADRCCS.RCOE="0" → "1") after the setting of the range comparison execution disable, while the A/D is not being requested.
- After two clocks with the peripheral clock from flag set (ADTCS.INT="1") of the interrupt control bit by the A/D conversion completion interrupt, the range comparison interrupt factor flag (ADRCIF.RCINT) is set in "1" by the continuous detection.

■ Range comparison over flag control

When confirmation outside the range of the range comparison (ADRCCS.RCOIRS="0"), the indication of larger than upper bound threshold or less than lower bound threshold can be confirmed for each activation channel with the range comparison threshold over flag bit (ADRCOT.RCOOF[n]).

Table 5-10 Range Comparison Upper and Lower Bound Threshold Select

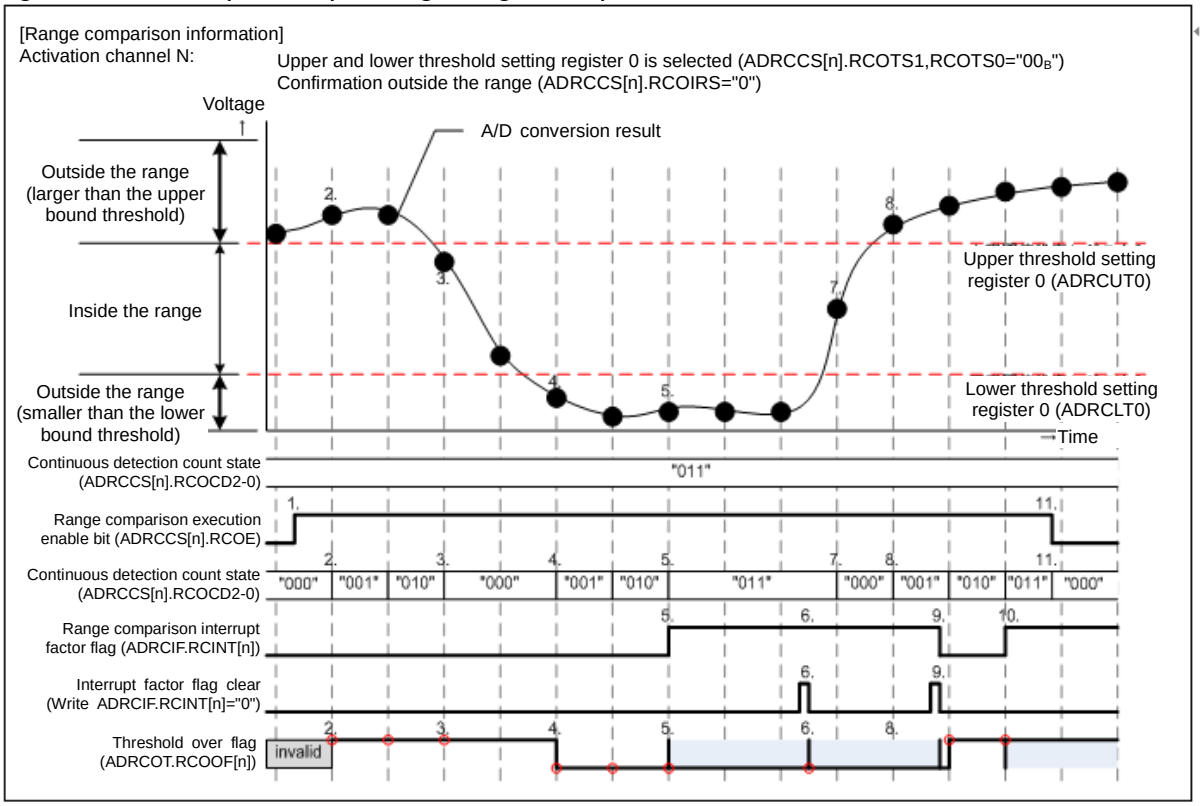
Range comparison result	Range comparison threshold over flag bit (ADRCOT:RCOOF)	
	Confirmation outside the range (ADRCCS: RCOIRS="0")	Confirmation inside the range (ADRCCS: RCOIRS="1")
Outside the range (larger than upper bound threshold) A/D data bit > Upper bound threshold setting register	"1"	The last value is maintained

Range comparison result	Range comparison threshold over flag bit (ADRCOT:RCOOF)	
	Confirmation outside the range (ADRCCS: RCOIRS="0")	Confirmation inside the range (ADRCCS: RCOIRS="1")
Inside the range A/D data bit ≤ Upper bound threshold setting register and A/D data bit ≥ Lower bound threshold setting register	The last value is maintained	The last value is maintained
Outside the range (Smaller than lower bound threshold) A/D data bit < Lower bound threshold setting register	"0"	The last value is maintained

Moreover, the range comparison threshold over flag bit (ADRCOT.RCOOF) maintains the content set in range comparison threshold over flag bit (ADRCOT.RCOOF) while range comparison interrupt factor flag (ADRCIF.RCINT) is being set to "1".

■ Example of operating range comparison

Figure 5-22 Example of Operating Range Comparison



1. When the range comparison execution enable setting (ADRCCS[n].RCOE="0") is effective, the continuous detection count state (ADRCCS[n].RCOCD2 to RCOCD0) is initialized to "000_B".
   The range comparison operation is started by the range comparison execution enable setting (ADRCCS[n].RCOE="1").

2. The continuous count detection state (ADRCCS[n].RCOCD2 to RCOCD0) is incremented when the range comparison result exceeds the upper bound threshold.
   Moreover, the threshold over flag reports that the upper bound threshold has been exceeded (ADRCOT.RCOOF[n]="1").
3. Since the range comparison result is detected within the range before the continuous detection count value (ADRCCS[n]. RCOCD2 to RCOCD0="011_B") is detected, the continuous detection count state is initialized (ADRCCS[n].RCOCD2 to RCOCD0="000_B").
   Moreover, the threshold over flag (ADRCOT.RCOOF[n]) maintains the last value.
4. The continuous count detection state (ADRCCS[n].RCOCD2 to RCOCD0) is incremented when the range comparison result is below the lower bound threshold.
   Moreover, the threshold over flag reports that the range comparison result is below the lower bound threshold (ADRCOT.RCOOF[n]="0").
5. Because the range comparison result continuously reached the continuous detection count specification value (ADRCCS[n].RCOCD2 to RCOCD0="011B"), the range comparison interrupt factor flag (ADRCIF.RCINT[n]) is set to "1".
   Moreover, when the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"), the threshold over state is set in the threshold over flag (ADRCOT.RCOOF[n]). It is maintained until the range comparison interrupt factor flag is clear (ADRCIF.RCINT[n]="0").
6. The set operation by the continuous detection state is given priority when the range comparison interrupt factor clear flag (ADRCIF.RCINT[n]="0") and the continuous detection state are generated at the same time. The range comparison interrupt factor flag is the set state (ADRCIF.RCINT[n]="1"), and the threshold over flag (ADRCOT.RCOOF[n]) sets the threshold over state again.
7. When the range comparison result is inside the range, the continuous detection count state is initialized (ADRCCS[n].RCOCD2 to RCOCD0="000_B") even if the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1").
8. The continuous count detection state (ADRCCS[n].RCOCD2 to RCOCD0) is incremented even in the range comparison interrupt factor flag set (ADRCIF.RCINT[n]="1") state, because the range comparison result exceeds the upper bound threshold.
   However, because of range comparison interrupt factor flag set state (ADRCIF.RCINT[n]="1"), the threshold over flag (ADRCOT.RCOOF[n]) maintains the last value.
9. The range comparison interrupt factor flag is cleared (ADRCIF.RCINT[n]="0") because of the range comparison interrupt factor flag clear (ADRCIF.RCINT[n]="0").
   Moreover, the maintenance of a threshold over flag (ADRCOT.RCOOF[n]) is released.
10. Because the range comparison result continuously reached the continuous detection count specification value (ADRCCS[n].RCOCD2 to RCOCD0="011_B"), the range comparison interrupt factor flag (ADRCIF.RCINT[n]) is set to "1".
    Moreover, when the range comparison interrupt factor flag is set (ADRCIF.RCINT[n]="1"), the threshold over state is set in the threshold over flag (ADRCOT.RCOOF[n]). It is maintained until the range comparison interrupt factor flag is cleared (ADRCIF.RCINT[n]="0").
11. When the range comparison execution enable is set (ADRCCS[n].RCOE="0"), the continuous detection count state (ADRCCS[n].RCOCD2 to RCOCD0) is initialized to "000_B".
    Moreover, neither the range comparison interrupt factor flag (ADRCIF.RCINT[n]) nor the threshold over flag (ADRCOT.RCOOF[n]) is cleared by the range comparison execution disable setting (ADRCCS[n].RCOE="0").

5.3. A/D Activation Arbitration Operation

This section explains the A/D activation arbitration operation.

A/D activation requests from A/D activation compare are arbitrated to generate an A/D activation trigger. In addition, the analog channel to be A/D-converted is determined.

5.3.1. A/D Activation Trigger Arbitration

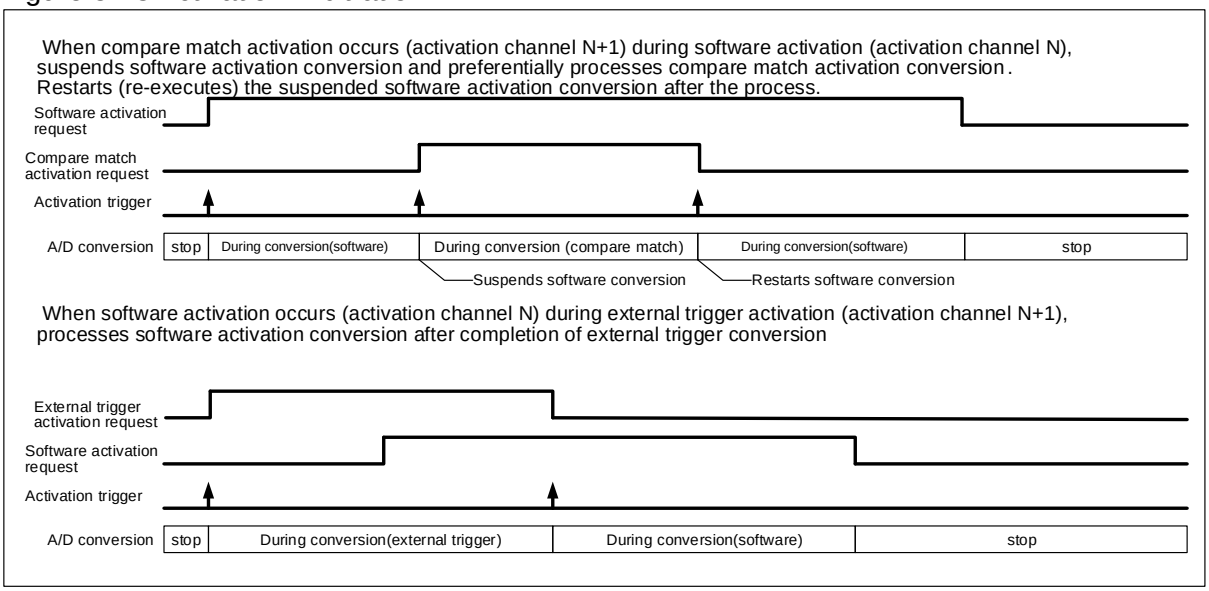
The A/D activation trigger arbitration is explained.

An A/D activation trigger is generated after one of activation requests from A/D activation compare channels is selected. Three types of activation requests from individual A/D activation compare channels are entered: software activation request, external trigger/reload timer activation request/PPG activation, and compare-match activation request. A/D activation trigger signals are thus generated. If contention occurs among multiple activation requests, the activation request with the lowest compare channel number takes precedence. Activation requests that are not selected are made to wait. When the current A/D conversion completes, arbitration restarts.

The sequence of activation arbitration priorities based on the activation factor is as follows: Compare-match activation request > external trigger/reload timer activation request/PPG activation request > software activation request. If two activation requests are caused by activation factors assigned the same priority, the request with the lower activation channel number takes precedence.

- If an activation factor with the same priority is encountered during A/D conversion suspension:
  The request from the activation channel with the lower number is processed first.
- If an activation factor with a different priority is encountered during A/D conversion suspension:
  The request based on the higher-priority activation factor is processed first.
- If an activation factor with a higher priority is encountered during A/D conversion:
  The current conversion is suspended, and the higher-priority activation factor is processed. After this processing completes, arbitration is performed again. The suspended activation factor is then processed.
- If an activation factor with a lower priority is encountered during A/D conversion:
  After the current conversion completes, arbitration is performed again. The activation factor with the lower priority is then processed.
- If an activation factor with the same priority is encountered during A/D conversion:
  After the current conversion completes, arbitration is performed again. The activation factor with the same priority is then processed.

Figure 5-23 Activation Arbitration



5.3.2. Analog Channel Select

The analog channel select is explained.

The A/D activation request and the A/D conversion target analog channel number are entered from the A/D activation compare.
The A/D activation arbitration selects the activation request analog channel number of the selected A/D activation compare channel.

5.3.3. A/D Conversion Cancel Function

The A/D conversion cancel function is explained.

When the activation request from the request source becomes inactive during the A/D conversion, an A/D conversion cancel signal is generated to abort the current conversion processing. If the activation factor from another activation channel is active when the activation request from the request source becomes inactive, an A/D conversion cancel signal is not generated, but an A/D activation trigger based on the active activation factor is generated.

5.4. 12-bit A/D Converter Operation

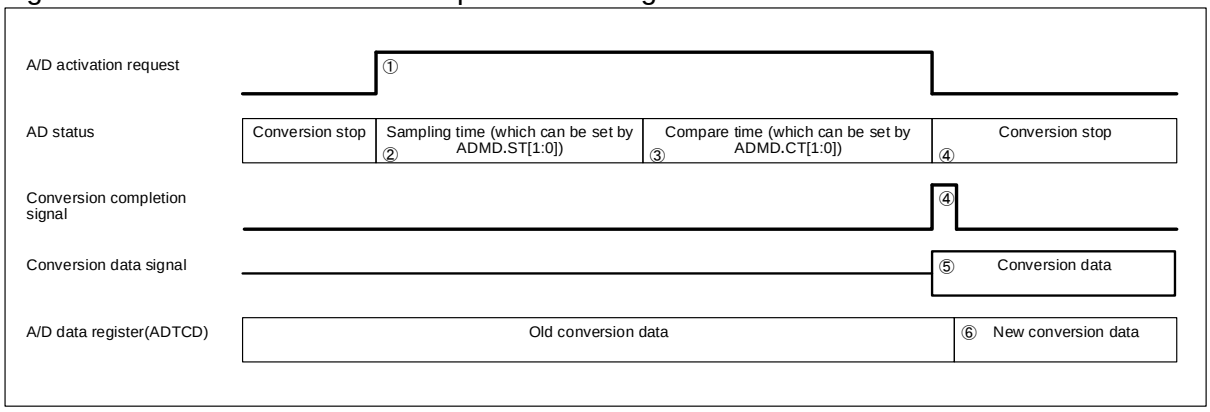
This section explains the 12-bit A/D converter operation.

The 12-bit A/D converter controls the A/D conversion.

5.4.1. Operation Timing

The operation timing is shown.

Figure 5-24 12-bit A/D Converter Operation Timing



(1) With the specified activation factor, the A/D conversion is started.

(2) After the activation request of (1) is received, the sampling operation is started.

(3) After the sampling time specified with ADMD.ST[1:0] passes, the compare operation is started.

(4) After the compare time specified with ADMD.CT[1:0] passes, the conversion completion signal rises, and the conversion is completed.

(5) A/D conversion data is output.

(6) New conversion data is stored in the A/D data register (ADTCD).

5.4.2. Activation Factors

The activation factor is shown.

A/D conversion activation factors include the software activation, external trigger activation, reload timer activation, compare match activation, and PPG activation. It is selected with ADTECS.STS2, ADTCS.STS[1:0].
If you do not want to issue an A/D activation request, set these bits to specify software activation ("000_B") and set the corresponding bit of ADTSE to disable software activation (ADTSE.ADT = 0).

5.4.3. A/D Conversion

The A/D conversion is shown.

One sequence of A/D conversion is performed in response to one activation trigger.

5.4.4. Re-activation

The re-activation is shown.

When an activation trigger signal is entered during the A/D conversion, the current conversion is stopped and initialized and the A/D conversion is re-activated.
For this reason, the A/D conversion re-activation starts with a delay of several cycles of the clock (12-bit A/D converter clock) in comparison with ordinary activation (start of A/D conversion while the A/D conversion is stopped).

5.4.5. A/D Conversion Cancel

The A/D conversion cancel is shown.

If an A/D conversion cancel signal is received during the A/D conversion, the current conversion stops and initialization takes place.

5.4.6. Analog Channel Select Control

The analog channel select control is shown.

Information on the analog channel to be A/D-converted is entered in addition to the activation trigger.
The analog channel information, which is effective when the activation trigger is active, is used to select an analog channel.

5.4.7. A/D Conversion Time

The A/D conversion time is shown.

The A/D conversion time is between total time of the sampling time and the compare time.

■ Sampling time

The sampling time selection can be set to either each channel or to a common setting by using the sampling time setting per channel enable bit (ADMD.STPCEN).

- When ADMD.STPCEN="0", the sampling time common to all channels is enabled. The sampling time is set by the sampling time set bit (ADMD.ST1, ST0).
- When ADMD.STPCEN="1", the sampling time of each channel can be set. The sampling time of each channel is set by the sampling time setting per channel enable bit (ADSTPCS.STCHn1, STCHn0: n=00 to 47).

Table 5-11 Sampling Time to Peripheral Clock Frequency

ST1, STCHn1	ST0, STCHn0	Function	Sampling time (Peripheral clock frequency)			
			(40MHz)	(32MHz)	(24MHz)	(16MHz)
0	0	12 Peripheral clock	300ns	375ns	500ns	750ns
0	1	18 Peripheral clock	450ns	562.5ns	750ns	1125ns
1	0	24 Peripheral clock	600ns	750ns	1000ns	1500ns
1	1	48 Peripheral clock	1200ns	1500ns	2000ns	3000ns

Note:

Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

Set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 47) bits to become 700ns(4.5V to 5.5V)/1000ns(2.7V to 5.5V) or more at the sampling time. When the sampling time is less than 700ns/1000ns, a normal value of the analog conversion value might not be obtained.

■ Compare time

The compare time is set to compare time set bit (ADMD.CT1, CT0).

Table 5-12 Compare Time to Peripheral Clock Frequency

CT1	CT0	Function	Compare time (Peripheral clock count)			
			(40MHz)	(32MHz)	(24MHz)	(16MHz)
0	0	28 Peripheral clock	700ns	875ns	1166.7ns	1750ns
0	1	42 Peripheral clock	1050ns	1312.5ns	1750ns	2625ns
1	0	56 Peripheral clock	1400ns	1750ns	2333.4ns	3500ns
1	1	112 Peripheral clock	2800ns	3500ns	4666.7ns	7000ns

Note:

Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

Set the CT1 and CT0 bits to become 700ns or more at the comparison time. When the compare time is less than 700ns, a normal value of the analog conversion value might not be obtained.

5.4.8. A/D Conversion Completion and A/D Data Retrieval

The A/D conversion completion and A/D data retrieval is shown.

When the A/D conversion completes normally without re-activation or cancel (the specified number of cycles have passed), the received conversion data is retrieved and output. At this time, an A/D conversion completion signal is generated.

5.4.9. Power down

The power down is shown.

Power down is in effect during the standby mode.

Note:

Stop the A/D conversion before the transition to standby mode occurs.

6. Notes

This section explains notes.

■ Notes on Using A/D Activation Compare

● Configuration of select

Be sure to configure select while the free-run timer is inactive.

● Configuration of protection of A/D data register

It is necessary to set the PRT bit and the PRTS bit before the A/D conversion starts. It is not allowed to set this bit during the A/D conversion and the state that the A/D data register is being protected.
When cancelling the protection function of the A/D data register, execute the protection cancellation which is set in the PRTS bit after the A/D conversion stops, or disable the protection function by the PRT bit.
If the PRTS bit is changed while the A/D data register is being protected, it is necessary to execute the protection cancellation operation which is set in the PRTS bit (for example reading of the A/D data register, or clearing operation by writing "0" to an interrupt request flag bit) after changing the PRTS bit in order to cancel the A/D data register protection. For example, PRTS is set to 0 after clearing an interrupt request flag bit while PRT=1 and PRTS=1, and the A/D data register is being protected, it is needed to read the A/D data register and clear by writing "0" to the interrupt request flag bit again in order to cancel the protection.

● Compare match activation

If the compare register (ADCOMP) is set to 0x0000 or the same value as the value set in the compare clear register of the free-run timer, an A/D activation request signal will be generated when a compare match is detected, regardless of whether the setting of the counting direction select bits (SEL0 and SEL1) of the A/D activation trigger control register (ADTCS) specifies counting-up or counting-down.

● Setting of the counting direction select bits of the A/D activation trigger control status register

When the free-run timer is in the up count mode, the counting direction select bits SEL0 and SEL1 of the A/D activation trigger control status register (ADTCS) must not be set respectively to "1" and "0" (SEL1, SEL0)=(1, 0) (only counting down).

● Range compare

The execution timing of the range comparison is after the A/D conversion result is set in A/D data register (ADTCD) when the A/D conversion is completed. Therefore, the range comparison result is reflected (RCOCD2 to RCOCD0 bit, RCOOF bit, and RCINT bit) delaying 2 cycles in the peripheral clock from the generation of the A/D conversion

completion interrupt request. Also, the range comparison interrupt request is generated.
When confirming that the comparison target is outside the range comparison range, the continuous detection measurement is not cleared to 0 times and continue the continuous detection even if the state of the range comparison result changes from larger than the upper bound threshold into smaller than lower bound threshold.
To initialize the continuous detection count state of the range comparison result, switch to enable range comparison execution (ADRCCS.RCOE="0" → "1") after range comparison execution disable has been set, while the A/D has not yet been requested.

About the continuous scan conversion that doesn't use the conversion count specification of each channel

When two or more repeat modes (ADTCS.RPT="1") are set between the activation channels corresponding to the same A/D converter unit, only a certain channel (channel with the highest priority level) will be processed in the A/D activation arbitration of latter part. Therefore, please make the protection function effective about those activation channels when the repeat mode is set with two or more activation channels.
When two or more repeat modes (ADTCS.RPT="1") are set between the activation channels corresponding to the same A/D converter unit, the conversion of the activation channel with low priority is not executed if the data protection state of the activation channel with high priority is released before converting the activation channel with low priority. Therefore, the activation channel of low priority is not executed and the scan conversion is not executed normally when the data protection state is released before the final activation channel of the continuous scanning conversion is executed.

About the continuous scan conversion when the conversion count of each channel is specified

Please switch the activation channel whose conversion count specification has been set to scan conversion execution enable, to repeat conversion mode.
When the continuous scan conversion mode with the conversion count of each channel specified, the EOCF bit set to the final channel of the scan conversion is cleared to "0" immediately even if set to "1" because it is restarted from the top immediately after the scan conversion is completed.
When an activation factor that is the same as the scan conversion target is set to an activation channel with a number larger than that of the final channel of continuous scan conversion (with the continuous scan conversion mode with the conversion count of each channel specified effective), the activation channel with the number larger than that of the final channel is evaluated once by the A/D activation arbitration of the latter part after completion of the final channel of the continuous scan conversion.
When an activation factor that is the same as the scan conversion target is set to an activation channel with a number larger than that of the final channel of stop scan conversion (with the stop scan conversion mode with the conversion count of each channel specified effective), the activation channel with the number larger than that of the final channel can execute A/D conversion only during the stop period.
The timing to release the data protection state does not have the restriction if the data protection function is effective. However, when the activation channel that does the scan conversion is in the data protection state, the scan conversion stops until the data protection state is released.

About the setting of the sampling time and the compare time

Set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 47) bits to become 700ns(4.5V to 5.5V)/1000ns(2.7V to 5.5V) or more at the sampling time. When the sampling time is less than 700ns/1000ns, a normal value of the analog conversion value might not be obtained.
Set the CT1 and CT0 bits to become 700ns or more at the comparison time. When the compare time is less than 700ns, a normal value of the analog conversion value might not be obtained.

About the setting of the ADMD register and ADSTPCS

Please rewrite the bits of the A/D mode setting register (ADMD) and the sampling time setting register (ADSTPCS) when the A/D operation has stopped before A/D conversion operation.

Chapter 45: Flash Memory

This chapter explains the flash memory.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FZ5C8-1v9-91520-3-E

1. Overview

This section explains an overview of the flash memory.

The size of the flash memory built in this series is up to 1088Kbytes (1024K+64Kbytes).

Error correction codes (ECC) are attached.

2. Features

This section explains features of the flash memory.

- **Usable capacity:**

  MB91F522 : 256Kbytes+64Kbytes (large sectors 128K × 2 + small sectors16K × 4)
  MB91F523 : 384Kbytes+64Kbytes (large sectors 128K × 3 + small sectors16K × 4)
  MB91F524 : 512Kbytes+64Kbytes (large sectors 128K × 4 + small sectors16K × 4)
  MB91F525 : 768Kbytes+64Kbytes (large sectors 128K × 6 + small sectors16K × 4)
  MB91F526 : 1024Kbytes+64Kbytes (large sectors 128K × 8 + small sectors16K × 4)

  Since this series has ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes described above.

  * 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 80MHz.

- **Write from external :**

  Possible from ROM writer

- **Operation mode :**

  1. CPU-ROM mode
     (CPU / DMA accesses the flash memory. Read-only)
  2. CPU programming mode
     (CPU accesses the flash memory. Read/Write/Erase)
  3. Flash memory mode (flash memory accessible from external)

- **Can be read, written, or erased (automatic algorithm*) by CPU**

- **Can be read, written, or erased (automatic algorithm*) by ROM writer**

- **Security function**

  · Operations after instruction fetch from external and write/erase except for chip erase are inhibited when security is on to avoid reading out flash memory data by an outsider.
  · The use of on-chip debugger (OCD) enables read from external by using OCD, even when security is on after

password authentication.

- **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.
  - An error is detected when data is read in the chip erase/sector erase mode. Always write "FFFF" and then read the data when the data in the erase mode (FFFF) needs to be read correctly.

*: Automatic algorithm = Embedded Algorithm™

3. Configuration

This section explains the configuration of the flash memory.

3.1 Block Diagram

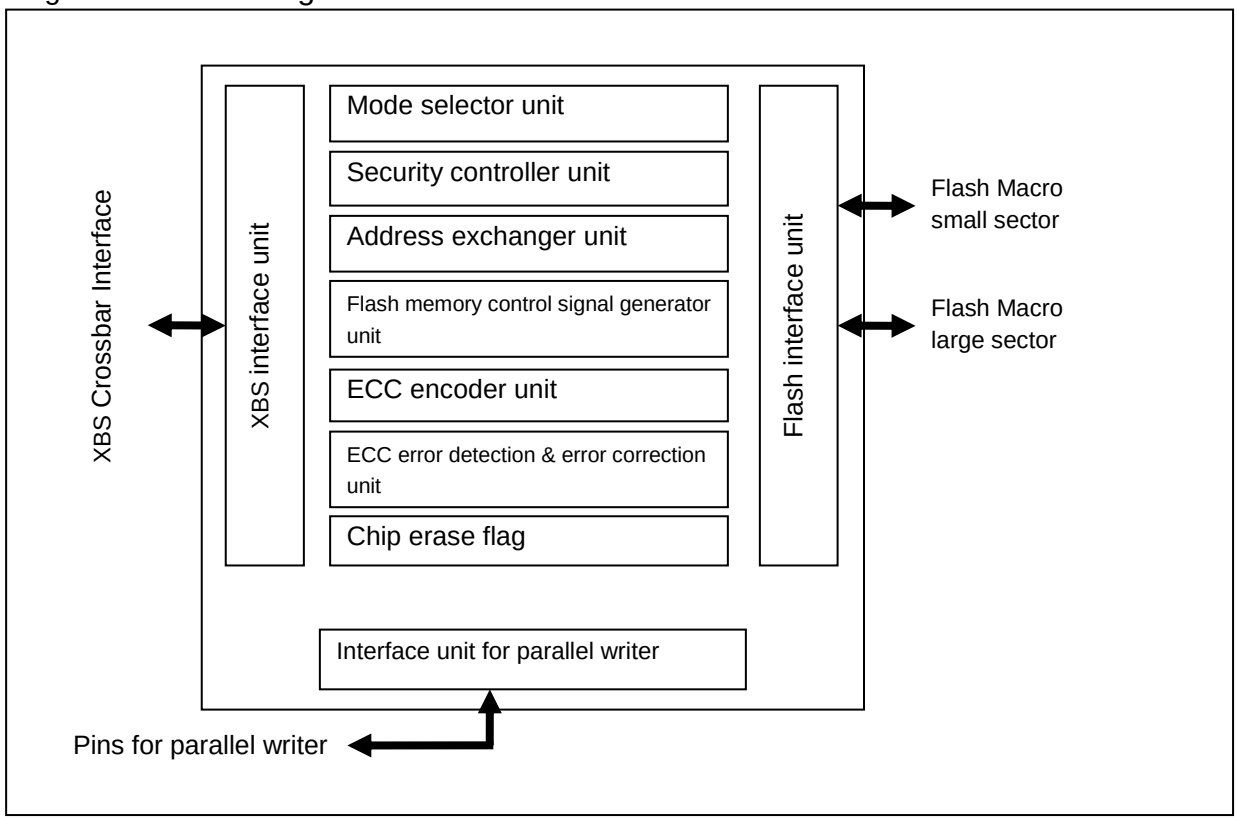
3.2 Sector Configuration Diagram

3.3 Sector Number and Flash Macro Number Correspondence Chart

3.1. Block Diagram

The block diagram of the flash memory is shown below.

Figure 3-1 Block Diagram



3.2. Sector Configuration Diagram

The sector configuration diagram of the flash memory is shown below.

Figure 3-2 Sector Configuration Diagram (MB91F522)

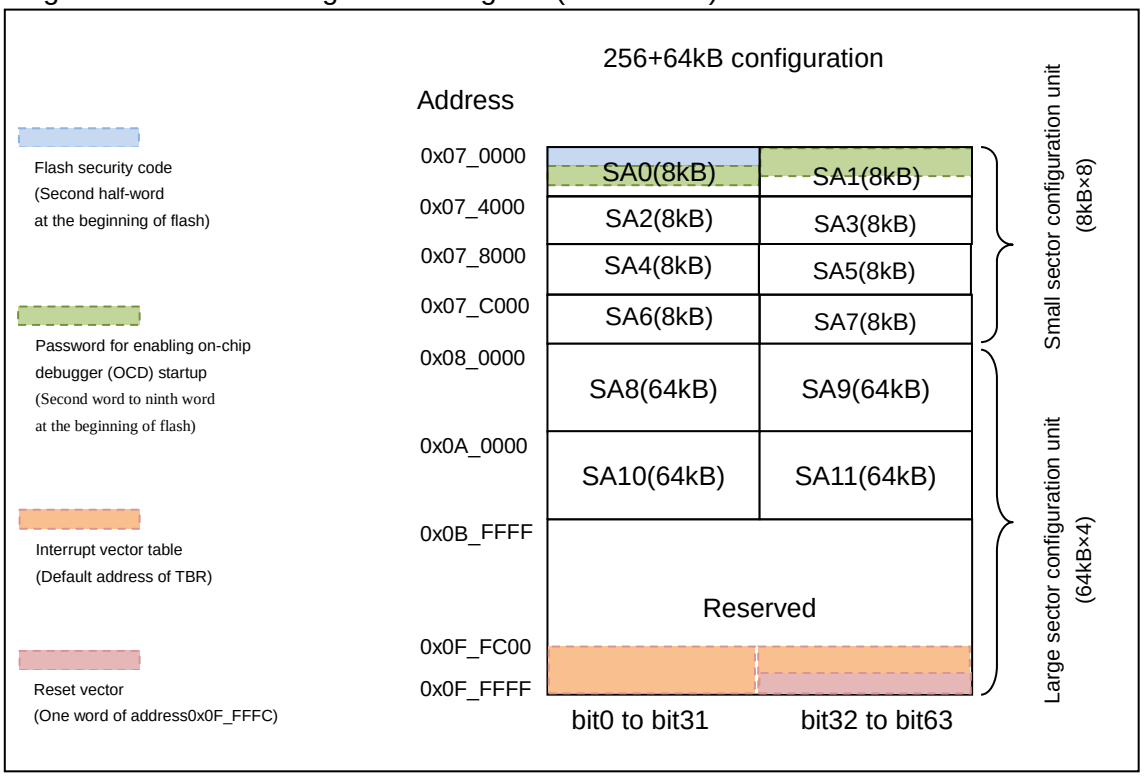


Figure 3-3 Sector Configuration Diagram (MB91F523)

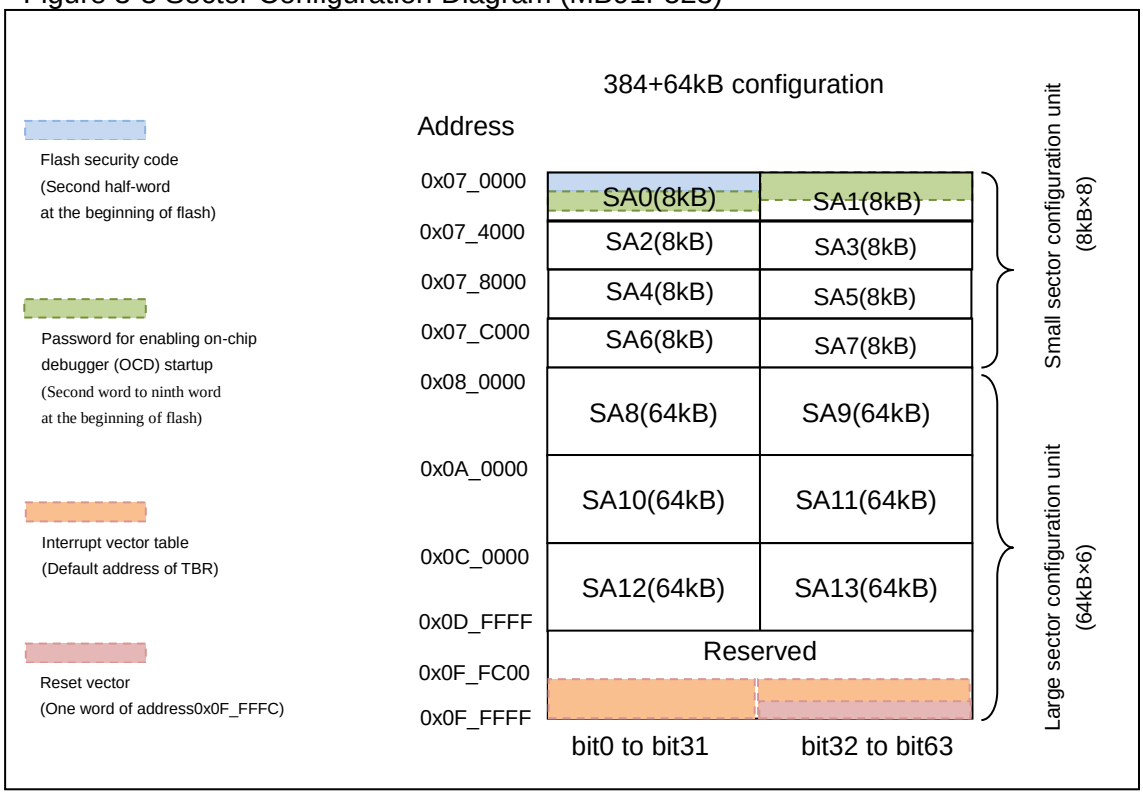


Figure 3-4 Sector Configuration Diagram (MB91F524)

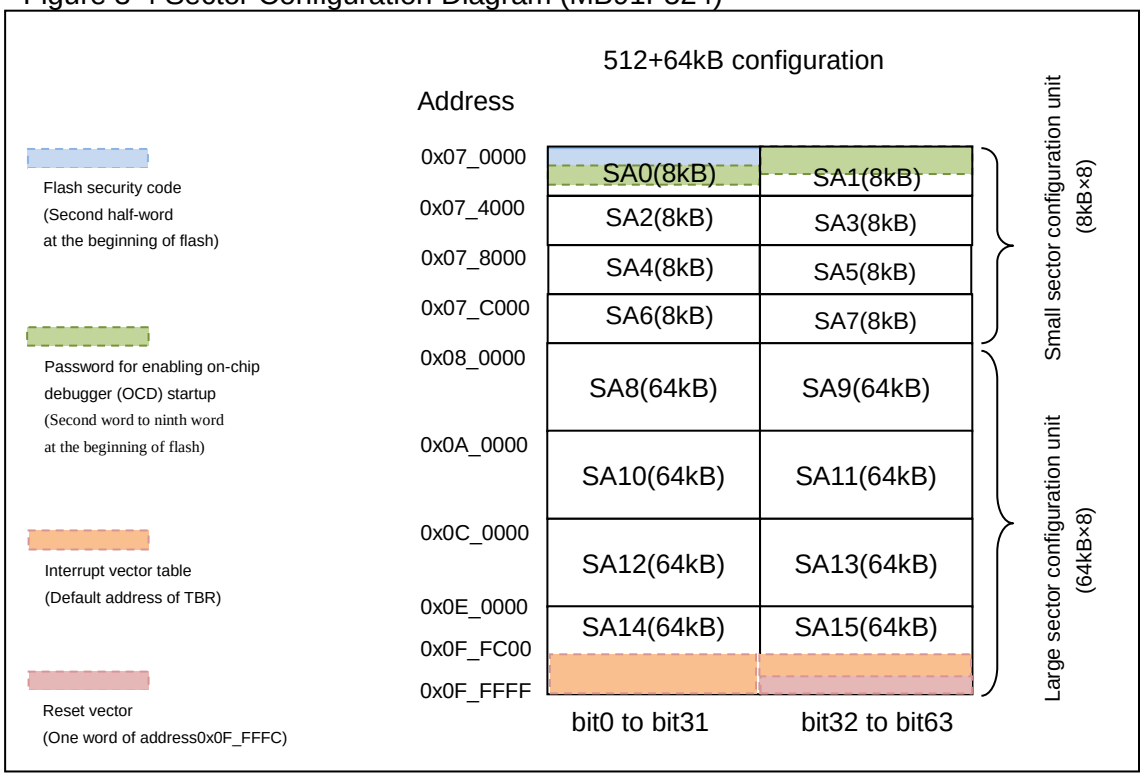


Figure 3-5 Sector Configuration Diagram (MB91F525)

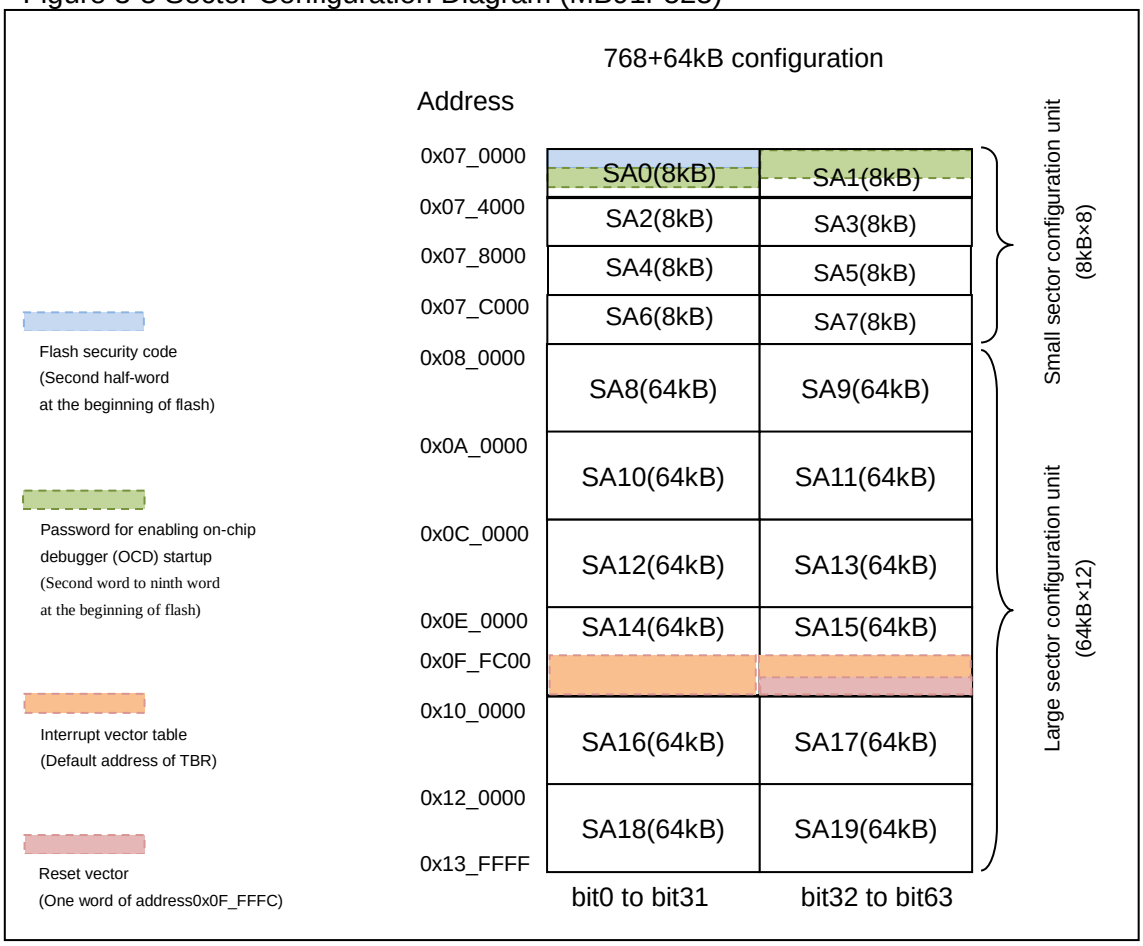
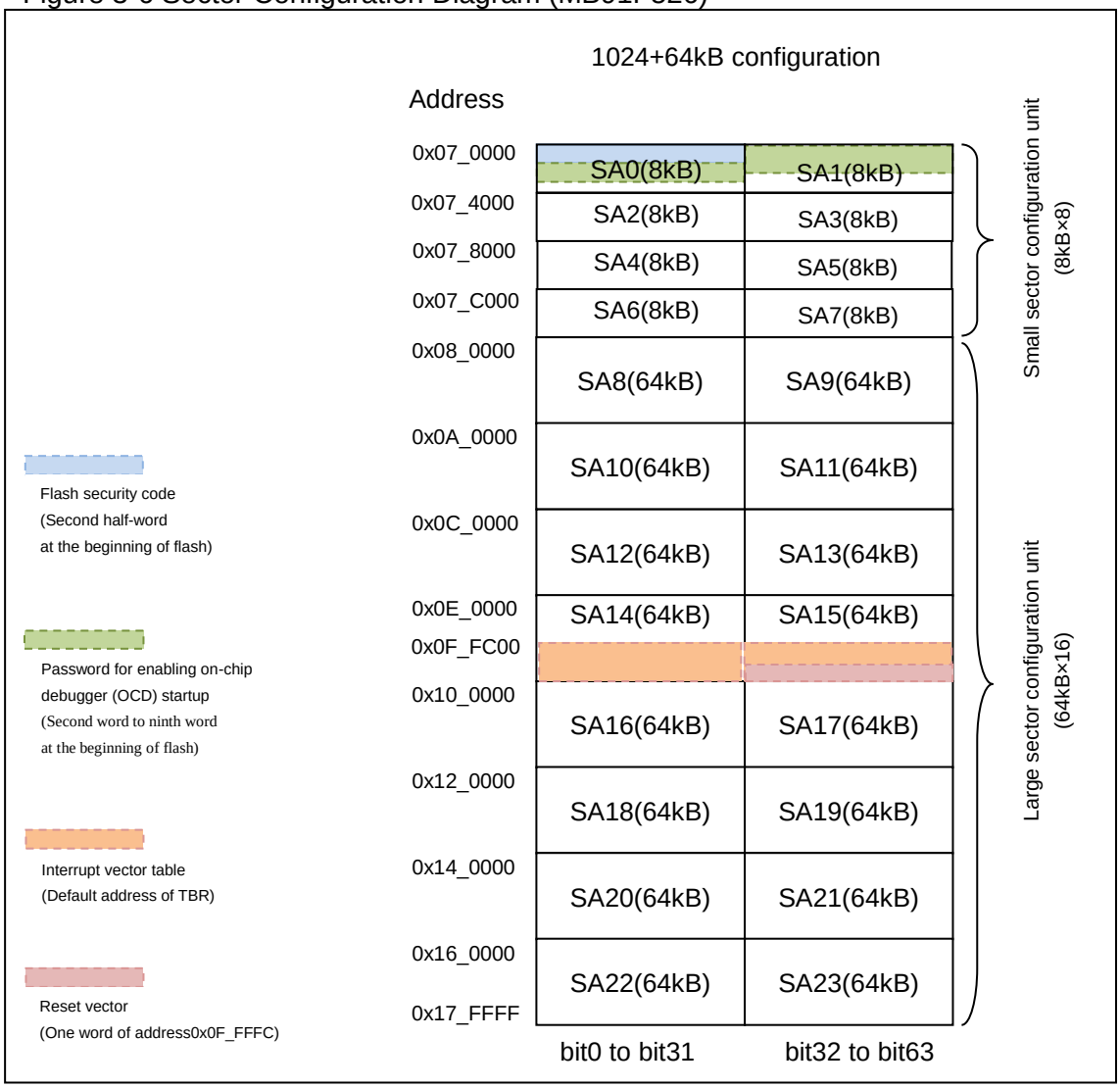


Figure 3-6 Sector Configuration Diagram (MB91F526)



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).

3.3. Sector Number and Flash Macro Number Correspondence Chart

Sector number and flash macro number correspondence charts are shown below.

Table 3-1 Sector Number Table MB91F522 (256+64KB Products)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32bits)	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 32bits)	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 32bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32bits)	64KB	
-	0x0C_0000 to 0x0F_FFFF	-	Reserved Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0xF_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Table 3-2 Sector Number Table MB91F523 (384+64KB Products)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32bits)	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 32bits)	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 32bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (Lower 32bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (Upper 32bits)	64KB	
-	0x0E_0000 to 0x0F_FFFF	-	Reserved Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0xF_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Table 3-3 Sector Number Table MB91F524 (512+64KB Products)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32bits)	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 32bits)	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 32bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (Lower 32bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (Upper 32bits)	64KB	
SA14	0x0E_0000 to 0x0F_FFFB (Lower 32bits)	64KB	
SA15	0x0E_0004 to 0x0F_FFFF (Upper 32bits)	64KB	Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0xF_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Table 3-4 Sector Number Table MB91F525 (768+64KB Products)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32bits)	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 32bits)	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 32bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (Lower 32bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (Upper 32bits)	64KB	
SA14	0x0E_0000 to 0x0F_FFFB (Lower 32bits)	64KB	
SA15	0x0E_0004 to 0x0F_FFFF (Upper 32bits)	64KB	Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0xF_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Sector number	Address	Sector size	Remark
SA16	0x10_0000 to 0x11_FFFB (Lower 32bits)	64KB	
SA17	0x10_0004 to 0x11_FFFF (Upper 32bits)	64KB	
SA18	0x12_0000 to 0x13_FFFB (Lower 32bits)	64KB	
SA19	0x12_0004 to 0x13_FFFF (Upper 32bits)	64KB	

Table 3-5 Sector Number Table MB91F526 (1024+64KB Products)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32bits)	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 32bits)	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 32bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32bits)	64KB	

Sector number	Address	Sector size	Remark
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (Lower 32bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (Upper 32bits)	64KB	
SA14	0x0E_0000 to 0x0F_FFFB (Lower 32bits)	64KB	
SA15	0x0E_0004 to 0x0F_FFFF (Upper 32bits)	64KB	Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0xF_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)
SA16	0x10_0000 to 0x11_FFFB (Lower 32bits)	64KB	
SA17	0x10_0004 to 0x11_FFFF (Upper 32bits)	64KB	
SA18	0x12_0000 to 0x13_FFFB (Lower 32bits)	64KB	
SA19	0x12_0004 to 0x13_FFFF (Upper 32bits)	64KB	
SA20	0x14_0000 to 0x15_FFFB (Lower 32bits)	64KB	
SA21	0x14_0004 to 0x15_FFFF (Upper 32bits)	64KB	
SA22	0x16_0000 to 0x17_FFFB (Lower 32bits)	64KB	
SA23	0x16_0004 to 0x17_FFFF (Upper 32bits)	64KB	

4. Registers

This section explains registers of the flash memory.

Table 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

4.1. Flash Control Register : FCTLR (Flash ConTroL Register)

The bit configuration of the flash control register is shown below.

This register configures the access control to flash.

■ FCTLR : Address 0840_H (Access : Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	FWE	Reserved		FSZ[1:0]		FAW[1:0]	
Initial value	1	0	-	-	1	0	0	0
Attribute	R1,WX	R/W	RX,W0	RX,W0	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FDSBL	Reserved		RDYF	Reserved			
Initial value	0	-	-	0	-	-	-	-
Attribute	R/W	RX,W0	RX,W0	R/W	RX,W0	RX,W0	RX,W0	RX,W0

[bit15] Reserved

This bit is reserved. This bit always reads out as "1". Writing has no effect on the operation.

[bit14] FWE (Flash Write Enable) : Flash Write Enable

It is the write enable bit to flash. Setting this bit configures CPU programming mode. Use the FSTR:FRDY bit to check whether writing is enabled.

If this bit is set, the ECC error detection and data correcting function will be disabled for data fetching to the flash memory.

FWE	Description
0	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 the reading time of the flash memory is 12.5ns, access to the flash memory at over 80MHz is disabled. Please set it to FAW=1(1wait) when you access it at over 80MHz.

Please set these bits before making the clock high-speed when you insert the wait cycle by FAW. Moreover, please set these bits after setting the clock low-speed when you delete the wait cycle.

FAW[1:0]	Description
00	0 cycle (Initial value)
01	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 during branch access

This bit directs the wait cycle insertion during branch access. During branch access, the wait cycle is inserted when this bit is set to "1". The purpose of this is to match the processing cycle when branching. When the branch access is generated, the control at the wait cycle is made by an internal state of 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.

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 0843_H (Access : Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					FECCERR	FHANG	FRDY
Initial value	-	-	-	-	-	0	0	1
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R,W	R,WX	R,WX

[bit7 to bit3] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] FECCERR (Flash ECC Error coRRection) : Data read ECC correction occurred

This bit is set if an ECC error correction occurs while reading flash memory other than CPU instruction read. This bit is cleared by writing "0".

FECCERR	Read	Write
0	An error correction by ECC has not occurred during data read (initial value)	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 "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.

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 2308_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0
Initial value	-	-	-	0	-	0	0	0
Attribute	RX,WX	RX,WX	RX,WX	R/W	RX,WX	R/W0	R/W	R/W

[bit7 to bit5] Reserved

These bits are reserved bits. The read value is undefined. Writing has no effect on the operation.

[bit4] DFWDSBL (Data Fetch Wait cycle DiSaBLe) : Data fetch wait cycle 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 this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved

This bit is reserved. 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

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.

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.

5. Operation

This section explains operations of the flash memory.

This section explains how to access flash area.

5.1 Access Mode Setting

5.2 Writing to Flash Memory by CPU

5.3 Automatic Algorithm

5.4 Reset Command

5.5 Write Command

5.6 Chip Erase Command

5.7 Sector Erase Command

5.8 Sector Erase Suspend Command

5.9 Security Function

5.10 Notes on Using Flash Memory

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.)

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.

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.

5.2. Writing to Flash Memory by CPU

This section explains writing to flash memory by CPU.

After configuring CPU programming mode, perform erasing and writing using the automatic algorithm. In this model, because error correction codes (ECC) are added to each single word, writing needs to be performed for each single word. In the following procedure, each word is written 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 ECC calculation.

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 "5.5. Write Command" for details on the write command.

3. Read the hardware sequence flag until the write has finished. See "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

5.3. Automatic Algorithm

This section explains the automatic algorithm.

When using CPU programming mode, writing and erasing of flash memory are performed by starting the automatic algorithm. This section explains the automatic algorithm.

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 5-1 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	$B0_H$										

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]
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:

- If an incorrect address value or data value is written, or an incorrect sequence is written, the commands written till then will be cleared.
- Do the following to the LSB 2-bit for the command address, and for the sector address (SA) issued at the generation of the sector erase command.
  - When half-word access : 2'b00
  - When byte access : 2’b01 or 2’b11

  Example 1:
  During byte access, if command address = (LSB 2-bit of the standard command address changed to 2’b01.)
  $yAA8_H \rightarrow yAA9_H$, $x554_H \rightarrow x555_H$, 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:
  During byte access, if command address = (LSB 2-bit of the standard command address 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)

■ Reset Command

By sending the reset command to the target flash memory, various commands as shown in Table 5-1that have been input so far can be cancelled and commands can be input again from the first time.

However, when each command is input up to the end 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 from the external.

See "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.
- With one 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, writing of flash is limited. See "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, the flash memory writes "0" to all of the cells in the chip to verify the margins (preprogramming) before erasing the entire chip, so there is no need to write to flash memory before erasing the chip. Furthermore, while verifying the margin, there is no need to control the flash memory from the external.

See "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, the flash memory writes "0" to the cells in the sector to be erased to verify the margins (preprogramming) before erasing the sector, so there is no need to write to flash memory before erasing the sector. Furthermore, while verifying the margin, there is no need to control the flash memory from the external.

See "5.7. Sector Erase Command" for details on the actual operation.

Note:

When security is ON, the sector erase procedure of flash is limited. See "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, neither a new write nor erase command is accepted.
To restart the interrupting erase operation from the sector erase suspend condition, send the erase restart command.

When the erase resume command is accepted, the state comes back to the sector erase condition, resuming the erase operation.
Even when the state has shifted from the command time-out state to this state, when the erase resume command is correctly written, the state does not shift to the command time-out state but shifts to the sector erase state, and the sector erase operation is immediately restarted.

See "5.8 Sector Erase Suspend Command" for actual operation.

Notes:

- 16.7μs + 2 cyc or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.
- At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.
- 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.

5.3.2. Automatic Algorithm Execution State

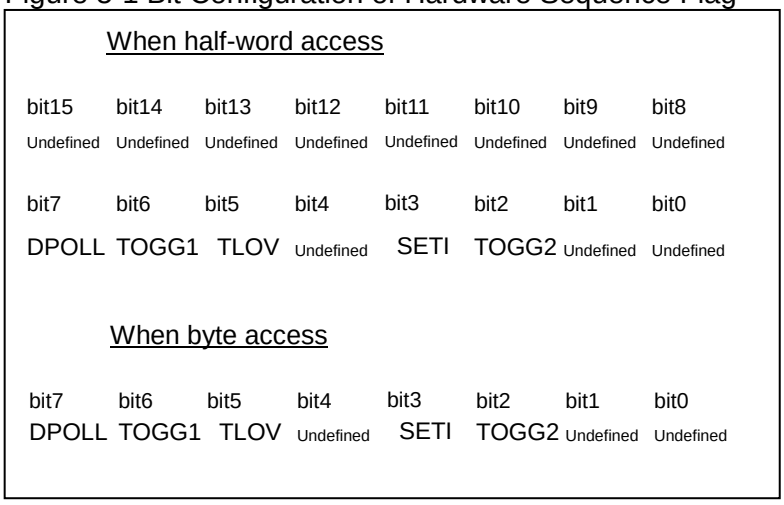
This section explains the automatic algorithm execution state.

Because flash memory performs writing and erasing with an automatic algorithm, it is possible to check whether the automatic algorithm is executed with the FRDY bit of the flash status register (FSTR), and to check the operating state with the hardware sequence flag.

■ Hardware Sequence Flag

This flag indicates the state of the automatic algorithm. When the FRDY bit of the flash status register (FSTR) is "0", the operating state can be checked by reading from an arbitrary address in flash memory. The following shows the bit configuration of the hardware sequence flag.

Figure 5-1 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 status of each bit of the hardware sequence flag and the flash memory status.

Table 5-2 Correspondence between Flags and Flash Memory Status

Status		DPOLL	TOGG1	TLOV	SETI	TOGG2
Run	Writing	Inverted data *	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data *	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

*: See "● Bit descriptions" for the values that are read out.

● Bit descriptions

[bit15 to bit8] Undefined bits

[bit7] DPOLL : Data POLLing flag bit

When the hardware sequence flag is read by specifying the write/erase target address, this bit indicates whether 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	The opposite value (inverted data) of the value of bit7 of the data last written is read out. The address specified to read out the hardware sequence flag is not accessed.
After writing finished	The value of bit7 of the address specified to read out the hardware sequence flag is read out.

2. During sector erase

During sector erase	"0" is read out from the sector being erased.
After sector erase	"1" is always read out.

3. During chip erase

During execution of chip erase	"0" is always read out.
After chip erase	"1" is always read out.

4. During sector erase suspend

State of suspend (incomplete end)	"0" is read out from the sector erase suspend sector.
Sector erase operation completion	"1" is read out 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 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 out alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
After write / sector erase / chip erase	The value of bit6 of the address specified to read out the hardware sequence flag is read out.

[bit5] TLOV : Timing Limit OVer flag bit

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether the execution time of the automatic algorithm exceeded the time specified 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 specified time
"1"	Exceeds specified 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 specified 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 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 under execution 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. When the read address is the target sector for erase during sector erase suspend, this flag toggles output data to indicate that the sector is the target sector for erase.

Read out target erase sector	If this bit is read sequentially, "1" and "0" are read out alternatively (toggle operation). The address specified to read out the hardware sequence flag is not accessed.
Read out non target erase sector	Data at the specified address is read out.

[bit1, bit0] Undefined bits

5.4. Reset Command

The reset command is explained.

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 completes successfully. When the power is turned on, there is no need to issue a reset command. Furthermore, in reset state, data can be read using normal read access and programs can be accessed by the CPU; thus there is no need to issue the reset command when reading data.

5.5. Write Command

The write command is shown below

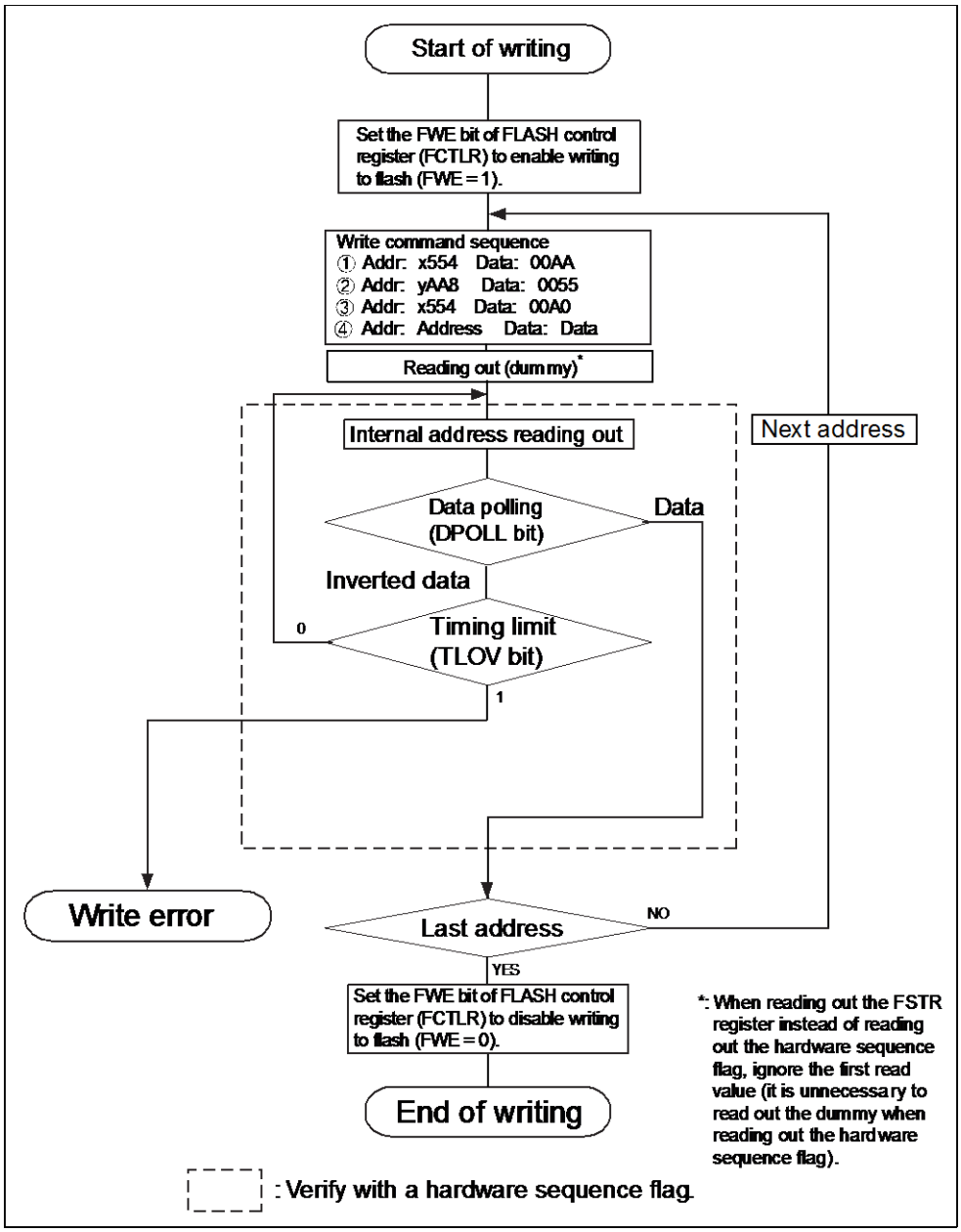
Writes are performed in the following order.

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 from external.

2. Perform read access to the written address
   The read data is the hardware sequence flag. Therefore, if bit7 (DPOLL bit) of the read data matches the written data, the write to the flash memory has finished. If the write has not finished, the opposite value (inverted data) of the value of bit7 of the last written data is read out.

The following shows an example of a write operation to the flash memory.

Figure 5-2 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 "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 specified time is exceeded, and the TLOV bit of the hardware sequence flag bit 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 being 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 "5.2 Writing to Flash Memory by CPU" for the procedure.

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 "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, the flash memory writes "0" to all of the cells in the chip to verify the margins (preprogramming) before erasing the entire chip, so there is no need to write to flash memory before erasing the chip. Furthermore, while verifying the margin, there is no need to control the flash memory from external.
- When security is on, there are restrictions in the procedure for erasing the flash. See "5.9.3 Unlocking Flash Security" for details.

5.7. Sector Erase Command

The sector erase command is shown below.

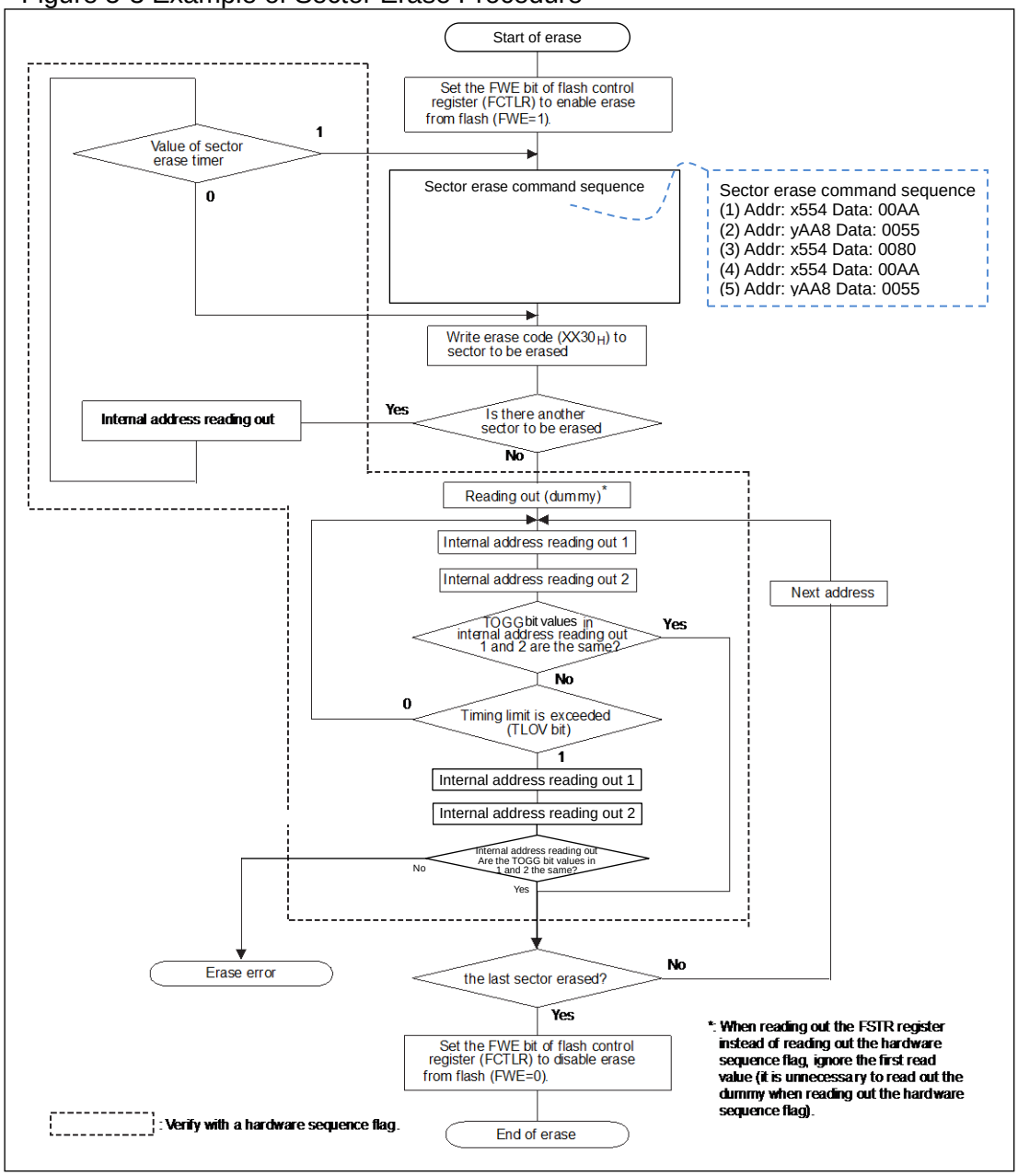
A sector in the flash memory can be selected and only data in the selected sector can be deleted. Multiple sectors can also be specified at the same time. Sector erase is performed in the following order.

1. Send sector erase commands sequentially to the target sector
   Once 40µs 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 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 5-3 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 mode.
- See "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 if the TLOV bit is "1".
- Only a command to add an erased sector and a command to suspend erasing are write commands acceptable in the command timeout state, while only a command to suspend erasing is an acceptable command in the sector erasing state.
- When the automatic erase algorithm is started, the flash memory writes "0" to the cells in the sector to be erased to verify the margins (preprogramming) before erasing the sector, so there is no need to write to flash memory before erasing the sector. Furthermore, while verifying the margin, there is no need to control the flash memory from external.

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, neither a new write nor erase command is accepted.

The sector erase suspend command is sent to an arbitrary address of target flash macro to suspend the sector erase.

After the sector erase stops, the reading operation from target flash macro is permitted. At this time, the hardware sequence flag is read out from the sector which is under the sector erase suspend condition.

The following conditions occur when the sector erase suspension state is entered.

- 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".

By using these, it is possible to verify that the sector erase suspension state has been entered.

Note:

- It takes up to 16.7µs + 2cyc from the time the sector erase suspend command is issued until the time when the sector erase operation is stopped and it becomes possible to read out from sectors that are not the erase target.
- At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.

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, send the sector erase restart command in Table 5-1.

The sector erase restart command is accepted only when the sector erase suspend condition is set.

Send the command after confirming that the sector erase suspend condition is set.

Accepting the erase restart command, the flash memory comes back to the sector erase condition, resuming the erase operation.

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, operations after instruction fetch from the external bus and write/erase except for chip erase are inhibited. See "5.9.4 Flash Access Restrictions When Security is ON" for details on the restrictions.

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.

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 3-2 to Figure 3-6 Sector Configuration), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.

5.9.3. Unlocking Flash Security

Unlocking flash security is shown below.

Execute the chip erase command to all flash macros in the following order.

(1) Erase WorkFlash.
(2) Erase the program flash which contains the flash security code.

Erase the program flash last, as shown above. Otherwise, the erase command to program flash is ignored. Furthermore, if a reset is input between (1) and (2), execute step (1) again before (2).

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 chip erase to each flash macro must be executed in the order shown above, from the viewpoint of the data protection stored in the flash macro.

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 generated according to the operating mode.

Table 5-3 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 factor. 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 reading operation masks data and returns 0xFFFF_FFFF. Write commands and sector erase commands are ignored. Chip erase commands are accepted. See "5.9.3 Unlocking Flash Security".

Furthermore, while the security is ON, when a data read is performed to the security information storage area (first 9 words of the flash memory),

- A data access error will occur, and an illegal instruction exception or data access error interrupt will occur. (See "FR Family FR81 32-bit microcontroller programming manual" and "10.3.1 MPU control register (MPUCR)" in CHAPTER:CPU for details.)
- 0xFFFFFFFF is returned as the read value.

However, when the OCD tool is connected, this restriction does not apply to access from OCDU or read during the debug state.

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 "5.2. Writing to Flash Memory by CPU" for the procedure.
- Do not issue commands to multiple macros simultaneously (i.e. in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- If authentication by password of on-chip debugger (OCD) completes, you can read the content of flash memory from external by using OCD even if security is ON. When you want to stop reading by an outsider, a password for the on-chip debugger (OCD) activation approval must be configured.
- Changing to the standby state is a prohibited during flash program/erase.
- Because this flash memory has a built-in ECC, it is impossible to overwrite data to the address where some values have already been written.

Chapter 46: WorkFlash Memory

This chapter explains the WorkFlash memory.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FZ5C0-1v9-91520-3-E

1. Overview

This section explains the overview of the WorkFlash memory.

The size of the WorkFlash in this series is 64 Kbytes. Error correction codes (ECC) are attached.

2. Features

This section explains features of the WorkFlash memory.

- Usable capacity:
  MB91F52x : 64K bytes (8K bytes × 8 sectors)
  For ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes.
- High-speed operation:
  Reading 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.
- Error correction code (ECC) function
  - There is an error correction code (ECC) function that corrects errors of up to one bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read. Furthermore, ECC codes are automatically added when writing to the flash memory. Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development.
  - An error is detected when data is read in the chip erase/sector erase state.
    Always write "FFFF" and then read data when data with erase state (FFFF) needs to be read without error detection.

*: Automatic Algorithm=Embedded Algorithm™

3. Configuration

This section explains the configuration of the WorkFlash memory.

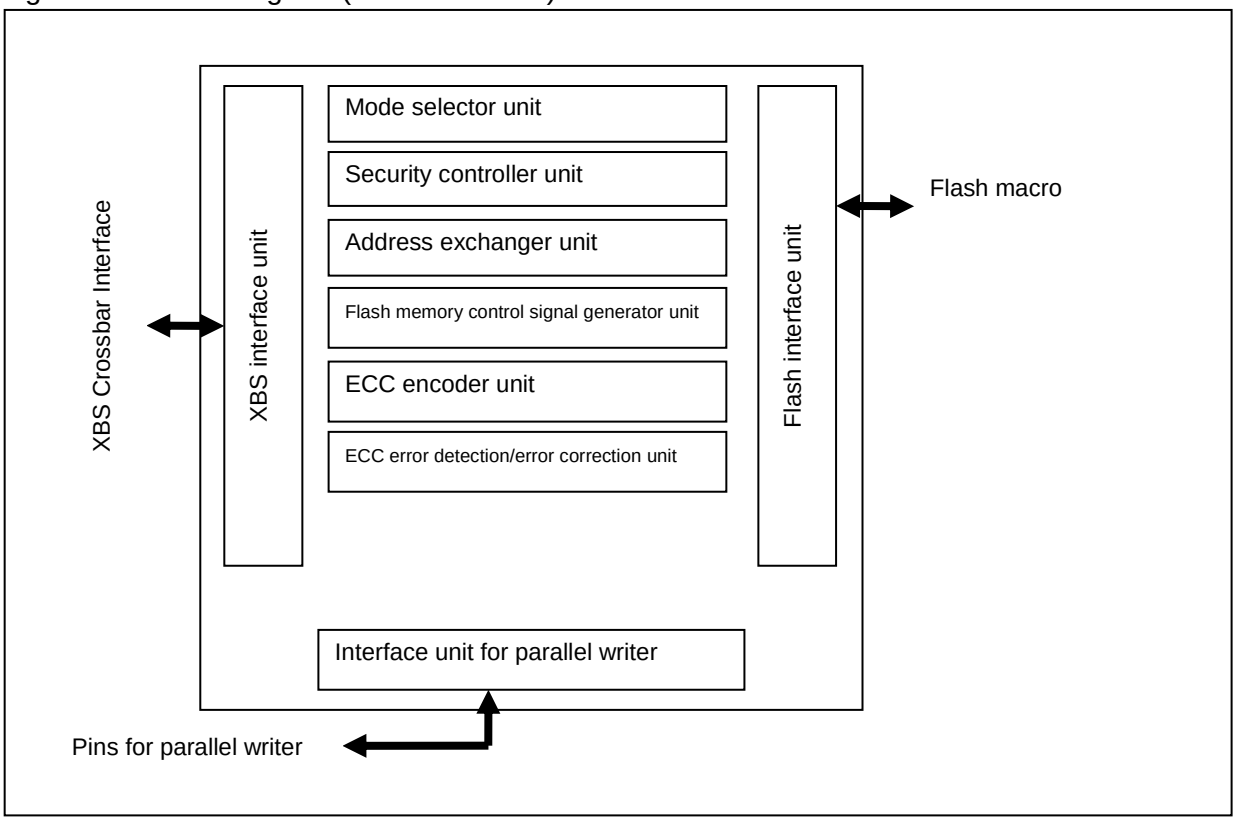
3.1 Block Diagram

3.2 Sector Configuration Diagram

3.1. Block Diagram

This section shows the block diagram of the WorkFlash memory.

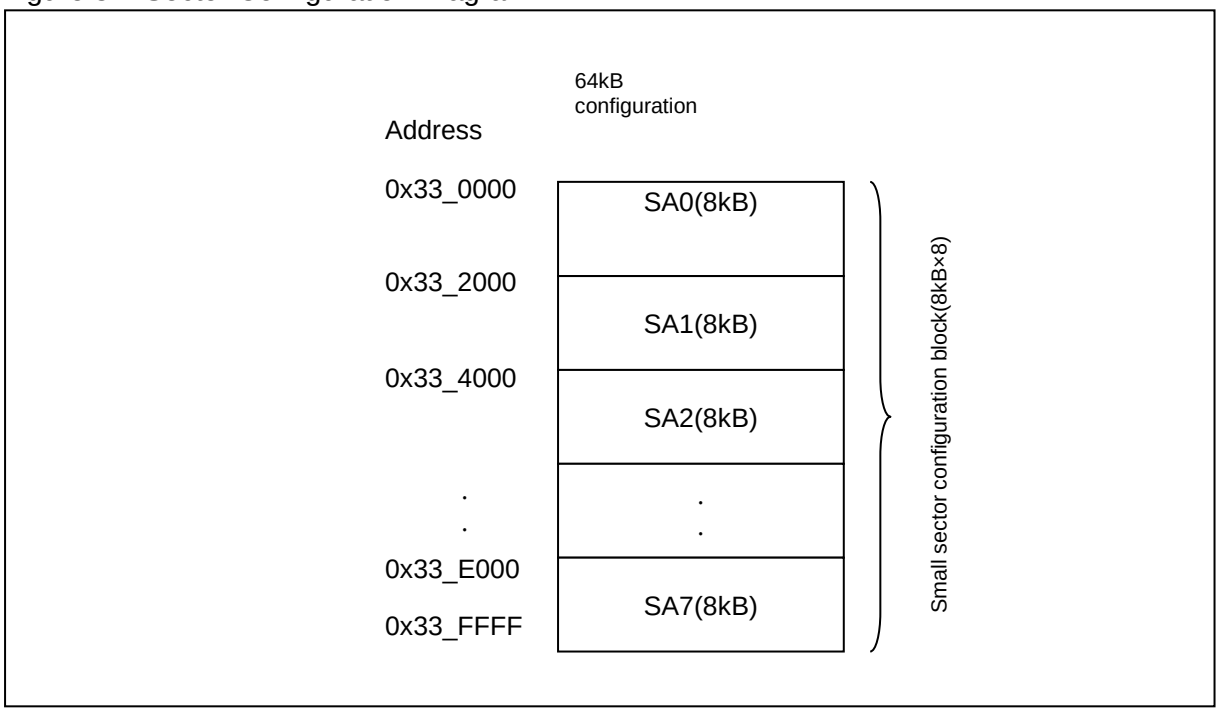
Figure 3-1 Block Diagram (64KB Products)



3.2. Sector Configuration Diagram

The sector configuration diagram of the WorkFlash memory is shown below.

Figure 3-2 Sector Configuration Diagram



4. Registers

This section explains registers of the WorkFlash memory.

Table 4-1 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

4.1. WorkFlash Control Register : DFCTLR (WorkFlash ConTroL Register)

The bit configuration of the WorkFlash control register is shown below.

This register configures the access control to the WorkFlash.

■ DFCTLR: Address 2300_H (Access : Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	FWE	Reserved					
Initial value	-	0	-	-	-	-	-	-
Attribute	RX,WX	R/W	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	-	-	-	-	-	-	-	-
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX

[bit15] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit14] FWE (Flash Write Enable) : Flash write enable

This bit is a control bit to enable write to the WorkFlash in the CPU mode.

If this bit is set, the ECC error detection and data correcting function will be disabled for data fetching to the WorkFlash memory.

FWE	Description
0	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

4.2. WorkFlash Status Register : DFSTR (WorkFlash STatus Register)

The bit configuration of the WorkFlash status register is shown below.

This register indicates the WorkFlash status.

■ DFSTR: Address 2303_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					DFECCERR	DFHANG	DFRDY
Initial value	-	-	-	-	-	0	0	1
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R/W	R,WX	R,WX

[bit7 to bit3] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] DFECCERR (WorkFlash ECC Error coRRection) : Data read ECC correction occurred

This bit indicates that ECC error occurs when reading data of WorkFlash in the CPU mode. This bit is cleared by writing "0". Writing "0" is prioritized when ECC error and writing "0" occur concurrently.

DFECCERR	Read	Write
0	An error correction by ECC has not occurred during data read (initial value)	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 "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.

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 2308_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0
Initial value	-	-	-	0	-	0	0	0
Attribute	RX,WX	RX,WX	RX,WX	R/W	RX,WX	R/W0	R/W	R/W

[bit7 to bit5] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit4] DFWDSBL (Data Fetch Wait cycle DiSaBLe) : Data fetch wait cycle disabled

If this bit is set to "1", the wait cycle inserted when setting wait at data fetch is disabled. However, you cannot disable the wait cycle to guarantee the cycle time.

DFWDSBL	Description
0	Wait cycle enabled (Initial value)
1	Wait cycle disabled

Note:

When this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.

[bit3] Reserved
This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved
This bit is reserved. 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 is performed 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 is performed to program flash memory in the CPU mode.

ECCDSBL0	Description
0	ECC function enabled (Initial value)
1	ECC function disabled

5. Operation

The section explains the operation of the WorkFlash memory.

This section explains the method for accessing the flash area.

5.1 Access Mode Setting

5.2 Writing Flash Memory by CPU

5.3 Automatic Algorithm

5.4 Reset Command

5.5 Write Command

5.6 Chip Erase Command

5.7 Sector Erase Command

5.8 Sector Erase Suspend Command

5.9 Security Function

5.10 Notes on Using WorkFlash Memory

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.)

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 reset is released, the mode will be the CPU-ROM mode.

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.

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 "5.5 Write Command" for details on the write command.

(3) Read the hardware sequence flag until the write has finished.
See "5.3.2 Automatic Algorithm Execution State" for details on hardware sequence flag read.

(4) Issue the write command. Write address = PA+2, write data = PD[15:0]
At this time, the hardware automatically calculates the ECC codes by combining with PD[31:16] from (2), and writing of ECC codes is also performed automatically at the same time.

(5) Read the hardware sequence flag until the write has finished.

(6) If there is more data to write, return to (2). Continue to (7) when all writes have finished.

(7) Set CPU-Rom mode.

(8) Read the value which has already been written, and check that the correct value can be read. Even if the correct value can be read, check the DFSTR:DFECCERR bit to make sure that there was no ECC correction. If ECC correction occurs, follow the same procedure again starting 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

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.

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 5-1 Command Sequence

Command	Number of writing	1st time		2nd time		3rd time		4th time		5th time		6th time	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	arbitrary	$F0_H$										
Read	1	RA	RD										
Write	4	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	$A0_H$	PA	PD				
Chip erase	6	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	80_H	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	10_H
Sector erase	6	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	80_H	$AA8_H$	AA_H	554_H	55_H	SA	30_H
Sector erase suspend	1	arbitrary	$B0_H$										
Sector erase resume	1	arbitrary	30_H										

* The data written in the table only shows the lower 8-bit. The upper 8-bit can be any value. The commands must be written as bytes or half-words.
* The addresses written in the table only show the lower 12-bit. Set the upper 20-bit to any address within the address range of the target flash macro.

PA: Write address (half-word alignment)
PD: Write data (Write as 16-bit.)
SA: Sector address (specify an arbitrary address within the address range of the sector to erase.)
RA: Read address
RD: Read data (the read width is arbitrary.)

Notes:

- When the wrong address value and data value are written or writing is performed in the wrong sequence, commands that have been written are cleared.
- Do the following to the LSB 2-bit for the command address, and for the sector address (SA) issued at the generation of the sector erase command.
  - When half-word access : 2'b00
  - When byte access : 2’b01 or 2’b11

  Example 1:
  During byte access, if command address = (LSB 2-bit of the standard command address changed to 2’b01.)
  $yAA8_H \rightarrow yAA9_H$, $x554_H \rightarrow x555_H$, 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:
  During byte access, if command address = (LSB 2-bit of the standard command address changed to 2’b11.)
  $yAA8_H \rightarrow yAAB_H$, $x554_H \rightarrow x557_H$, 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)

■ 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 "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 "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 "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 time-out 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 "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 "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 "5.8 Sector Erase Suspend Command" for actual operation.

Notes:

- 16.7µs +2cyc 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.
- At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.
- 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.

5.3.2. Automatic Algorithm Execution State

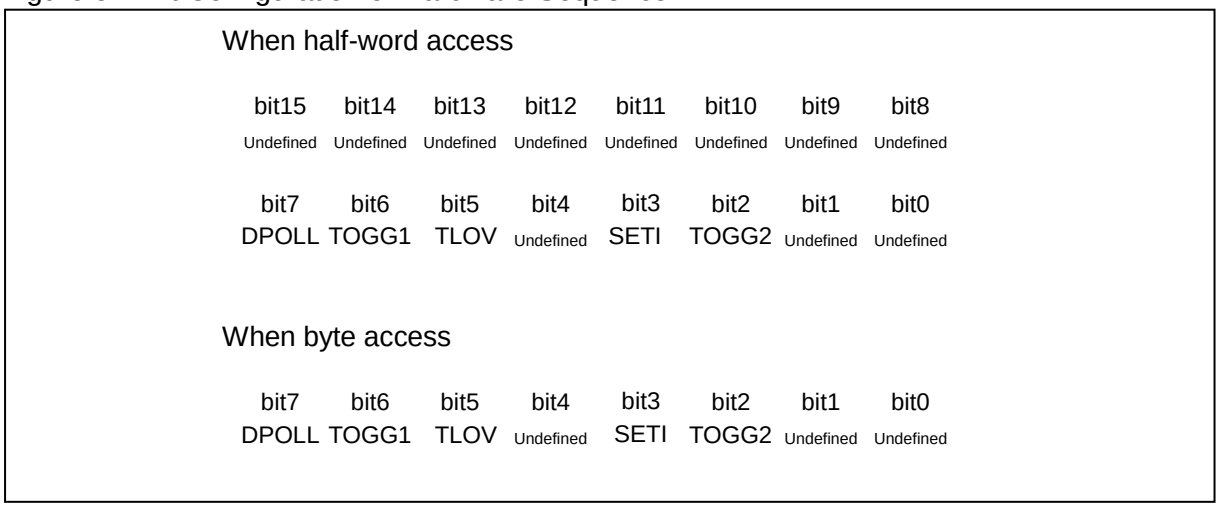
The automatic algorithm execution state is shown below.

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 5-1 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 5-2 Correspondence between Flags and Flash Memory status

State		DPOLL	TOGG1	TLOV	SETI	TOGG2
Run	Writing	Inverted data *	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data *	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

*: See "● Bit descriptions" for the values that are read out.

● Bit descriptions

[bit15 to bit8] Undefined bits

[bit7] DPOLL (Data polling flag bit)

When the hardware sequence flag is read by specifying the write/erase target address, this bit indicates whether or not the automatic algorithm is running using a data polling function.
The value that is read differs depending on the state.

1. When writing

During execution of writing	Reads out the opposite value (inverted data) of the value of bit7 of the last data to be written. The address specified for reading the hardware sequence flag is not accessed.
After writing finished	Reads out the value of bit7 of the address specified for reading the hardware sequence flag.

2. During sector erase

During execution of sector erase	Reads "0" from the sector being erased.
After sector erase	This bit always reads out as "1".

3. During chip erase

During execution of chip erase	This bit always reads out as "0".
After chip erase	This bit always reads out as "1".

4. During sector erase suspend

State of suspend (incomplete end)	"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 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. 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

5.4. Reset Command

The reset command is shown below.

The flash memory can be reset by sending reset commands to the target flash memory. Because this state is the flash memory initial state, the flash memory always returns to the reset state when the power is turned on or a command finishes successfully. When the power is turned on, there is no need to issue a reset command. Furthermore, in reset state, data can be read using normal read access and programs can be accessed by the CPU; thus there is no need to issue a reset command when reading data.

5.5. Write Command

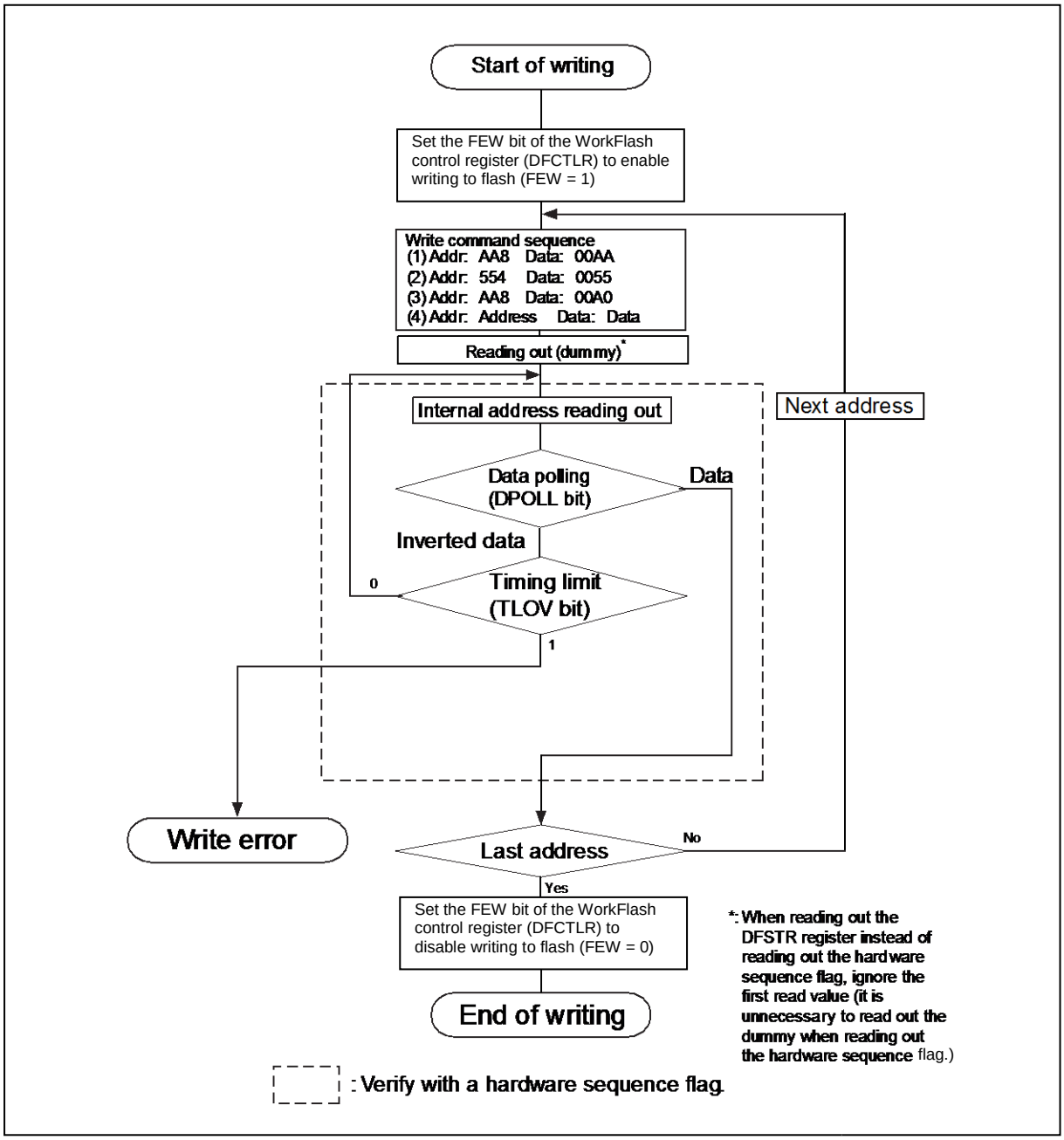
The write command is shown below.

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 5-2 Example of Write Procedure



Notes:

- If write completes, the write address is not accepted because the flash memory returns to the read mode.
- See "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 "5.2 Writing Flash Memory by CPU" for procedure.

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 "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 "5.9.3 Unlocking Flash Security" for details.

5.7. Sector Erase Command

The sector erase command is shown below.

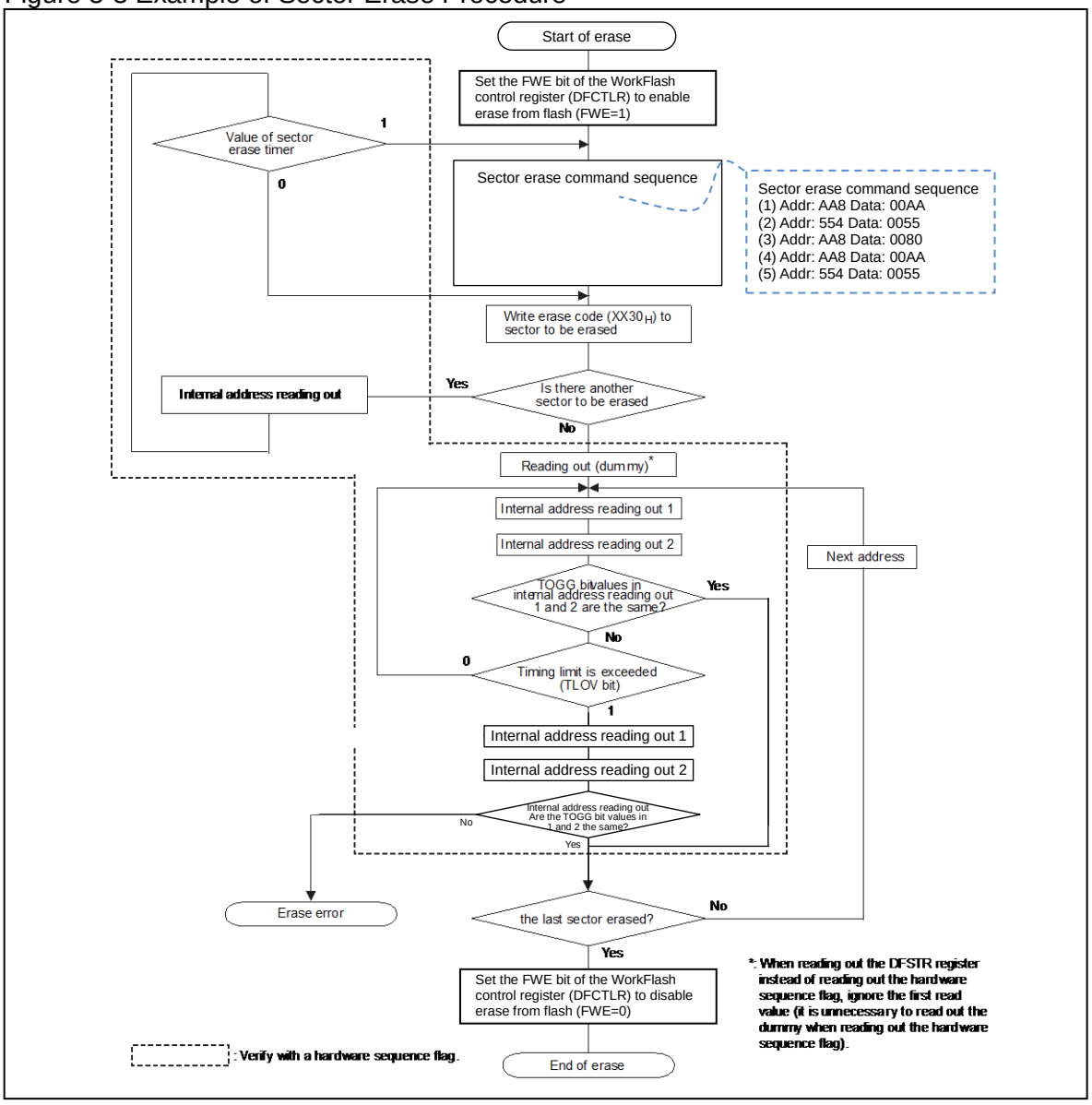
A sector in the flash memory can be selected and only data in the selected sector can be deleted. Multiple sectors can also be specified at the same time. Sector erase is performed in the following order.

1. Send sector erase commands sequentially to the target sector
   Once 40µs 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 time-out 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 5-3 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 "5.3 Automatic Algorithm" for details on the sector erase command.
- Because the value of the DPOLL bit of the hardware sequence flag changes concurrently with the TLOV bit, check this bit again even if the value of the TLOV bit is "1".
- When the TOGG1 bit and the TLOV bit of the hardware sequence flag change to "1", toggle operation stops at the same time. Therefore, it is necessary to confirm the TOGG1 bit again even if the TLOV bit is "1".
- Only a command to add an erased sector and a command to suspend erasing are write commands acceptable in the command time-out state, while only a command to suspend erasing is an acceptable command in the sector erasing state.
- 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.

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:

- 16.7µs + 2cyc 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.
- At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.

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 5-1 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.

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 "5.9.4 Flash Access Restrictions When Security is ON" for details of the restrictions.

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.

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 3-2 Sector Configuration in "CHAPTER : FLASH MEMORY"), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.

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. Execute chip erase to each flash macro in the above order from the viewpoint of the data protection stored in the flash macro.

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 5-3 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 "5.9.3 Unlocking Flash Security".

5.10. Notes on Using WorkFlash Memory

Notes on using the workflash memory are shown below.

- If this device is reset during a write, the data that was written cannot be guaranteed.
- If CPU programming mode is set (FWE=1) using the FWE bit of the WorkFlash control register (DFCTLR), do not execute the program in workflash 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 workflash memory, do not generate interrupt requests. The program runs out of control without fetching the correct values.
- Because this model has the ECC bit added, data always needs to be written as 32-bit by two 16-bit writes. See "5.2 Writing Flash Memory by CPU" for procedure.
- Do not issue commands to multiple macros simultaneously (i.e., in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- If authentication by password of on-chip debugger (OCD) completes, you can read the content of work 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 workflash memory, the data superscription to the address where some values have already been written cannot be done.

Chapter 47: On-Chip Debugger : OCD

This chapter explains the on-chip debugger (OCD).

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FR81S10_OCD-1v2-91520-2-E

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.

2. Features

This section explains features of the on-chip debugger (OCD).

- **One-wire debug tool I/F**
  - Initial communication type: Manchester code
  - MDI wake up and standby function
  - Auto negotiation mode function
  - The clock calibration is unnecessary on the tool side.
- **Debug security function**
- **Debug mode control function**
- **Execution control function**
  - Status display functions (chip status, CPU status, etc.)
  - Debug command execution control function
  - Small-scale debug main memory (8bytes=4 instructions)
  - CPU register save register (PC/PS)
  - PC monitor function
  - Reset function
    - Chip reset (INT)
    - CPU reset (RST)
- **Semi-hosting function**
- **Break function**
  - Step execution break
  - Event trigger break
  - Forced break
  - Guarded access break
  - Trace end break
  - Control on interrupt acceptance immediately after the execution start address
- **Debug DMA function (DDMA function)**

  Support of transfer modes (address mode, verify mode, DEBUG I/F burst transfer)

- **Event function**
  - Code event: 8
  - Conditional code event: 2
  - Data event: 8
  - Interrupt event: 2
  - User event: 2
  - Event sequencer: 2 levels + reset
- **Execution time measurement timer function**
  - Go-Break measurement
  - Inter-trigger measurement (single measurement/cumulative measurement, MIN/MAX value measurement when accumulation is measured)
- **Trace function**
  - Special state trace
  - Branch trace
  - Data trace
  - Trace delay
  - Number of trace frames: 512

3. Configuration

This section shows the configuration of the on-chip debugger (OCD).

Figure 3-1 Block Diagram of OCDU

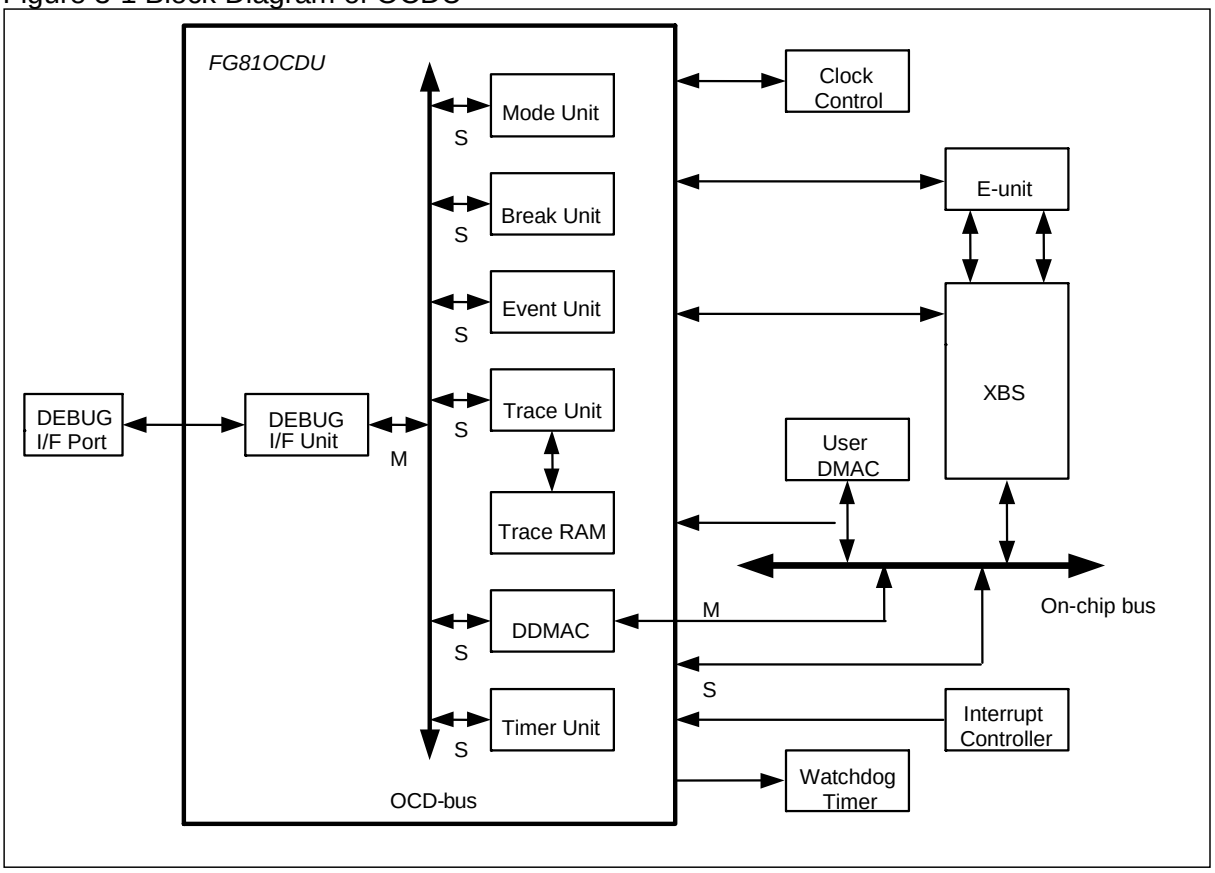
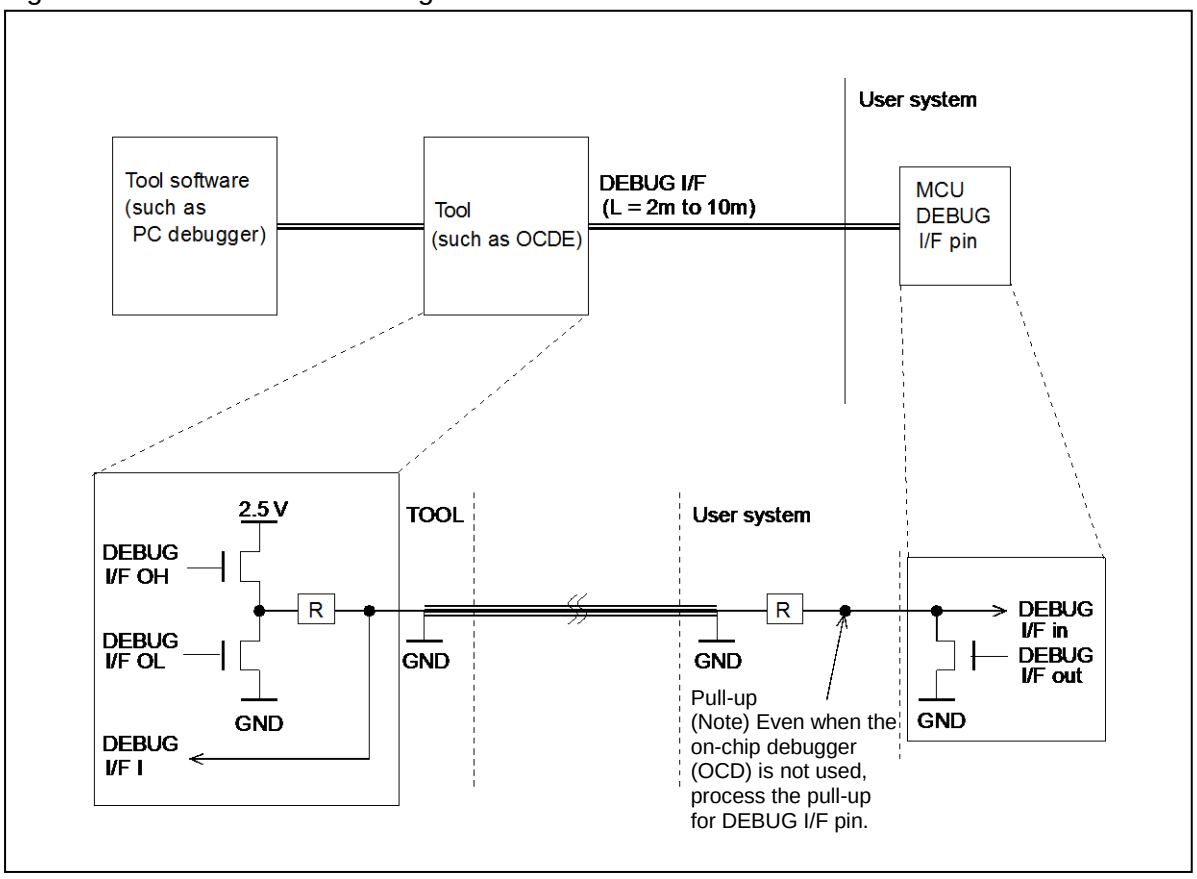


Figure 3-2 OCD Connection Diagram



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.

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.

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.

4. Registers

This section explains the registers of the on-chip debugger (OCD).

4.1 DBG Register

4.2 User IO Register

4.1. DBG Register

The bit configuration of the DBG register is shown.

Table 4-1 Register Map (DBG Register)

Address	Register				Register function
	+0	+1	+2	+3	
0xFF00	DSUCR		Reserved		DSU control register

4.1.1. DSU Control Register : DSUCR

The DSU control register is shown below.

This register is used to control DSU in the free-run mode.

For details, contact our sales representative.

- **DSUCR: Address $FF00_H$ (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DSU
Initial value	X	X	X	X	X	X	X	0
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R,W

4.2. User IO Register

The bit configuration of the User IO register is shown.

Table 4-2 Register Map (User IO Register)

Address	Register				Register function
	+0	+1	+2	+3	
0x0BF0	HSCFR				High-speed communication frequency register
0x0BF8	Reserved		MBR		Massage buffer
0x0BFC	Reserved		UER		User event register

4.2.1. User Event Register : UER

The user event register is shown below.

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

4.2.2. High-Speed Communication Frequency Register : HSCFR

The high-speed communication frequency register is shown below.

This register is used to set frequency information on PLL clock used.

For details, contact their representatives.

- **HSCFR: Address $0BF0_H$ (Access: Byte, Half-word, Word)**

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved						FREQ[17:16]	
Initial value	X	X	X	X	X	X	0	0
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FREQ[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

4.2.3. Message BuffeR : MBR

The message buffer is shown below.

This register is used to control writing and requesting of the message for a semi-hosting.

- **MBR: Address $0BFA_H$ (Access: Byte, Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	DMAREQ	Reserved					
Initial value	0	0	X	X	X	X	X	X
Attribute	R,WX	R,W1	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MB[7:0]							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W

[bit15] BUSY (Message Buffer BUSY) : Message Buffer Busy

This bit detects the busy state of the message buffer. It becomes "1" by writing in MBR.MB[7:0] bits, and it becomes "0" by the message reading from the tool side. This bit is for reading only, and writing doesn't have the influence in operation.

BUSY	Message buffer busy state
0	Non-busy state (initial value)
1	Busy state

[bit14] DMAREQ (DDMA Message Handling REQuest) : DDMA message processing request bit

This bit requests the method of message handling that uses DDMA. Only "1" writing of this bit is effective, and "0" writing doesn't have the influence in operation.

DMAREQ	DDMA message processing
0	Doesn't request (initial value)
1	Requests

[bit7 to bit0] MB[7:0] (Message Buffer) : Message Buffer

These bits are for writing message data of one byte. These bits are only for writing. The reading value is undefined.

5. Operation

This section explains the operation of the on-chip debugger (OCD).

5.1 OCDU Operating Mode

5.2 Overview of DEBUG I/F

5.3 Specification Restrictions at Connection to OCD Tool of This Series

5.4 OCD-DSU ID Code and Mount Type Information on This Series

5.1. OCDU Operating Mode

OCDU operating mode is shown.

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

5.1.2. Operating Mode Status Transition

Operating mode status transition is shown.

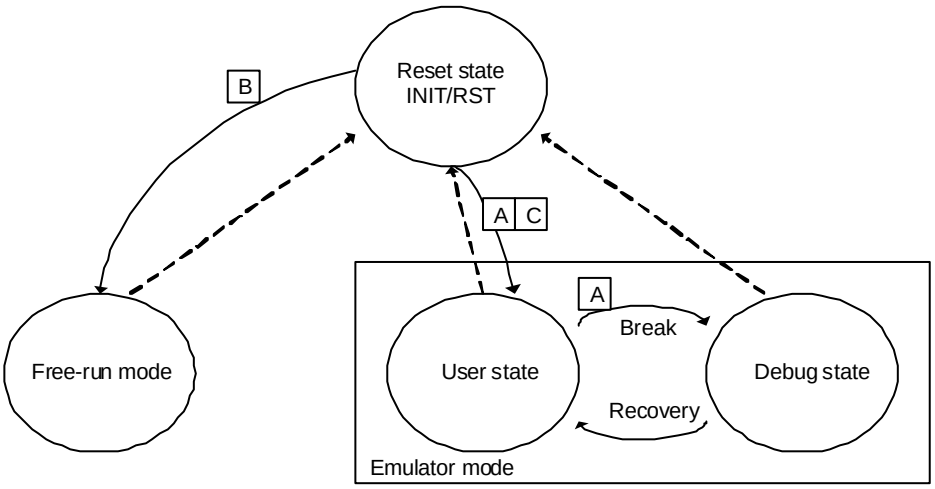
At INIT releasing (including RST accompanied by INIT), control transits to the debug state of the emulator mode or to the free-run mode according to the mode command from DEBUG I/F in the chip reset sequence.

At RST releasing (not accompanied by INIT), control transits to the operating mode before RST generation. However, if a forced break request is issued after RST occurs in the user state, control transits to the debug state of the emulator mode at RST releasing.

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 5-1 OCDU Operating Mode Transition Diagram



[Transition indicated by broken lines]
Transition for when a reset (INIT, RST) occurs. All states transit to the reset state.
[Transition indicated by solid lines]
A: (1) Transition for when a mode command is received successfully in the MDI chip reset sequence at INIT release after the reset for which INIT is involved occurs.
(2) Transition at RST release after the reset for which INIT is not involved occurs in the debug state.
(3) Transition at RST release when a forced break is requested before RST release after the reset without INIT occurs in the user state or pseudo on-the-fry break is requested.

B: (1) Transition for when a mode command is not received successfully in the MDI chip reset sequence at INIT release after the reset for which INT is involved occurs.
(2) Transition at RST release after the reset for which INIT is not involved occurs in the free-run mode.
C: Transition at RST release after the reset for which INIT is not involved occurs in the user state

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)
- Auto negotiation mode function (communication form notification)

The two-way pin of DEBUG I/F is accomplished by Nch open-drain output. The DEBUG I/F pin is pulled up on a user system. It is pulled up with a tool during tool connection.

For the tool connection, see Figure 3-2 OCD Connection Diagram.

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
- Idle phase
- Mode entry phase
- End phase

● Start phase

Start phase is the interval when the generated INIT is released until 32 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase.

● INIT notification phase

INIT notification phase is the interval when the start phase is ended until 568 sampling clock cycles of the normal UART is counted. OCDU outputs "L" of 280 cycles to DEBUG I/F twice while it is in this phase (The idle of eight cycles is inserted among), and notifies the tool the generation of INIT.

● Idle phase

Level sense phase is the interval when the INIT notification phase is ended until 256 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase.

● Mode entry phase

Mode entry phase is the interval when the Idle phase is ended until 256 sampling clock cycles of the normal UART is counted. OCDU starts the reception of the mode command from the tool in this phase.

When starting reception of the mode command is detected (start bit detected in the UART reception) in this phase, OCDU activates in the emulator mode (debug state). Then, if the normal mode command (no reception error and mode command match) is received, OCDU can receive the subsequent register access command after this. If the normal mode command (reception error and no mode command match) is not received, OCDU generates INIT request and executes the chip reset sequence again after INIT is released.

When starting reception of the mode command is not detected (start bit detected in the UART reception) in this phase, OCDU activates in the free-run mode.

If the mode command is received immediately after starting the mode entry phase, the mode command must be received after waiting for input of "H" to DEBUG I/F for more than one cycle width using the UART reception sampling clock. If this condition is not met, the start bit of the mode command reception cannot be detected normally, the mode may not be entered correctly.

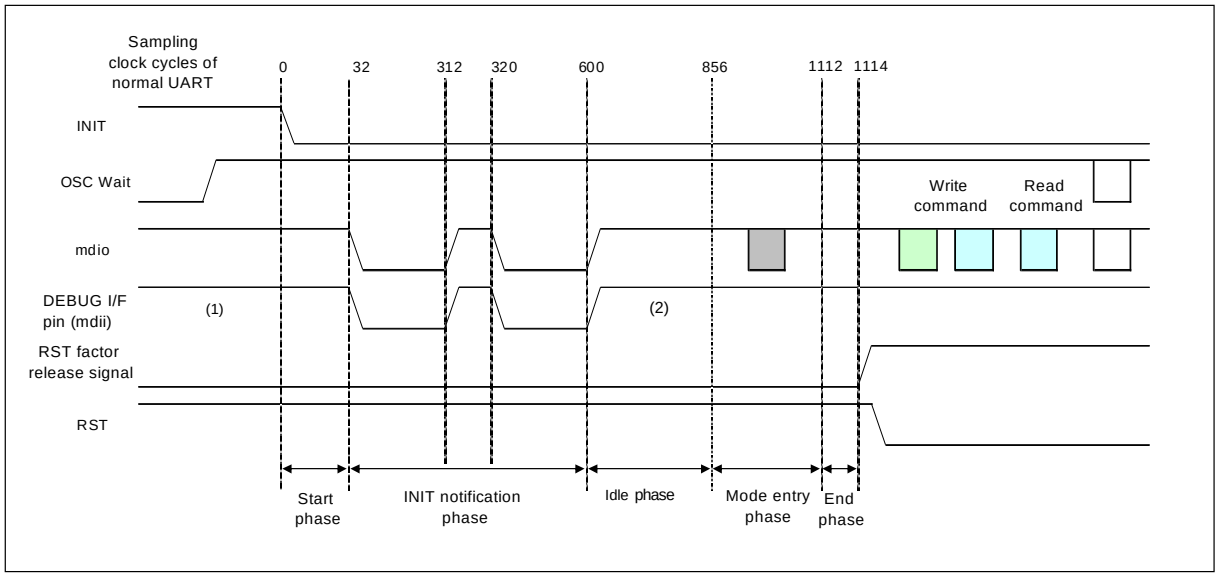
End phase

End phase is the interval when the mode entry phase is ended until 2 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase. OCDU executes the reset issuance sequence described in "5.4.3 Reset (RST)" of "CHAPTER : RESET" when the end phase is ended. The RST factor is released.

The relationship between the number of sampling clock cycles of the normal UART and the phase for the chip reset sequence is as follows.

Phase of chip reset sequence	Start phase	INIT notification phase	Idle phase	Mode entry phase	End phase
Sampling clock cycles of the normal UART from INIT release	1 to 32	33 to 600	601 to 856	857 to 1112	1113 to 1114

The following shows the chip reset sequence.



OSC Wait: Oscillation of main source oscillation clock is stabilized. INIT is released after the oscillation stabilization is confirmed.

(1): DEBUGIF is set to H level by pull-up processing of the tool.

(2): DEBUGIF becomes the level of pull-up processing on the user system.

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 dedicated OCD register for the number of times and specified length.

■ Security Information

The debug security area is allocated at 30 bytes of built-in flash start address+4 to +33. 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.

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:

5.3.1. Clock Setting

Clock setting is shown.

- When the device is connected to the OCD tool, the main clock does not stop oscillating. It is, however, possible to read from the register CMONR.MCRDY and write to the register CSELR.MCEN, similarly to the case when not connected.
- When the device is communicating via high-speed UART or phase modulation UART, PLL does not stop oscillating. It is, however, possible to read from the register CMONR.PCRDY and write to the register CSELR.PCEN, similarly to the case when not connected. In addition, since PLL continues to oscillate, changes in settings of the registers shown below will have no effect. It is, however, possible to read and write the registers shown below, similarly to the case when not connected.
  - PLLCR.ODS
  - PLLCR.PMS
  - PLLCR.PDS
  - CCPSDIVR.PODS
  - CCPLLFBR.IDIV

5.3.2. Standby Mode

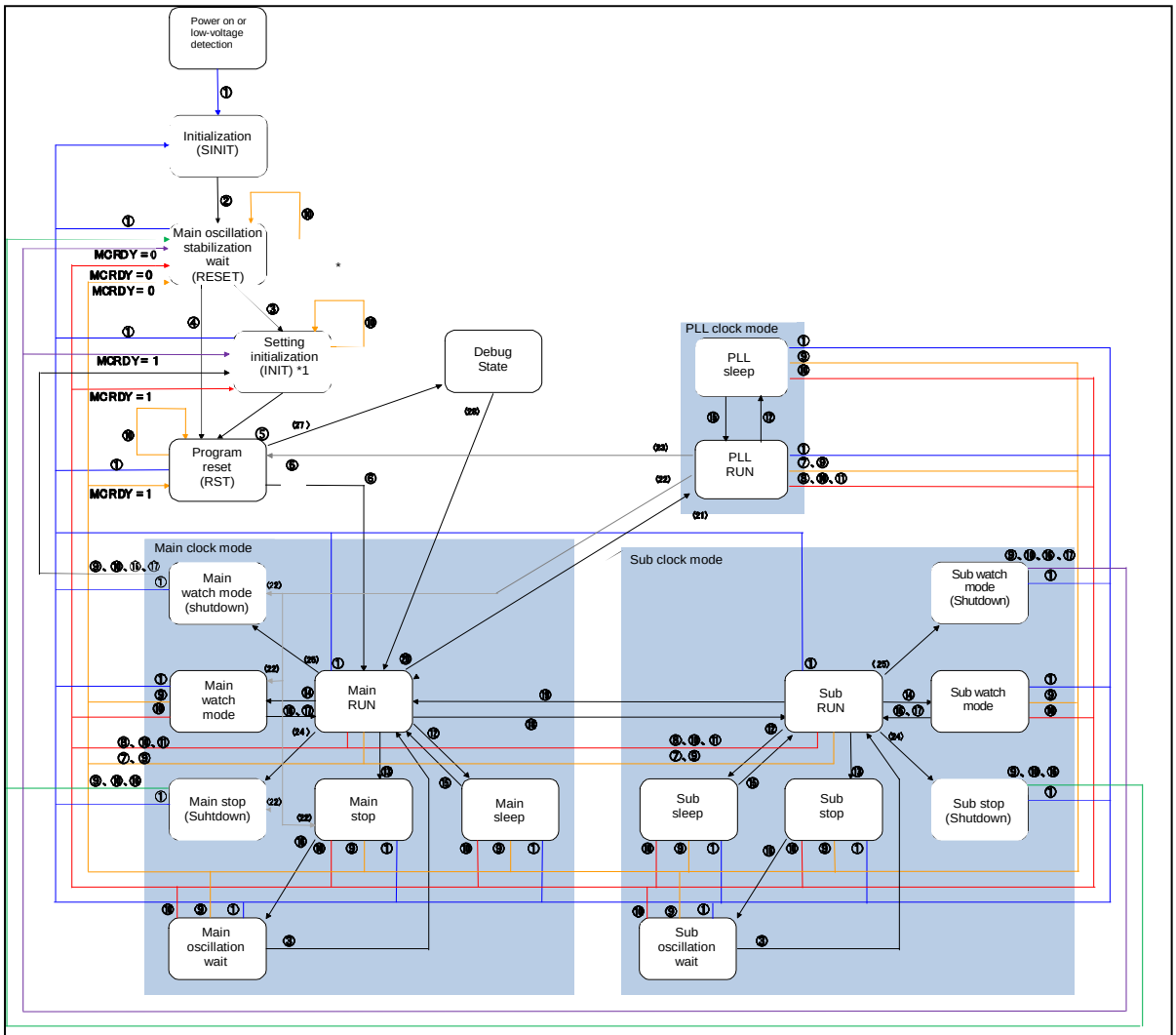
Standby mode is shown.

- Even if the watch mode is entered, PLL does not stop oscillating when OCD high-speed UART and phase modulation UART communication are enabled.
- The main clock does not stop oscillating even if the stop mode is entered. PLL does not stop oscillating when OCD high-speed UART and phase modulation UART communication are enabled. 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.)
  - 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.

5.3.3. Clock Reset State Transitions

Clock reset state transitions is shown.

Figure 5-2 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.

5.3.4. Summary of Specification Restrictions

Summary of specification restrictions is shown.

1) Communication mode (*1): Normal UART

Note:

When debugging from shutdown standby mode, be sure to keep the device unconnected to the OCD tool.

When starting to debug the device by connecting the OCD tool, the debug can be started by enabling OCDU operation by the MDI wakeup function. If, however, the device is being in stop mode (power-shutdown) or in watch mode (power-shutdown), it is impossible to start debugging. This is because, in both the modes, power supply to the OCDU is cut off and consequently the OCDU cannot receive a signal to enable its operation. Start debugging in other mode than the standby mode (power-shutdown).

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
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input (irregular)			
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input			No voltage step-down circuit switch stabilization wait time (*2) *: Only recovery from main/sub stop mode or main/sub watch mode
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input (+NMIX pin input) [MB91F52xxxD] RSTX pin input			Causes a transition to the emulator mode (debug state) after reset is released
Watchdog reset 0 (irregular)			
Watchdog reset 0			
Watchdog reset 1 (irregular)			
Watchdog reset 1			
External power supply low-voltage detection reset (irregular)			
External power supply low-voltage detection reset			No voltage step-down circuit switch stabilization wait time (*2) *: 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	
Low-voltage detection (internal power supply low-voltage detection) reset		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Flash security violation reset (irregular)			
Flash security violation reset		No	
Software reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Software reset		No	

Interrupt factor	Processing time difference from when the OCD tool is not connected	Remarks
All interrupts	Yes	No voltage step-down circuit switch stabilization wait time (*2) *: 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 (real-time clock, RTC/WDT1 calibration counter operation)
Main/sub watch mode		Voltage step-down circuit is fixed Main oscillation continues *: Sub watch mode

*1: For communication mode settings, see the "SOFTUNE Workbench Operating Manual".

*2: Voltage step-down circuit stabilization waits time: about 6µs

2) Communication mode (*1): High-speed UART/phase modulation UART

Reset factor	Difference from when the OCD tool is not connected		Remarks
	Initialization range	Processing time	
Power-on reset	No	Yes	Causes a transition to the emulator mode (debug state) after reset is released
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input (irregular)			
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input			No voltage step-down circuit switch stabilization wait time (*2) *: Only recovery from main/sub stop mode or main/sub watch mode
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] RSTX pin input (+NMIX pin input) [MB91F52xxxD] RSTX pin input			Causes a transition to the emulator mode (debug state) after reset is released
Watchdog reset 0 (irregular)			
Watchdog reset 0			
Watchdog reset 1 (irregular)			
Watchdog reset 1			
External power supply low-voltage detection reset (irregular)			
External power supply low-voltage detection reset			No voltage step-down circuit switch stabilization wait time (*2) *: Only recovery from main/sub stop mode or main/sub watch mode
Illegal standby mode transition detection reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Illegal standby mode transition detection reset		No	
Low-voltage detection (internal power supply low-voltage detection) reset		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Flash security violation reset (irregular)			
Flash security violation reset		No	
Software reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Software reset		No	

Interrupt factor	Processing time difference from when the OCD tool is not connected	Remarks
All interrupts	Yes	No voltage step-down circuit switch stabilization wait time (*2) *: 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, RTC/WDT1 calibration counter operation)
Main/sub watch mode		Voltage step-down circuit is fixed Main oscillation continues *: Watch mode Operation continues (CAN)

*1: For communication mode settings, see "SOFTUNE Workbench Operating Manual".

*2: Voltage step-down circuit stabilization waits time: about 6µs

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 5-1 OCD-DSU ID Code of This Series

ID name	bit width	Associated ID register name	Address in the OCD space	Value	Remarks
Manufacturer ID	16	E_IDMCR	0x000	0x0400	
CPU family ID	16	E_IDFCR	0x001	0x0200	FR81E/FR81S
DSU type ID	8	E_IDVCR	0x003	0x0X	
DSU version ID	4	E_IDVCR	0x003	0x1	
Device ID	16	E_IDDCR	0x002	0x0030	MB91F52xBW/Y/J/L
				0x0031	MB91F52xBS/U/H/K
				0x0032	MB91F52xDW/Y/J/L
				0x0033	MB91F52xDS/U/H/K
				0x0034	MB91F52xFW/Y/J/L
				0x0035	MB91F52xFS/U/H/K
				0x0036	MB91F52xJW/Y/J/L
				0x0037	MB91F52xJS/U/H/K
				0x0038	MB91F52xKW/Y/J/L
				0x0039	MB91F52xKS/U/H/K
				0x003a	MB91F52xLW/Y/J/L
				0x003b	MB91F52xLS/U/H/K
Device version ID	4	E_IDVCR	0x003	0x2 0x3 0x4 0x5	Revision : B Revision : C Revision : D Revision : E

Table 5-2 Mount Type Information of This Product Type

Product name	Number of code events	Number of data events	Data event (Compare)	Sequencer event	Trace
MB91F526 MB91F525 MB91F524 MB91F523 MB91F522	8	8	○	○	512 frames

Chapter 48: Waveform Generator

This chapter explains the waveform generator.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Notes

Code : FS20-1v1-91520-2-E

1. Overview

This section explains the overview of the waveform generator.

When you use the waveform generator together with PPG, you will be able to output six distinct waveforms from the 16-bit free-run timer. In addition, you will be able to measure the input pulse width and external clock cycle.

2. Features

This section explains the features of the waveform generator.

■ Function of the Waveform Generator

- The waveform generator consists of three 16-bit dead timer registers, three timer state control registers, 16-bit dead timer reload interrupt register, 16-bit dead timer minus control register, one 16-bit waveform control register, and PPG output control register.
- You will be able to generate real-time output, 16-bit PPG waveform output, non-overlap 3-phase waveform output (for inverter control), and DC chopper waveform output using the waveform generator.
- You will be able to generate non-overlap waveform output based on the dead time of the 16-bit dead timer (dead time timer function).
- In the 2-channel mode, you will be able to generate non-overlap waveform output by operating the real-time output (dead time timer function).
- When a real-time output compare match is detected, GATE signal will be generated. This signal starts or stops the PPG timer operation (GATE function).
- When a real-time output compare match is detected, the 16-bit dead timer becomes active. You will be able to start or stop the PPG timer easily by generating GATE signal that controls the PPG operation (GATE function).

■ Function of DTTI

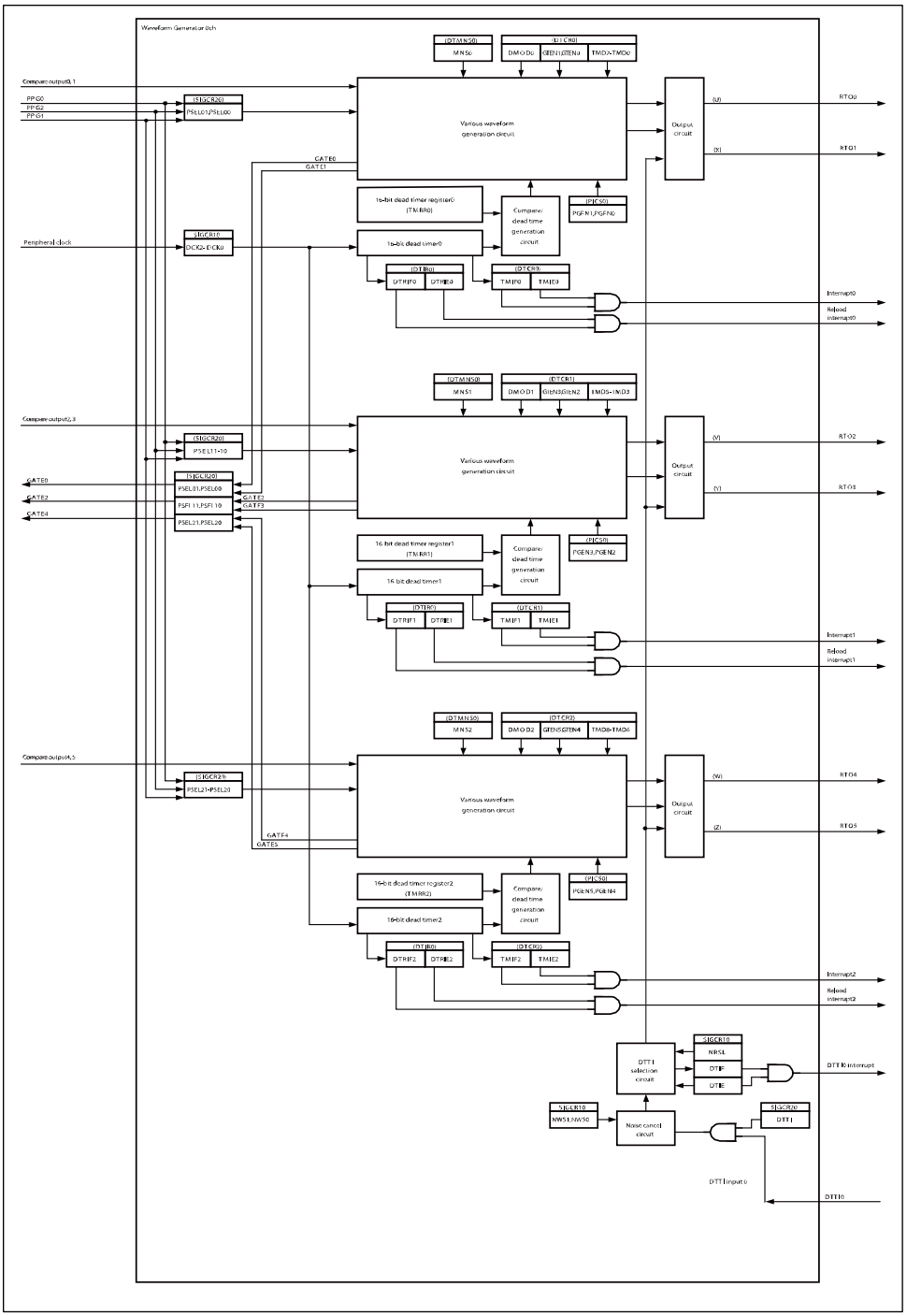
- By using the DTTI pin, you can forcibly stop the waveform generator output.
- By using the DTTI register, you can forcibly stop the waveform generator output.
- You can forcibly disable the waveform generator output, and can handle it as a general-purpose port, regardless of the setting of the extended port function registers (EPFR).

3. Configuration

This section explains the configuration of the waveform generator.

■ Configuration of the Waveform Generator

Figure 3-1 Configuration of the Waveform Generator



4. Registers

This section explains the register of the waveform generator.

■ Table of External Pins

Table 4-1 Table of External Pins

Channel	External pin			
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ, MB91F52xK, MB91F52xL
DTTI	DTTI_0/ DTTI_1	DTTI_0/ DTTI_1	DTTI_0/ DTTI_1	DTTI_0/ DTTI_1/ DTTI_2
0	RTO0_1	RTO0_1	RTO0_1	RTO0_0/ RTO0_1
1	RTO1_1	RTO1_1	RTO1_1	RTO1_0/ RTO1_1
2	RTO2_1	RTO2_0/ RTO2_1	RTO2_0/ RTO2_1	RTO2_0/ RTO2_1
3	RTO3_0/ RTO3_1	RTO3_0/ RTO3_1	RTO3_0/ RTO3_1	RTO3_0/ RTO3_1
4	RTO4_0/ RTO4_1	RTO4_0/ RTO4_1	RTO4_0/ RTO4_1	RTO4_0/ RTO4_1
5	RTO5_1	RTO5_1	RTO5_0/ RTO5_1	RTO5_0/ RTO5_1

■ List of Waveform Generator Register

Table 4-2 List of Waveform Generator Register

Address	+0	+1	+2	+3
0x12A0	16-bit dead timer register 0 (TMRR0)		16-bit dead timer register 1 (TMRR1)	
0x12A4	16-bit dead timer register 2 (TMRR2)		Reserved	Reserved
0x12A8	16-bit dead timer state control register 0 (DTSCR0)	16-bit dead timer state control register 1 (DTSCR1)	16-bit dead timer state control register 2 (DTSCR2)	Reserved
0x12AC	Reserved	16-bit dead timer reload interrupt register 0 (DTIR0)	Reserved	16-bit dead timer minus control register 0 (DTMNS0)
0x12B0	Reserved	Waveform control register 10 (SIGCR10)	Reserved	Waveform control register 20 (SIGCR20)
0x12B4	PPG output control register 0 (PICS0)	-	-	-

4.1. Registers for the Waveform Generator

This section explains the register of the waveform generator.

The waveform generator consists of 16-bit dead timer register, 16-bit dead timer state control register, 16-bit dead timer reload interrupt register, 16-bit dead timer minus control register, waveform control register, and PPG output control register.

4.1.1. 16-bit dead TIMeR Register (TMRR)

The bit configuration for the 16-bit dead timer register is shown below.

The 16-bit dead timer register (TMRR) holds the reload value of the 16-bit dead timer.

- **TMRR0: Address $12A0_H$ (Access: Half-word, Word)**
- **TMRR1: Address $12A2_H$ (Access: Half-word, Word)**
- **TMRR2: Address $12A4_H$ (Access: Half-word, Word)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00
Initial values	0	0	0	0	0	0	0	1
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TR15 to TR00: 16-bit Dead timer reload value bits

TR15 to TR00	Function
	16-bit dead timer reload value

- These bits are used to store reload value of the 16-bit dead timer.
- The value of these registers will be reloaded when the 16-bit dead timer starts operating.
- If the value is rewritten to these registers while the timer is active, this new value will become valid when the timer starts/operates next time.
- In the dead time timer mode, these registers are used to configure non-overlap time.
  Non-overlap time = (set value) × selected clock
- In the timer mode, these registers are used to configure GATE time for PPG timer operation.
  GATE time = (set value) × selected clock

Note:

When accessing these registers, use a half-word or word access instruction.

Note:

Do not set "0000_H" to these registers.

4.1.2. 16-bit Dead Timer State Control Register (DTSCR)

The bit configuration for the 16-bit dead timer state control register is shown below.

The 16-bit dead timer state control register (DTSCR) is used to control operation mode, interrupt request enable, interrupt request flag, GATE signal enable, and output level polarity of the waveform generator.

■ DTSCR0: Address $12A8_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMOD0	GTEN1	GTEN0	TMIF0	TMIE0	TMD2	TMD1	TMD0
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R(RM1), W	R/W	R/W	R/W	R/W

[bit7] DMOD0: Output polarity control bit

DMOD0	Function
0	Normal polarity output
1	Inverted polarity output

- This bit is used to configure U/V/W output in the dead time timer mode.
- When this bit is set, output polarity of U/V/W will be inverted.

Note:

This bit does not mean anything if the dead time timer mode is not selected. (TMD2: bit26=0)

[bit6] GTEN1: GATE signal control bit1

GTEN1	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.

- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20.

[bit5] GTEN0: GATE signal control bit0

GTEN0	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.
- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20.

[bit4] TMIF0: Interrupt request flag bit

TMIF0	Function	
	Read	Write
0	No counter underflow has been detected.	This bit is cleared.
1	A counter underflow has been detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the 16-bit dead timer.
- This bit is set to "1" when an underflow occurs at the 16-bit dead timer.
- When this bit is set to "0", this bit is cleared. Writing "1" to this bit does not have any effects on this bit.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.

This bit functions only when the TMD2 to TMD0: bit26 to bit24 are "000_B" or "001_B". Otherwise, the bit is always "0".

If a software clear (write of "0") or an interrupt clear signal ("H") and a hardware set (an underflow occurs at the 16-bit dead timer 0) occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit3] TMIE0: Interrupt request enable bit, software trigger bit

TMIE0	Function
0	An interrupt will not be generated even when an underflow occurs at the 16-bit dead timer.
1	An interrupt will be generated when an underflow occurs at the 16-bit dead timer.

- This bit is used as a software trigger bit and an interrupt enable bit for the 16-bit dead timer.
- When TMD2 to TMD0: bit26 to bit24 are "000_B" or "001_B", this bit will be used as a software trigger for the 16-bit dead timer. When this bit is changed from "0" to "1", it will function as a trigger for the 16-bit dead timer and reload the value to start the down count.
- When this bit is set to "1" and the interrupt request flag bit (TMIF0: bit28) is "1", an interrupt request will be transmitted to the CPU.

Note:

If you have the 16-bit dead timer triggered again, make sure to write "0" to this bit before writing "1" to it.

[bit2 to bit0] TMD2 to TMD0: Operation mode bits

TMD2	TMD1	TMD0	Function
0	0	0	OUT signal will be output.
0	0	1	If PPG output is disabled, OUT signal will be output. If PPG output is enabled, PPG pulse will be output while the OUT signal is "H".
0	1	0	The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer. If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)
1	0	0	Non-overlap signal will be generated at OUT signal. (Dead time timer mode)
1	1	1	Setting prohibited
Others			Setting prohibited

- These bits are used to select the operation mode for the waveform generator.
- If the TMD2 to TMD0: bit26 to bit24 are "000_B", the compare output 0 and 1 signals of the output compare will be output from RTO0 and RTO1 respectively. The 16-bit dead timer can also be used as a reload timer.
- If the TMD2 to TMD0: bit26 to bit24 are "001_B", the compare output 0 and 1 signals of the output compare will be output from RTO0 and RTO1 respectively when PPG output is disabled (PGEN0:bit26=0, PGEN1:bit27=0 of the PPG output control register (PICS0)). When PPG output is enabled (PGEN0:bit26=1, PGEN1:bit27=1 of PPG output control register (PICS0)), PPG pulse will be output from RTO0 and RTO1 respectively while the compare output 0 and 1 signals of the output compare are "H". The 16-bit dead timer can also be used as a reload timer.
- The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer when TMD2 to TMD0: bit26 to bit24 are "010_B". If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)
- When TMD2 to TMD0: bit26 to bit24 are "100_B", non-overlap signal will be generated at OUT signal. (Dead time timer mode)

Note:

To operate the waveform generator in the dead time timer mode, be sure to select the 2-channel mode (compare control register (OCS01); CMOD: bit28=1) for the compare output 1 of the output compare.

■ DTSCR1: Address $12A9_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMOD1	GTEN3	GTEN2	TMIF1	TMIE1	TMD5	TMD4	TMD3
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R(RM1), W	R/W	R/W	R/W	R/W

[bit7] DMOD1: Output polarity control bit

DMOD1	Function
0	Normal polarity output
1	Inverted polarity output

- This bit is used to configure U/V/W output in the dead time timer mode.
- When this bit is set, output polarity of U/V/W will be inverted.

Note:

This bit does not mean anything if the dead time timer mode is not selected. (TMD5: bit18=0)

[bit6] GTEN3: GATE signal control bit3

GTEN3	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.
- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20.

[bit5] GTEN2: GATE signal control bit2

GTEN2	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.
- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20.

[bit4] TMIF1: Interrupt request flag bit

TMIF1	Function	
	Read	Write
0	No counter underflow has been detected.	This bit is cleared.
1	A counter underflow has been detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the 16-bit dead timer.
- This bit is set to "1" when an underflow occurs at the 16-bit dead timer.
- When this bit is set to "0", this bit is cleared. Writing "1" to this bit does not have any effects on this bit.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
This bit functions only when the TMD5 to TMD3: bit18 to bit16 are "000_B" or "001_B". Otherwise, the bit is always "0".
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set (an underflow occurs at the 16-bit dead timer 1) occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit3] TMIE1: Interrupt request enable bit, software trigger bit

TMIE1	Function
0	An interrupt will not be generated even when an underflow occurs at the 16-bit dead timer.
1	An interrupt will be generated when an underflow occurs at the 16-bit dead timer.

- This bit is used as a software trigger bit and an interrupt enable bit for the 16-bit dead timer.
- When TMD5 to TMD3: bit18 to bit16 are "000_B" or "001_B", this bit will be used as a software trigger for the 16-bit dead timer. When this bit is changed from "0" to "1", it will function as a trigger for the 16-bit dead timer and reload the value to start the down count.
- When this bit is set to "1" and the interrupt request flag bit (TMIF1: bit20) is "1", an interrupt request will be transmitted to the CPU.

Note:

If you have the 16-bit dead timer triggered again, make sure to write "0" to this bit before writing "1" to it.

[bit2 to bit0] TMD5 to TMD3 : Operation mode bits

TMD5	TMD4	TMD3	Function
0	0	0	OUT signal will be output.
0	0	1	If PPG output is disabled, OUT signal will be output. If PPG output is enabled, PPG pulse will be output while the OUT signal is "H".
0	1	0	The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer. If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)
1	0	0	Non-overlap signal will be generated at OUT signal. (Dead time timer mode)
1	1	1	Setting prohibited
Others			Setting prohibited

- These bits are used to select the operation mode for the waveform generator.
- If the TMD5 to TMD3: bit18 to bit16 are "000_B", the compare output 2 and 3 signals of the output compare will be output from RTO2 and RTO3 respectively. The 16-bit dead timer can also be used as a reload timer.
- If the TMD5 to TMD3: bit18 to bit16 are "001_B", the compare output 2 and 3 signals of the output compare will be output from RTO2 and RTO3 respectively when PPG output is disabled (PGEN2:bit28=0, PGEN3:bit29=0 of the PPG output control register (PICS0)). When PPG output is enabled (PGEN2:bit28=1, PGEN3:bit29=1 of PPG output control register (PICS0)), PPG pulse will be output from RTO2 and RTO3 respectively while the compare output 2 and 3 signals of the output compare are "H". The 16-bit dead timer can also be used as a reload timer.
- The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer when TMD5 to TMD3: bit18 to bit16 are "010_B". If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)
- When TMD5 to TMD3: bit18 to bit16 are "100_B", non-overlap signal will be generated at the OUT signal. (Dead time timer mode)

Note:

To operate the waveform generator in the dead time timer mode, be sure to select the 2-channel mode (compare control register (OCS23); CMOD: bit28=1) for the compare output 3 of the output compare.

■ DTSCR2: Address $12AA_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMOD2	GTEN5	GTEN4	TMIF2	TMIE2	TMD8	TMD7	TMD6
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R(RM1), W	R/W	R/W	R/W	R/W

[bit7] DMOD2: Output polarity control bit

DMOD2	Function
0	Normal polarity output
1	Inverted polarity output

- This bit is used to configure U/VW output in the dead time timer mode.
- When this bit is set, output polarity of U/VW will be inverted.

Note:

This bit does not mean anything if the dead time timer mode is not selected. (TMD8: bit10=0)

[bit6] GTEN5: GATE signal control bit5

GTEN5	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.
- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20.

[bit5] GTEN4: GATE signal control bit4

GTEN4	Function
0	GATE signal will not be controlled by a compare output of the output compare. (asynchronous mode)
1	GATE signal will be controlled by a compare output of the output compare. (synchronous mode)

- This bit is used to control the PPG timer's GATE signal output for the compare output of the output compare.
- If it is set to 0, GATE signal will not be output.
- If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20.

[bit4] TMIF2: Interrupt request flag bit

TMIF2	Function	
	Read	Write
0	No counter underflow has been detected.	This bit is cleared.
1	A counter underflow has been detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the 16-bit dead timer.
- This bit is set to "1" when an underflow occurs at the 16-bit dead timer.
- When this bit is set to "0", this bit is cleared. Writing "1" to this bit does not have any effects on this bit.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
This bit functions only when the TMD8 to TMD6: bit10 to bit8 are "000_B" or "001_B". Otherwise, the bit is always "0".
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set (an underflow occurs at the 16-bit dead timer 2) occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit3] TMIE2: Interrupt request enable bit, software trigger bit

TMIE2	Function
0	An interrupt will not be generated even when an underflow occurs at the 16-bit dead timer.
1	An interrupt will be generated when an underflow occurs at the 16-bit dead timer.

- This bit is used as a software trigger bit and an interrupt enable bit for the 16-bit dead timer.
- When TMD8 to TMD6: bit10 to bit8 are "000_B" or "001_B", this bit will be used as a software trigger for the 16-bit dead timer. When this bit is changed from "0" to "1", it will function as a trigger for the 16-bit dead timer and reload the value to start the down count.
- When this bit is set to "1" and the interrupt request flag bit (TMIF2: bit12) is "1", an interrupt request will be transmitted to the CPU.

Note:

If you have the 16-bit dead timer triggered again, make sure to write "0" to this bit before writing "1" to it.

[bit2 to bit0] TMD8 to TMD6: Operation mode bits

TMD8	TMD7	TMD6	Function
0	0	0	OUT signal will be output.
0	0	1	If PPG output is disabled, OUT signal will be output. If PPG output is enabled, PPG pulse will be output while the OUT signal is "H".
0	1	0	The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer. If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)

TMD8	TMD7	TMD6	Function
1	0	0	Non-overlap signal will be generated at OUT signal. (Dead time timer mode)
1	1	1	Setting prohibited
Others			Setting prohibited

- These bits are used to select the operation mode for the waveform generator.
- If the TMD8 to TMD6: bit10 to bit8 are "000_B", the compare output 4 and 5 signals of the output compare will be output from RTO4 and RTO5 respectively. The 16-bit dead timer can also be used as a reload timer.
- If the TMD8 to TMD6: bit10 to bit8 are "001_B", the compare output 4 and 5 signals of the output compare will be output from RTO4 and RTO5 respectively when PPG output is disabled (PGEN4:bit30=0, PGEN5:bit31=0 of the PPG output control register (PICS0)). When PPG output is enabled (PGEN4:bit30=1, PGEN5:bit31=1 of PPG output control register (PICS0)), PPG pulse will be output from RTO4 and RTO5 respectively while the compare output 4 and 5 signals of the output compare are "H". The 16-bit dead timer can also be used as a reload timer.
- The rising edge of each OUT signal functions as a trigger and activates the 16-bit dead timer when TMD8 to TMD6: bit10 to bit8 are "010_B". If PPG output is disabled, "H" will be output until the 16-bit dead timer stops. If PPG output is enabled, PPG pulse will be output until the 16-bit dead timer stops. (Timer mode)
- When TMD8 to TMD6: bit10 to bit8 are "100_B", non-overlap signal will be generated at the OUT signal. (Dead time timer mode)

Note:

To operate the waveform generator in the dead time timer mode, be sure to select the 2-channel mode (compare control register (OCS45); CMOD: bit28=1) for the compare output 5 of the output compare.

4.1.3. 16-bit Dead Timer reload Interrupt Register (DTIR)

The bit configuration for the 16-bit dead timer reload Interrupt register is shown below.

The 16-bit dead timer reload interrupt register (DTIR) is used to control the interrupt request when the timer is reloaded before it underflows as well as to control the interrupt request enable.

■ DTIR0: Address $12AD_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTRIF2	DTRIE2	DTRIF1	DTRIE1	DTRIF0	DTRIE0	Reserved	
Initial values	0	0	0	0	0	0	0	0
Attributes	R(RM1), W	R/W	R(RM1), W	R/W	R(RM1), W	R/W	R0,W0	R0,W0

[bit7] DTRIF2: 16-bit dead timer 2 reload interrupt flag bit

DTRIF2	Function	
	Read	Write
0	No reload of the dead timer has been detected.	This bit is cleared.
1	A reload of the dead timer has been detected.	This bit remains unaffected.

- For the 16-bit dead timer 2, if the timer is reloaded before it underflows, this bit will be set to "1".
- An interrupt request is generated when this bit and the interrupt request enable bit (DTIR: DTRIE2) are "1".
- This bit is cleared when "0" is written to it. A write of "1" does not change this bit and has no influence on others.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If this bit is read by a read-modify-write (RMW) instruction, "1" is always read.
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit6] DTRIE2: 16-bit dead timer 2 reload interrupt enable bit

DTRIE2	Function
0	An interrupt will not be generated even when a reload occurs at the 16-bit dead timer.
1	An interrupt will be generated when a reload occurs at the 16-bit dead timer.

- This bit enables or disables the output of interrupts to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (DTIR: DTRIF2) are "1".

[bit5] DTRIF1: 16-bit dead timer 1 reload interrupt flag bit

DTRIF1	Function	
	Read	Write
0	No reload of the dead timer has been detected.	This bit is cleared.
1	A reload of the dead timer has been detected.	This bit remains unaffected.

- For the 16-bit dead timer 1, if the timer is reloaded before it underflows, this bit will be set to "1".
- An interrupt request is generated when this bit and the interrupt request enable bit (DTIR: DTRIE1) are "1".
- This bit is cleared when "0" is written to it. A write of "1" does not change this bit and has no influence on others.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If this bit is read by a read-modify-write (RMW) instruction, "1" is always read.
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit4] DTRIE1: 16-bit dead timer 1 reload interrupt enable bit

DTRIE1	Function
0	An interrupt will not be generated even when a reload occurs at the 16-bit dead timer.
1	An interrupt will be generated when a reload occurs at the 16-bit dead timer.

- This bit enables or disables the output of interrupts to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (DTIR: DTRIF1) are "1".

[bit3] DTRIF0: 16-bit dead timer 0 reload interrupt flag bit

DTRIF0	Function	
	Read	Write
0	No reload of the dead timer has been detected.	This bit is cleared.
1	A reload of the dead timer has been detected.	This bit remains unaffected.

- For the 16-bit dead timer 0, if the timer is reloaded before it underflows, this bit will be set to "1".
- An interrupt request is generated when this bit and the interrupt request enable bit (DTIR: DTRIE0) are "1".
- This bit is cleared when "0" is written to it. A write of "1" does not change this bit and has no influence on others.
- This bit will be cleared when the dead timer interrupt clear signal is "H".

Note:

If this bit is read by a read-modify-write (RMW) instruction, "1" is always read.
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit2] DTRIE0: 16-bit dead timer 0 reload interrupt enable bit

DTRIE0	Function
0	An interrupt will not be generated even when a reload occurs at the 16-bit dead timer.
1	An interrupt will be generated when a reload occurs at the 16-bit dead timer.

- This bit enables or disables the output of interrupts to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (DTIR: DTRIF0) are "1".

[bit1, bit0] Reserved

Always write 0 to these bits.

4.1.4. 16-bit Dead Timer Minus Control Register (DTMNS)

The bit configuration for the 16-bit dead timer minus control register is shown below.

The 16-bit dead timer minus control register (DTMNS) is used to configure the minus control of the dead time function.
You will need to be careful when setting the dead time function selection bits (MNS2 to MNS0) since this register has key enable bits (KEY1, KEY0).
To set values to the MNS2 to MNS0 bits, write the values in the following sequence:
KEY1, KEY0=00 and MNS2 to MNS0="value you wish to set",
→ KEY1, KEY0=01 and MNS2 to MNS0="value you wish to set (the same value as the last time)"
→ KEY1, KEY0=10 and MNS2 to MNS0="value you wish to set (the same value as the last time)"
→ KEY1, KEY0=11 and MNS2 to MNS0="value you wish to set (the same value as the last time)".
For the MNS2 to MNS0, the value will be reflected when you write the value for the fourth time (when writing to KEY1, KEY0=11). Without following this process (if you write values to or read values from other registers in the middle of the writing process, the value written is incorrect, or if you read values from this register in the middle of the writing process), the value written to this register will be invalid.

■ DTMNS0: Address $12AF_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	KEY1	KEY0	Reserved			MNS2	MNS1	MNS0
Initial values	0	0	0	0	0	0	0	0
Attributes	R0,W	R0,W	R0,W0	R0,W0	R0,W0	R,W	R,W	R,W

[bit7, bit6] KEY1, KEY0: Key enable bits

KEY1, KEY0	Function
	Key code

- It is a key code register used for the settings of MNS2 to MNS0.
- To set values to the MNS2 to MNS0 bits, write the values in the following sequence:
  KEY1, KEY0=00 and MNS2 to MNS0="value you wish to set",
  → KEY1, KEY0=01 and MNS2 to MNS0="value you wish to set (the same value as the last time)"
  → KEY1, KEY0=10 and MNS2 to MNS0="value you wish to set (the same value as the last time)"
  → KEY1, KEY0=11 and MNS2 to MNS0="value you wish to set (the same value as the last time)".
  For the MNS2 to MNS0, the value will be reflected when you write the value for the fourth time (when writing to KEY1, KEY0=11).
- Without following this process (if you write values to or read values from other registers in the middle of the writing process, the value written is incorrect, or if you read values from this register in the middle of the writing process), the value written to this register will be invalid.
- These bits are always "0" when read.

[bit5 to bit3] Reserved

Always write 0 to these bits.

[bit2] MNS2: Dead time function selection bit (RTO4 and RTO5)

MNS2	Function
0	Minus control of the dead timer function will not be executed.
1	Minus control of the dead timer function will be executed.

- Select the control of the dead timer function for RTO4 and RTO5.
- When you set this bit to "0":
  Minus control of the dead timer function will not be executed.
- When you set this bit to "1":
  Minus control of the dead timer function will be executed.
- When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits.
- The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.

[bit1] MNS1: Dead time function selection bit (RTO2 and RTO3)

MNS1	Function
0	Minus control of the dead timer function will not be executed.
1	Minus control of the dead timer function will be executed.

- Select the control of the dead timer function for RTO2 and RTO3.
- When you set this bit to "0":
  Minus control of the dead timer function will not be executed.
- When you set this bit to "1":
  Minus control of the dead timer function will be executed.
- When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits.
- The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.

[bit0] MNS0: Dead time function selection bit (RTO0 and RTO1)

MNS0	Function
0	Minus control of the dead timer function will not be executed.
1	Minus control of the dead timer function will be executed.

- Select the control of the dead timer function for RTO0 and RTO1.
- When you set this bit to "0":
  Minus control of the dead timer function will not be executed.

- When you set this bit to "1":
  Minus control of the dead timer function will be executed.
- When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits.
- The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.

4.1.5. Waveform Control Register 1/2 (SIGCR1, SIGCR2)

The bit configuration for the waveform control register 1/2 is shown below.

The waveform control register 1/2 (SIGCR1, SIGCR2) is used to control the operating clock frequency, noise cancel function valid settings, DTTI input valid settings, and DTTI interrupts.

■ SIGCR10: Address $12B1_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R(RM1), W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] DTIE: DTTI input validating bit

DTIE	Function
0	Invalidate the DTTI input
1	Validate the DTTI input

- This bit is used to validate the output level control DTTI signals for the RTO0 to RTO5 pins.

[bit6] DTIF: DTTI Interrupt flag bit

DTIF	Function	
	Read	Write
0	No interrupt request	This bit is cleared.
1	Interrupt request present	This bit remains unaffected.

- This bit is an interrupt flag for the DTTI.
- When the DTTI input becomes valid (DTIE: bit23=1) and "L" level of the DTTI is detected, this bit will be set and an interrupt request will be generated.
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- This bit will be cleared when the DTTI interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
When the noise cancel function becomes valid (NRSL: bit21=1) and a noise pulse is generated, this bit will be set to "1".
If a software clear (write of "0") or a clear by an interrupt clear signal ("H") and a hardware set (DTTI "L" level detection) occur at the same time, the hardware set has a priority over the software clear or the clear by interrupt clear signal. Then, this bit will be set.

[bit5] NRSL: Noise cancel function validating bit

NRSL	Function
0	Noise cancel circuit of the DTTI input will be invalidated
1	Noise cancel circuit of the DTTI input will be validated

- This bit is used for validating the noise cancel function.
- The noise cancel circuit receives DTTI input signals until an overflow occurs at the counter while it remains at the "L" level. The counter is a N-bit counter operated by the "L" level input.
  Value for N will be either 2, 3, 4, or 5 based on the settings of the NWS1, NWS0: bit17, bit16.

Note:

To cancel the noise pulse width, you will need an approximately 2n peripheral clock.
If you select the noise cancel circuit, the input will be invalidated while it is in the mode that the peripheral clock is inactive (i.e. stop mode).

[bit4 to bit2] DCK2 to DCK0: Operating clock selection bits

DCK2	DCK1	DCK0	Function
0	0	0	ϕ
0	0	1	ϕ/2
0	1	0	ϕ/4
0	1	1	ϕ/8
1	0	0	ϕ/16
1	0	1	ϕ/32
1	1	0	ϕ/64
1	1	1	Setting prohibited

ϕ : Peripheral clock

- These bits are used to select the operating clock for the 16-bit dead timer.

[bit1, bit0] NWS1, NMS0: DTTI noise width selection bits

NWS1	NWS0	Function
0	0	Cancel the 4-peripheral clock cycle noise
0	1	Cancel the 8-peripheral clock cycle noise
1	0	Cancel the 16-peripheral clock cycle noise
1	1	Cancel the 32-peripheral clock cycle noise

- These bits are used to select the DTTI pin noise pulse width to be removed.

■ SIGCR20: Address $12B3_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PSEL21	PSEL20	PSEL11	PSEL10	PSEL01	PSEL00	Reserved	DTTI
Initial values	0	0	0	0	0	0	0	1
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W

[bit7, bit6] PSEL21, PSEL20: PPG input channel selection bits (RTO4, RTO5)

PSEL21	PSEL20	Function
0	0	PPG0
0	1	PPG2
1	0	PPG4
1	1	Setting prohibited (operation is not guaranteed)

- These bits are used to select the PPG input for the RTO4 and RTO5.
- These bits are also used to select the GATE output destination for PPG.
- Settings of "11_B" is prohibited.

[bit5, bit4] PSEL11, PSEL10: PPG input channel selection bits (RTO2, RTO3)

PSEL11	PSEL10	Function
0	0	PPG0
0	1	PPG2
1	0	PPG4
1	1	Setting prohibited (operation is not guaranteed)

- These bits are used to select the PPG input for the RTO2 and RTO3.
- These bits are also used to select the GATE output destination for PPG.
- Settings of "11_B" is prohibited.

[bit3, bit2] PSEL01, PSEL00: PPG input channel selection bits (RTO0, RTO1)

PSEL01	PSEL00	Function
0	0	PPG0
0	1	PPG2
1	0	PPG4
1	1	Setting prohibited (operation is not guaranteed)

- These bits are used to select the PPG input for the RTO0 and RTO1.
- These bits are also used to select the GATE output destination for PPG.
- Settings of "11_B" is prohibited.

[bit1] Reserved

Always write 0 to this bit.

[bit0] DTTI: Software DTTI setting bit

DTTI	Function
0	DTTI set
1	DTTI clear

- Writing "0" will set the DTTI.
- Writing "1" will clear the DTTI.

Note:

Since it computes the logical OR with an external input DTTI, DTTI depends on the external input level.

4.1.6. PPG Output Control Register (PICS)

The bit configuration for the PPG output control register is shown below.

The PPG output control register (PICS) is used to control the PPG output.

■ PICS0: Address $12B4_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	PGEN5	PGEN4	PGEN3	PGEN2	PGEN1	PGEN0	Reserved	
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	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 values	0	0	0	0	0	0	0	0
Attributes	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							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W0	R/W0	R/W0

[bit31] PGEN5: PPG output enable bit

PGEN5	Function
0	Disable PPG output to the RTO5
1	Enable PPG output to the RTO5

This bit is used to select PPG output to the RTO5.

[bit30] PGEN4: PPG output enable bit

PGEN4	Function
0	Disable PPG output to the RTO4
1	Enable PPG output to the RTO4

This bit is used to select PPG output to the RTO4.

[bit29] PGEN3: PPG output enable bit

PGEN3	Function
0	Disable PPG output to the RTO3
1	Enable PPG output to the RTO3

This bit is used to select PPG output to the RTO3.

[bit28] PGEN2: PPG output enable bit

PGEN2	Function
0	Disable PPG output to the RTO2
1	Enable PPG output to the RTO2

This bit is used to select PPG output to the RTO2.

[bit27] PGEN1: PPG output enable bit

PGEN1	Function
0	Disable PPG output to the RTO1
1	Enable PPG output to the RTO1

This bit is used to select PPG output to the RTO1.

[bit26] PGEN0: PPG output enable bit

PGEN0	Function
0	Disable PPG output to the RTO0
1	Enable PPG output to the RTO0

This bit is used to select PPG output to the RTO0.

[bit25 to bit0] Reserved

Always write 0 to these bits.

5. Operation

This section explains the operation.

5.1 Interrupts for Waveform Generator

5.2 Operation of the Waveform Generator

5.1. Interrupts for Waveform Generator

The interrupts for waveform generator is explained.

Table 5-1 and Table 5-2 show the interrupt control bits and interrupt factors of the waveform generator.

Table 5-1 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-1

	16-bit dead timer 0	16-bit dead timer 1	16-bit dead timer 2
Interrupt request flag bit	16-bit dead timer state control register 0 (DTSCR0) TMIF0: bit28	16-bit dead timer state control register 1 (DTSCR1) TMIF1: bit20	16-bit dead timer state control register 2 (DTSCR2) TMIF2: bit12
Interrupt request enable bit	16-bit dead timer state control register 0 (DTSCR0) TMIE0: bit27	16-bit dead timer state control register 1 (DTSCR1) TMIE1: bit19	16-bit dead timer state control register 2 (DTSCR2) TMIE2: bit11
Interrupt factor	16-bit dead timer 0 underflow	16-bit dead timer 1 underflow	16-bit dead timer 2 underflow

Table 5-2 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-2

	DTTI0
Interrupt request flag bit	Waveform control register 10 (SIGCR10) DTIF: bit22
Interrupt request enable bit	-
Interrupt factor	Detection of "L" level at DTTI0

For the waveform generator, TMIF0/TMIF1/TMIF2 (bit28/bit20/bit12) of the 16-bit dead timer state control register (DTSCR) will be set to "1" when an underflow occurs at the 16-bit dead timer and TMD2 to TMD0/TMD5 to TMD3/TMD8 to TMD6 (bit26 to bit24/bit18 to bit16/bit10 to bit8) of the DTSCR register is either "000_B" or "001_B". If interrupt requests are enabled (DTSCR register; TMIE0/TMIE1/TMIE2(bit27/bit19/bit11)=1) in this state, an interrupt request is output to the interrupt controller.

Table 5-3 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #2-1

	16-bit dead timer 0/3	16-bit dead timer 1/4	16-bit dead timer 2/5
Interrupt request flag bit	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIF0	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIF1	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIF2
Interrupt request enable bit	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIE0	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIE1	16-bit dead timer reload interrupt register 0/1 (DTIR0/1) DTRIE2
Interrupt factor	While 16-bit dead timer 0/3 is in operation, reload occurs before the timer underflows	While 16-bit dead timer 1/4 is in operation, reload occurs before the timer underflows	While 16-bit dead timer 2/5 is in operation, reload occurs before the timer underflows

If a reload occurs before the timer underflows while the 16-bit dead timer is active, the interrupt request flag bit will be set. An interrupt request will be output to the interrupt controller when the corresponding interrupt request enable bit is enabled.

5.2. Operation of the Waveform Generator

The operation of the waveform generator is explained.

The waveform generator can generate various types of waveforms (including dead time) using the real-time output (RTO0 to RTO5), 16-bit PPG timer 0/2/4, and 16-bit dead timer 0/1/2.

Output state of RTO0 to RTO5 and GATE

Table 5-4 RTO/GATE Output State and Bit Settings

TMD2	TMD1	TMD0	GTEN	PGEN	RTO	GATE
0	0	0	X	X	Compare output OUT (16-bit output compare output)	Always "0"
0	0	1	0/1	0	Compare output OUT (16-bit output compare output)	(OUT and GTEN) *3
			0	1	Outputs the PPG0/PPG2/PPG4 pulses while the OUT is "H" *1	Always "0"
			1	1	Outputs the PPG0/PPG2/PPG4 pulses activated by GATE signals while the OUT is "H"	OUT
0	1	0	0/1	0	Activates the 16-bit dead timer by the rising edge of OUT, and "H" is being output until the 16-bit dead timer underflows	When GTEN and the timer is active, "H" is being output *4
			0	1	Activates the 16-bit dead timer by the rising edge of OUT, and PPG0/PPG2/PPG4 pulses are being output until the 16-bit dead timer underflows *1	Always "0"
			1	1	Activates the 16-bit dead timer by the rising edge of OUT, and PPG0/PPG2/PPG4 pulses activated by GATE signals are being output until the 16-bit dead timer underflows	When the timer is active, "H" is being output *4
1	0	0	X	X	Generates a non-overlap signal at OUT *2	Always "0"
1	1	1	0	X	Setting prohibited	-
			1	X	Setting prohibited	-
Others					Always "0"	Always "0"

*1: You will need to select a channel you wish to use from the PPG0/PPG2/PPG4 and activate the PPG in advance.

*2: To generate a non-overlap signal, make sure to select the 2-channel mode (compare control register (OCS1, OCS3, OCS5); CMOD=1) for the OUT1, OUT3, and OUT5.

*3: GATE signal will be generated from the OUT of which GTEN bit is set to "1".

*4: GATE signal will be generated during the operation of the timer activated by the OUT of which GTEN bit is set to "1". If more than one GATE bit is set to "1", GATE signal will be the OR signal of signals in each active timer operation.

■ Pins and Bit Setting

Pin application table for RTO/GATE output state and bit settings is as follows:

Table 5-5 Pins and Bit Setting Application Table

GATE signal control bit	PPG output enable bit	Real time output	Compare output	16-bit dead timer	Non-overlap signal
GTEN0	PGEN0	RTO0	OUT0	16-bit dead timer 0	OUT1
GTEN1	PGEN1	RTO1	OUT1	16-bit dead timer 0	OUT1
GTEN2	PGEN2	RTO2	OUT2	16-bit dead timer 1	OUT3
GTEN3	PGEN3	RTO3	OUT3	16-bit dead timer 1	OUT3
GTEN4	PGEN4	RTO4	OUT4	16-bit dead timer 2	OUT5
GTEN5	PGEN5	RTO5	OUT5	16-bit dead timer 2	OUT5

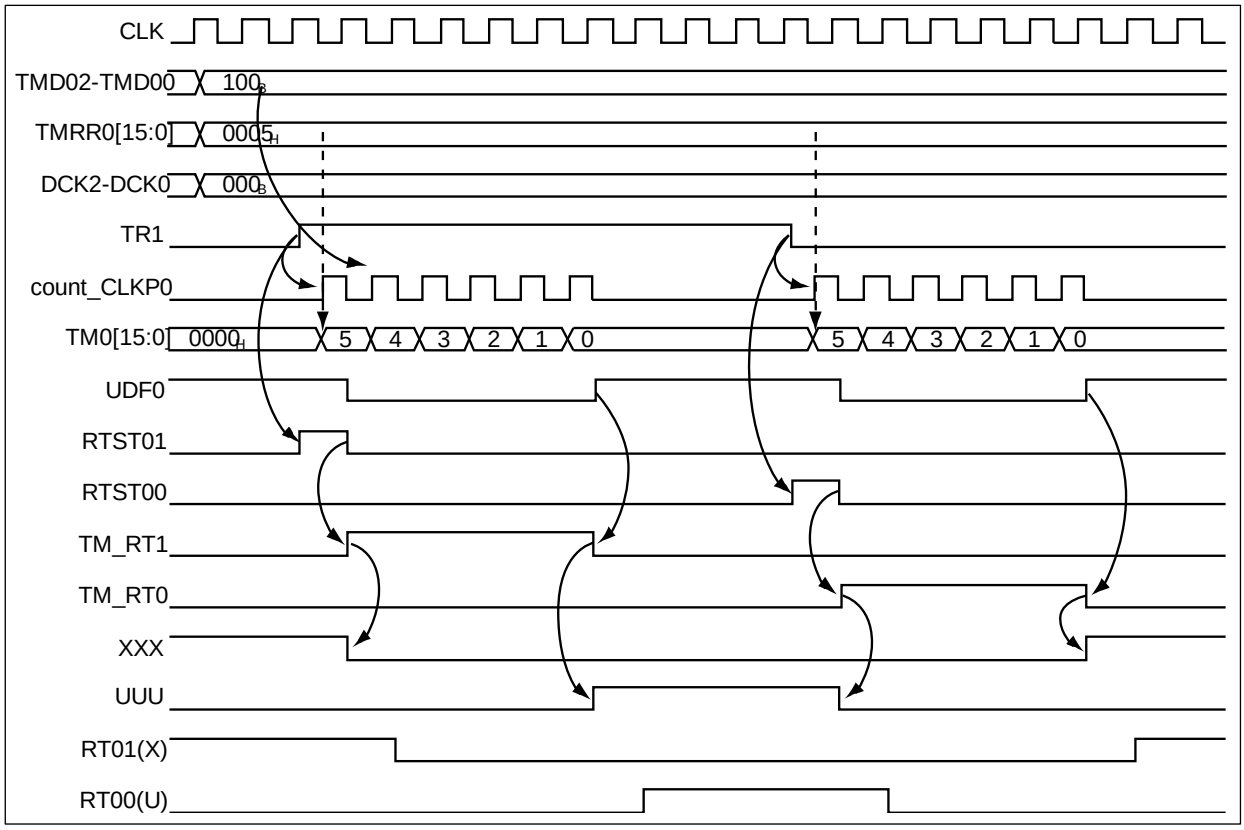
Note:

The RTO0 and RTO1 are controlled by the TMD2 to TMD0 of the 16-bit dead timer state control register (DTSCR0). The RTO2 and RTO3 are controlled by the TMD5 to TMD3 of the DTSCR1 register. The RTO4 and RTO5 are controlled by the TMD8 to TMD6 of the DTSCR2 register.

■ Operation Timing

Operation timing of the waveform generator dead timer

Figure 5-1 Operation Timing of the Waveform Generator Dead Timer



PPG output control

PPG output to the RTO0 to RTO5 pins can be enabled by the PGEN5 to PGEN0 of the PPG output control register (PICS).

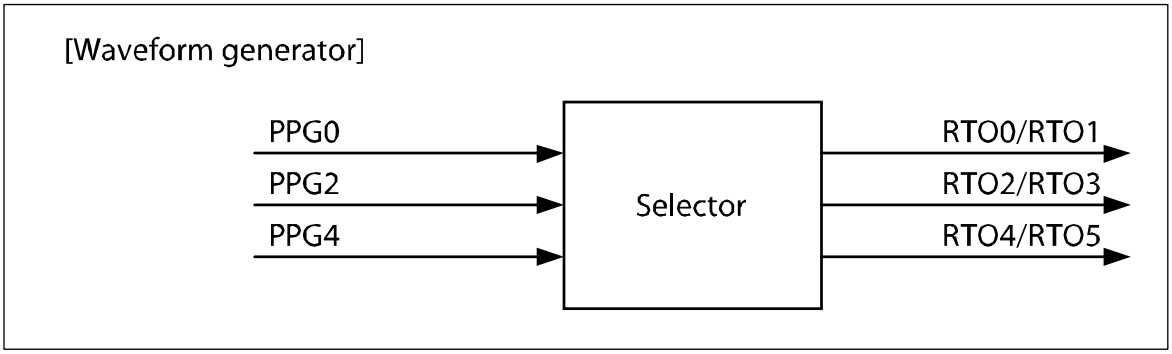
Gate-triggered PPG output

The waveform generator can generate GATE signals by the real-time output RTO0 to RTO5. Each of the 16-bit dead timer 0, 1, and 2 operates two real-time outputs (RTO0/RTO2/RTO4, RTO1/RTO3/RTO5), which generates six distinct gate signals. These six gate signals generate OR GATE signal, which triggers the PPG count.

In addition, when you use the PGEN0 to PGEN5 signals, you will be able to output six types of waveforms to the RTO0 to RTO5 pins by using the PPG only.

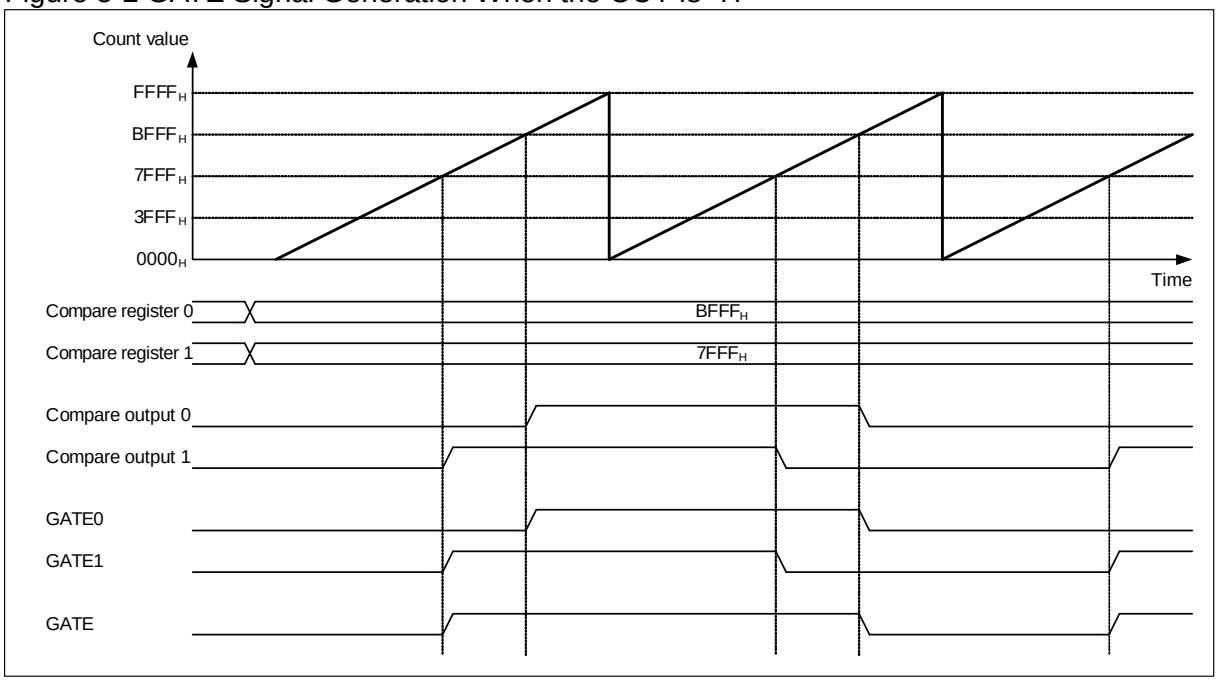
Note:

The case of waveform generator 0 is introduced as an example. For waveform generator 1, PPG channels you will be able to select are shown in the table below.



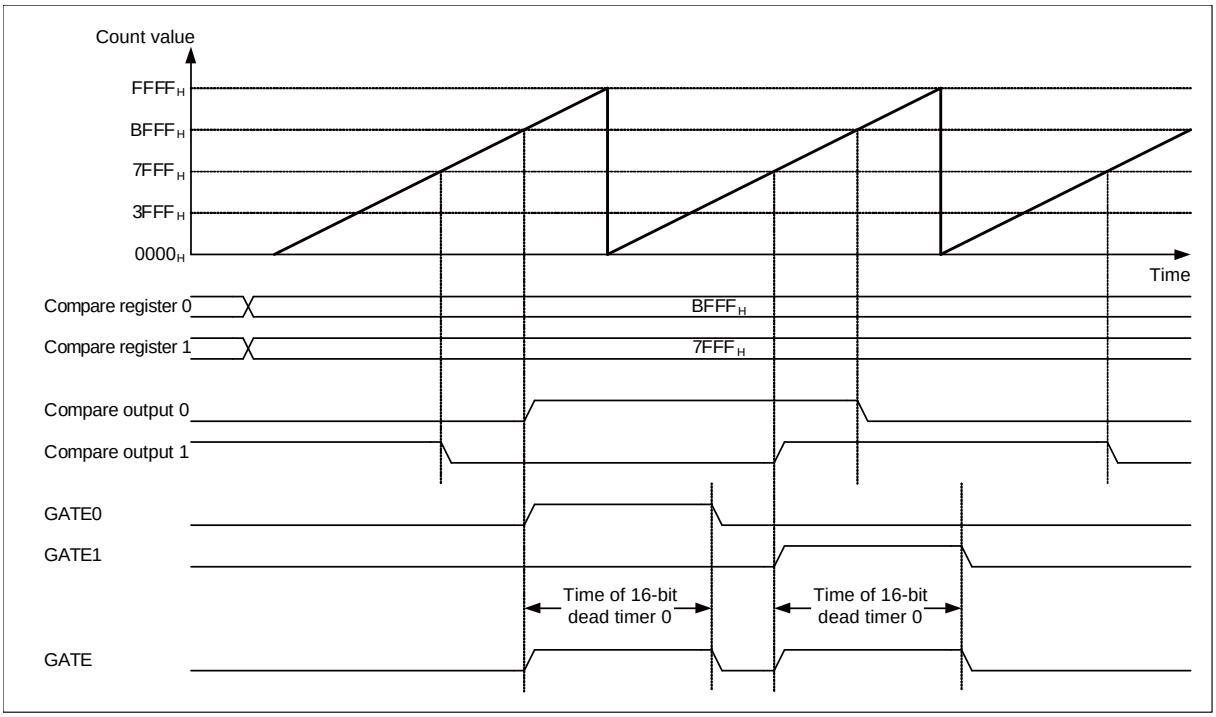
- **GATE signal generation when the GATE is active and each OUT is "H" (TMD8 to TMD0 of the 16-bit dead timer state control (DTSCR0, DTSCR1, DTSCR2) are "001_B")**

Figure 5-2 GATE Signal Generation When the OUT Is "H"



- **GATE signal generation from the rising edge of the OUT through the underflow of the 16-bit dead timer 0, 1, and 2 when the GTEN is active (TMD8 to TMD0 of the DTSCR0, DTSCR1, and DTSCR2 registers are 010_B)**

Figure 5-3 GATE Signal Generation from the Rising Edge of the OUT through the 16-bit Dead Timer Underflow



Note:

Each 16-bit dead timer will be used for two OUTs. In other words, the 16-bit dead timer 0 is used for OUT0 and OUT1, the 16-bit dead timer 1 is used for OUT2 and OUT3, and the 16-bit dead timer 2 is used for OUT4 and OUT5. Thus, do not activate the timer which is already active, using the OUT. If you activate such timer, the GATE signal output will be extended, which may cause a malfunction as a result.

Operation in timer mode

If a rising edge of the OUT0 to OUT5 pins has been detected, the value will be reloaded to the 16-bit dead timer and the timer starts counting down. The PPG timer continues to output to the RTO0 to RTO5 pins until the 16-bit dead timer underflows.

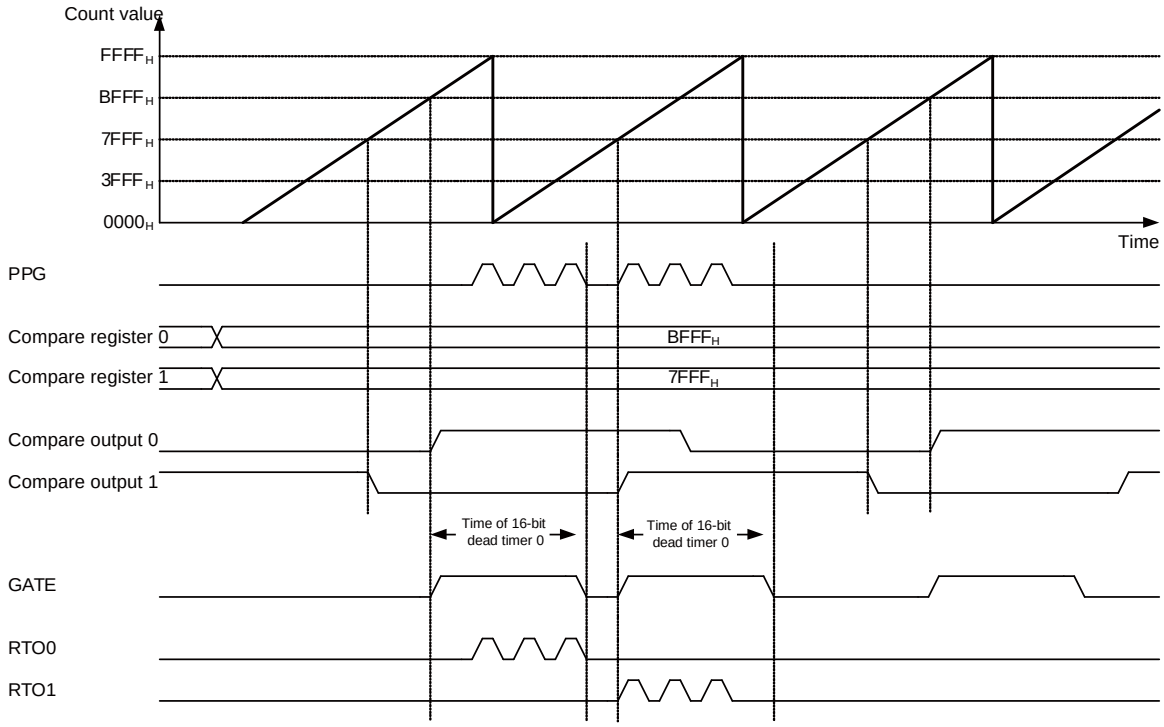
- **PPG output pulse generation from the OUT rising edge through the 16-bit dead timer underflow (TMD8 to TMD0 of the DTSCR0, DTSCR1, DTSCR2 registers are 010_B)**

Figure 5-4 Waveform Generated When the TMD2 to TMD0 are "010_B"

<Register setting>

TCDT	:"XXXX_H"
TCCS	:"XXXXXXXX X0X0XXXX ----XXXX_B"
CPCLR	:"XXXX_H" (cycle setting)
OCCP0 to OCCP5	:"XXXX_H" (compare value)
OCS01 to OCS45	:"-XX0XXXX XXXXXX11_B"
DTCR0 to DTCR2	:"011XX010_B"
TMRR0 to TMRR2	:"XXXX_H" (non-overlap timing setting)
SIGCR10	:"XXXXX00_B" (DTTI input and 16-bit dead timer count clock setting)
PICS0	:"XX_H" (PPG0 output selection)

(Note) "X": Make a setting according to the operation.



Note:

Each 16-bit dead timer will be used for two OUTs. In other words, the 16-bit dead timer 0 is used for OUT0 and OUT1, the 16-bit dead timer 1 is used for OUT2 and OUT3, and the 16-bit dead timer 2 is used for OUT4 and OUT5. Thus, do not activate the timer which is already active, using the OUT. If you activate such timer, the GATE signal output will be extended, which may cause a malfunction as a result.

Operation of the dead time timer mode

The dead time generator inputs the compare output (OUT1, OUT3, OUT5) and outputs non-overlap signal (inverted signal) to the external pins (RTO0 to RTO5).

- **Non-overlap signal generation by the normal polarity OUT1, OUT3, and OUT5 (TMD8 to TMD0 of the 16-bit dead timer control registers (DTSCR0, DTSCR1, DTSCR2) are 100_B)**

  If you select the non-overlap signal of which DMOD2 to DMOD0 of the DTSCR0, DTSCR1, and DTSCR2 registers are "0" (normal polarity), the delay that corresponds to the non-overlap time configured at the 16-bit dead timer registers (TMRR0 to TMRR2) will be applied. This delay will be applied to the rising edge or the falling edge of the OUT1, OUT3, and OUT5 pins.

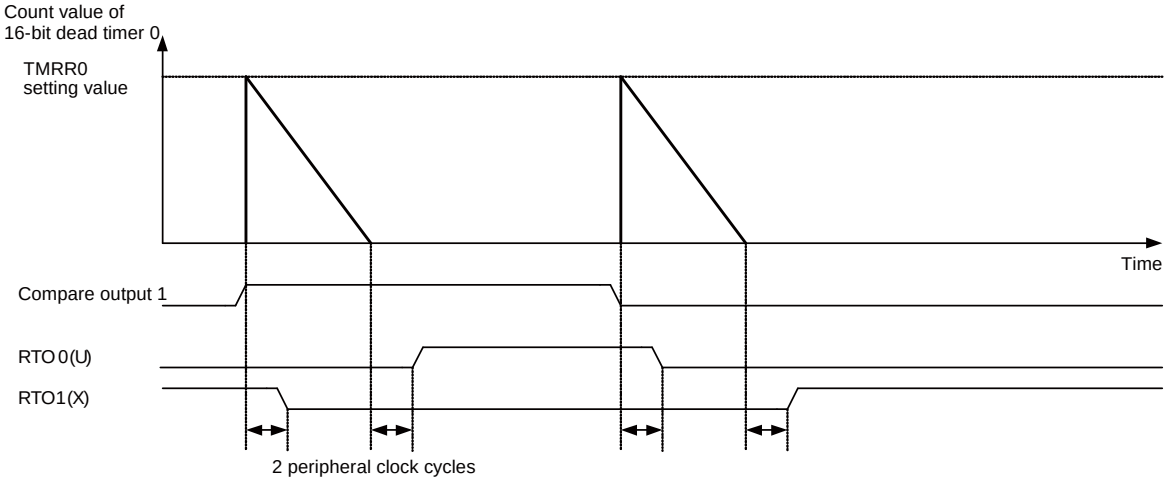
  If the edge transition time of the OUT1, OUT3, and OUT5 is less than the non-overlap time configured, the 16-bit dead timer restarts counting down from the value of time from the dead timer start at the next RT edge to the restart. When the dead timer is activated once again before the counting down of the restarted dead timer ends, the counting down will be restarted with the values of registers TMRR0 to TMRR2.

Figure 5-5 Non-overlap Signal Generation by the Normal Polarity Compare Output

<Register setting>

TCDT	:"XXXX$_H$"
TCCS	:"X-XXXXXX X0X0XXXX ----XXXX$_B$"
CPCLR	:"XXXX$_H$" (cycle setting)
OCCP0 to OCCP5	:"XXXX$_H$" (compare value)
OCS01 to OCS45	:"-XX1XXXX XXXXXX11$_B$"
DTCR0 to DTCR2	:"0XXXX100$_B$"
TMRR0 to TMRR2	:"XXXX$_H$" (non-overlap timing setting)
DTMNS0	:"XX---000$_B$"
SIGCR10	:"XXXXXX00$_B$" (DTTI input and 16-bit dead timer count clock setting)

(Note) "X": Make a setting according to the operation.



Pin name	Output signal
RTO0 (U)	Delayed signals are applied to the rising edge of compare output 1.
RTO2 (V)	Delayed signals are applied to the rising edge of compare output 3.
RTO4 (W)	Delayed signals are applied to the rising edge of compare output 5.
RTO1 (X)	Delayed inverted signals are applied to the falling edge of compare output 1.
RTO3 (Y)	Delayed inverted signals are applied to the falling edge of compare output 3.
RTO5 (Z)	Delayed inverted signals are applied to the falling edge of compare output 5.

- **Non-overlap signal generation of minus control by the inverted polarity OUT1, OUT3, and OUT5 (TMD8 to TMD0 of the 16-bit dead timer control registers (DTSCR0, DTSCR1, DTSCR2) are 100_B)**

  If you select the non-overlap signal of which DMOD2 to DMOD0 of the DTSCR0, DTSCR1, and DTSCR2 registers are "1" (inverted polarity), the delay that corresponds to the non-overlap time configured at the 16-bit dead timer registers (TMRR0 to TMRR2) will be applied. This delay will be applied to the rising edge or the falling edge of the OUT1, OUT3, and OUT5 pins.

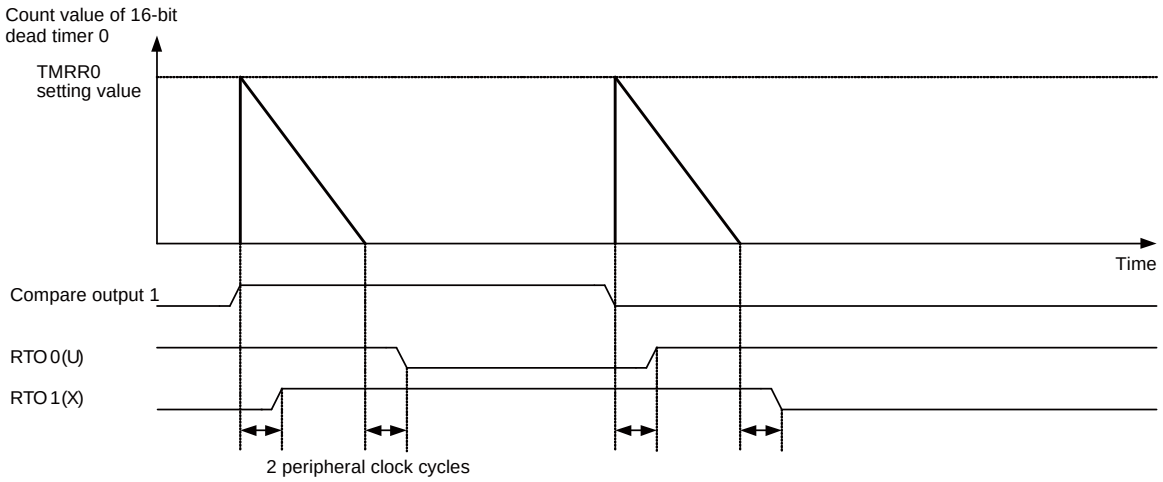
  If the edge transition time of the OUT1, OUT3, and OUT5 is less than the non-overlap time configured, the 16-bit dead timer restarts counting down from the value of time from the dead timer start at the next RT edge to the restart. When the dead timer is activated once again before the counting down of the restarted dead timer ends, the counting down will be restarted with the values of registers TMRR0 to TMRR2.

Figure 5-6 Non-overlap Signal Generation by the Inverted Polarity Compare Output

<Register setting>

TCDT	:"XXXX_H"
TCCS	:"X-XXXXXX X0X0XXXX ----XXXX_B"
CPCLR	:"XXXX_H" (cycle setting)
OCCP0 to OCCP5	:"XXXX_H" (compare value)
OCS01 to OCS45	:"-XX1XXXX XXXXXX11$_B$"
DTCR0 to DTCR2	:"1XXXX100$_B$"
TMRR0 to TMRR2	:"XXXX_H" (non-overlap timing setting)
DTMNS0	:"XX---000$_B$"
SIGCR10	:"XXXXXX00$_B$" (DTTI input and 16-bit dead timer count clock setting)

(Note) "X": Make a setting according to the operation.



Pin name	Output signal
RTO0(U)	Delayed inverted signals are applied to the rising edge of compare output 1.
RTO2(V)	Delayed inverted signals are applied to the rising edge of compare output 3.
RTO4(W)	Delayed inverted signals are applied to the rising edge of compare output 5.
RTO1(X)	Delayed signals are applied to the falling edge of compare output 1.
RTO3(Y)	Delayed signals are applied to the falling edge of compare output 3.
RTO5(Z)	Delayed signals are applied to the falling edge of compare output 5.

Operation of the dead time timer mode (minus control)

You will be able to execute the minus control function of the non-overlap time (MNSx bit of the DTMNS register is 1) using the 16-bit dead timer minus control register (DTMNS).

- **Non-overlap signal generation of minus control by the normal polarity OUT1, OUT3, and OUT5 (TMD8 to TMD0 of the 16-bit dead timer control registers (DTSCR0, DTSCR1, DTSCR2) are 100_B)**

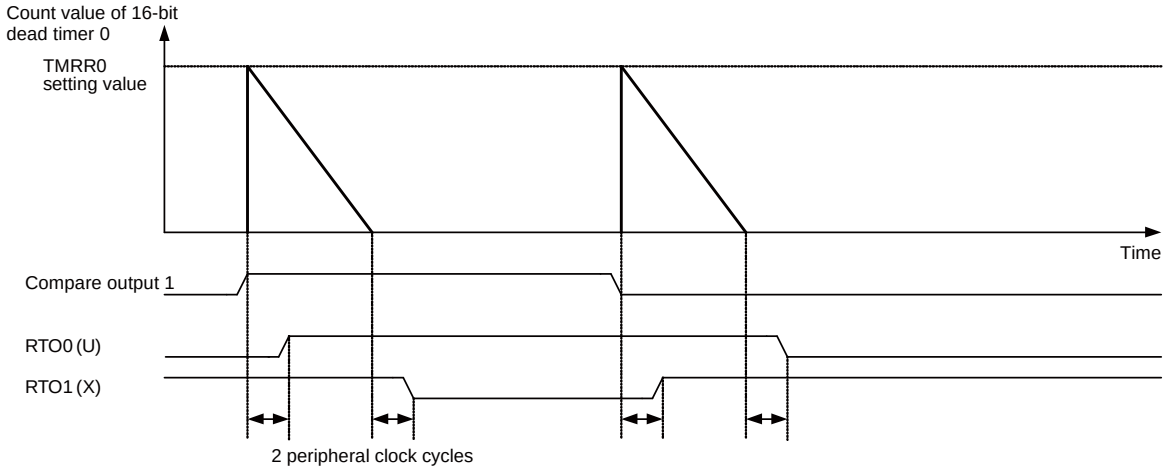
  Signal generation is executed by outputting U/V/W and X/Y/Z of the inverted polarity non-overlap signal that does not operate by minus control as X/Y/Z and U/V/W.

Figure 5-7 Non-overlap Signal Generation When MNSx Bit of the Normal Polarity DTMNS Register Is 1 (Minus Setting)

<Register setting>

TCDT	: "XXXX_H"
TCCS	: "X-XXXXXX X0X0XXXX----XXXX_B"
CPCLR	: "XXXX_H" (cycle setting)
OCCP0 to OCCP5	: "XXXX_H" (compare value)
OCS01 to OCS45	: "- XX1XXXX XXXXXX11_B"
DTCR0 to DTCR2	: "1XXX$X100_B$"
TMRR0 to TMRR2	: "XXXX_H" (non-overlap timing setting)
DTMNS0	: "XX---111_B"
SIGCR10	: "XXXXX$X00_B$" (DTTI input and 16-bit dead timer count clock setting)

(Note) "X": Make a setting according to the operation.



Pin name	Output signal
RTO0 (U)	Delayed signals are applied to the falling edge of compare output 1.
RTO2 (V)	Delayed signals are applied to the falling edge of compare output 3.
RTO4 (W)	Delayed signals are applied to the falling edge of compare output 5.
RTO1 (X)	Delayed inverted signals are applied to the rising edge of compare output 1.
RTO3 (Y)	Delayed inverted signals are applied to the rising edge of compare output 3.
RTO5 (Z)	Delayed inverted signals are applied to the rising edge of compare output 5.

- **Non-overlap signal generation by the inverted polarity OUT1, OUT3, and OUT5 control (TMD8 to TMD0 of the 16-bit dead timer control registers (DTSCR0, DTSCR1, DTSCR2) are 100_B)**

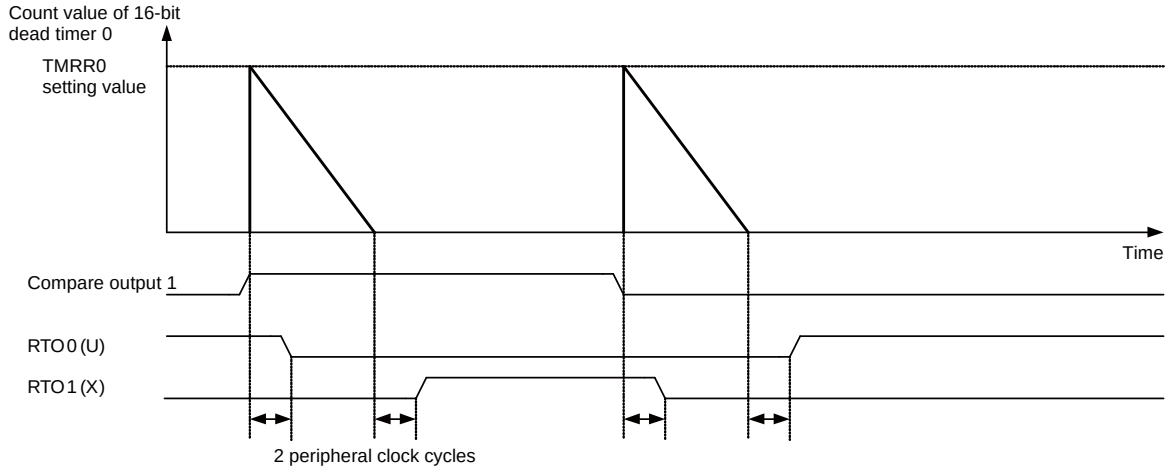
  Signal generation is executed by outputting U/V/W and X/Y/Z of the normal polarity non-overlap signal that does not operate by minus control as X/Y/Z and U/V/W.

Figure 5-8 Non-overlap Signal Generation When MNS Bit of the Inverted Polarity DTMNS Register Is 1 (Minus Setting)

<Register setting>

TCDT	: "XXXX_H"
TCCS	: "X-XXXXXX X0X0XXX---XXXX_B"
CPCLR	: "XXXX_H" (cycle setting)
OCCP0 to OCCP5	: "XXXX_H" (compare value)
OCS01 to OCS45	: "- XX1XXXX XXXXXX11_B"
DTCR0 to DTCR2	: "0XXXX100_B"
TMRR0 to TMRR2	: "XXXX_H" (non-overlap timing setting)
DTMNS0	: "XX---111_B"
SIGCR10	: "XXXXXX00_B" (DTTI input and 16-bit dead timer count clock setting)

(Note) "X": Make a setting according to the operation.



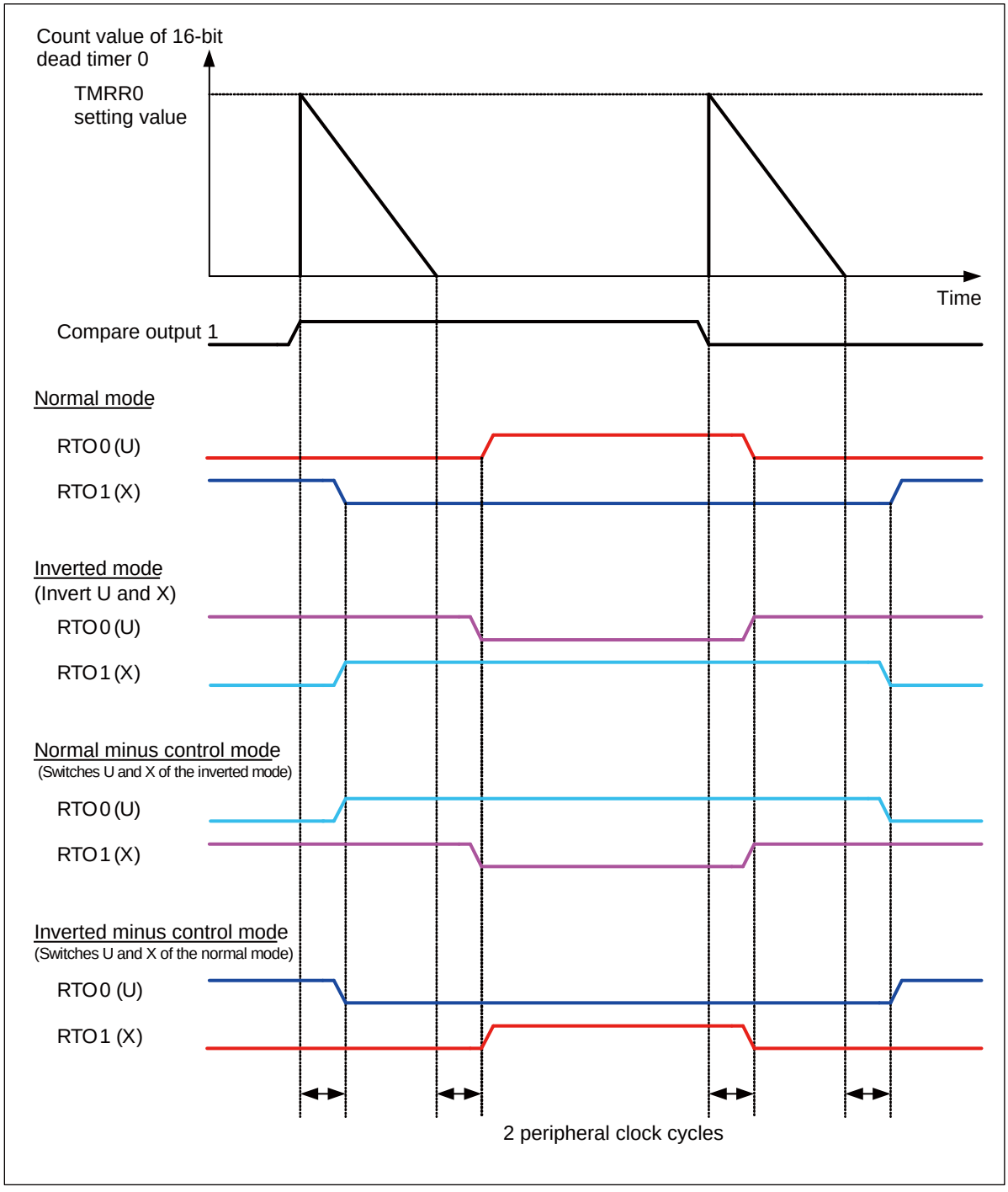
Pin name	Output signal
RTO0 (U)	Delayed inverted signals are applied to the falling edge of compare output 1.
RTO2 (V)	Delayed inverted signals are applied to the falling edge of compare output 3.
RTO4 (W)	Delayed inverted signals are applied to the falling edge of compare output 5.
RTO1 (X)	Delayed signals are applied to the rising edge of compare output 1.
RTO3 (Y)	Delayed signals are applied to the rising edge of compare output 3.
RTO5 (Z)	Delayed signals are applied to the rising edge of compare output 5.

Operation of the dead time timer mode (precautions)

■ Signal generation

Default is the normal mode.
In the inverted mode, each U and X of the normal mode will be output after being inverted.
In the normal minus control mode, each U and X of the normal mode will be output after being inverted and replaced.
In the inverted minus control mode, each U and X of the normal mode will be output after being replaced.

Figure 5-9 Signal Generation in the Dead Time Timer Mode



- **When the "H" interval of the compare output is long (or short) and a reload occurs before the dead timer underflows (the reload of the dead timer is once)**

  In the normal mode or inverted minus control mode, output of the X (or U) is fixed to "L".
  In the normal minus control mode or inverted mode, output of the U (or X) is fixed to "H". While the dead timer is counting down, both U and X are output as "H". The interval is 2 times of the time from the start to the restart of the dead timer (compare output transition time).
  In addition, when a reload occurs before the timer underflows while the dead timer is active, the interrupt request flag bit of the 16-bit dead timer reload interrupt register (DTIR) will be set. If interrupts are enabled, the interrupt will be notified.

Figure 5-10 When the "H" Interval of the Compare Output Is Long and a Reload Occurs Before the Dead Timer Underflows

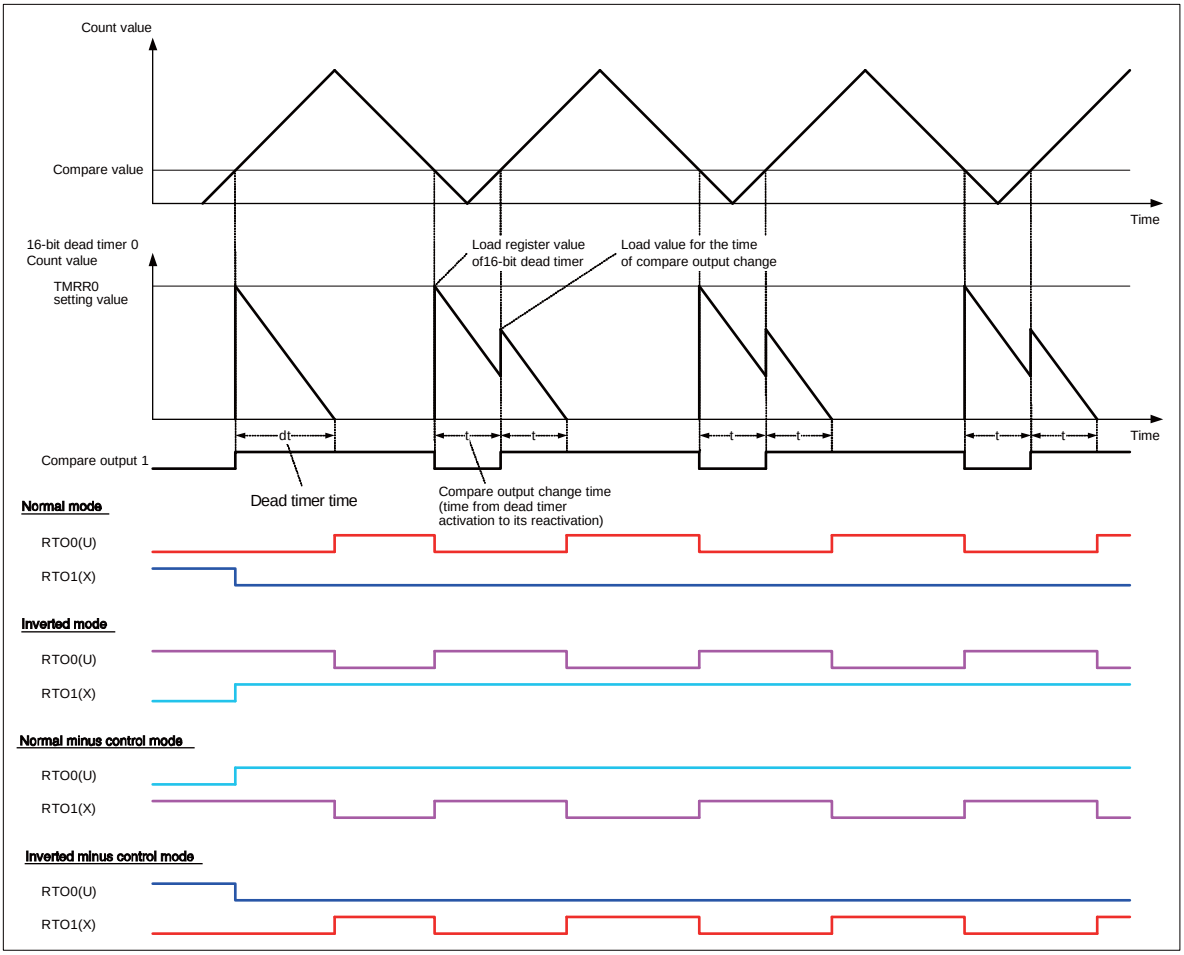
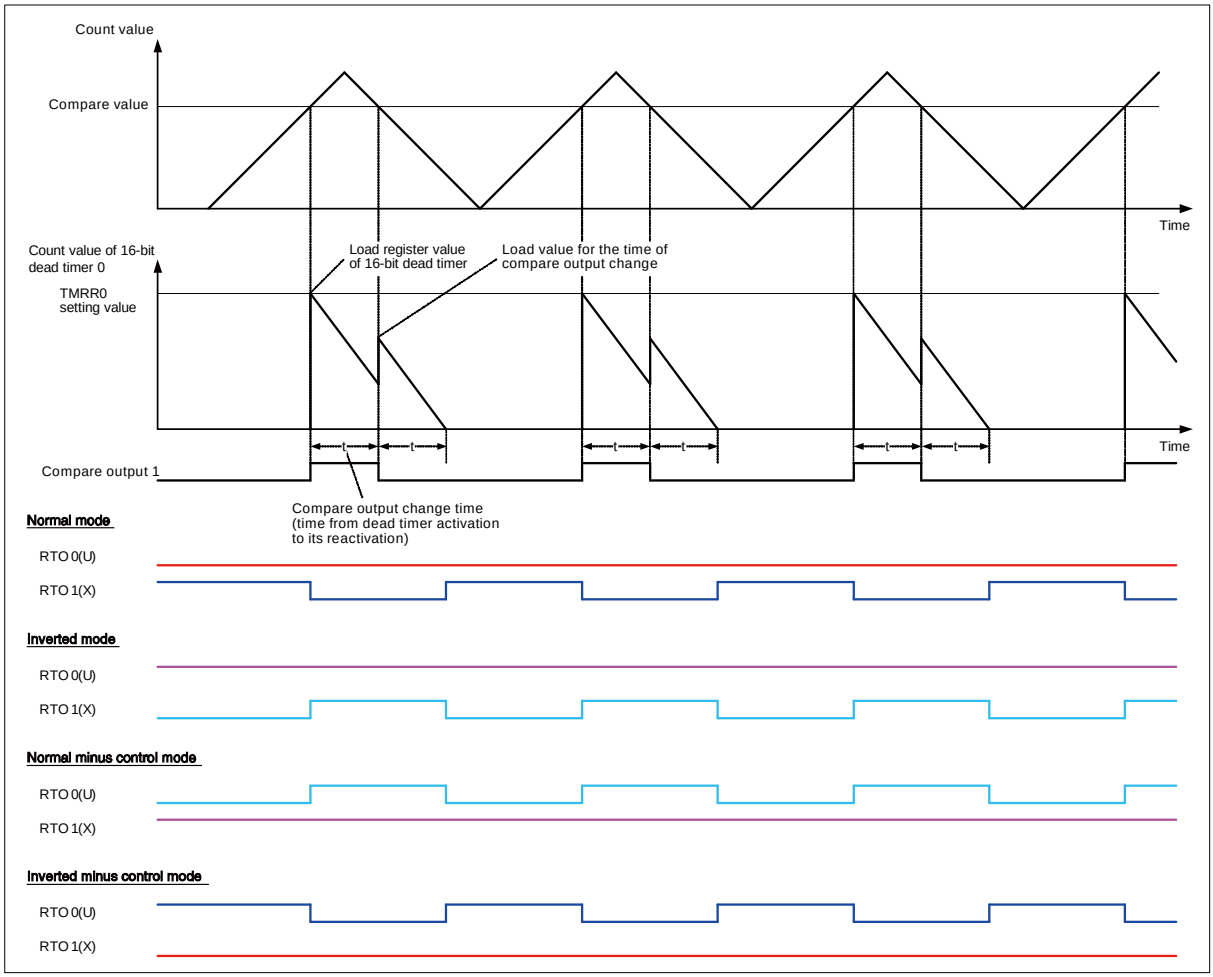


Figure 5-11 When the "H" Interval of the Compare Output Is Short and a Reload Occurs Before the Dead Timer Underflows



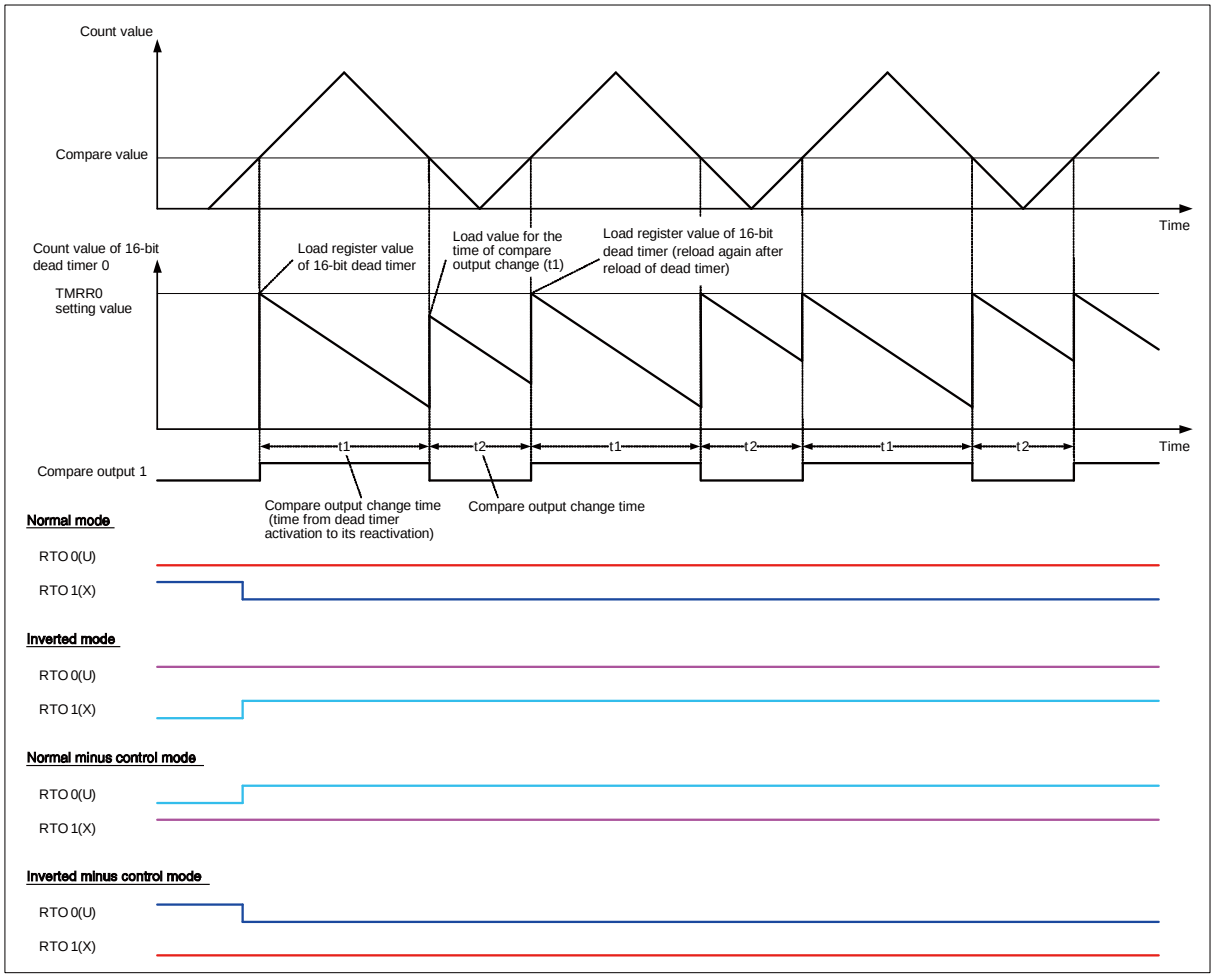
- **When the compare output transition time is short and reload continues before the dead timer underflows (the reload of the dead timer continues)**

  A setting of 16-bit dead timer register (TMRR) with a short compare output transition time that allows continuous reloading before the dead timer underflows is prohibited.

  In the case with the setting above in normal mode or inverted minus control mode, output of the X and U will be fixed to "L". In the normal minus control mode or inverted mode, output of the U and X will be fixed to "H".

  In addition, when a reload occurs before the timer underflows while the dead timer is active, the interrupt request flag bit of the 16-bit dead timer reload interrupt register (DTIR) will be set. If interrupts are enabled, the interrupt will be notified.

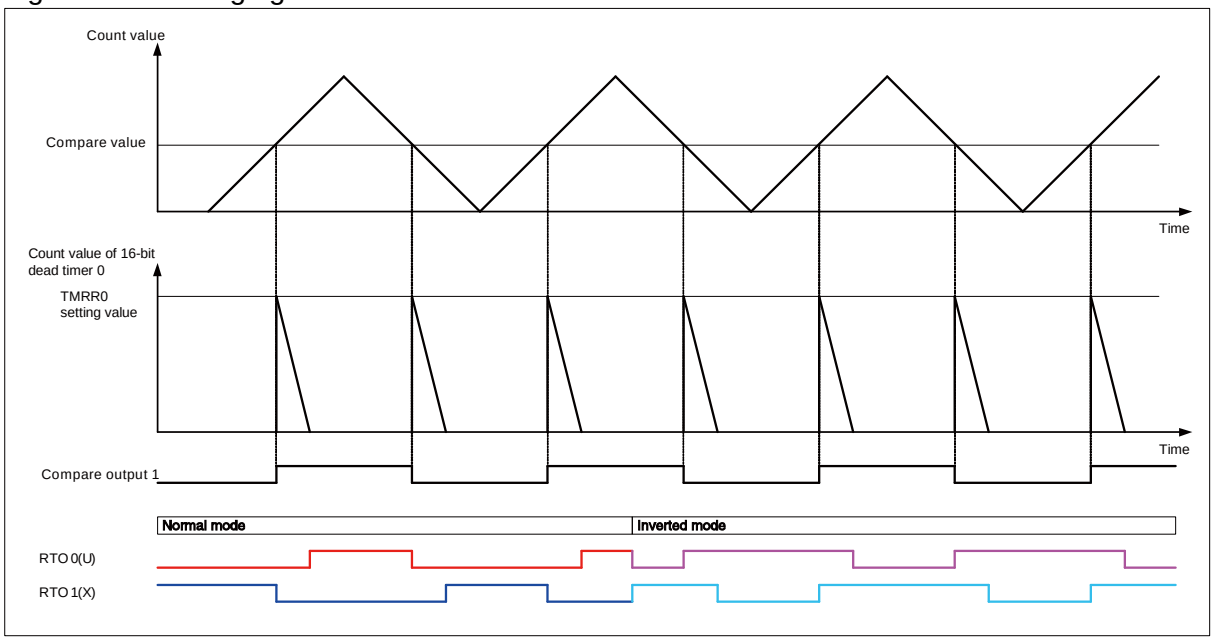
Figure 5-12 When the Compare Output Transition Time is Short and Reload Continues Before the Dead Timer Underflows



- **Changing from normal mode to inverted mode while the dead time timer mode is active**

  If you change the operation mode from the normal mode to the inverted mode while the dead time timer mode is active, the point of variation of U and X overlap. Note that the operation mode will be changed from the normal mode to the inverted mode immediately.

Figure 5-13 Changing from Normal Mode to Inverted Mode While the Dead Time Timer Mode Is Active



- **Changing the minus control mode while the dead time timer mode is active**

  When you change the minus control mode while the dead time timer mode is active, you will need to reflect the settings of minus control mode while the dead timer is inactive and the trigger input (compare output) is "L" in order to prevent the point of variation of U and X from being overlapped.

Figure 5-14 Changing from Normal Mode to Normal Minus Control Mode While the Dead Time Timer Mode Is Active #1

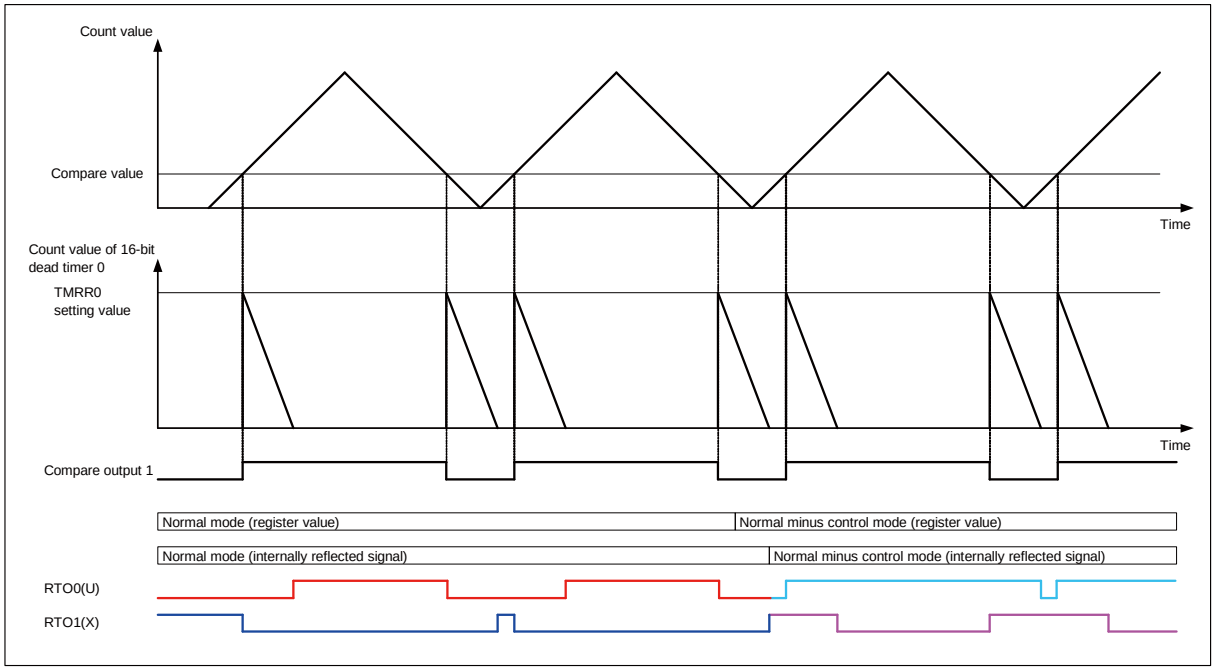
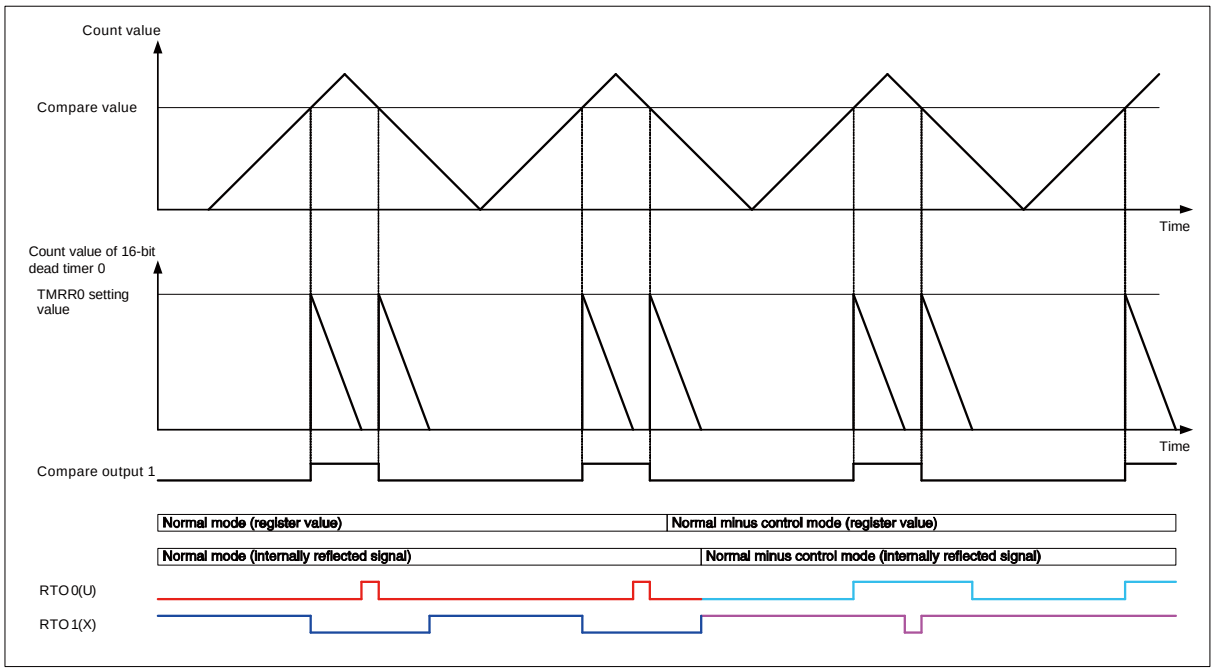


Figure 5-15 Changing from Normal Mode to Normal Minus Control Mode While the Dead Time Timer Mode Is Active #2



Operation of the DTTI pin control

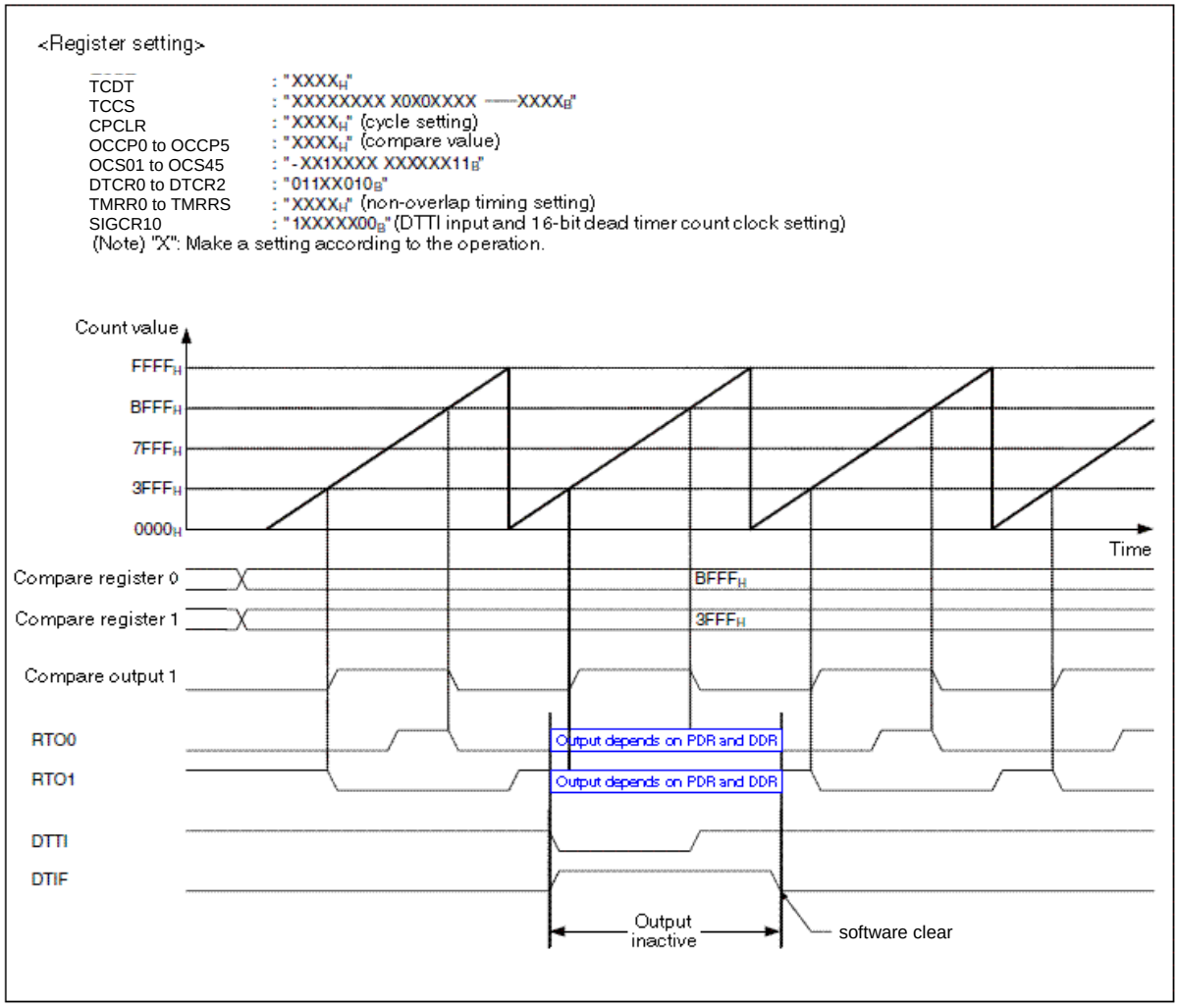
You will be able to control the RTO0 to RTO5 outputs by DTTI pins when you set the DTIE of the waveform control register 1 (SIGCR1) to "1". If "L" level of the DTTI pin is detected, RTO0 to RTO5 function as general-purpose ports until the interrupt flag (DTIF of SIGCR1 register) is cleared.

The state of general-purpose ports RT00 to RT05 during the detection of interrupt flag (DTIF) is independent of the setting of the extended port function registers (EPFR).

See "CHAPTER: I/O PORTS" for the controlling method of general-purpose ports.

Even if "L" level of the DTTI pin input is detected, the timer continues its operation while the waveform generator is active. No waveform will be output to the external pins RTO0 to RTO5.

Figure 5-16 Operation of the Valid DTTI Input



DTTI Operation of the Waveform Control Register 2 (SIGCR2)

DTTI output of the waveform control register 2 is designed to compute OR with the DTTI pin input and generate DTTI input. Therefore, if you set this register to "0", the operation is always in the DTTI input state and any inputs to the DTTI pin will be ignored.

When you clear this register by writing "1" to this register, the value of the DTTI input pin will be used.

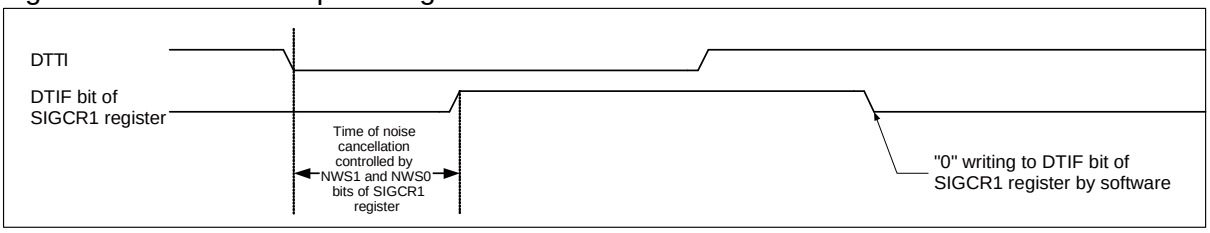
■ DTTI Pin Noise Cancellation Feature

When you set the NRSL of the waveform control register 1 (SIGCR1) to "1", the noise cancellation feature of the DTTI pin input will become effective. Once the noise cancellation feature becomes effective, the time required to have the output pins (RTO0 to RTO5) fixed to port control state will be delayed by the 4, 8, 16, or 32 peripheral clock cycles (to be selected by NWS1 and NWS0 of the SIGCR1 register). Since the noise cancellation circuit uses resources, the input will be invalid even if the DTTI input becomes effective in a mode where the oscillation stops (i.e. stop mode).

■ DTTI Interrupt

When "L" level of the DTTI is detected, the DTTI interrupt flag (DTIF of the SIGCR1 register) is set to "1" after the noise cancellation time has elapsed, and the interrupt request will be transmitted to the interrupt controller.

Figure 5-17 DTTI Interrupt Timing



Note:

If the value of NWS1 and NWS0 bits of the SIGCR1 register changes during the noise cancellation time, the greater noise cycle value (NWS1, NWS0) will become effective.
DTIF of the SIGCR1 register can be cleared by software only.

6. Notes

This section explains the notes.

Notes on Using the Waveform Generator

■ Notes on setting by a program

- When you change the value of TMD8, TMD5, TMD2, TMD7, TMD4, TMD1, TMD6, TMD3, and TMD0 bits for the 16-bit dead timer state control register (DTSCR) while the waveform generator is active (TMD2 to TMD0/TMD5 to TMD3/TMD8 to TMD6 of the DTSCR register are "001_B", "010_B" or "100_B"), make sure that counting operation of the trigger source and the 16-bit dead timer is not in progress. Without following this procedure, an unexpected waveform may be output from the RTO pin due to the output scheduled in the previous trigger. In the case of RTO output, however, it will return to the normal operation when the timer underflows or it is triggered by a new trigger source again.
- When the TMD8 to TMD0 of the DTSCR register are "001_B", the trigger source is "H" level of OUT". When the TMD8 to TMD0 bits are "010_B", it is the "rising edge of OUT". When the TMD8 to TMD0 bits are "100_B", it is the "rising or falling edge of OUT". For example, when the TMD8 to TMD0 bits change from "100_B" to "010_B", you will be able to execute the following procedure:

1. Set the 16-bit dead timer register (TMRR) to an extremely small value as "0001_H".
2. Set the output of RTO1, RTO3, and RTO5 to "L" or "H" and wait until an underflow occurs at timer 0, 1, and 2.
3. Change the mode bits (TMD8 to TMD0) and the corresponding settings.
4. The modified output waveform will be appeared at RTO pin after 1 machine cycle.

- When a value is written to the 16-bit dead timer register (TMRR) while the timer is counting, this new value will become effective in the next timer trigger. When accessing the timer register, use a half-word or word transfer instruction.
- Change the DCK2 to DCK0 of the waveform control register 1 (SIGCR1) only when the timer is not counting.
- Change the NWS1 and NWS0 of the waveform control register 1 (SIGCR1) only when the noise cancellation feature is disabled.

■ Notes on interrupts

- The interrupt flags (TMIF) must be cleared before setting "1" to the interrupt request enable bit (TMIE) of the 16-bit dead timer state control register (DTSCR).
- The interrupt flags (DTIF) must be cleared before setting "1" to the DTTI input validating bit (DTIE) of the waveform control register 1 (SIGCR1).

Chapter 49: Bus Diagnosis Function



This chapter explains the bus diagnosis function.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS44-1v0-91520-2-E

1. Overview

This section explains the overview of BUS diagnosis function.

The bus diagnosis function prevents an LSI malfunction by checking data that was output on a bus during access to each resource.

		Function
1	Diagnosis target	· Addresses and data to be output to the address and data buses, and control signals (read, write, read-modify-write, and bus access size signals) for controlling buses
2	Diagnosis bus	· AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2), Rbus(PCLK2)
3	Diagnosis method	· Diagnosis based on parity The output side calculates parity for each 8-bit group and outputs that parity. The input side checks that parity.
4	Parity	· Odd parity
5	Test function	· A parity error is generated to support program debug at bus diagnosis.
6	Error detection	· Control parity error · Address parity error · Data parity error At error occurrence, the relevant address is displayed in the register.
7	Effect	· Detection of a bus disconnection · Detection of a bus transistor failure · Detection of garbled data resulting from a bus short caused by dirt · Detection of garbled data caused by a defective contact
8	NMI notification	· Control parity error, address parity error, data parity error

2. Features

This section explains the features of BUS diagnosis function.

The internal bus of the CPU consists of AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2) and Rbus(PCLK2). The bus diagnosis function is realized by adding parity to address, data, and bus control signal for each of them, and by checking the parities. However, if a parity error occurs at write access to the bus diagnosis register, the NMI request signal cannot be cleared. So, the write access to bus diagnosis register is excluded from diagnosis.
When the parity error is detected at write access, write access to peripheral resource is shut off.

NMI output resulting from bus diagnosis is generated by one of the following factors:

- A control parity error (CNER) was detected.
- A data parity error (DER[3:0]) was detected.
- An address parity error (AER[3:0]) was detected.

3. Configuration

This section explains the Configuration of BUS diagnosis function.

Figure 3-1 Configuration Diagram

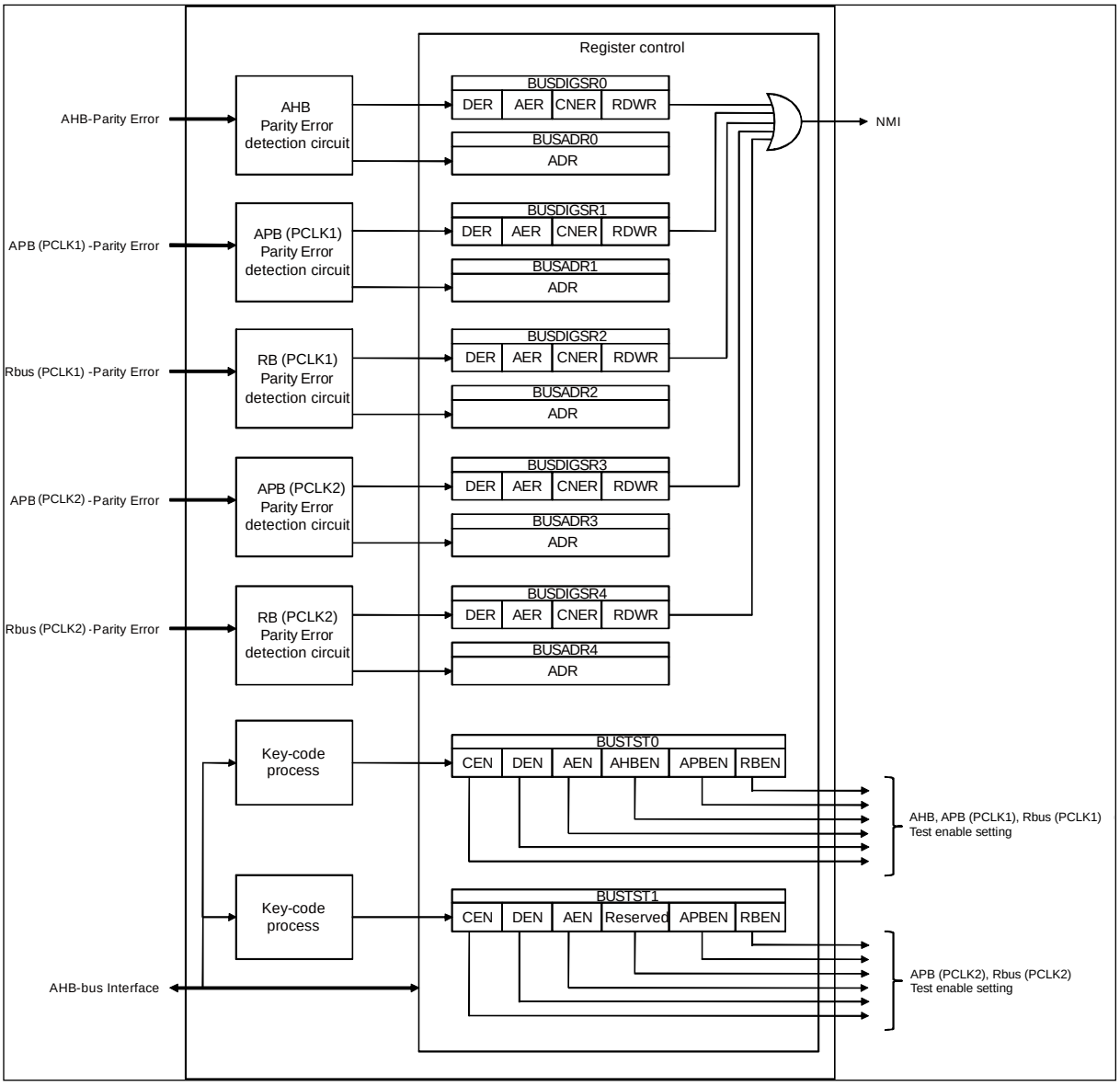
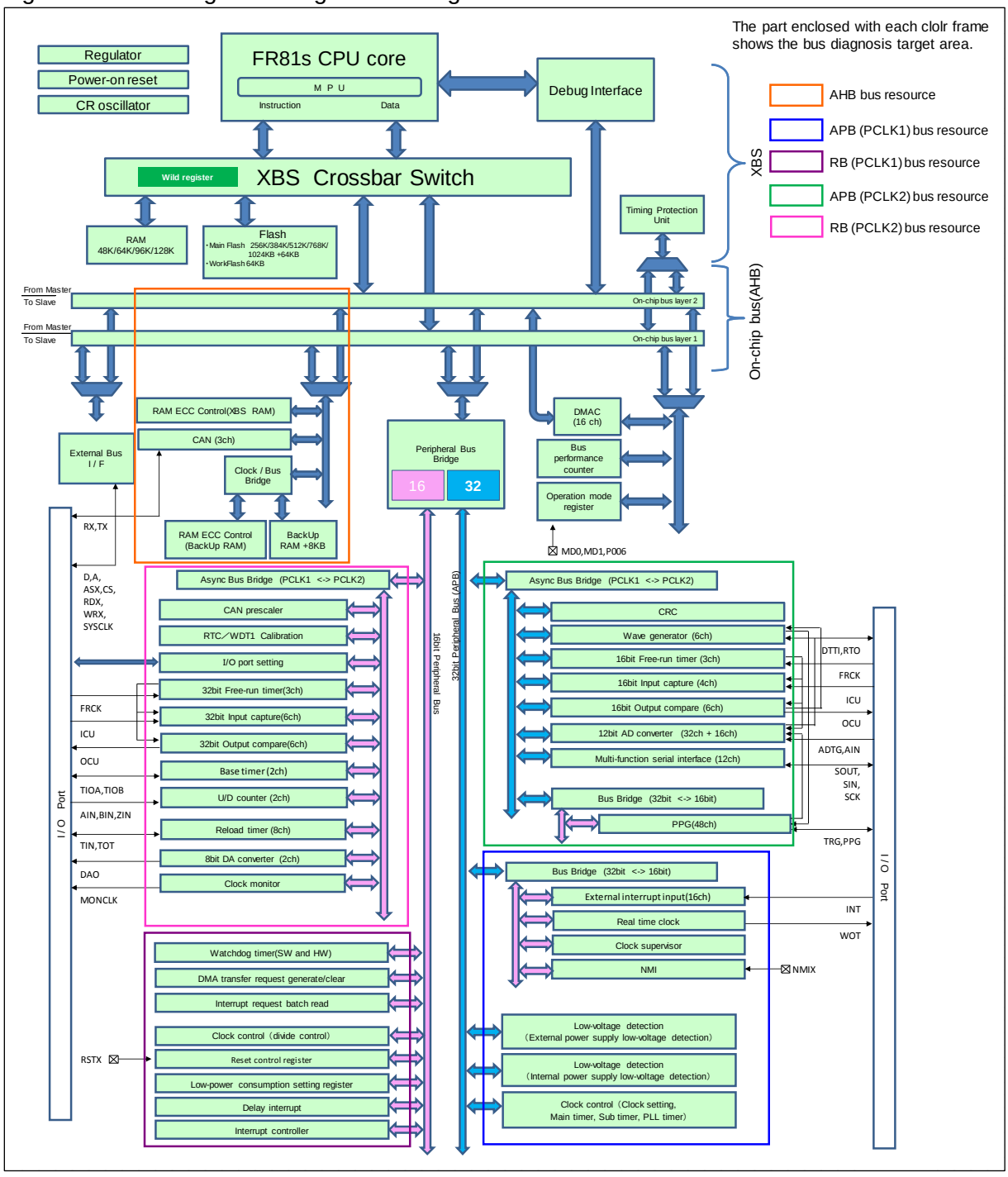


Figure 3-2 Bus Diagnosis Target Area Diagram



Note:

The following address areas are excluded from bus diagnosis target.

Watchdog timer	0x00000038 to 0x0000003F *	
Delay interrupt	0x00000044 *	
MPU	0x00000310 to 0x000003AC	To diagnosis target of reload timer 0/1/2
Interrupt controller	0x00000440 to 0x0000046F *	
Reset control/Low-power consumption control	0x00000480 to 0x00000482 *	
Clock control	0x00000488 to 0x0000048A *	
DMA transfer request by peripheral	0x00000490 to 0x0000049F *	
External bus interface	0x00000600 to 0x000006CF	
Bus performance counter	0x00000710 to 0x0000071F	
Flash memory register	0x00000840 to 0x00000843	
Wild register	0x0000085A to 0x0000085B	
	0x00000880 to 0x000008FF	
OCDU	0x00000BF0 to 0x00000BFF	
Synchronous/Asynchronous setting	0x00001000 to 0x00001001	
DMA controller	0x00000C00 to 0x00000DFF *	
Bus diagnosis	0x00003100 to 0x00003127 (Read data is a diagnosis target.)	

* : Area connected to R-bus that excluded from diagnosis target may cause error detection under some condition of register setting or failure condition.

For detail information, see "5.3 Notes".

4. Registers

This section explains the registers of bus diagnosis function.

Table 4-1 Register Map

Address	bit31 bit16	bit15 bit0	Initial value
003100_H	BUSDIGSR0 (Bus diagnosis status register 0)	BUSDIGSR1 (Bus diagnosis status register 1)	0x00000000
003104_H	BUSDIGSR2 (Bus diagnosis status register 2)	BUSTSTR0 (Bus diagnosis test register 0)	0x00000000
003108_H	BUSADR0 (Bus diagnosis address register 0)		0x00000000
$00310C_H$	BUSADR1 (Bus diagnosis address register 1)		0x00000000
003110_H	BUSADR2 (Bus diagnosis address register 2)		0x00000000

Address	bit31 bit16	bit15 bit0	Initial value
003114_H	Reserved	BUSDIGSR3 (Bus diagnosis status register 3)	0x00000000
003118_H	BUSDIGSR4 (Bus diagnosis test register 4)	BUSTSTR1 (Bus diagnosis test register 1)	0x00000000
$00311C_H$	Reserved		0x00000000
003120_H	BUSADR3 (Bus diagnosis address register 3)		0x00000000
003124_H	BUSADR4 (Bus diagnosis address register 4)		0x00000000

4.1. BUS DIaGnosis Status Register : BUSDIGSR

This section explains the bus diagnosis status register.

The bus diagnosis status register (BUSDIGSR) consists of a data parity error, address parity error, control parity error, data direction, and error flag clear.
Bus diagnosis status register 0 indicates AHB error status. Bus diagnosis status register 1 indicates APB(PCLK1) error status. Bus diagnosis status register 2 indicates Rbus(PCLK1) error status. Bus diagnosis status register 3 indicates APB(PCLK2) error status. Bus diagnosis status register 4 indicates Rbus(PCLK2) error status.

- **BUSDIGSR0: Address 3100_H (Access: Half-word, Word)**
- **BUSDIGSR1: Address 3102_H (Access: Half-word, Word)**
- **BUSDIGSR2: Address 3104_H (Access: Half-word, Word)**
- **BUSDIGSR3: Address 3116_H (Access: Half-word, Word)**
- **BUSDIGSR4: Address 3118_H (Access: Half-word, Word)**

BIT	15	14	13	12	11	10	9	8
	DER[3]	DER[2]	DER[1]	DER[0]	AER[3]	AER[2]	AER[1]	AER[0]
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

BIT	7	6	5	4	3	2	1	0
	PECLR	Reserved					CNER	RDWR
Initial values	0	0	0	0	0	0	0	0
Attributes	R0/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX

[bit15 to bit12] DER3 to DER0: Data parity error

Data parity error flag. Parity is calculated for each piece of 8-bit data. If an error occurs, the associated bit is set to "1".

For DER[3]=1, a parity error occurs in bit7 to bit0 of data.
For DER[2]=1, a parity error occurs in bit15 to bit8 of data.

For DER[1]=1, a parity error occurs in bit23 to bit16 of data.
For DER[0]=1, a parity error occurs in bit31 to bit24 of data.

If these bits are "0", this indicates that no error occurs. These bits are not updated while "1" is set in any one of these bits. If these bits are set to "1", NMI occurs.
These bits are read-only. To clear them to "0", write "1" in the PECLR bit.

Notes:

- These bits are not updated while any one of them is "1", any of the AER bits is "1", or the CNER bit is "1".
- Regarding the data parity error, error detection and notification are made only for data with a valid access size.
- Note that it may be led to infinite loop when an error is detected after the status register is read immediately after an error of the bus diagnosis status register is cleared.

[bit11 to bit8] AER3 to AER0: Address parity error

Address parity error flag. Parity is calculated for each piece of 8-bit address. If an error occurs, the associated bit is set to "1".

For AER[3]=1, a parity error occurs in bit7 to bit0 of address.
For AER[2]=1, a parity error occurs in bit15 to bit8 of address.
For AER[1]=1, a parity error occurs in bit23 to bit16 of address.
For AER[0]=1, a parity error occurs in bit31 to bit24 of address.

If these bits are "0", this indicates that no error occurs. If any one of these bits is set to "1", NMI occurs.
These bits are read-only. To clear them to "0", write "1" in the PECLR bit.

Note:

These bits are not updated while any one of them is "1", any one of the DER bits is "1", or the CNER bit is "1".

[bit7] PECLR: Parity error clear

Parity error clear bit. If "1" is written in this bit, the DER, AER, and CNER bits are set to "0". This bit is always "0" during reading. Writing "0" in this bit is ignored. This bit is write-only.

[bit6 to bit2] Reserved

Always write "0" to these bits.

[bit1] CNER: Control parity error

Control parity error bit. Read/write signals for controlling buses and the data size control signal are handled as data. If a parity error occurs, this bit is set to "1".
If this bit is "0", this indicates that no error occurs. This bit is not updated while "1" is set in this bit. If this bit is set to "1", NMI occurs.
This bit is read-only. To clear it to "0", write "1" in the PECLR bit.

Note:

This bit is not updated while this bit is "1", any one of the AER bits is "1", or the DER bit is "1".

[bit0] RDWR: Data direction

Data direction flag. If a data or address parity error occurs, this bit indicates that the error has occurred during reading or writing.

The read direction is indicated when this bit is "0".
The write direction is indicated when this bit is "1".

Notes:

- Writing "1" in the PECLR bit does not influence this bit.
- This bit is not updated while the DER, AER, or CNER bit is "1".
- This bit is valid when any one of the DER, AER, and CNER bits is "1".

Notes:

- If a diagnosis error occurs during writing to bus diagnosis status register, writing continues and the target error flag is not set to "1".

4.2. BUS diagnosis TeST Register : BUSTSTR0/1

This section explains the bus diagnosis test register.

The bus diagnosis test register (BUSTSTR0, BUSTSTR1) sets the bus diagnosis test function.

■ BUSTSTR0: Address 3106_H (Access: Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	KEY1	KEY0	-		CEN	RBEN	APBEN	AHBEN
Initial values	0	0	0	0	0	0	0	0
Attributes	R0/W	R0/W	R0,WX	R0,WX	R/W	R/W	R/W	R/W

BIT	7	6	5	4	3	2	1	0
	DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 bit14] KEY1, KEY0: Key bits

Key bits. If "00", "01", "10", and "11" are continuously written in these bits, the data is updated to the data written in bit11 to bit0. However, during this continuous writing, data in bit11 to bit0 is not updated unless the same value is written in bit11 to bit0 four times. Moreover, data is not updated if the bus diagnosis register is read during writing. In this case, writing to this register continuously four times is required again.

Example:
Write $07AA_H$ in BUSTSTR.
Next, write $47AA_H$ in BUSTSTR.
Next, write $87AA_H$ in BUSTSTR.
Next, write $C7AA_H$ in BUSTSTR. -> With this writing, BUSTSTR is set to $07AA_H$.

[bit13, bit12] Undefined
"0" is always read. Writing does not affect operation.

[bit11] CEN: Control error
Control error setting bit

If this bit is "0", the control parity is properly generated.
If this bit is "1", a control parity error occurs.

Note:

For RBEN=0, APBEN=0, and AHBEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit10] RBEN: Rbus parity error generation enable
This bit enables the generation of a Rbus (PCLK1) parity error.

If this bit is "0", the Rbus parity is generated as correct one (odd parity).
If this bit is "1", the Rbus parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]= 0000, AEN[3:0]= 0000, CEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit9] APBEN: APB parity error generation enable
This bit enables the generation of an APB (PCLK1) parity error.

If this bit is "0", the APB parity is generated as correct one (odd parity).
If this bit is "1", the APB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]= 0000, AEN[3:0]= 0000, CEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit8] AHBEN: AHB parity error generation enable
This bit enables the generation of an AHB parity error.

If this bit is "0", the AHB parity is generated as correct one (odd parity).

If this bit is "1", the AHB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]= 0000, AEN[3:0]= 0000, CEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit7 to bit4] DEN3 to DEN0: Data error

Data error setting bits.

If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated.
If DEN[3] is "1", a parity error for bit7 to bit0 of the data bus is generated.
If DEN[2] is "0", parity in bit15 to bit8 of the data bus is properly generated.
If DEN[2] is "1", a parity error for bit15 to bit8 of the data bus is generated.
If DEN[1] is "0", parity in bit23 to bit16 of the data bus is properly generated.
If DEN[1] is "1", a parity error for bit23 to bit16 of the data bus is generated.
If DEN[0] is "0", parity in bit31 to bit24 of the data bus is properly generated.
If DEN[0] is "1", a parity error for bit31 to bit24 of the data bus is generated.

Notes:

- For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.
- Only value set in valid data range of access size is set as an error.

[bit3 to bit0] AEN3 to AEN0: Address error

Address error setting bits

If AEN[3] is "0", parity in bit7 to bit0 of the address bus is properly generated.
If AEN[3] is "1", a parity error for bit7 to bit0 of the address bus is generated.
If AEN[2] is "0", parity in bit15 to bit8 of the address bus is properly generated.
If AEN[2] is "1", a parity error for bit15 to bit8 of the address bus is generated.
If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.
If AEN[1] is "1", a parity error for bit23 to bit16 of the address bus is generated.
If AEN[0] is "0", parity in bit31 to bit24 of the address bus is properly generated.
If AEN[0] is "1", a parity error for bit31 to bit24 of the address bus is generated.

Note:

For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.

Notes:

- Any interrupt is disabled during writing to the bus diagnosis test register.
- The bus diagnosis test register is used for debugging the bus diagnosis functions.
- Writing to the bus diagnosis test register is performed even if diagnosis error occurs, but the error flag of the target status register is not set to "1".

- Continue two times the same access in order to generate a parity error against an access to RB-bus resource.
- Set those bits in the bus diagnosis test register only when the test function is used.

■ BUSTSTR1: Address 311A$_H$ (Access: Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	KEY1	KEY0	-		CEN	RBEN	APBEN	Reserved
Initial values	0	0	0	0	0	0	0	0
Attributes	R0/W	R0/W	R0,WX	R0,WX	R/W	R/W	R/W	R/W0

BIT	7	6	5	4	3	2	1	0
	DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15, bit14] KEY1, KEY0: Key bits

Key bits. If "00", "01", "10", and "11" are continuously written in these bits, the data is updated to the data written in bit11 to bit0. However, during this continuous writing, data in bit11 to bit0 is not updated unless the same value is written in bit11 to bit0 four times. Moreover, data is not updated if the bus diagnosis register is read during writing. In this case, writing to this register continuously four times is required again.

Example:

Write 07AA$_H$ in BUSTSTR.
Next, write 47AA$_H$ in BUSTSTR.
Next, write 87AA$_H$ in BUSTSTR.
Next, write C7AA$_H$ in BUSTSTR. -> With this writing, BUSTSTR is set to 07AA$_H$.

[bit13, bit12] Undefined

"0" is always read. Writing does not affect operation.

[bit11] CEN: Control error

Control error setting bit

If this bit is "0", the control parity is properly generated.
If this bit is "1", a control parity error occurs.

Note:

For RBEN=0 and APBEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit10] RBEN: Rbus parity error generation enable

This bit enables the generation of a Rbus (PCLK2) parity error.

If this bit is "0", the Rbus parity is generated as correct one (odd parity).

If this bit is "1", the Rbus parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]= 0000, AEN[3:0]= 0000, CEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit9] APBEN: APB parity error generation enable

This bit enables the generation of an APB (PCLK2) parity error.

If this bit is "0", the APB parity is generated as correct one (odd parity).
If this bit is "1", the APB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]= 0000, AEN[3:0]= 0000, CEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit8] Reserved

Always write "0".

[bit7 to bit4] DEN3 to DEN0: Data error

Data error setting bits.

If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated.
If DEN[3] is "1", a parity error for bit7 to bit0 of the data bus is generated.
If DEN[2] is "0", parity in bit15 to bit8 of the data bus is properly generated.
If DEN[2] is "1", a parity error for bit15 to bit8 of the data bus is generated.
If DEN[1] is "0", parity in bit23 to bit16 of the data bus is properly generated.
If DEN[1] is "1", a parity error for bit23 to bit16 of the data bus is generated.
If DEN[0] is "0", parity in bit31 to bit24 of the data bus is properly generated.
If DEN[0] is "1", a parity error for bit31 to bit24 of the data bus is generated.

Notes:

- For RBEN= 0 and APBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.
- Only value set in valid data range of access size is set as an error.

[bit3 to bit0] AEN3 to AEN0: Address error

Address error setting bits

If AEN[3] is "0", parity in bit7 to bit0 of the address bus is properly generated.
If AEN[3] is "1", a parity error for bit7 to bit0 of the address bus is generated.
If AEN[2] is "0", parity in bit15 to bit8 of the address bus is properly generated.
If AEN[2] is "1", a parity error for bit15 to bit8 of the address bus is generated.
If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.
If AEN[1] is "1", a parity error for bit23 to bit16 of the address bus is generated.
If AEN[0] is "0", parity in bit31 to bit24 of the address bus is properly generated.

If AEN[0] is "1", a parity error for bit31 to bit24 of the address bus is generated.

Note:

For RBEN= 0 and APBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.

4.3. BUS diagnosis ADdRess Register : BUSADR

This section explains the bus diagnosis address register.

If an address parity error, data parity error, or control parity error is detected, the bus diagnosis address register (BUSADR) stores the relevant address. This register is valid when the DER, AER, or CNER bit of the bus diagnosis status register (BUSDIGSR) is "1".

Bus diagnosis address register 0 indicates an address in which an AHB diagnosis error was detected. Bus diagnosis address register 1 indicates an address in which an APB (PCLK1) diagnosis error was detected. Bus diagnosis address register 2 indicates an address in which an Rbus (PCLK1) diagnosis error was detected. Bus diagnosis address register 3 indicates an address in which an APB (PCLK2) diagnosis error was detected. Bus diagnosis address register 4 indicates an address in which an Rbus (PCLK2) diagnosis error was detected.

- **BUSADR0: Address 3108_H (Access: Word)**
- **BUSADR1: Address $310C_H$ (Access: Word)**
- **BUSADR2: Address 3110_H (Access: Word)**
- **BUSADR3: Address 3120_H (Access: Word)**
- **BUSADR4: Address 3124_H (Access: Word)**

31	30	29	28	27	26	25	24	BIT
ADR[31]	ADR[30]	ADR[29]	ADR[28]	ADR[27]	ADR[26]	ADR[25]	ADR[24]	
0	0	0	0	0	0	0	0	Initial values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

23	22	21	20	19	18	17	16	BIT
ADR[23]	ADR[22]	ADR[21]	ADR[20]	ADR[19]	ADR[18]	ADR[17]	ADR[16]	
0	0	0	0	0	0	0	0	Initial values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

15	14	13	12	11	10	9	8	BIT
ADR[15]	ADR[14]	ADR[13]	ADR[12]	ADR[11]	ADR[10]	ADR[9]	ADR[8]	
0	0	0	0	0	0	0	0	Initial values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

7	6	5	4	3	2	1	0	BIT
ADR[7]	ADR[6]	ADR[5]	ADR[4]	ADR[3]	ADR[2]	ADR[1]	ADR[0]	
0	0	0	0	0	0	0	0	Initial values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit31 to bit0] ADR31 to ADR0: Bus address

If an address or data parity error is detected, these bits indicate an address that is accessed at that detection. This register is read-only.

Notes:

- This register is valid when any one of the DER, AER, or CNER bit of the bus diagnosis status register is "1".
- This register is not updated when any one of the DER, AER, or CNER bit of the bus diagnosis status register is "1".

5. Operation

This section explains the operation of bus diagnosis function.

If an access is made from AHB, APB, or Rbus, the bus diagnosis performs parity check for the address, data, and control buses to diagnose the address, data, and control bus as being correct.
If a bus is determined as failure by AHB, APB, or Rbus, the content of each error is notified to the bus diagnosis status register and address of resource to the bus diagnosis address register, and the content of failure can be determined.
During writing, no writing is made to a peripheral resource if an address parity error, data parity error and control parity error are generated.

5.1. Error Detection

This section explains the Error detection of bus diagnosis function.

When the error is detected by the bus diagnosis, access address and access direction at the time of error detection are stored in the RDWR bit of the bus diagnosis address register (BUSADR) and of the bus diagnosis status register (BUSDISR), respectively.
Moreover, when the error by the write access is detected, the write to the resource is not done.

Address error detection

When the parity operation result of the bus address is an error, The address error detection sets "1" to AER[3:0] bit in the bus diagnosis status register (BUSDISR).
AER[3:0] bit can be cleared by writing "1" to PECLR bit in this register.

Control error detection

When the parity operation result of the bus control is an error, the control error detection sets "1" to CNER bit in the bus diagnosis status register (BUSDISR).
The CNER bit can be cleared by writing "1" to PECLR bit in this register.

Data error detection

When the parity operation result of the bus data is an error, the data error detection sets "1" in the corresponding DER[3:0] bit of the bus diagnosis status register (BUSDISR) according to the access size of the word, the half-word, and the byte.
DER[3:0] bit can be cleared by writing "1" to PECLR bit in this register.

The error detection part of the bus diagnosis status register (BUSDIGSR).DER[3:0] is shown as follows.
(❍: Error detection is done. , -: Error detection is not done.)

Access size	Address	BUSDIGSR0 (AHB: On-chip bus) BUSDIGSR1/3 (APB: 32bit peripheral bus) BUSDIGSR2/4 (R-bus: 16bit peripheral bus)			
		DER[0]	DER[1]	DER[2]	DER[3]
		Data bit31-24	Data bit23-16	Data bit15-8	Data bit7-0
Word access	Addr+0	❍*	❍*	❍*	❍*
Half-word access	Addr+0	❍	❍	-	-
Half-word access	Addr+2	-	-	❍	❍
Byte access	Addr+0	❍	-	-	-
Byte access	Addr+1	-	❍	-	-
Byte access	Addr+2	-	-	❍	-
Byte access	Addr+3	-	-	-	❍

*:R-bus: The word access to 16bit peripheral bus are treated as 2-times half-word access. Therefore, only the half-word access error detected first is notified to the register.

It is accessed in following order usually.
Upper half-word access (data bit31-16) --> lower half-word access (data bit15-0)

However, only the word access to the PPG and up/down counter is accessed in following order.
Lower half-word access (data bit15-0) --> upper half-word access (data bit31-16)

NMI request generation/stop

NMI request is continued while either error (address error (AER[3:0]), control error (CNER) or data error (DER)) of each bus has been detected.
NMI request is discontinued when all errors (address error (AER[3:0]), control error (CNER) and data error (DER)) of each bus are cleared.

Note:

BUS diagnostic function generates NMI interrupt upon detecting of bus error. NMI is an interrupt that cannot be masked (i.e., cannot be suppressed).
On software side, you must always configure an NMI process routine. If NMI process routine is undefined, and if bus failure occurs, program execution runs out of control after NMI is generated.

5.2. Test Function

This section explains the test function of bus diagnosis function.

In this function, a pseudo error can be generated if bus diagnosis test register (BUSTSTR) is used.
The key code processing is necessary for being set to this register.
If "00", "01", "10", and "11" are not continuously written in the KEY1 and KEY0 bits, the register is not set. At this time, if the same value is not written four times, the register value is not updated.
However, even if the bus diagnosis test register is set, the pseudo data error is not detected when the reading data is all "1".

Note:

ALL "1" read might detect the error according to the register setting and the access requirement.
Please refer to Notes in "■ Data error setting" for details.
Only when the test function is used, the foregoing limitation is applied.

■ Bus error setting

A pseudo error can be caused for the set bus by setting "1" to RBEN, APBEN, and AHBEN.
However, it is necessary to set "1" to either of AEN[3:0], CEN or DEN[3:0] bit at the same time.

■ Address error setting

The pseudo address error can be caused in the corresponding address bit by setting "1" to AEN[3:0].
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

■ Control error setting

The pseudo control error can be caused in the control bit by setting "1" to CEN.
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

■ Data error setting

The pseudo data error can be caused in the corresponding data bit by setting "1" to DEN[3:0].
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

Notes:

- Set only the bit corresponding to the access size when you set DEN[3:0].
- When DEN[3:0] is set to the bit that doesn't correspond to the access size, the error of the data not accessed might be detected.
- When you set DEN[3:0] to the bit that doesn't correspond to the access size, it might be different from the method of detecting the data error described in "■ Data error detection".

■ Bus diagnosis pseudo error generation procedure

The procedure that causes a pseudo error is the following.

1. Set the type of the bus error to be diagnosed in the bus diagnosis test register (BUSTSTR).
   - In the key code, write the same error setting four continuous times, "00" → "01" → "10" → "11".
2. Access the resource in the diagnosis area with the bus that does the pseudo error setting.
   - When you set the address error and the control error (When you do not set the data error.)

   ⇒ A pseudo error occurs by accessing the resource in the diagnosis area.

   - When you set the data error

   ⇒ A pseudo error occurs by accessing the resource in the diagnosis area by the access size corresponding to DEN[3:0]. (See Notes in "■ Data error setting."

5.3. Notes

This section explains the notes of bus diagnosis function.

- When bus diagnosis test function of R-bus is used

  When it accesses the following resource area in the state (RBEN was set to "1" by the bus diagnosis test register, and either AEN[3:0] and CEN or DEN[3:0] was set to "1"), the error is detected. At this time, it might seem that the bus was diagnosed.

At this time, the write access to the resource is executed.

- Watchdog timer
- Delay interrupt
- Reload timer 0/1/2
- Interrupt controller
- Reset control/power consumption control
- Clock control
- DMA transfer request by peripheral
- DMA controller

5.4. Example of Operating Bus Diagnosis

This section explains the example of operating bus diagnosis.

The example of operating the bus diagnosis is shown as follows.

- Data error detection operation
  The data bus breaks down when the write (read) is accessed to the resource by byte access[7:0].
  ⇒ Only DER[3] of the bus diagnosis status register corresponding to the resource is set to "1".

Notes:

- Even if the bus is out of order at this time by data bus[31:8], neither DER[2], DER[1] nor DER[0] are set to "1".
- Only the bit corresponding to the access size is similarly set as for the word and the half-word access.

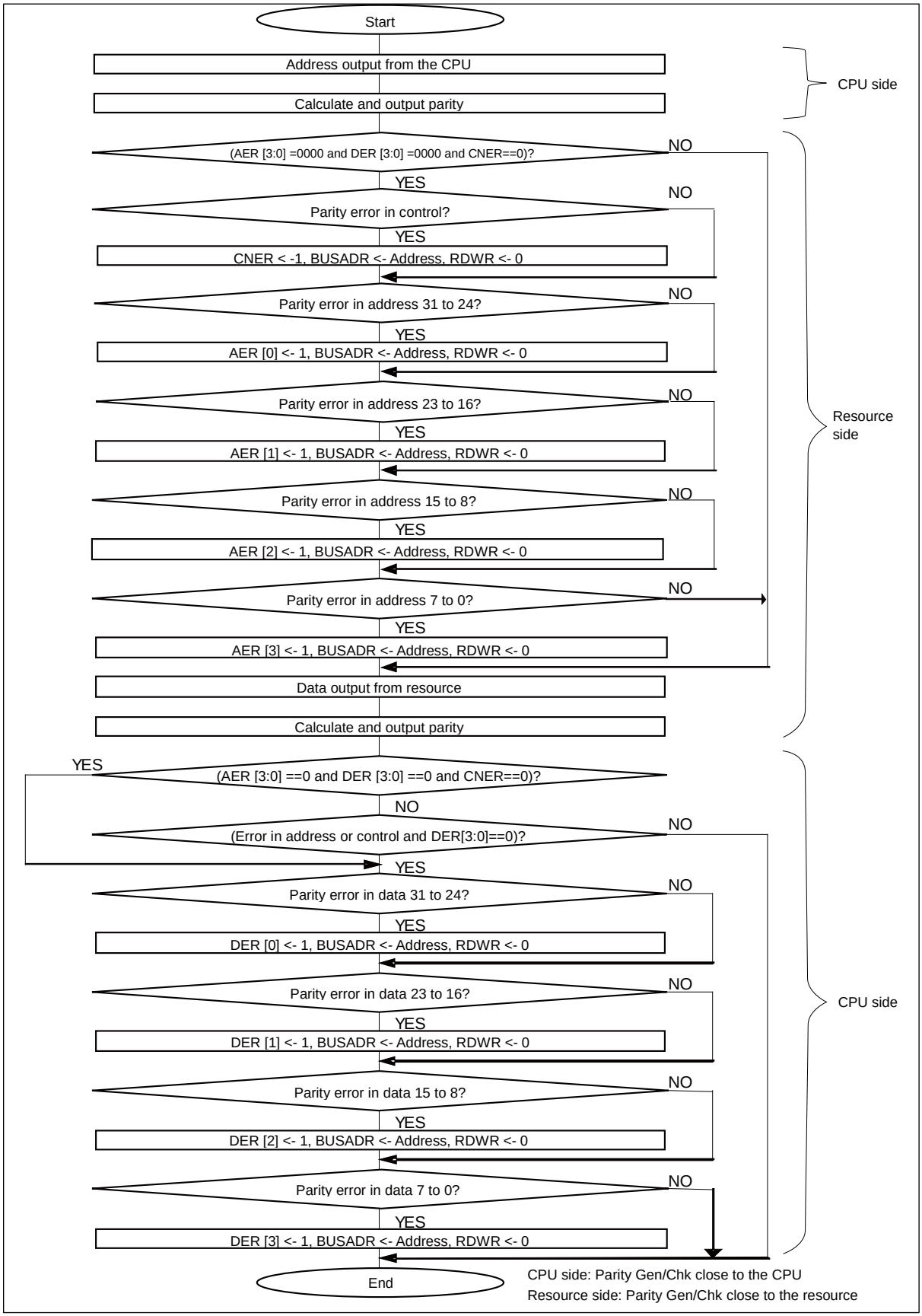
- Pseudo data error setting operation
  When you want to make the error detected in bus diagnosis status register DER[0] in the test function
  ⇒ Set "1" to bus error setting and data error setting DER[0] by the key code access, and do byte access[31:24].

When DEN[3:1] is set, and byte access [31:24] is done, the error might be detected in bus diagnosis status register DER[3:1].

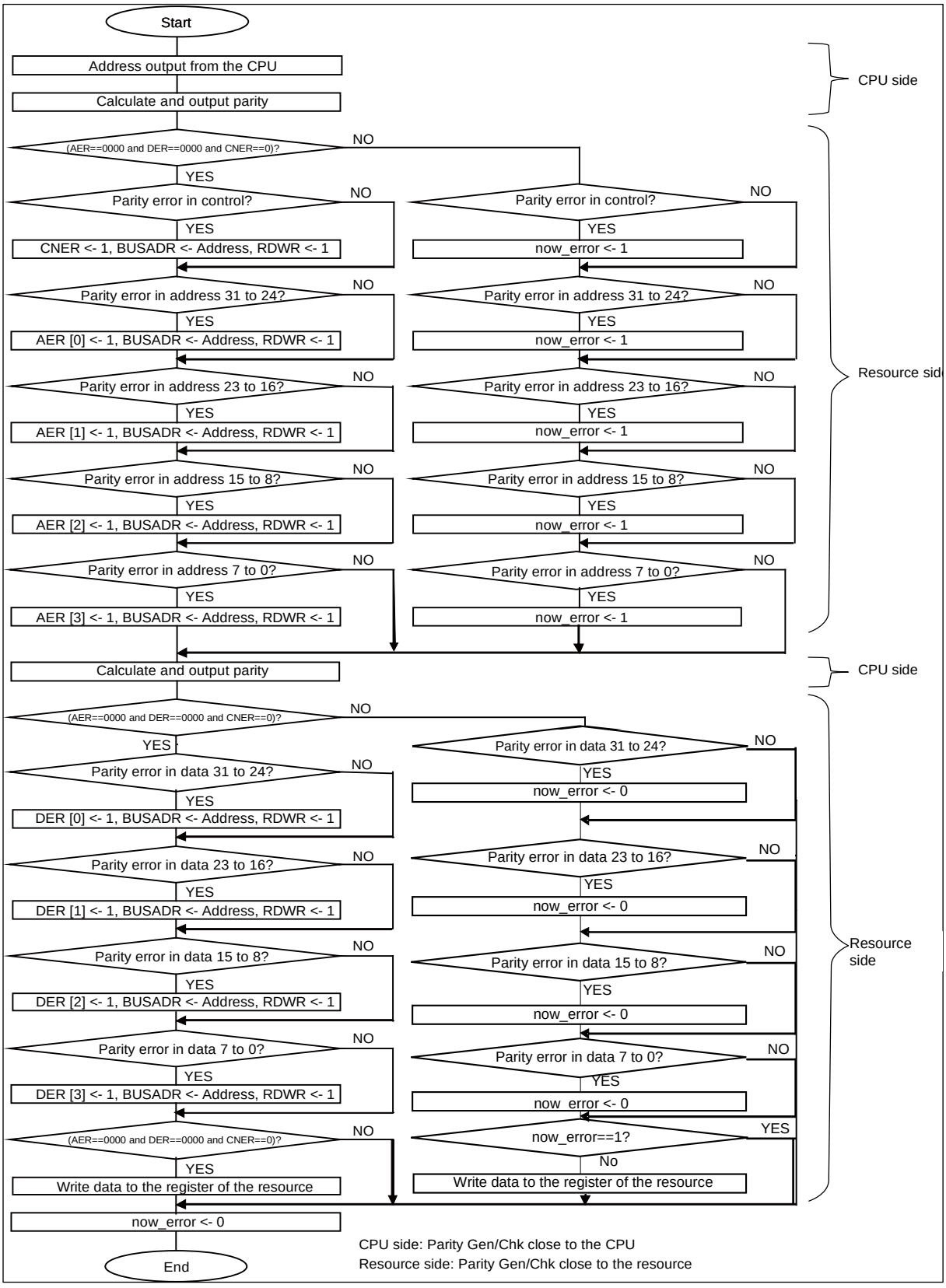
■ Bus Diagnosis Operation Flow

The operation flow of the bus diagnosis is shown below.

(1) At register reading


Start
Address output from the CPU
Calculate and output parity
CPU side
(AER [3:0] =0000 and DER [3:0] =0000 and CNER==0)?
NO
YES
Parity error in control?
NO
YES
CNER < -1, BUSADR <- Address, RDWR <- 0
Parity error in address 31 to 24?
NO
YES
AER [0] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in address 23 to 16?
NO
YES
AER [1] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in address 15 to 8?
NO
YES
AER [2] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in address 7 to 0?
NO
YES
AER [3] <- 1, BUSADR <- Address, RDWR <- 0
Data output from resource
Calculate and output parity
Resource side
YES
(AER [3:0] ==0 and DER [3:0] ==0 and CNER==0)?
NO
(Error in address or control and DER[3:0]==0)?
NO
YES
Parity error in data 31 to 24?
NO
YES
DER [0] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in data 23 to 16?
NO
YES
DER [1] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in data 15 to 8?
NO
YES
DER [2] <- 1, BUSADR <- Address, RDWR <- 0
Parity error in data 7 to 0?
NO
YES
DER [3] <- 1, BUSADR <- Address, RDWR <- 0
CPU side
End
CPU side: Parity Gen/Chk close to the CPU
Resource side: Parity Gen/Chk close to the resource

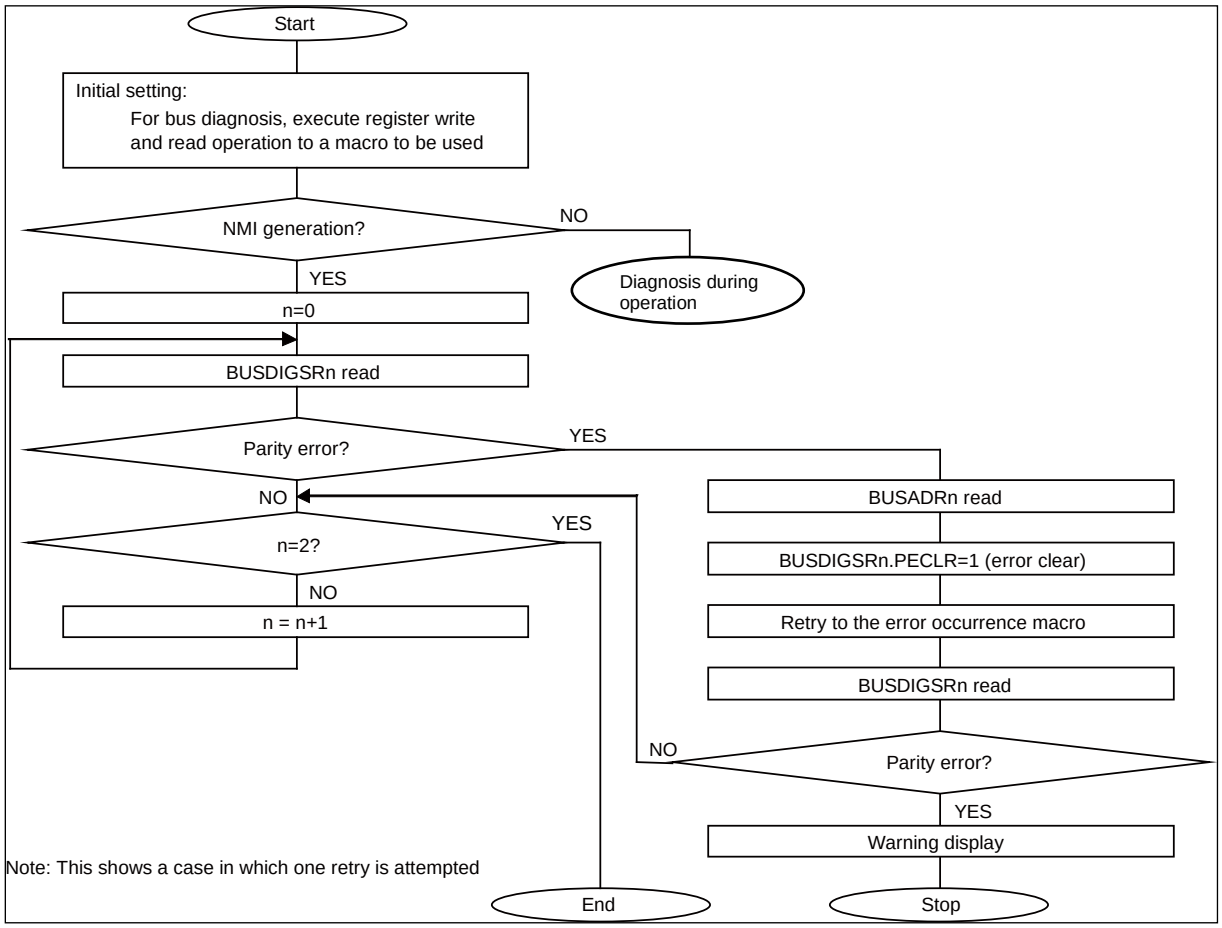
(2) At register writing


Start
Address output from the CPU
Calculate and output parity
CPU side
(AER==0000 and DER==0000 and CNER==0)?
NO
YES
Parity error in control?
NO
YES
CNER <- 1, BUSADR <- Address, RDWR <- 1
Parity error in control?
NO
YES
now_error <- 1
Parity error in address 31 to 24?
NO
YES
AER [0] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in address 31 to 24?
NO
YES
now_error <- 1
Parity error in address 23 to 16?
NO
YES
AER [1] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in address 23 to 16?
NO
YES
now_error <- 1
Parity error in address 15 to 8?
NO
YES
AER [2] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in address 15 to 8?
NO
YES
now_error <- 1
Parity error in address 7 to 0?
NO
YES
AER [3] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in address 7 to 0?
NO
YES
now_error <- 1
Resource sid
Calculate and output parity
CPU side
(AER==0000 and DER==0000 and CNER==0)?
NO
YES
Parity error in data 31 to 24?
NO
YES
DER [0] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in data 31 to 24?
NO
YES
now_error <- 0
Parity error in data 23 to 16?
NO
YES
DER [1] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in data 23 to 16?
NO
YES
now_error <- 0
Parity error in data 15 to 8?
NO
YES
DER [2] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in data 15 to 8?
NO
YES
now_error <- 0
Parity error in data 7 to 0?
NO
YES
DER [3] <- 1, BUSADR <- Address, RDWR <- 1
Parity error in data 7 to 0?
NO
YES
now_error <- 0
(AER==0000 and DER==0000 and CNER==0)?
NO
YES
Write data to the register of the resource
now_error==1?
YES
No
Write data to the register of the resource
Resource side
now_error <- 0
End
CPU side: Parity Gen/Chk close to the CPU
Resource side: Parity Gen/Chk close to the resource

(3) Processing at error detection

The following gives an example of processing at error detection.

· At Initialization



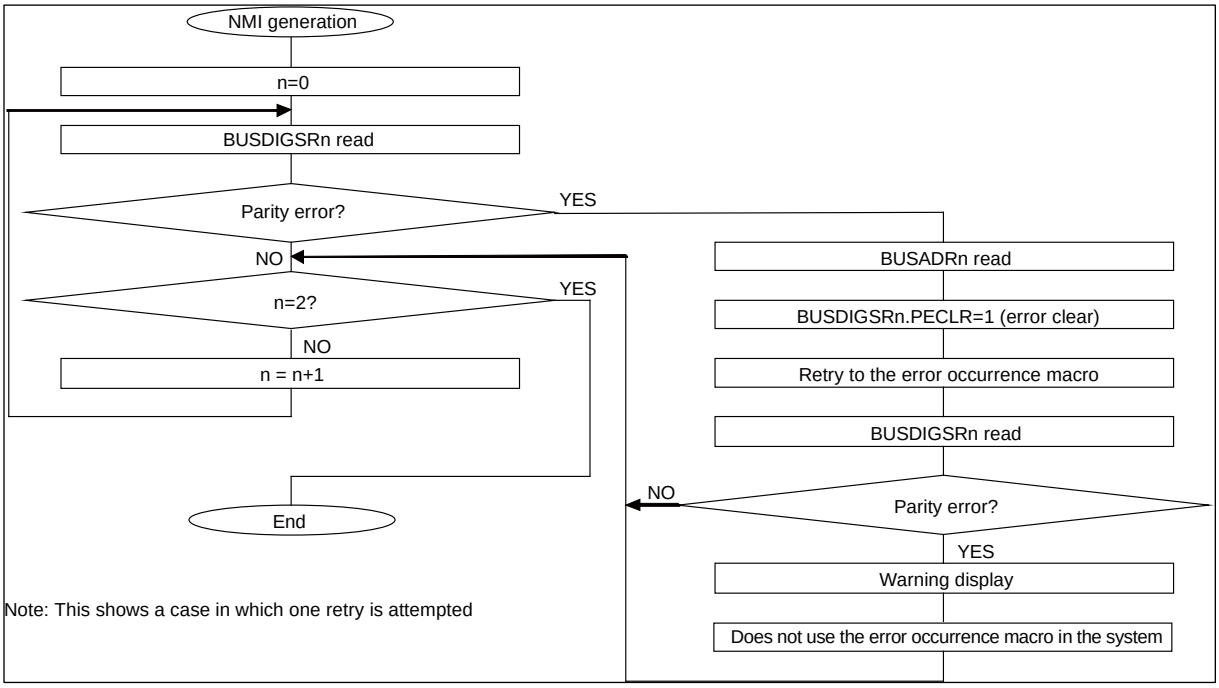
Note:

Before setting the ILM interrupt level mask register to 16 (10000_B) or more, set the address of the interrupt function to the interrupt vector.
If the program is executed in following steps while the bus signal continuously malfunctions, CPU refers to the interrupt function address immediately after the ILM setting.

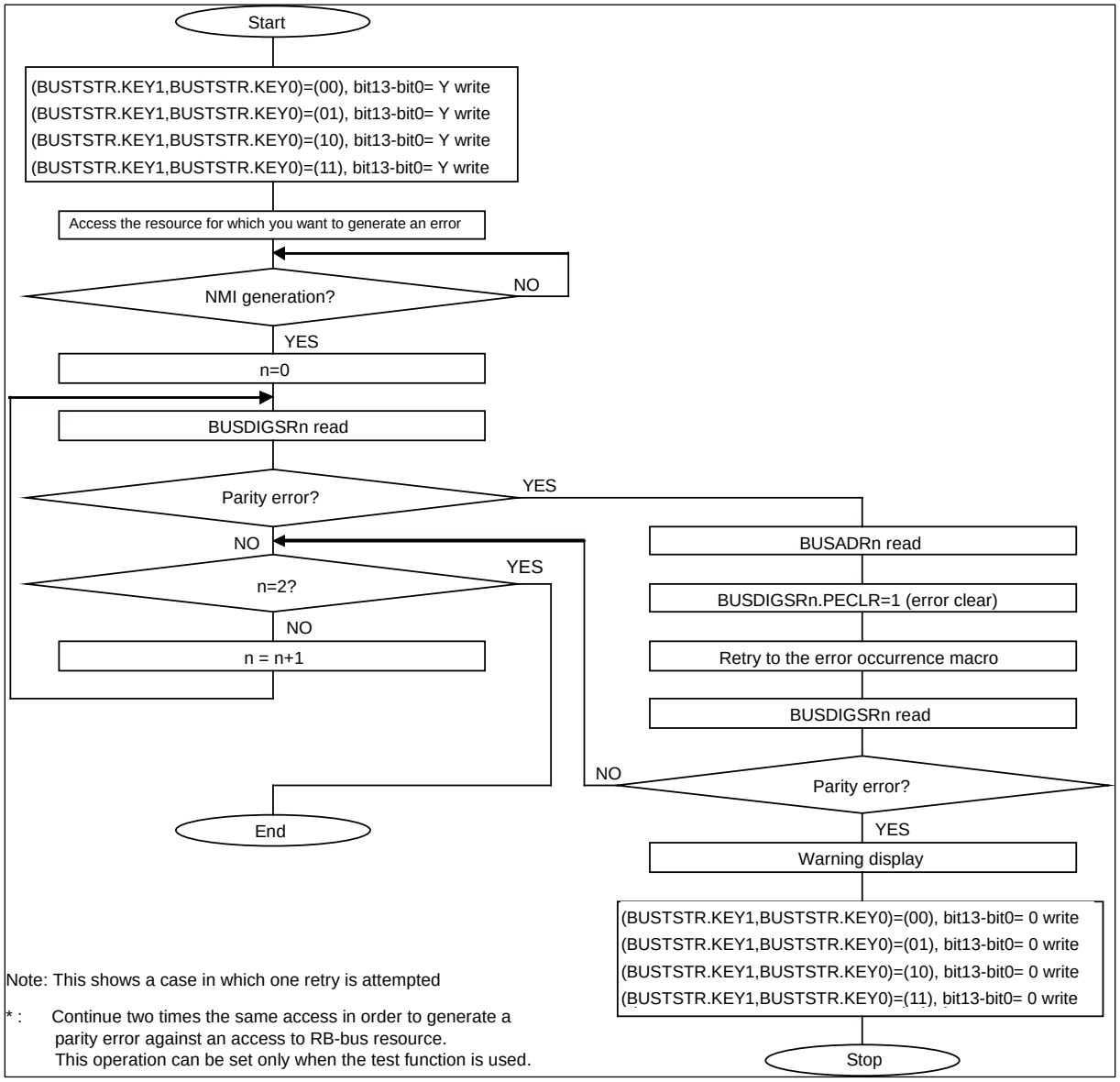
Example:

1. Define the vector area.
2. Clear the Stack
3. Set the pin register
4. Set the interrupt mask register (ILM: interrupt mask register)
   The NMI of the bus diagnosis is detected in this step, then the program is terminated.
5. Set the interrupt function address for the interrupt vector.

- During Operation



- In the Test Mode



Note: This shows a case in which one retry is attempted

* : Continue two times the same access in order to generate a parity error against an access to RB-bus resource. This operation can be set only when the test function is used.

Chapter 50: RAM Diagnosis Function



This chapter explains the RAM diagnosis function.

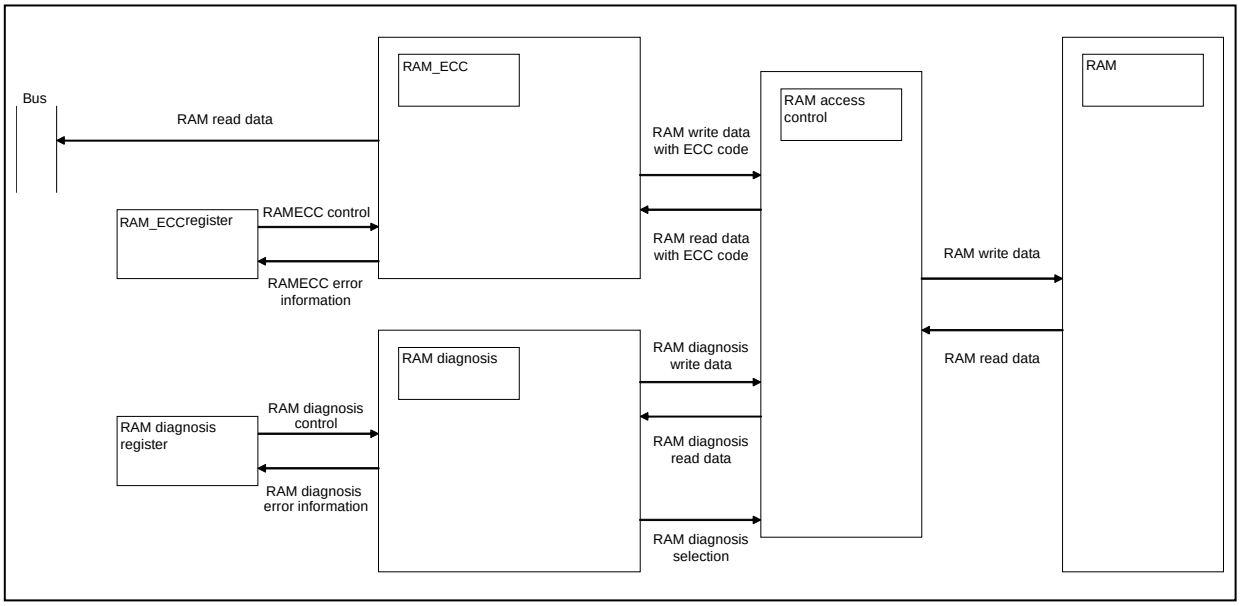
1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FS28_DIAG-3v1-91520-3-E

1. Overview

This section is explains overview of RAM diagnosis function.

The RAM diagnosis block perform diagnosis of RAM and initialization of RAMs.



For RAMECC functions, see "CHAPTER: RAMECC FUNCTION".

2. Features

This section explains feature of RAM diagnosis function.

■ Target RAM

- XBS RAM
  - MB91F522 : 48K bytes
  - MB91F523 : 48K bytes
  - MB91F524 : 64K bytes
  - MB91F525 : 96K bytes
  - MB91F526 : 128K bytes
- BackupRAM : 8K bytes

■ RAM Diagnosis

The RAM diagnoses listed below are selected and executed. (Two or more items selectable)

Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
Checker

March (all "0" -> all "1" are executed in that order.)

Interrupt function

An interrupt signal for a diagnosis end factor is generated. (RAM diagnosis end interrupt)
An interrupt signal is generated when an error is detected. (RAM diagnosis error interrupt)

■ RAM Initialization

One of the following RAM initializations is selected and executed.

Write all "0"
Write all "1"

Interrupt function

An interrupt signal for an initialization end factor is generated. (RAM initialization complete interrupt)

■ Test mode

In this mode, fake TEST error will be generated for software debugging.

3. Configuration

This section explains the configuration of RAM diagnosis function.

Figure 3-1 Block Diagram of XBS RAM Diagnosis Function (Configuration)

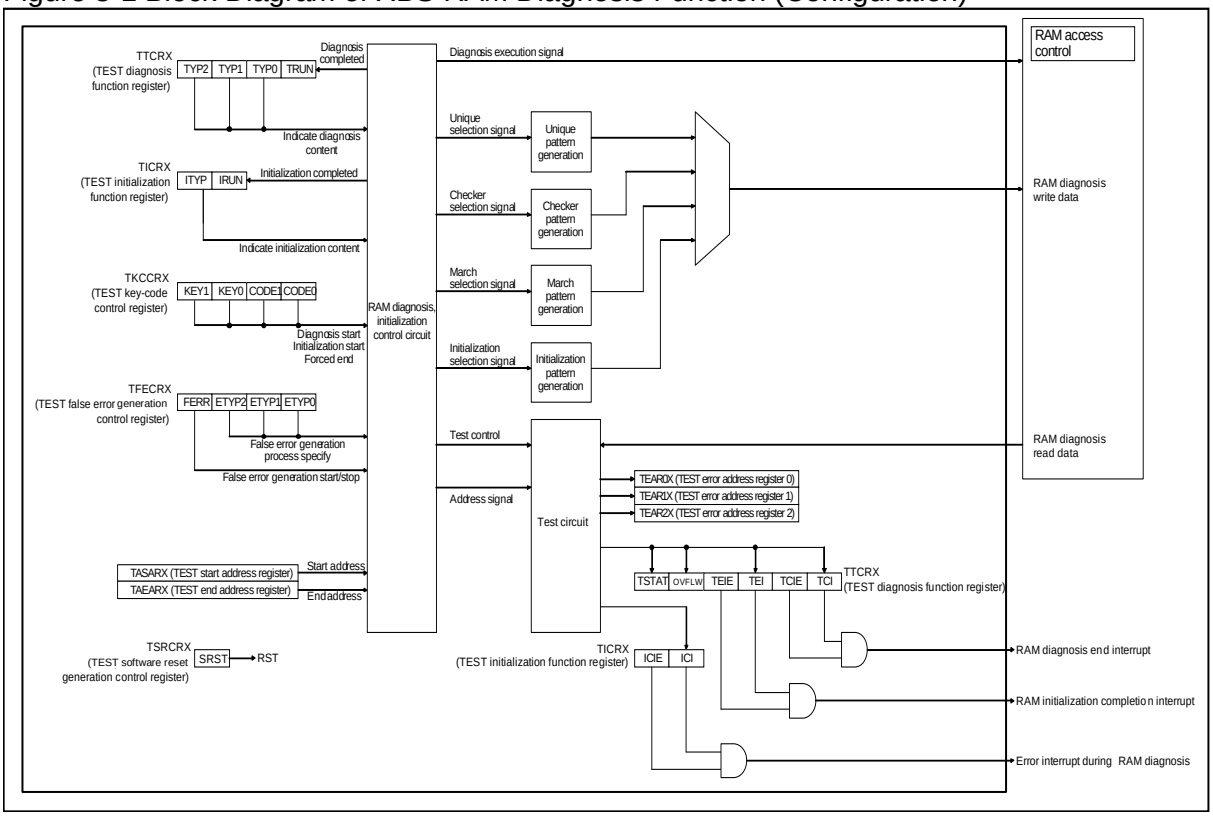
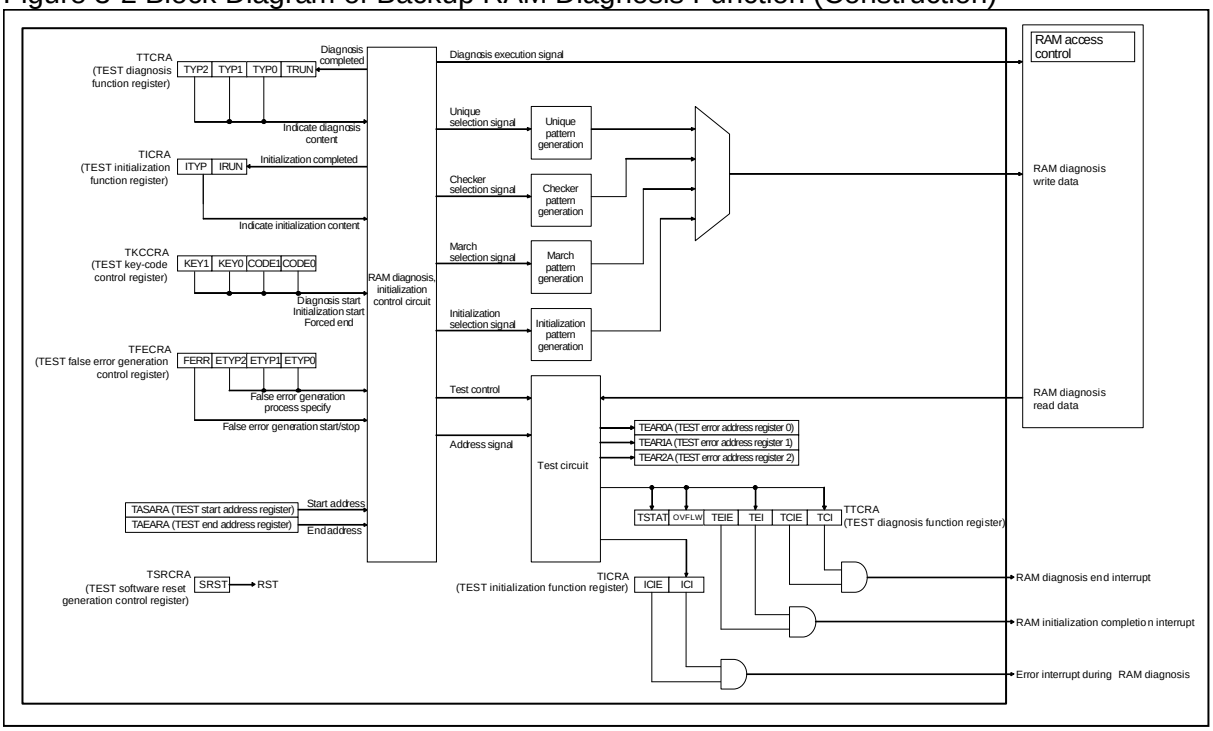


Figure 3-2 Block Diagram of Backup RAM Diagnosis Function (Construction)



4. Registers

This section explains the registers of RAM diagnosis function.

Table 4-1 Register Map

Address	Registers				Functions
	+0	+1	+2	+3	
0x300C	TEAR0X				TEST error address register 0 XBS RAM
0x3010	TEAR1X				TEST error address register 1 XBS RAM
0x3014	TEAR2X				TEST error address register 2 XBS RAM
0x3018	TAEARX		TASARX		TEST end address register XBS RAM TEST start address register XBS RAM
0x301C	TFECRX	TICRX	TTCRX		TEST fake error generation control register XBS RAM TEST initialization function register XBS RAM TEST diagnosis function register XBS RAM
0x3020	TSRCRX	Reserved	Reserved	TKCCRX	TEST software reset generation control register XBS RAM TEST key code control register XBS RAM
0x3030	TEAR0A				TEST error address register 0 BACKUP-RAM
0x3034	TEAR1A				TEST error address register 1 BACKUP-RAM
0x3038	TEAR2A				TEST error address register 2 BACKUP-RAM
0x303C	TAEARA		TASARA		TEST end address register BACKUP-RAM TEST start address register BACKUP-RAM
0x3040	TFECRA	TICRA	TTCRA		TEST fake error generation control register BACKUP-RAM TEST initialization function register BACKUP-RAM TEST diagnosis function register BACKUP-RAM
0x3044	TSRCRA	Reserved	Reserved	TKCCRA	TEST software reset generation control register BACKUP-RAM TEST key code control register BACKUP-RAM

4.1. TEST Error Address Register 0 XBS RAM : TEAR0X

This section explains the bit structure of TEST error address register 0 XBS RAM.

If an error occurs during RAM diagnosis for XBS RAM, TEST error address register 0 (TEAR0X) holds the relevant address.

■ TEAR0X: Address $300C_H$ (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit15] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis for XBS RAM, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0001_0000_H) + (Offset address set with TEAR0X + 2'b00)

4.2. TEST Error Address Register 1 XBS RAM : TEAR1X

This section explains the bit structure of TEST Error Address Register 1 XBS RAM.

Only when an error occurs in the address different from that held in TEAR0X during RAM diagnosis for XBS RAM, TEST error address register 1 (TEAR1X) holds that address.

■ TEAR1X: Address 3010_H (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER31 to TER29: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit15] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0001_0000_H) + (Offset address set with TEAR1X + 2'b00)

4.3. TEST Error Address Register 2 XBS RAM : TEAR2X

This section explains the bit structure of TEST Error Address Register 2 XBS RAM.

Only when an error occurs in the address different from that held in TEAR0X and TEAR1X during RAM diagnosis for XBS RAM, TEST error address register 2 (TEAR2X) holds that address.

■ TEAR2X: Address 3014_H (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit15] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0001_0000_H) + (Offset address set with TEAR2X + 2'b00)

4.4. TEST Start Address Register XBS RAM : TASARX

This section explains the bit structure of TEST Start Address Register XBS RAM.

TEST start address register (TASARX) specifies the start address of RAM diagnosis and initialization for XBS RAM.

■ TASARX: Address $301A_H$ (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved	ST14	ST13	ST12	ST11	ST10	ST9	ST8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	7	6	5	4	3	2	1	0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Reserved bit. These bits read out "0". Write "0" when writing.

[bit14 to bit0] ST14 to ST0: RAM diagnosis start address bits

These bits are used to specify the address from which the RAM diagnosis and initialization start for XBS RAM.

Note:

Setting of a value outside the XBS RAM area and a value that sets TASARX.ST14 to ST0 > TAEARX.ED14 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0001_0000_H) + (Offset address set with TASARX + 2'b00)

4.5. TEST End Address Register XBS RAM : TAEARX

This section explains the bit structure of TEST End Address Register XBS RAM.

TEST end address register (TAEARX) specifies the end address of RAM diagnosis and initialization for XBS RAM.

■ TAEARX: Address 3018_H (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved	ED14	ED13	ED12	ED11	ED10	ED9	ED8
Initial values	0	1	1	1	1	1	1	1
Attributes	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	7	6	5	4	3	2	1	0
	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Initial values	1	1	1	1	1	1	1	1
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Reserved bit. These bits read out "0". Write "0" when writing.

[bit14 to bit0] ED14 to ED0: RAM diagnosis end address bits

These bits are used to specify the address with which the RAM diagnosis and initialization end for XBS RAM.

Note:

Setting of a value outside the XBS RAM area and a value that sets TASARX.ST14 to ST0 > TAEARX.ED14 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0001_0000_H) + (Offset address set with TAEARX + 2'b11)

4.6. TEST Diagnosis Function Register XBS RAM : TTCRX

This section explains the bit structure of TEST Diagnosis Function Register XBS RAM.

The TEST diagnosis function register (TTCRX) specifies the RAM diagnosis content for XBS RAM, and holds the diagnosis result and its status.

■ TTCRX: Address 301E$_H$ (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved						TSTAT	OVFLW
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	TEIE	TEI	TCIE	TCI	TTYP2	TTYP1	TTYP0	TRUN
Initial values	0	0	0	0	1	1	0	0
Attributes	R/W	R(RM1),W	R/W	R (RM1), W	R/W	R/W	R/W	R, WX

[bit15 to bit10] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit9] TSTAT: RAM diagnosis error detection bit

TSTAT	Function
0	No error is detected with the RAM diagnosis
1	An error is detected with the RAM diagnosis

If an error occurs during RAM diagnosis for XBS RAM, this bit is set to "1".

This bit is initialized (cleared to "0") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit8] OVFLW: RAM diagnosis error overflow bit

OVFLW	Function
0	During the RAM diagnosis, an error occurs in three or less addresses
1	During the RAM diagnosis, an error occurs in four or more addresses

If a RAM diagnosis error for XBS RAM occurs in four or more addresses, this bit is set to "1".

This bit is initialized (cleared to "0") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit7] TEIE: Interrupt enable bit resulting from a diagnosis error

TEIE	Function
0	Prohibition of an interrupt resulting from a diagnosis error
1	Enabling of an interrupt resulting from a diagnosis error

This bit is used to enable an interrupt resulting from a RAM diagnosis error for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis error.

"1": Enables an interrupt resulting from a RAM diagnosis error. If TTCRX.TEI=1 is set and the RAM diagnosis ends, the interrupt signal (RAM diagnosis error interrupt) is output.

[bit6] TEI: Diagnosis error generation bit

TEI	Function
0	Read: No error occurrence during the RAM diagnosis. Write: Flag clearing.
1	Read: Error occurred during the RAM diagnosis. Write: No influence on the operation.

If TTCRX.TSTAT=1 is set when RAM diagnosis end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set TTCRX.TSTAT=1 when RAM diagnosis is ended.
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit5] TCIE: Interrupt enable bit for a diagnosis end factor

TCIE	Function
0	Prohibition of an interrupt for the diagnosis end factor
1	Enabling of an interrupt for the diagnosis end factor

This bit is used to enable an interrupt for the RAM diagnosis end factor for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis end.
"1": Enables an interrupt resulting from a RAM diagnosis end. The interrupt signal (RAM diagnosis end interrupt) is output with TTCRX.TCI= 1.

[bit4] TCI: Diagnosis end bit

TCI	Function
0	Read: The RAM diagnosis does not end. Write: Flag clearing.
1	Read: The RAM diagnosis ended. Write: No influence on the operation.

If RAM diagnosis end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM diagnosis is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit3 to bit1] TTYP2 to TTYP0: RAM diagnosis content indication bit

These bits are used to set the RAM diagnosis type for XBS RAM to be executed.
The RAM diagnosis types are executed in the following order.

1. Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
2. Checker
3. March (all "0" -> all "1" are executed in that order.)

These bits are used to determine whether or not each type is executed.

TTYP2	TTYP1	TTYP0	Function
1	1	0	Execution of unique and checker
-	-	1	Execution of march
-	1	-	Execution of checker
1	-	-	Execution of unique

By default, the unique and checker diagnoses are executed (110_B). However, to change the RAM diagnosis content, be sure to specify this change before the RAM diagnosis operation start instruction.
If march is executed last, the RAM content is all "1".

[bit0] TRUN: RAM diagnosis operation status bit

TRUN	Function
0	The RAM diagnosis is stopping
1	The RAM diagnosis is in progress

This bit is used to set or hold the RAM diagnosis status for XBS RAM.

"1": Set when a RAM diagnosis is started by the key code setting.
"0": Set when all diagnoses are complete or forcibly terminated by the key code.

4.7. TEST Initialization Function Register XBS RAM : TICRX

This section explains the bit structure of TEST Initialization Function Register XBS RAM.

The TEST initialization function register (TICRX) specifies the RAM initialization content, and holds the initialization result and its status for XBS RAM.

■ TICRX: Address $301D_H$ (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	Reserved				ICIE	ICI	ITYP	IRUN
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R (RM1), W	R/W	R, WX

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit3] ICIE: Interrupt enable bit for a RAM initialization end factor

ICIE	Function
0	Prohibition of an interrupt for a RAM initialization end factor
1	Enabling of an interrupt for a RAM initialization end factor

This bit is used to enable an interrupt for the RAM initialization end factor for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM initialization end.
"1": Enables an interrupt resulting from a RAM initialization end. The interrupt signal (RAM initialization complete interrupt) is output with TICRX.ICI= 1.

[bit2] ICI: RAM initialization end bit

ICI	Function
0	Read: The RAM initialization does not end. Write: Flag clearing.
1	Read: The RAM initialization ended. Write: No influence on the operation.

If RAM initialization end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM initialization is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit1] ITYP: RAM initialization content indication bit

ITYP	Function
0	Initialization to All "0"
1	Initialization to All "1"

This bit is used to set the type to be executed during RAM initialization for XBS RAM.

"0": Initializes to all "0".
"1": Initializes to all "1".

[bit0] IRUN: RAM initialization operation status bit

IRUN	Function
0	RAM Initialization is stopping
1	RAM Initialization is in progress

This bit is used to set or hold the RAM initialization status for XBS RAM.

"1": Set when RAM initialization is started by the key code setting.
"0": Set when all initialization is completed or forcibly terminated by the key code.

4.8. TEST Software Reset Generation Control Register XBS RAM : TSRCRX

This section explains the bit structure of TEST Soft Reset Generation Control Register XBS RAM.

The TEST software reset generation control register (TSRCRX) specifies the generation of the software reset for initializing internal circuits for XBS RAM's RAM diagnosis.

■ TSRCRX: Address 3020_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	SRST	Reserved						
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

[bit7] SRST: Software reset enabling bit

SRST	Function
0	Prohibition of a software reset
1	Enabling of a software reset

This bit is used to enable a software reset for the internal circuit for RAM diagnosis for XBS RAM.

This bit reads out "0".
"1": Reset pulses occur for 4τ only and the internal circuit for RAM diagnosis except this register is reset.
τ: Peripheral clock

[bit6 to bit0] Reserved
Reserved bits. These bits read out "0". Write "0" when writing.

4.9. TEST Fake Error Generation Control Register XBS RAM : TFECRX

This section explains the bit structure of TEST Fake Error Generation Control Register XBS RAM.

TEST fake error generation control register (TFECRX) generates a fake error in RAM diagnosis operation for XBS RAM. You can specify RAM diagnosis operations for which you want to generate an error.

■ TFECRX: Address $301C_H$ (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	Reserved				FERR	ETYP2	ETYP1	ETYP0
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved
Reserved bits. These bits read out "0". Write "0" when writing.

[bit3] FERR: Fake error enable bit for RAM diagnosis

FERR	Function
0	Prohibition of a fake error (normal operation)
1	Enabling of a fake error

This bit is used to enable a fake error for RAM diagnosis for XBS RAM.

"0": Prohibits a fake error. (normal operation)
"1": Enables a fake error. Data write including intentional error is enabled following ETYP2 to ETYP0.

[bit2 to bit0] ETYP2 to ETYP0: Fake error process specification bits

These bits are used to specify a process to generate a fake error.

ETYP2	ETYP1	ETYP0	Process to generate a fake error
-	-	1	March diagnosis
-	1	-	Checker diagnosis
1	-	-	Unique diagnosis

4.10. TEST Key Code Control Register XBS RAM : TKCCRX

This section explains the bit structure of TEST Key Code Control Register XBS RAM.

The TEST key code control register (TKCCRX) is used to start or forcibly terminate the RAM diagnosis or initialization for XBS RAM.

■ TKCCRX: Address 3023_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	KEY1	KEY0	Reserved				CODE1	CODE0
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W

[bit7, bit6] KEY1, KEY0: Key code control bits

Key code control bits. Set the operation instruction content to CODE[1:0] (no change during operation) and perform the operation.

The procedure is:

1. 00 -> 01 -> 10 -> 11 : Write in this order.
2. Same values in CODE[1:0].
3. Different operations (access or read other registers for RAM diagnosis, or continuous write in the different order other than the above) within the procedure will be invalid.

Note:

The key code process will be continued even if any access to the registers in the RAMECC is made in the procedure.

[bit5 to bit2] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit1, bit0] CODE1, CODE0: RAM diagnosis/initialization control bits

These bits specify operational direction for the key code procedure above.

CODE1, CODE0	Function
00	Forced termination
01	Initialization start
10	Diagnosis start
11	Setting prohibited

If this value is changed or set to "11" during operating the key code above, the key code procedure itself will be invalid.

4.11. TEST Error Address Register 0 BACKUP-RAM : TEAR0A

This section explains the bit structure of TEST Error Address Register 0 BACKUP-RAM.

If an error occurs during RAM diagnosis for Backup RAM, TEST error address register 0 (TEAR0A) holds the relevant address.

■ TEAR0A: Address 3030_H (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved					D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for Backup RAM, these bits hold a diagnosis pattern for which the error occurred. D10 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D10 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis

TER2	TER1	TER0	Function
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit11] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit10 to bit0] D10 to D0: Error generation address bits

During RAM diagnosis for Backup RAM, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0000_4000_H) + (Offset address set with TEAR0A + 2'b00)

4.12. TEST Error Address Register 1 BACKUP-RAM : TEAR1A

This section explains the bit structure of TEST Error Address Register 1 BACKUP-RAM.

Only when an error occurs in the address different from that held in TEAR0A during RAM diagnosis for Backup RAM, TEST error address register 1 (TEAR1A) holds that address.

■ TEAR1A: Address 3034_H (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved					D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER31 to TER29: Diagnosis error factor specification bits

During RAM diagnosis for Backup RAM, these bits hold a diagnosis pattern for which the error occurred. D10 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D10 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit11] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit10 to bit0] D10 to D0: Error generation address bits

During RAM diagnosis for Backup RAM, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.

The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0000_4000_H) + (Offset address set with TEAR1A + 2'b00)

4.13. TEST Error Address Register 2 BACKUP-RAM : TEAR2A

This section explains the bit structure of TEST Error Address Register 2 BACKUP-RAM.

Only when an error occurs in the address different from that held in TEAR0A and TEAR1A during RAM diagnosis for Backup RAM, TEST error address register 2 (TEAR2A) holds that address.

■ TEAR2A: Address 3038_H (Access: Byte, Half-word, Word)

BIT	31	30	29	28	27	26	25	24
	TER2	TER1	TER0	Reserved				
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	23	22	21	20	19	18	17	16
	Reserved							
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

BIT	15	14	13	12	11	10	9	8
	Reserved					D10	D9	D8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial values	0	0	0	0	0	0	0	0
Attributes	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for Backup RAM, these bits hold a diagnosis pattern for which the error occurred. D10 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D10 to D0 are invalid with no error generated
-	-	1	An error occurs during march diagnosis
-	1	-	An error occurs during checker diagnosis
1	-	-	An error occurs during unique diagnosis

These bits are initialized (cleared to "000") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit28 to bit11] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit10 to bit0] D10 to D0: Error generation address bits

During RAM diagnosis, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.

(Absolute address) = (0000_4000_H) + (Offset address set with TEAR2A + 2'b00)

4.14. TEST Start Address Register BACKUP-RAM : TASARA

This section explains the bit structure of TEST Start Address Register BACKUP-RAM.

TEST start address register (TASARA) specifies the start address of RAM diagnosis and initialization for Backup RAM.

■ TASARA: Address $303E_H$ (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved					ST10	ST9	ST8
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W

BIT	7	6	5	4	3	2	1	0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Initial values	0	0	0	0	0	0	0	0
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit11] Reserved

Reserved bit. These bits read out "0". Write "0" when writing.

[bit10 to bit0] ST10 to ST0: RAM diagnosis start address bits

These bits are used to specify the address from which the RAM diagnosis and initialization start for Backup RAM.

Note:

Setting of a value outside the Backup RAM area and a value that sets TASARA.ST10 to ST0 > TAEARA.ED10 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0000_4000_H) + (Offset address set with TASARA + 2'b00)

4.15. TEST End Address Register BACKUP-RAM : TAEARA

This section explains the bit structure of TEST End Address Register BACKUP-RAM.

TEST end address register (TAEARA) specifies the end address of RAM diagnosis and initialization for Backup RAM.

■ TAEARA: Address $303C_H$ (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved					ED10	ED9	ED8
Initial values	0	0	0	0	0	1	1	1
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W

BIT	7	6	5	4	3	2	1	0
	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Initial values	1	1	1	1	1	1	1	1
Attributes	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit11] Reserved

Reserved bit. These bits read out "0". Write "0" when writing.

[bit10 to bit0] ED10 to ED0: RAM diagnosis end address bits

These bits are used to specify the address with which the RAM diagnosis and initialization end for Backup RAM.

Note:

Setting of a value outside the Backup RAM area and a value that sets TASARA.ST10 to ST0 > TAEARA.ED10 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0000_4000_H) + (Offset address set with TAEARA + 2'b11)

4.16. TEST Diagnosis Function Register BACKUP-RAM : TTCRA

This section explains the bit structure of TEST Diagnosis Function Register BACKUP-RAM.

The TEST diagnosis function register (TTCRA) specifies the RAM diagnosis content for Backup RAM, and holds the diagnosis result and its status.

■ TTCRA: Address 3042_H (Access: Byte, Half-word, Word)

BIT	15	14	13	12	11	10	9	8
	Reserved						TSTAT	OVFLW
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX

BIT	7	6	5	4	3	2	1	0
	TEIE	TEI	TCIE	TCI	TTYP2	TTYP1	TTYP0	TRUN
Initial values	0	0	0	0	1	1	0	0
Attributes	R/W	R(RM1),W	R/W	R (RM1), W	R/W	R/W	R/W	R, WX

[bit15 to bit10] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit9] TSTAT: RAM diagnosis error detection bit

TSTAT	Function
0	No error is detected with the RAM diagnosis
1	An error is detected with the RAM diagnosis

If an error occurs during RAM diagnosis for Backup RAM, this bit is set to "1".
This bit is initialized (cleared to "0") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit8] OVFLW: RAM diagnosis error overflow bit

OVFLW	Function
0	During the RAM diagnosis, an error occurs in three or less addresses
1	During the RAM diagnosis, an error occurs in four or more addresses

If a RAM diagnosis error for Backup RAM occurs in four or more addresses, this bit is set to "1".
This bit is initialized (cleared to "0") by hardware, using the RAM diagnosis start instruction as the trigger.

[bit7] TEIE: Interrupt enable bit resulting from a diagnosis error

TEIE	Function
0	Prohibition of an interrupt resulting from a diagnosis error
1	Enabling of an interrupt resulting from a diagnosis error

This bit is used to enable an interrupt resulting from a RAM diagnosis error for Backup RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis error.
"1": Enables an interrupt resulting from a RAM diagnosis error. If TTCRA.TEI=1 is set and the RAM diagnosis ends, the interrupt signal (RAM diagnosis error interrupt) is output.

[bit6] TEI: Diagnosis error generation bit

TEI	Function
0	Read: No error occurrence during the RAM diagnosis. Write: Flag clearing.
1	Read: Error occurred during the RAM diagnosis. Write: No influence on the operation.

If TTCRA.TSTAT=1 is set when RAM diagnosis end for Backup RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set TTCRA.TEI=1 when RAM diagnosis is ended.
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit5] TCIE: Interrupt enable bit for a diagnosis end factor

TCIE	Function
0	Prohibition of an interrupt for the diagnosis end factor
1	Enabling of an interrupt for the diagnosis end factor

This bit is used to enable an interrupt for the RAM diagnosis end factor for Backup RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis end.
"1": Enables an interrupt resulting from a RAM diagnosis end. The interrupt signal (Backup RAM diagnosis end interrupt) is output with TTCRA.TCI= 1.

[bit4] TCI: Diagnosis end bit

TCI	Function
0	Read: The RAM diagnosis does not end. Write: Flag clearing.
1	Read: The RAM diagnosis ended. Write: No influence on the operation.

If RAM diagnosis end for Backup RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM diagnosis is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit3 to bit1] TTYP2 to TTYP0: RAM diagnosis content indication bit

These bits are used to set the RAM diagnosis type for Backup RAM to be executed.

The RAM diagnosis types are executed in the following order.

1. Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
2. Checker
3. March (all "0" -> all "1" are executed in that order.)

These bits are used to determine whether or not each type is executed.

TTYP2	TTYP1	TTYP0	Function
1	1	0	Execution of unique and checker
-	-	1	Execution of march
-	1	-	Execution of checker
1	-	-	Execution of unique

By default, the unique and checker diagnoses are executed (110_B). However, to change the RAM diagnosis content, be sure to specify this change before the RAM diagnosis operation start instruction.
If march is executed last, the RAM content is all "1".

[bit0] TRUN: RAM diagnosis operation status bit

TRUN	Function
0	The RAM diagnosis is stopping
1	The RAM diagnosis is in progress

This bit is used to set or hold the RAM diagnosis status for Backup RAM.

"1": Set when a RAM diagnosis is started by the key code setting.
"0": Set when all diagnoses are complete or forcibly terminated by the key code.

4.17. TEST Initialization Function Register BACKUP-RAM : TICRA

This section explains the bit structure of TEST Initialization Function Register BACKUP-RAM.

The TEST initialization function register (TICRA) specifies the RAM initialization content, and holds the initialization result and its status for Backup RAM.

■ TICRA: Address 3041_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	Reserved				ICIE	ICI	ITYP	IRUN
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R (RM1), W	R/W	R, WX

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit3] ICIE: Interrupt enable bit for a RAM initialization end factor

ICIE	Function
0	Prohibition of an interrupt for a RAM initialization end factor
1	Enabling of an interrupt for a RAM initialization end factor

This bit is used to enable an interrupt for the RAM initialization end factor for Backup RAM.

"0": Prohibits an interrupt resulting from a RAM initialization end.
"1": Enables an interrupt resulting from a RAM initialization end. The interrupt signal (Backup RAM initialization complete interrupt) is output with TICRA.ICI= 1.

[bit2] ICI: RAM initialization end bit

ICI	Function
0	Read: The RAM initialization does not end. Write: Flag clearing.
1	Read: The RAM initialization ended. Write: No influence on the operation.

If RAM initialization end for Backup RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM initialization is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit1] ITYP: RAM initialization content indication bit

ITYP	Function
0	Initialization to All "0"
1	Initialization to All "1"

This bit is used to set the type to be executed during RAM initialization for Backup RAM.

"0": Initializes to all "0".
"1": Initializes to all "1".

[bit0] IRUN: RAM initialization operation status bit

IRUN	Function
0	RAM Initialization is stopping
1	RAM Initialization is in progress

This bit is used to set or hold the RAM initialization status for Backup RAM.

"1": Set when RAM initialization is started by the key code setting.
"0": Set when all initialization is completed or forcibly terminated by the key code.

4.18. TEST Software Reset Generation Control Register BACKUP-RAM : TSRCRA

This section explains the bit structure of TEST Software Reset Generation Control Register BACKUP-RAM.

The TEST software reset generation control register (TSRCRA) specifies the generation of the software reset for initializing internal circuits for Backup RAM's RAM diagnosis.

■ TSRCRA: Address 3044_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	SRST	Reserved						
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0

[bit7] SRST: Software reset enabling bit

SRST	Function
0	Prohibition of a software reset
1	Enabling of a software reset

This bit is used to enable a software reset for the internal circuit for RAM diagnosis for Backup RAM.

This bit reads out "0".
"1": Reset pulses occur for 4τ only and the internal circuit for RAM diagnosis except this register is reset.
τ: Peripheral clock

[bit6 to bit0] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

4.19. TEST Fake Error Generation Control Register BACKUP-RAM : TFECRA

This section explains the bit structure of TEST Fake Error Generation Control Register BACKUP-RAM.

TEST fake error generation control register (TFECRA) generates a fake error in RAM diagnosis operation for Backup RAM. You can specify the RAM diagnosis operations for which you want to generate an error.

■ TFECRA: Address 3040_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	Reserved				FERR	ETYP2	ETYP1	ETYP0
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit3] FERR: Fake error enable bit for RAM diagnosis

FERR	Function
0	Prohibition of a fake error (normal operation)
1	Enabling of a fake error

This bit is used to enable a fake error for RAM diagnosis for Backup RAM.

"0": Prohibits a fake error. (normal operation)
"1": Enables a fake error. Data write including intentional error is enabled following ETYP2 to ETYP0.

[bit2 to bit0] ETYP2 to ETYP0: Fake error process specification bits

These bits are used to specify a process to generate a fake error

ETYP2	ETYP1	ETYP0	Process to generate a fake error
-	-	1	March diagnosis
-	1	-	Checker diagnosis
1	-	-	Unique diagnosis

4.20. TEST Key Code Control Register BACKUP-RAM : TKCCRA

This section explains the bit structure of TEST Key Code Control Register BACKUP-RAM

The TEST key code control register (TKCCRA) is used to start or forcibly terminate the RAM diagnosis or initialization for Backup RAM.

■ TKCCRA: Address 3047_H (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	KEY1	KEY0	Reserved				CODE1	CODE0
Initial values	0	0	0	0	0	0	0	0
Attributes	R0, W	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W

[bit7, bit6] KEY1, KEY0: Key code control bits

Key code control bits. Set operation instruction content to CODE[1:0] (no change during operation) and perform operation.

The procedure of the state transition is:

1. 00 -> 01 -> 10 -> 11 : Write in this order.
2. Same values in CODE[1:0].
3. Different operations (access or read other registers for RAM diagnosis, or continuous write in the different order other than the above) within the procedure will be invalid.

Note:

The key code process will be continued even if any access to the registers in the RAMECC is made in the procedure.

[bit5 to bit2] Reserved

Reserved bits. These bits read out "0". Write "0" when writing.

[bit1, bit0] CODE1, CODE0: RAM diagnosis/initialization control bits

These bits specify operational direction for the key code procedure above.

CODE1, CODE0	Function
00	Forced termination
01	Initialization start
10	Diagnosis start
11	Setting prohibited

If this value is changed or set to "11" during operating the key code above, the key code procedure itself will be invalid.

5. Operation

This section explains the Operation of RAM diagnosis.

5.1 RAM Diagnosis

5.2 RAM Initialization

5.3 Interrupt-Related Register

5.4 RAM Diagnosis Fake Error Generation Procedure

5.5 Number of Required Cycles

5.6 Note

5.1. RAM Diagnosis

This section explains the RAM diagnosis.

XBS RAM diagnosis is performed only in the following order.

1. Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
2. Checker
3. March (all "0" -> all "1" are executed in that order.)

The RAM diagnosis is performed following the settings in the TTYP[2:0] bits of the TEST diagnosis function register (TTCRX). By default, unique and checker are executed.

The coverage of the RAM diagnosis for XBS RAM is specified by the TEST start address register (TASARX) and TEST end address register (TAEARX).

Following procedure is required for RAM diagnosis for XBS RAM.

1. Before start diagnosing, read TRUN of the TEST diagnosis function register (TTCRX) and IRUN of the TEST initialization function register (TICRX), and check that they are "0".

   In the case where TTCRX.TRUN or TICRX.IRUN is not "0":

   - Wait for TTCRX.TRUN="0", then clear TTCRX.TCI.
   - Wait for TICRX.IRUN="0", then clear TICRX.ICI.
2. Write continuously "02_H" -> "42_H" -> "82_H" -> "$C2_H$" to the TEST key code control register (TKCCRX), then start diagnosing.

When all RAM diagnosis types for XBS RAM are completed, TRUN bit of the TEST diagnosis function register (TTCRX) becomes "0" to finish the RAM diagnosis. The results of the diagnosis is retained in the TEST error address register 0 to 2 (TEAR0X to TEAR2X) and TEST diagnosis function register (TTCRX). The RAM holds the diagnosis data.

In addition, write continuously "00_H" -> "40_H" -> "80_H" -> "$C0_H$" to the TEST key code control register (TKCCRX) to terminate the RAM diagnosis for XBS RAM forcibly. RAM diagnosis ends even if it is in progress. In this case, the diagnosis result is not reliable.

Perform the same procedure for RAM diagnosis for Backup RAM.

5.2. RAM Initialization

This section explains the RAM initialization.

Only either of the following RAM initialization operation types for XBS RAM is specified with the ITYP bit of the TEST initialization function register (TICRX).

- Write all "0" (default)
- Write all "1"

ECC area has the values depending on the written values.
The coverage of the RAM initialization for XBS RAM is specified by the TEST start address register (TASARX) and TEST end address register (TAEARX).

Following procedure is required for XBS RAM's RAM diagnosis.

1. Before start diagnosing, read TRUN of the TEST diagnosis function register (TTCRX) and IRUN of the TEST initialization function register (TICRX), and check that they are "0".

   In the case where TTCRX.TRUN or TICRX.IRUN is not "0":

   - Wait for TTCRX.TRUN="0", then clear TTCRX.TCI.
   - Wait for TICRX.IRUN="0", then clear TICRX.ICI.

2. Write continuously "01_H" -> "41_H" -> "81_H" -> "$C1_H$" to the TEST key code control register (TKCCRX), then start diagnosing.

When RAM initialization is completed, IRUN bit of the TEST initialization function register (TICRX) becomes "0" to finish the RAM initialization.

In addition, write continuously "00_H" -> "40_H" -> "80_H" -> "$C0_H$" to the TEST key code control register (TKCCRX) to terminate the RAM initialization forcibly. RAM initialization ends even if it is in progress. In this case, the initialization results is not guaranteed.

Perform the same procedure for RAM diagnosis for Backup RAM.

5.3. Interrupt-Related Register

This section explains the interrupt-related register.

To generate an interrupt, write "1" to the interrupt generation enabling bits (TEIE, TCIE, and ICIE) according to the purposes, and set the RAM diagnosis interrupt vector and RAM diagnosis interrupt level.

Interrupt factor	Interrupt vector	Interrupt level
TTCRX.TEI (RAM diagnosis error interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TTCRX.TCI (RAM diagnosis end interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TICRX.ICI (RAM initialization complete interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TTCRA.TEI (Backup RAM diagnosis error interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TTCRA.TCI (Backup RAM diagnosis end interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TICRA.ICI (Backup RAM initialization complete interrupt)	#35($000FFF70_H$)	ICR19(0453_H)

For details of the interrupt levels and interrupt vectors, see "CHAPTER: INTERRUPT CONTROL (INTERRUPT CONTROLLER)".
The interrupt request flags (TEI, TCI, ICI) are not automatically cleared. So, to clear them, use software before return from interrupt processing. (Write "0" in the TEI, TCI, and ICI bits.)

5.4. RAM Diagnosis Fake Error Generation Procedure

This section explains the RAM diagnosis fake error generation procedure.

This function intentionally generates fake errors for software debugging.
Set the RAM diagnosis fake error generation for XBS RAM as following procedure:

1. Specify the error type with the TEST fake error generation control register (TFECRX).
   (1) Set a diagnosis pattern to the TFECRX.ETYP[2:0] to generate a fake error.
   (2) Specify a diagnosis pattern to generate a fake error by writing TFECRX.FERR="1".

2. Set the diagnosis start with the TEST diagnosis function register (TTCRX).
   (1) Set a diagnosis pattern to operate with the TTCRX.TTYP[2:0].
   (2) Write continuously "02_H" -> "42_H" -> "82_H" -> "$C2_H$" four times to the TEST key code control register (TKCCRX), then start diagnosis pattern (See "5.1. RAM Diagnosis")

Perform the same procedure for RAM diagnosis for Backup RAM.

5.5. Number of Required Cycles

This section explains the number of required cycles.

The following shows the estimation of the cycle count required for various RAM diagnosis and initialization for XBS RAM, Backup RAM.

XBS RAM : 48kByte = 12k word address (MB91F522, MB91F523)
64kByte = 16k word address (MB91F524)
96kByte = 24k word address (MB91F525)

128kByte = 32k word address (MB91F526)

Backup RAM : 8kByte = 2k word address

(1) "RAM diagnosis (unique)"

- Write (1 cycle)
- Read 1 (1 cycle)
- Read 2 (1 cycle)

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses, and the entire number of cycles is as follows:

(	1	+	1	+	1	) x	12288(12k)	+1	=	36865
(	1	+	1	+	1	) x	16384(16k)	+1	=	49153
(	1	+	1	+	1	) x	24576(24k)	+1	=	73729
(	1	+	1	+	1	) x	32768(32k)	+1	=	98305
	Write		Read 1		Read 2		Word			Total

(2) "RAM diagnosis (checker)"

- Write 1 (1 cycle) : W1
- Read 1 (1 cycle) : R1

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses. To perform the partial write function diagnosis, five write processes and four read processes are provided for each word address. So, the following is obtained.

- Write 2 (1 × 5 cycles) : W2
- Read 2 (2 × 4 cycles) : R2

Moreover, the same processing is repeated with data different from above data. The entire number of cycles is as follows:

((	1	+	1	) x	12288(12k)	+1	+	5	+	8	) x	2	=	49180
((	1	+	1	) x	16384(16k)	+1	+	5	+	8	) x	2	=	65564
((	1	+	1	) x	24576(24k)	+1	+	5	+	8	) x	2	=	98332
((	1	+	1	) x	32768(32k)	+1	+	5	+	8	) x	2	=	131100
	W1		R1		Word			W2		R2		Repetition		Total

(3) "RAM diagnosis (march)"

- Write (1 × 3 cycles)

· Read (2 × 2 cycles)

This diagnosis has 3 writes and 2 reads per a word address so that the processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses, Moreover, the same processing is repeated with data different from above data. So, the entire number of cycles is as follows:

(	3	+	4	) x	12288(12k)	x	2	=	172032
(	3	+	4	) x	16384(16k)	x	2	=	229376
(	3	+	4	) x	24576(24k)	x	2	=	344064
(	3	+	4	) x	32768(32k)	x	2	=	458752
	Write		Read		Word		Repetition		Total

(4) "RAM initialization"

· Write (1 cycle)

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses, The entire number of cycles is as follows:

1	x	12288(12k)	=	12288
1	x	16384(16k)	=	16384
1	x	24576(24k)	=	24576
1	x	32768(32k)	=	32768
Write		Word		Total

The time required for 128kByte RAM diagnosis of 2MHz and 80MHz operations is obtained as follows:

Table 5-1 Time Required for RAM Diagnosis and Initialization for 128kByte

	Unique	Checker	March	Initialization	Total
Number of cycles	98305	131100	458752	32768	720925
2[MHz] (=500[ns])	49152.5[µs]	65550[µs]	229376[µs]	16384[µs]	360[ms]
80MHz (=12.5[ns])	1228.8[µs]	1638.6[µs]	5734.4[µs]	409.6[µs]	9.1[ms]

Moreover, the time required for the diagnosis with initial register values is obtained as follows:

Table 5-2 Time Required of Diagnosis (Initial Setting) after Power-on Reset is Released (128kByte)

	Unique	Checker	Total
Number of cycles	98305	131100	229405
2[MHz] (=500[ns])	49152.5[µs]	65550[µs]	115[ms]

5.6. Note

This section explains the note.

Accessing to RAM is prohibited during RAM diagnosis or during initialization.
RAM diagnosis and initialization cannot be used during debugging with the on-chip debugger (OCD).
While performing diagnostic or initialization process, start setting is ignored and the currently running process will continue. To start anything, perform steps below to make sure that no operation is being performed.

1. Make sure that all of TTCR:TRUN, TICR and IRUN are "0".
2. Start the key code operation with TKCCR (TKCCRX and TKCCRA) for diagnosis or initialization.

Forced termination can be performed by key code operation. Perform as shown below.

- Input "00_H"-> "40_H"-> "80_H"-> "$C0_H$" to TKCCR continuously.

Chapter 51: Timing Protection Unit

This chapter explains the Timing Protection Unit.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

Code : FR81S10_TPU-1v1-91520-2-E

1. Overview

This section explains the overview of the Timing Protection Unit.

Timing Protection Unit (TPU) is a timer that OS uses for the ensuring safety of the system by the watch of the time of Task/ISR. The control target that OS supervises is as follows.

- Resource lock time
- Global interrupt lock time
- Task/ISR dead line
- Task/ISR runtime
- Inter-arrival time (Interrupt Frequency)

The control registers can be accessed only in a privileged mode because the built-in timers of TPU are controlled by OS, and the controls are all programmable.

2. Features

This section explains features of the Timing Protection Unit.

- Count with system clock (HCLK)
- Built-in timers. Max: 8 timers
- 24-bit Up-counter
- Two operational modes of Normal/Overflow
  - Normal Mode: When a count value is exceeded to the setting value, interrupt is generated.
  - Overflow Mode: The interrupt is generated by the counter overflow.
- Automatic restart function
- Global prescaler (division factor 1/1 to 1/64)
- Prescaler by timer (1, 1/2, 1/4, 1/16)
- Reading function of counter value
- Software control of Start/ Stop/ Continue
- Status display of each timer (stopped/ active)
- Debug mode support
- Access protection function of TPU control register

3. Configuration

This section explains the configuration of the Timing Protection Unit.

There is no block diagram.

4. Registers

This section explains registers of the Timing Protection Unit.

For all registers, writing is permitted only for privileged mode/debugging access.
The area of 0x00000900-0x00009FF is TPU register area. The area not shown in the following is all reserved.

Table 4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0900	TPUUNLOCK				TPU Unlock Register
0x0904	TPULST	Reserved	TPUVST	Reserved	TPU Lock Status Register
0x0908	TPUCFG				TPU Configuration Register
0x090C	TPUTIR	Reserved	Reserved	Reserved	TPU Timer Interrupt Request Register
0x0910	TPUTST	Reserved	Reserved	Reserved	TPU Timer Status Register
0x0914	TPUTIE	Reserved	Reserved	Reserved	TPU Timer Interrupt Enable Register
0x0918	TPUTMID				TPU Module ID Register
0x0930	TPUTCN00				TPU Timer Control Register 0 ch.0
0x0934	TPUTCN01				TPU Timer Control Register 0 ch.1
0x0938	TPUTCN02				TPU Timer Control Register 0 ch.2
0x093C	TPUTCN03				TPU Timer Control Register 0 ch.3
0x0940	TPUTCN04				TPU Timer Control Register 0 ch.4
0x0944	TPUTCN05				TPU Timer Control Register 0 ch.5
0x0948	TPUTCN06				TPU Timer Control Register 0 ch.6
0x094C	TPUTCN07				TPU Timer Control Register 0 ch.7
0x0950	TPUTCN10	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.0
0x0954	TPUTCN11	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.1
0x0958	TPUTCN12	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.2
0x095C	TPUTCN13	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.3
0x0960	TPUTCN14	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.4
0x0964	TPUTCN15	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.5
0x0968	TPUTCN16	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.6
0x096C	TPUTCN17	Reserved	Reserved	Reserved	TPU Timer Control Register 1 ch.7
0x0970	TPUTCC0				TPU Timer Current Count Register ch.0
0x0974	TPUTCC1				TPU Timer Current Count Register ch.1
0x0978	TPUTCC2				TPU Timer Current Count Register ch.2
0x097C	TPUTCC3				TPU Timer Current Count Register ch.3
0x0980	TPUTCC4				TPU Timer Current Count Register ch.4
0x0984	TPUTCC5				TPU Timer Current Count Register ch.5
0x0988	TPUTCC6				TPU Timer Current Count Register ch.6
0x098C	TPUTCC7				TPU Timer Current Count Register ch.7

4.1. TPU Unlock Register : TPUUNLOCK

The bit configuration of TPU unlock register is shown below.

■ TPUUNLOCK : Address 0900_H (Access : Word)

	bit31							bit0
	UNLOCK[31:0]							
Initial value	0000	0000	0000	0000	0000	0000	0000	0000
Attribute	R0,W							

This register is used to specify access prohibition/permission to the TPU control register (TPUCFG and TPUTCN1n (n: timer channel number)).
It is required to prevent the illegal update of TPU control registers due to the malfunction of system.
Writing to this register is permitted only at the privileged mode. The readout value is always "0".
Be sure to keep access within 32-bit width (word) because Lock/Unlock is judged with 32-bit.

[bit31 to bit0] UNLOCK[31:0] : LOCK/UNLOCK value
If present value of UNLCOK is written to the register, access to the TPU control register is permitted.
To prohibit accessing, write the values other than a present value of UNLOCK.

4.2. TPU Lock Status Register : TPULST

The bit configuration of TPU lock status register is shown below.

■ TPULST : Address 0904_H (Access : Byte, Half-word, Word)

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

This register is used to indicate the lock status of TPU.
This register is read only, and writing to the register has no influence in operation.

[bit7 to bit1] (Reserved) : (Reserved bit)
These bits are reserved bits. When writing to those bits, "0" must be set. The readout value is always "0".

[bit0] LST (Lock Status) : Lock status display

This bit indicates whether access to the TPU control register is locked.

LST	Lock Status
0	Access permission
1	Access prohibition

4.3. TPU Access Violation Status Register : TPUVST

The bit configuration of TPU access violation status register is shown below.

■ TPUVST : Address 0906_H (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					IULST	ULVST	AVST
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R,W0

An illegal access to the TPU register is detected, and the factor is maintained. When an illegal access to the register is detected, the corresponding bit of the detected violation factor is set, and it is processed as an illegal instruction exception.
Writing to this register is permitted only at the privileged mode.

[bit7 to bit3] (Reserved) : (Reserved bit)
These bits are reserved bits. When writing to those bits, "0" must be set. The readout value is always "0".

[bit2] IULST (Illegal Unlock Access Status) : Illegal unlock operation detection
When an illegal unlock access is detected, this bit becomes "1". Writing to this bit is effective only the value is "0".

When a value other than the value set for UNLOCK is written in the TPUUNLOCK register in privileged mode when TPU control register access is prohibited (TPULST.LST=1) (including cases other than word access), an illegal unlock operation is detected.

[bit1] ULVST (Unlock Access Violation Status) : Control register access violation detection while access prohibiting
When writing in TPU control register (TPUCFG, TPUTCN1n) is detected while prohibiting the TPU control register access, this bit becomes "1". Only when "0" is written, it becomes effective.
When there is a write operation to TPUCFG, TPUTCN1n in privileged mode when TPU control register access is prohibited (TPULST.LST=1), an illegal access is detected.

[bit0] AVST (Access Violation Status) : Access violation detection
When the access violations other than IULST and ULVST are detected, this bit becomes "1". Only when "0" is written, it becomes effective. It concretely becomes a register access by the instruction fetch.

4.4. TPU Configuration Register : TPUCFG

The bit configuration of TPU configuration register is shown below.

■ TPUCFG : Address 0908_H (Access : Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							DBGE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	GLBPSE	Reserved	GLBPS[5:0]					
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							INTE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

It is a register that controls the entire TPU.

[bit31 to bit25] (Reserved) : (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is "0".

[bit24] DBGE (Debug Mode Enable) : Debug mode transition

This bit is used to control transition to debug mode.

When debug mode is permitted, all timers stop operating. Each timer restarts operation when coming off debug mode.

DBGE	Debug Mode
0	All timer operation permission (Normal mode)
1	All timer operation suppression (Debug mode)

[bit23] GLBPSE (Global Prescaler Enable) : Global prescaler operation permission

The operation of global prescaler is controlled. When the operation is prohibited, all timers do not perform count operation.

DLBPSE	Global Prescaler
0	Operation prohibition
1	Operation permission

[bit22] (Reserved) : (Reserved bit)

This is a reserved bit. Be sure to write "0". The readout value is "0".

[bit21 to bit16] GLBPS[5:0] (Global Prescaler Bits) : Global prescaler frequency setting

These bits are used to specify the frequency of the clock that supplied to all timers in common. Update of the bits has to be done when TPUCFG.GLBPSE=0 (timer operation disabled).

In TPU, the system clock (HCLK) is divided with global prescaler and the clock is supplied to each timer.

GLBPS[5:0] indicates the value of dividing frequency as it is.

GLBPS[5:0]	Global Prescaler Output
000000	HCLK / 1
000001	HCLK / 2
000010	HCLK / 3
…	…
111111	HCLK / 64

[bit15 to bit1] (Reserved) : (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is "0".

[bit0] INTE (TPU Interrupt Enable) : TPU interrupt enable

This bit is used to enable the interrupt request from TPU.

INTE	TPU Interrupt
0	Interrupt disable
1	Interrupt enable

4.5. TPU Timer Interrupt Request Register : TPUTIR

The bit configuration of TPU timer interrupt request register is shown below.

■ TPUTIR : Address 0090C$_H$ (Access : Byte, Half-word, Word)

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

This register indicates interrupt request status from each timer in TPU . This register is read only. Writing to the register causes no influence in operation.

[bit7 to bit0] IR[7:0] (Interrupt Request) : Interrupt request

These bits indicate presence of the interrupt request for each channel. These bits show that there is an interrupt request factor regardless of timer interrupt enable register (TPUTIE). The requests are actually used as interrupt requests only when they are from channels where TPUTIE is effective.
Bit0 to 7 corresponds to channel 0 to 7 respectively.

IRn	Interrupt Request
0	Ch.n no Interrupt request
1	Ch.n Interrupt request

(n = 0 to 7)

4.6. TPU Timer Status Register : TPUTST

The bit configuration of TPU timer status register is shown below.

■ TPUTST : Address 00910$_H$ (Access : Byte, Half-word, Word)

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

This register indicates the operation status of each timer in TPU. This register is read only. Writing to the register causes no influence in operation.

[bit7 to bit0] TS[7:0] (Timer Status) : Timer operation status

These bits indicate timer operation status of each channel.

Bit0 to 7 corresponds to channel 0 to 7 respectively.

TSn	Operation Status
0	Ch.n Stopped
1	Ch.n Operating

(n = 0 to 7)

4.7. TPU Timer Interrupt Enable Register : TPUTIE

The bit configuration of TPU timer interrupt enable register is shown below.

■ TPUTIE : Address 00914_H (Access : Byte, Half-word, Word)

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

This register is used to enable interrupt of each timer in TPU.

[bit7 to bit0] IE[7:0] (Interrupt Enable): Timer interrupt enable

These bits are used to enable timer interrupt request for each channel.
Bit0 to 7 corresponds to channel 0 to 7 respectively.

IEn	Interrupt enable
0	Ch.n Interrupt disable
1	Ch.n Interrupt enable

(n = 0 to 7)

4.8. TPU Module ID Register : TPUTMID

The bit configuration of TPU module ID register is shown below.

■ TPUTMID : Address 00918_H (Access : Byte, Half-word, Word)

	bit31							bit0
	MID[31:0]							
Initial value	0000	0000	0000	0000	0000	0000	0000	0000
Attribute	R,WX							

This register is used to indicate the TPU module ID. It is read only. Writing to the register causes no influence in operation.
It is used to identify the function of built-in TPU. In the OS, it is used to distinguish the type of TPU.

4.9. TPU Timer Control Register 00 to 07 : TPUTCN00 to 07

The bit configuration of TPU timer control register 00 to 07 is shown below.

■ TPUTCN00 to TPUTCN07 : Address 00930_H to $0094C_H$ (Access : Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	START	STOP	CONT	IES	IEC	IRC	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ECPL[23:16]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ECPL[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

It is a control register of each timer.

[bit31] START (Start) : Timer operation start

This bit is used to instruct to start the timer operation. The timer operation is started to write "1" to this bit. The readout value is always "0".
When operation is started by this bit at a normal mode, the timer starts counting from "0".
When operation is started by this bit at the overflow mode, the timer starts counting from "0" or ECPL[23:0](TPUTCN1n.PL=1).
Writing "0" to the bit causes no influence in operation.

[bit30] STOP (Stop) : Timer operation stop

This bit is used to instruct to stop the timer operation. The timer operation is stopped to write "1" to this bit.
The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit29] CONT (Continue) : Timer operation restart

This bit is used to instruct to restart the timer operation. The readout value is always "0".
When the operation is restarted by this bit, operation is restarted from the count value that has stopped.
When START, STOP, and the CONT bit are set at the same time, priority is judged in order of START > CONT > STOP.
Writing "0" to the bit causes no influence in operation.

[bit28] IES (Interrupt Enable Set) : Interrupt enable bit Set

This bit is used to instruct to set timer interrupt enable. The interrupt enable bit (TPUTIE.IE[n]) is set by writing "1" in this bit.
The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit27] IEC (Interrupt Enable Clear) : Interrupt enable bit clear

This bit is used to instruct to clear timer interrupt enable. The interrupt enable bit (TPUTIE.IE[n]) is cleared by writing "1" in this bit.
The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit26] IRC (Interrupt Request Clear) : Interrupt request bit clear

This bit is used to instruct timer interrupt clear request . The interrupt request bit (TPUIR.IR[n]) is cleared by writing "1" to this bit.
The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit25, bit24] (Reserved) : (Reserved bit)
These bits are reserved bit. Be sure to write "0". The readout value is "0".

[bit23 to bit0] ECPL[23:0] (End Count or Pre-load) : Counter End value or pre load value
The value used as the end value or pre-load value of the counter is set.
ECPL[23:0] is used as the end value of the counter in the normal mode.
ECPL[23:0] is used as pre-load value in the overflow mode.

4.10. TPU Timer Control Register 10 to 17 : TPUTCN10 to 17

The bit configuration of TPU timer control register 10 to 17 is shown below.

■ TPUTCN10 to TPUTCN17 : Address 00950_H to $0096C_H$ (Access : Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			PL	FRT	TMOD	PS[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

It is a control register for each timer.

[bit7 to bit5] (Reserved) : (Reserved bit)
These bits are reserved bit. Be sure to write "0". The readout value is "0".

[bit4] PL (Pre-load) : Pre-Load instructions
This bit is used to specify pre-load of ECPL[23:0] when the timer operation is started. This bit is effective when the timer is in the overflow mode.

PL	Pre-load
0	Pre-load invalid
1	Pre-load valid

[bit3] FRT (Free-Running Timer) : Free-Running Timer instructions
This bit is used to instruct free-run operation. It is effective in both normal mode/overflow mode.
After the interrupt is generated by the end value of the counter, the count is restarted from "0" automatically when this bit is made effective in the normal mode.
After the interrupt is generated by the counter overflow, the count is restarted from "0" (TPUTCN1n.PL=0) or

ECPL[23:0] (TPUTCN1n.PL=1) automatically when this bit is made effective in the overflow mode.

FRT	Free-run
0	Free-run invalid
1	Free-run valid

[bit2] TMOD (TPU Mode) : TPU operation mode

This bit is used to specify TPU operation mode. In the operation mode of the timer, there is the normal mode in which the count is incremented from "0" to ECPL[23:0], or the overflow mode in which the count is started from "0" (TPUTCN1n.PL=0) or ECPL[23:0] (TPUTCN1n.PL=1) and the counter overflow is detected.

TMOD	Timer Operation Mode
0	Normal mode
1	Overflow mode

[bit1, bit0] PS[1:0] (Individual Prescaler) : Timer prescaler setting

The prescaler value of each timer is set. The output of global prescaler is input to each timer, and this input is divided and used as the operating frequency of each timer.

PS[1:0]	Prescaler
00	1/1
01	1/2
10	1/4
11	1/16

4.11. TPU Timer Current Count Register 0 to 7 : TPUTCC0 to 7

The bit configuration of TPU timer current count register 0 to 7 is shown below.

■ TPUTCC0 to TPUTCC7 : Address 00970$_H$ to 0098Cn$_H$ (Access : Byte, Half-word, Word)

	bit31	bit24	bit23					bit0
	Reserved		TCC[23:0]					
Initial value	0000	0000	0000	0000	0000	0000	0000	0000
Attribute	R0,W0	R0,W0	R,WX					

This register indicates the present counter value of the timer. This register is read only.

[bit31 to bit24] (Reserved) : (Reserved bit)

These bits are reserved bit. Be sure to write "0". The readout value is "0".

[bit23 to bit0] TCC[23:0] (Timer Current Count) : Timer Count value

These bits indicate the present counter value.

5. Operation

This section explains the operation.

5.1 TPU Control Register Access Protection

5.2 Global Prescaler

5.3 Interrupt Control

5.4 Timer Operation

5.5 Free-run Function

5.6 Individual Prescaler Function

5.7 Debug Support Function

5.8 Operation Flow

5.1. TPU Control Register Access Protection

This section explains the TPU control register access protection.

The TPU register is permitted to be accessed only in the privileged mode because all TPU registers are the system registers. The illegal instruction exception (data access error) is generated if accessing it in the user mode.

The TPU register not only has a function for access protection as the system register, it also has a function for register access protection with the Lock code to prevent writing of the TPU control register as a result of malfunctioning.

The target registers of the access protection are the following two registers.

- TPU configuration register (TPUCFG)
- TPU timer control register 10 to 17 (TPUTCN 10 to 17)

To make the TPU configuration register access protection effective, write the values other than a present set value of UNLOCK[31:0] in the TPU unlock register (TPUUNLOCK). When the access protection function becomes effective, the LST bit of the TPU lock status register is set to indicate the lock state.

When writing it in the control register of the protection target, write the value set last time in UNLOCK[31:0]. It becomes TPULST.LST=0 and the unlock state when the lock is released.

After generating reset, the register access protection function in the invalid state (TPULST.LST=0).

When the TPU control register access protection function is effective(TPULST.LST=1), and when the values other than UNLOCK[31:0] are written in TPU unlock register (TPUUNLOCK), the error reply is returned to AHB and the data access error is generated in CPU as an illegal access.
Then, the violation factor is set in the TPU access violation detection register (TPUVST.IULST=1) .

When the TPU control register access protection function is effective(TPULST.LST=1), and when there is a write request to the TPU control register (TPUCFG) and TPU timer control register (TPUTCFG1n), the data access error is generated in CPU as an illegal access.
Then, the violation factor is set in the TPU access violation detection register (TPUVST.ULVST =1) .

Moreover, it is judged that the access by the instruction fetch is a malfunction and generates the illegal instruction exception. Then, it becomes TPUVST.AVST=1.

5.2. Global Prescaler

This section explains the global prescaler.

The global prescaler is a common prescaler used with all timers of TPU. The global prescaler divides HCLK (input clock of TPU) according to a set value of TPUCFG.GLBPS[5:0]. The value of division can be set by 1 to 64.

The global prescaler function controls operation by the TPUCFG.GLBPSE bit. The global prescaler function is enabled by writing "1" in TPUCFG.GLBPSE and it is disabled by writing "0" in TPUCFG.GLBPSE.

When TPUCFG.GLBPSE=0, the prescaler function is disabled and the clock of all timers doesn't become valid.

Please, update TPUCFG.GLBPS[5:0] after setting the global prescaler function disabled (TPUCFG.GLBPSE=0).

5.3. Interrupt Control

This section explains the Interrupt Control.

The generation of the interrupt request is controlled by the TPUCFG.INTE bit that controls the interrupt request by TPU and the TPUTIE.IE[n] bits that controls the interrupt of each timer. When an effective interrupt request by each timer exists in TPUCFG.INTE=1((TPUTIE.IE[n]=1)&(TPUTIR.IR[n]=1)), NMI is generated in TPU.

The interrupt factor of each channel can be confirmed with TPUTIR.IR[n].

Interrupt enable/disable of each channel is controlled with TPUTCN0n.IES/TPUTCN0n.IEC.
If 1 is written in TPUTCN0n.IES, the interrupt is permitted and it becomes TPUTIE.IE[n]=1. If "1" is written in TPUTCN0n.IEC, the interrupt is prohibited and it becomes TPUTIE.IE[n]=0. If "1" is written to TPUTCN0n.IES and TPUTCN0n.IEC at the same time, it gives higher priority to "clear" than "set".

Please write "1" in TPUTCN0n.IRC when you clear the interrupt request of each channel.

5.4. Timer Operation

This section explains the timer operation.

Each timer is configured by 24-bit Up-counter.

The timer has two operation modes of normal mode/overflow mode. The operation mode is controlled by the TPUTCN1n.TMOD bit. The operation mode becomes the normal mode when TPUTCN1n.TMOD=0, and becomes the overflow mode when TPUTCN1n.TMOD=1.

● Normal Mode

The timer operates as the up-counter when the normal mode. When the counter value is equal or larger than TPUTCN0n.ECPL [23:0], the interrupt flag (TPUTIR.IR[n] (n: the timer channel)) is set. An actual interrupt request is generated when TPUTIE.IE [n]=1.

The timer starts counting from "0" by writing "1" to the TPUTCN0n.START bit.

During timer count, timer operation is indicated with TPUTST.TS[n]=1. If the interrupt flag is set (TPUTIR.IR[n]=1), the count is stopped and TPUTST.TS[n]=0 is indicated.

If "1" is written in the TPUTCN0n.STOP bit, the counter stops operating, and TPUTST.TS[n]=0 is indicated. The counter value at this time (When stopping) is maintained, and does not become 0.

If "1" is written in the TPUTCN0n.CONT bit, the counter operation is restarted, and becomes TPUTST.TS[n]=1.

● Overflow Mode

When the overflow of the timer is detected in the overflow mode, the interrupt request flag (TPUTIR.IR[n]) is set. An actual interrupt request is generated when TPUTIE.IE[n]=1.

Pre-load to the counter is possible in the overflow mode. The value of TPUTCN0n.ECPL[23:0] is pre-loaded and the count is started after TPUTCN1n.PL=1 is set and operation starts. The timer starts counting from "0" if TPUTCN1n.PL=0.

5.5. Free-run Function

This section explains the free-run function.

Each timer can set free-run operation. The free-run operation is a function to restart the count automatically after the timer counts to the interrupt generation factor. In this case, because the counter operation doesn't stop, it keeps operating as timer operation status TPUTST.TS[n]=1.

The free-run function becomes effective if you set TPUTCN1n.FRT=1. The free-run function can use both normal mode/overflow mode.

At the normal mode, the count is restarted from "0". At the overflow mode, the count is restarted from "0" when TPUTCN1n.PL=0, and the count is restarted after the value of TPUTCN0n.ECPL[23:0] is loaded when TPUTCN1n.PL=1.

5.6. Individual Prescaler Function

This section explains the individual prescaler function.

TPU has the individual prescaler for each timer, and it can divide by 1, 2, 4, or 16 the global prescaler output. Individual prescaler is set with TPUTCN1n.PS[1:0].

5.7. Debug Support Function

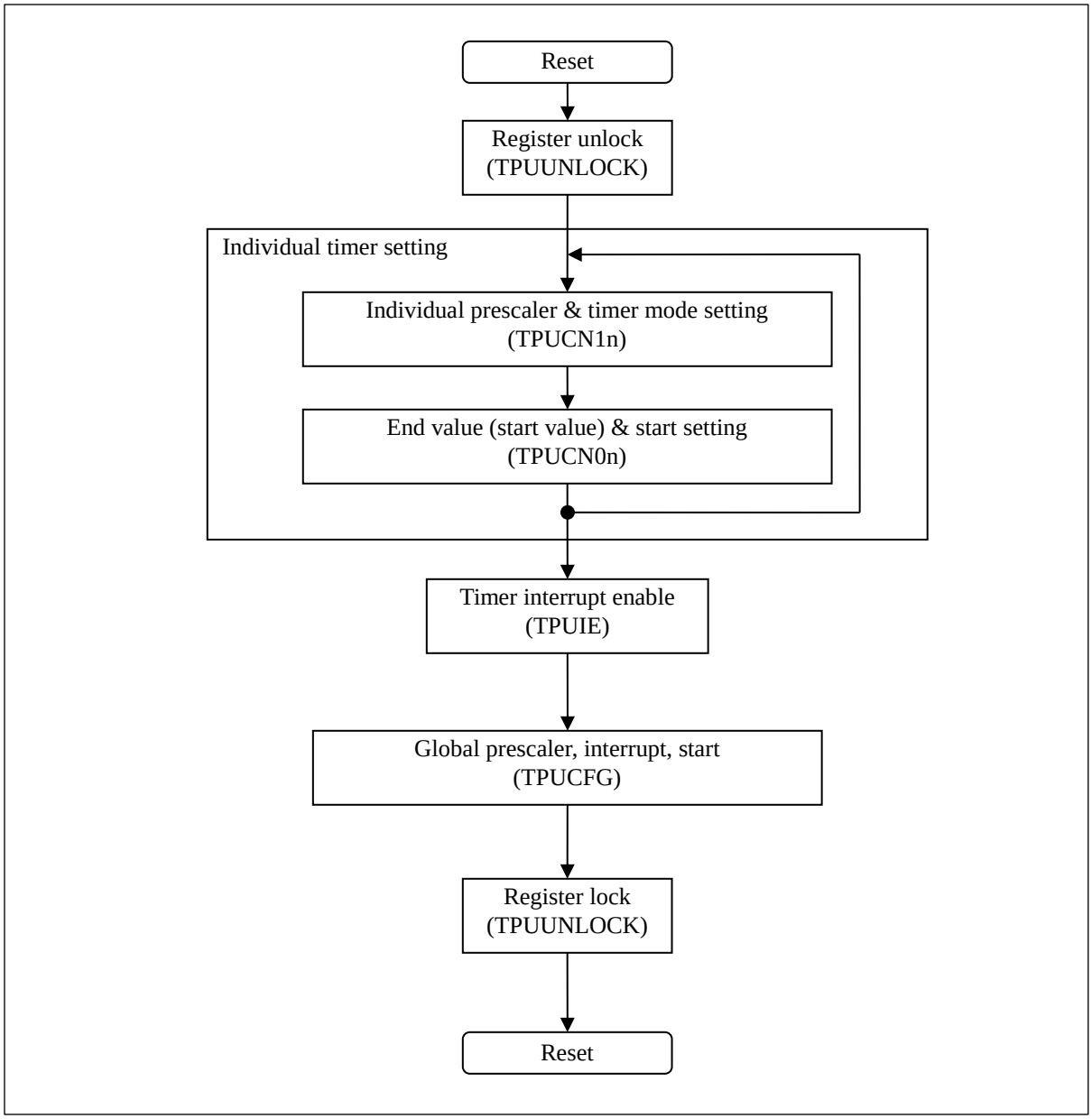
This section explains the debug support function.

TPU can be stopped by writing "1" in the debug mode control bit of the TPU control register (TPUCFG.DBGE=1) with software. The debug mode of TPU is released when "0" is written in TPUCFG.DBGE and operation is restarted.

5.8. Operation Flow

This section explains the operation flow.

■ Initialization Flow



Chapter 52: Clock Monitor

This chapter explains the clock monitor.

1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Setting
7. Q&A
8. Notes

Code : FJ43-1v0-91520-2-E

1. Overview

This section explains the overview of clock monitor.

The clock monitor is a macro that outputs and monitors internal clock signals to external pins. The clock monitor has a function for dividing the frequency of a clock signal before output to the pin, allowing clock signals to be used for synchronization of external circuits with MCU functions.

2. Features

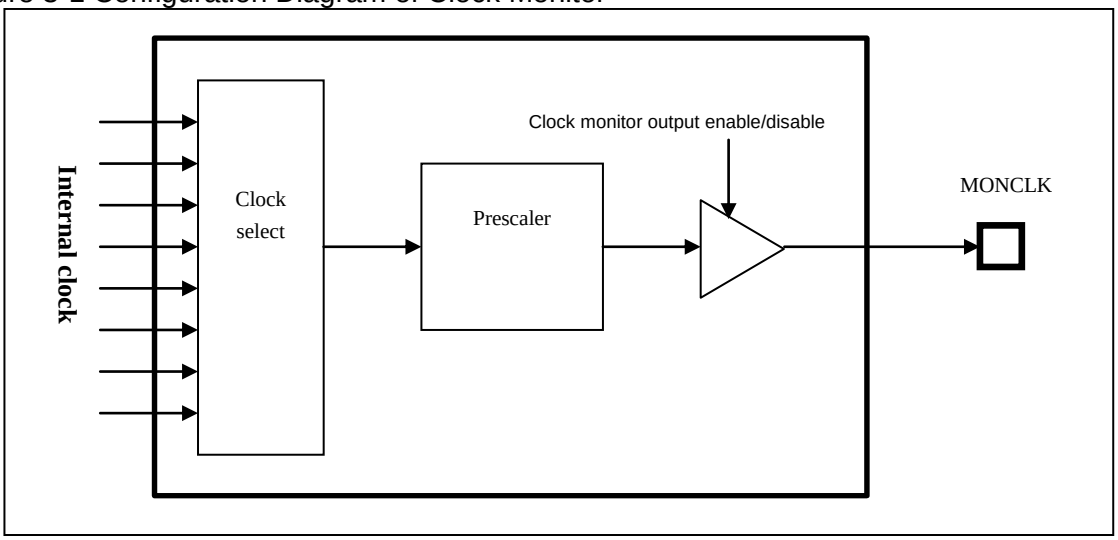
This section explains the features of clock monitor.

- Format: Divide the internal clock signal and output to a pin (MONCLK)
- Channels: 1
- Division ratio: CLK/1, CLK/2, CLK/3 to CLK/16
- Allows for glitch-less output
- Programmable mark level (outputs "L" or "H" before the clock output is enabled)
- Interrupts: None
- Stops clock output in stop mode and becomes high impedance

3. Configuration

This section explains the configuration of clock monitor.

Figure 3-1 Configuration Diagram of Clock Monitor



MB No. (number of pin)	Pin number of MONCLK
MB91F52xB(64pin)	26
MB91F52xD(80pin)	33
MB91F52xF(100pin)	41
MB91F52xJ(120pin)	47
MB91F52xK(144pin)	55
MB91F52xL(176pin)	69

4. Registers

This section explains the registers of clock monitor

Table 4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x04A8	Reserved	Reserved	CSCFG	CMCFG	Clock Monitor Configuration Registers

4.1. Clock Monitor Configuration Registers : CMCFG

THE clock monitor configuration registers are shown.

■ CMCFG: Address $04AB_H$ (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	CMPRE3	CMPRE2	CMPRE1	CMPRE0	CMSEL3	CMSEL2	CMSEL1	CMSEL0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit4] CMPRE3 to CMPRE0 (Output Frequency Prescaler Bits)

Division ratio setting of selected source clock by CMSEL bits.

CMPRE3	CMPRE2	CMPRE1	CMPRE0	Clock frequency output to the MONCLK pin
0	0	0	0	Source clock divided by 1 (Initial value)
0	0	0	1	Source clock divided by 2
0	0	1	0	Source clock divided by 3
0	0	1	1	Source clock divided by 4
0	1	0	0	Source clock divided by 5
0	1	0	1	Source clock divided by 6
0	1	1	0	Source clock divided by 7
0	1	1	1	Source clock divided by 8
1	0	0	0	Source clock divided by 9
1	0	0	1	Source clock divided by 10
1	0	1	0	Source clock divided by 11
1	0	1	1	Source clock divided by 12

CMPRE3	CMPRE2	CMPRE1	CMPRE0	Clock frequency output to the MONCLK pin
1	1	0	0	Source clock divided by 13
1	1	0	1	Source clock divided by 14
1	1	1	0	Source clock divided by 15
1	1	1	1	Source clock divided by 16

[bit3 to bit0] CMSEL3 to CMSEL0 (Output Source Clock Selection Bits)

Selected source clock for output signal of MONCLK pin.

CMSEL3	CMSEL2	CMSEL1	CMSEL0	Clock source output to MONCLK pin
0	0	0	0	MONCLK output disabled (high impedance state) (initial value)
0	0	0	1	Main oscillation before CSV input
0	0	1	0	CR oscillation
0	0	1	1	Main oscillation output from CSV
0	1	0	0	Setting Prohibited
0	1	0	1	Setting Prohibited
0	1	1	0	Setting Prohibited
0	1	1	1	Setting Prohibited
1	0	0	0	PLL output
1	0	0	1	SSCG output
1	0	1	0	PLL output after CAN prescaler (CAN system clock)
1	0	1	1	CCLK
1	1	0	0	HCLK
1	1	0	1	PCLK1(Spread peripheral clock)
1	1	1	0	PCLK2 (Peripheral clock after spread/ non spread selection)
1	1	1	1	TCLK

■CSCFG: Address $04AA_H$ (Access: Byte, Half-word, Word)

BIT	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	MONCKI	Reserved			
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0	R,WX	R,WX	R/W	R/W0	R/W0	R/W0	R/W0

[bit7] Reserved

This bit is reserved. Always set this bit to "0" when writing.

[bit6, bit5] Reserved

These bits are reserved. Writing to these bits has no influence on operation.

[bit4] MONCKI : Clock Monitor MONCLK Inverter

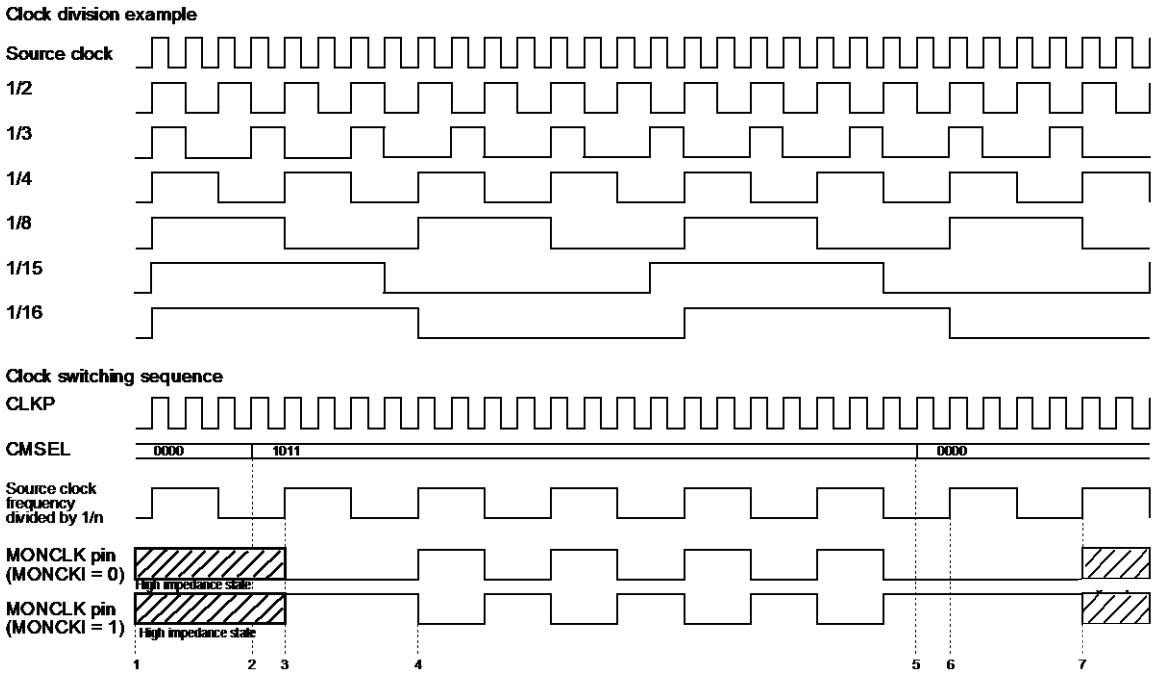
MONCKI	Function
0	MONCLK mark level is low level (initial value)
1	MONCLK mark level is high level

[bit3 to bit0] Reserved

This bit is reserved. Always set this bit to "0" when writing.

5. Operation

This section explains operation description of clock monitor



1. The MONCLK pin is in the high impedance state.
2. CMSEL is set to the selected clock (prescaler) from 0000_B (no clock selected).
3. The MONCLK pin is set to the output "L" status (or output "H" if MONCKI is set to "1") for the duration of one internal (prescaled) clock.
4. After one period of the selected (prescaler) internal clock, MONCLK outputs the selected (prescaler) internal clock.
5. CMSEL is set to 0000_B (no clock selected) from the selected clock (prescaler).
6. The MONCLK pin is set to the output "L" status (or output "H" if MONCKI is set to "1") for the duration of one internal (prescaled) clock.
7. The MONCLK pin switches to the high impedance state.

6. Setting

This section explains the setting of clock monitor.

Setting	Setting register	Setting procedure
Setting of the prescaler value	Output frequency prescaler (CMCFG.CMPRE3 to CMPRE0)	Section 7.2

Setting	Setting register	Setting procedure
Setting of the source clock	Output source clock select (CMCFG.CMSEL3 to CMSEL0)	Section 7.1
Setting of the mark level	Clock monitor inverter (CSCFG.MONCKI)	Section 7.4
Enabling of clock monitor output	Output source clock select (CMCFG.CMSEL3 to CMSEL0)	Section 7.3

7. Q&A

This section explains the Q&A of clock monitor

7.1 How to Configure the Output Pin (MONCLK)

7.2 How to Select the Output Frequency

7.3 How to Enable or Disable Clock Monitor Output

7.4 How to Set the Clock Output Mark Level

7.1. How to Configure the Output Pin (MONCLK)

Setting of the output pin (MONCLK) is shown.

Use the output source clock selection bits (CMCFG.CMSEL3 to CMSEL0).

7.2. How to Select the Output Frequency

Selection of the output frequency is shown.

Use the output frequency prescaler bits (CMCFG.CMPRE3 to CMPRE0).

Clock division ratio	When output frequency (Example) HCLK is selected		Output frequency prescaler (CMCFG.CMPRE3 to CMPRE0)
	HCLK=32MHz	HCLK=40MHz	
1/2	16.0MHz	20.0MHz	Set to 0001_B
1/3	10.7MHz	13.3MHz	Set to 0010_B
1/4	8.0MHz	10.0MHz	Set to 0011_B
1/8	4.0MHz	5.0MHz	Set to 0111_B
1/15	2.1MHz	2.7MHz	Set to 1110_B
1/16	2.0MHz	2.5MHz	Set to 1111_B

7.3. How to Enable or Disable Clock Monitor Output

Enabling or disabling clock monitor output is shown.

Use the output source clock selection bits (CMCFG.CMSEL3 to CMSEL0).

Operation Description	Output source clock select bits (CMCFG.CMSEL3 to CMSEL0)
Disable clock monitor output (Set the pin to the high impedance state)	Set to 0000_B
Enable clock monitor output	Set to 0001_B to 1111_B (However, setting to 0100_B to 0111_B are prohibited)

7.4. How to Set the Clock Output Mark Level

Setting of the clock output mark level is shown.

Use the clock monitor MONCLK inverter bit (CSCFG.MONCKI).

8. Notes

This section explains the notes on clock monitor.

In order to achieve glitch-free switching, use the following procedure when changing the source clock (CMSEL3 to CMSEL0) or prescaler ratio (CMPRE3 to CMPRE0).

- The CMPRE3 to CMPRE0 registers can be written only when the CMSEL3 to CMSEL0 registers are "0_H".
- The CMPRE3 to CMPRE0 registers can be written only when "0H" is written to the CMSEL3 to CMSEL0 registers during the same write access.
- At least two cycles of the monitor clock division are required during the two write accesses to CMPRE and CMCFG.
- When selecting another effective clock while a clock is already selected as the clock source (CMSEL is not "0_H"), first set CMSEL to "0_H" and check that CMSEL returns "0_H" on read before writing the target clock setting value to CMSEL.
- If the clock selected as the monitor clock is stopped during monitoring, rewriting to any registers have no effect until the selected clock is started again or the unit is reset.

(Access example)

1. Access
   CNCFG.CMSEL = 0
   CMCFG.CMPRE = Prescaler
2. Access
   CMCFG.CMSEL = Clock

The CSCFG.MONCKI flag can also be written in the same procedure as above only when CMSEL3 to CMSEL0 are "0_H".

Appendix

Appendix is shown.

A. I/O Map

B. List of Interrupt Vector

C. Pins Statuses in State of CPU

D. Major Changes

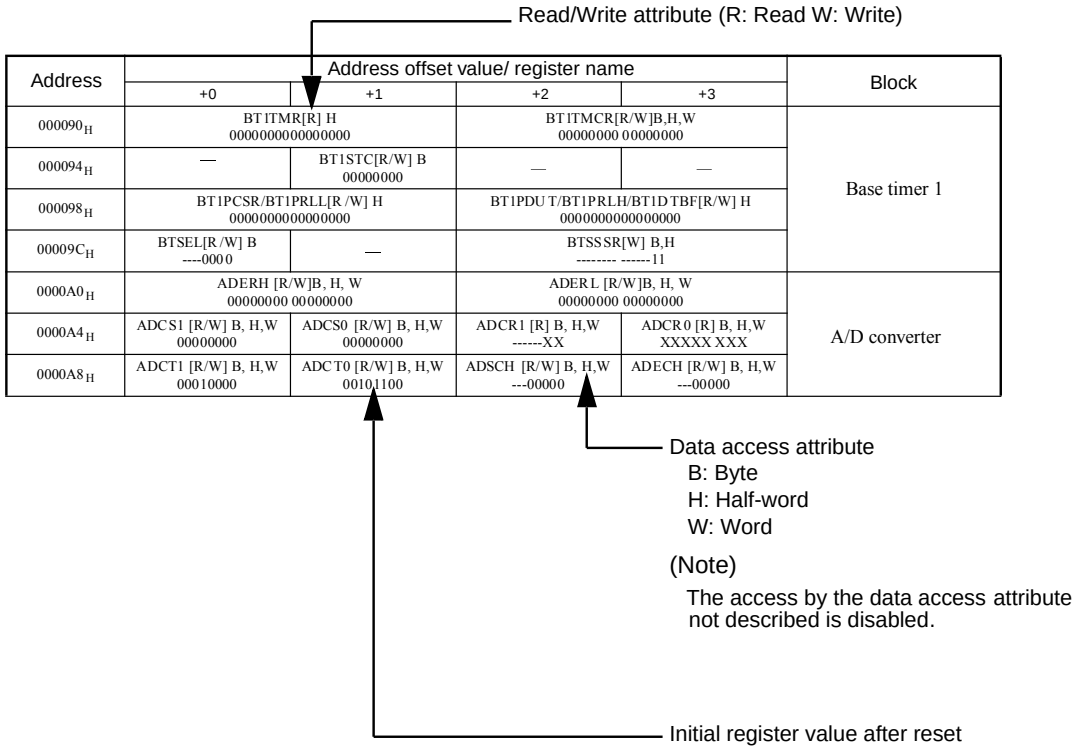
Code : APDX-1v0-91520-2-E

A. 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-1 Legend of I/O Map



The initial register value after reset indicates as follows:

- "1": Initial value "1"
- "0": Initial value "0"
- "X": Initial value undefined
- "-": Reserved bit/Undefined bit
- "*": Initial value "0" or "1" according to the setting

Note:

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 -XXXXXXX	PDR14 [R/W] B,H,W ---XXX--	PDR15 [R/W] B,H,W --XXXXXX	
000010_H	—	—	—	—	
000014_H	—	—	—	—	
000018_H	PDR16 [R/W] B,H,W XXXXXXXX	PDR17 [R/W] B,H,W XXXXXXXX	PDR18 [R/W] B,H,W XXXXXXXX	PDR19 [R/W] B,H,W XXXXXXXX	
$00001C_H$ to 000034_H	—	—	—	—	Reserved
000038_H	WDTECR0 [R/W] B,H,W ---00000	—	—	—	Watchdog Timer [S]
$00003C_H$	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	
000040_H	—	—	—	—	Reserved
000044_H	DICR [R/W] B -------0	—	—	—	Delayed Interrupt
000048_H to $00005C_H$	—	—	—	—	Reserved
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	TMRLRA7 [R/W] H XXXXXXXX XXXXXXXX		TMR7 [R] H XXXXXXXX XXXXXXXX		Reload Timer 7
$00006C_H$	TMRLRB7 [R/W] H XXXXXXXX XXXXXXXX		TMCSR7 [R/W] B,H,W 00000000 0-000000		
000070_H	—	FRS8 [R/W] B,H,W --00--00 --00--00 --00--00			Free-run timer selection register 8
000074_H	—	FRS9 [R/W] B,H,W --00--00 --00--00 --00--00			Free-run timer selection register 9
000078_H	—	—	—	OCLS67 [R/W] B,H,W ----0000	OCU67 Output level control register
$00007C_H$	—	—	—	OCLS89 [R/W]	OCU89 Output

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
				B,H,W ----0000	level control register
000080_H	BT0TMR [R] H 00000000 00000000		BT0TMCR [R/W] H -000--00 -000-000		Base Timer 0
000084_H	BT0TMCR2 [R/W] B -------0	BT0STC [R/W] B -0-0-0-0	—	—	
000088_H	BT0PCSR/BT0PRLL [R/W] H 00000000 00000000		BT0PDUT/BT0PRLH/BT0DTBF [R/W] H 00000000 00000000		
$00008C_H$	—	—	—	—	Reserved
000090_H	BT1TMR [R] H 00000000 00000000		BT1TMCR [R/W] H -000--00 -000-000		Base Timer 1
000094_H	BT1TMCR2 [R/W] B -------0	BT1STC [R/W] B -0-0-0-0	—	—	
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
$0000A0_H$ to $0000FC_H$	—	—	—	—	Reserved
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	MSCY4 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 4,5 Cycle measurement data register 45
$00011C_H$	MSCY5 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000120_H	OCCP6 [R/W] W 00000000 00000000 00000000 00000000				Output Compare 6,7 32-bit OCU
000124_H	OCCP7 [R/W] W 00000000 00000000 00000000 00000000				
000128_H	—	—	OCSH67 [R/W] B,H,W ---0--00	OCSL67 [R/W] B,H,W 0000--00	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$00012C_H$	OCCP8 [R/W] W 00000000 00000000 00000000 00000000				Output Compare 8,9 32-bit OCU
000130_H	OCCP9 [R/W] W 00000000 00000000 00000000 00000000				
000134_H	—	—	OCSH89 [R/W] B,H,W ---0--00	OCSL89 [R/W] B,H,W 0000--00	
000138_H to $0001B4_H$	—	—	—	—	Reserved
$0001B8_H$	EPFR64 [R/W] B,H,W -----00-	EPFR65 [R/W] B,H,W 0000-000	EPFR66 [R/W] B,H,W --000000	EPFR67 [R/W] B,H,W ----0000	Extended port function register
$0001BC_H$	EPFR68 [R/W] B,H,W ----0000	EPFR69 [R/W] B,H,W ----0000	EPFR70 [R/W] B,H,W ---00000	EPFR71 [R/W] B,H,W -0-0-0-0	
$0001C0_H$	EPFR72 [R/W] B,H,W 000000-0	EPFR73 [R/W] B,H,W 00000000	EPFR74 [R/W] B,H,W 00000000	EPFR75 [R/W] B,H,W 00000000	
$0001C4_H$	EPFR76 [R/W] B,H,W 00000000	EPFR77 [R/W] B,H,W --000000	EPFR78 [R/W] B,H,W ------00	EPFR79 [R/W] B,H,W 00000000	
$0001C8_H$	EPFR80 [R/W] B,H,W ---00000	EPFR81 [R/W] B,H,W 00000000	EPFR82 [R/W] B,H,W 00000000	EPFR83 [R/W] B,H,W -0000000	
$0001CC_H$	EPFR84 [R/W] B,H,W 00000000	EPFR85 [R/W] B,H,W --000000	EPFR86 [R/W] B,H,W ---00000	EPFR87 [R/W] B,H,W ------00	
$0001D0_H$	EPFR88 [R/W] B,H,W -------0	—	—	—	
$0001D4_H$	—	—	—	—	Reserved
$0001D8_H$	TMRLRA4 [R/W] H XXXXXXXX XXXXXXXX		TMR4 [R] H XXXXXXXX XXXXXXXX		Reload Timer 4
$0001DC_H$	TMRLRB4 [R/W] H XXXXXXXX XXXXXXXX		TMCSR4 [R/W] B, H,W 00000000 0-000000		
$0001E0_H$ to $0001EC_H$	—	—	—	—	Reserved
$0001F0_H$	TMRLRA5 [R/W] H XXXXXXXX XXXXXXXX		TMR5 [R] H XXXXXXXX XXXXXXXX		Reload Timer 5
$0001F4_H$	TMRLRB5 [R/W] H XXXXXXXX XXXXXXXX		TMCSR5 [R/W] B, H,W 00000000 0-000000		
$0001F8_H$	TMRLRA6 [R/W] H XXXXXXXX XXXXXXXX		TMR6 [R] H XXXXXXXX XXXXXXXX		Reload Timer 6

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0001FC_H$	TMRLRB6 [R/W] H XXXXXXXX XXXXXXXX		TMCSR6 [R/W] B, H,W 00000000 0-000000		Reload Timer 6
000200_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
000240_H	CPCLR3 [R/W] W 11111111 11111111 11111111 11111111				Free-run Timer 3 32-bit FRT
000244_H	TCDT3 [R/W] W 00000000 00000000 00000000 00000000				
000248_H	TCCSH3 [R/W] B,H,W 0-----00	TCCSL3 [R/W] B,H,W -1-00000	—	—	
$00024C_H$	CPCLR4 [R/W] W 11111111 11111111 11111111 11111111				Free-run Timer 4 32-bit FRT
000250_H	TCDT4 [R/W] W 00000000 00000000 00000000 00000000				
000254_H	TCCSH4 [R/W] B,H,W 0-----00	TCCSL4 [R/W] B,H,W -1-00000	—	—	
000258_H to $0002C0_H$	—	—	—	—	Reserved
$0002C4_H$ to $0002FC_H$	—	—	—	—	Reserved
000300_H to $00030C_H$	—	—	—	—	Reserved
000310_H	—	—	MPUCR [R/W] H 000000-0 ----0100		MPU [S] (Only CPU core 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		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000338_H	PABR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				MPU [S] (Only CPU core can access this area)
$00033C_H$	—	—	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		
000348_H	PABR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00034C_H$	—	—	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 to $0003AC_H$	—				Reserved [S]
$0003B0_H$ 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	DMA request generation and clear
000404_H	—	ICSEL5 [R/W] B,H,W -----000	ICSEL6 [R/W] B,H,W ----0000	ICSEL7 [R/W] B,H,W ----0000	
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 -----000	
$00040C_H$	—	ICSEL13 [R/W] B,H,W ------00	ICSEL14 [R/W] B,H,W ------00	ICSEL15 [R/W] B,H,W ------00	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000410_H	ICSEL16 [R/W] B,H,W ----0000	ICSEL17 [R/W] B,H,W ------00	ICSEL18 [R/W] B,H,W ---00000	ICSEL19 [R/W] B,H,W -----000	DMA request generation and clear
000414_H	ICSEL20 [R/W] B,H,W -----000	ICSEL21 [R/W] B,H,W ------00	ICSEL22 [R/W] B,H,W ------00	ICSEL23 [R/W] B,H,W ------00	
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 Reading Register
$00041C_H$	—	—	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 --00----	IRPR6L [R] B,H,W 0000----	IRPR7H [R] B,H,W -0-00---	IRPR7L [R] B,H,W ------00	
000428_H	IRPR8H [R] B,H,W --0-----	IRPR8L [R] B,H,W -00-----	IRPR9H [R] B,H,W -0------	IRPR9L [R] B,H,W -0------	
$00042C_H$	IRPR10H [R] B,H,W -0------	IRPR10L [R] B,H,W -0------	IRPR11H [R] B,H,W 0-------	IRPR11L [R] B,H,W 0-------	
000430_H	IRPR12H [R] B,H,W --0000--	IRPR12L [R] B,H,W ----00--	IRPR13H [R] B,H,W 00------	IRPR13L [R] B,H,W 00------	
000434_H	IRPR14H [R] B,H,W 00000000	IRPR14L [R] B,H,W 00000000	IRPR15H [R] B,H,W 000-----	IRPR15L [R] B,H,W 0000000-	
000438_H	ICSEL24 [R/W] B,H,W ------00	ICSEL25 [R/W] B,H,W ---00000	ICSEL26 [R/W] B,H,W -------0	ICSEL27 [R/W] B,H,W -------0	DMA request generation and clear
$00043C_H$	—	—	—	—	Reserved [S]
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 * 000---11	—	Reset Control [S] Power Control [S] * Writing STBCR by DMA is forbidden
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 request by 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 ---00000	—	—	—	CAN prescaler
$0004A8_H$	—	—	CSCFG[R/W]B,H,W ---0----	CMCFG[R/W]B,H,W 00000000	Clock monitor control register
$0004AC_H$	ADERH0[R/W] B,H 11111111 11111111		ADERL0[R/W] B,H 11111111 11111111		Analog input control register 0
$0004B0_H$	—		ADERL1[R/W] B,H 11111111 11111111		Analog input control register 1
$0004B4_H$	—	—	—	—	Reserved
$0004B8_H$	CUCR0 [R/W] B,H,W -------- ---0--00		CUTD0 [R/W] B,H,W 10000000 00000000		RTC/WDT1 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		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0004C8_H$	CUTR1 [R] B,H,W -------- 00000000 00000000 00000000				RTC/WDT1 calibration
$0004CC_H$ 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 Control [S]
$00051C_H$	—	—	—	—	Reserved [S]
000520_H	CCPSSELR [R/W] B,H,W -------0	—	—	CCPSDIVR [R/W] B,H,W -000-000	Clock Control 2 [S]
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 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 7)
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 15)
000558_H	—	—	—	—	Reserved
$00055C_H$	—	—	WTDR [R/W] H 00000000 00000000		Real Time Clock (RTC)
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 000111--	—	—	Clock Supervisor

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000570_H to $00057C_H$	—	—	—	—	Reserved
000580_H	REGSEL [R/W] B,H,W 0110011-	—	—	—	Regulator Control / Low-Voltage Detection
000584_H	LVD5R [R/W] B,H,W -------1	LVD5F [R/W] B,H,W 00000001	LVD [R/W] B,H,W 01000--0	—	
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 $0005BC_H$	—	—	—	—	Reserved
$0005C0_H$ 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]
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]

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000684_H	AWR1 [R/W] W ----XXXX XXXXXXXX XXXXXXXX XXXXX-X-				External Bus Interface [S]
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 $0006FC_H$	—	—	—	—	Reserved [S]
000700_H to $00070C_H$	—	—	—	—	Reserved
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	—	—	—	Mode Register
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 to 000854_H	—	—	—	—	Reserved [S]
000858_H	—	—	WREN [R/W] H 00000000 00000000		Wild Register [S]
$00085C_H$ to $00087C_H$	—	—	—	—	Reserved [S]
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				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000898_H	WRAR03 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild Register [S]
$00089C_H$	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				
$0008C0_H$	WRAR08 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008C4_H$	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				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0008F0_H$	WRAR14 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild Register [S]
$0008F4_H$	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	TPUUNLOCK [R/W] W 00000000 00000000 00000000 00000000				Time Protection Unit [S]
000904_H	TPULST [R] B,H,W -------0	—	TPUVST [R/W] B,H,W -----000	—	
000908_H	TPUCFG [R/W] B,H,W -------0 0-000000 -------- -------0				
$00090C_H$	TPUTIR [R] B,H,W 00000000	—	—	—	
000910_H	TPUTST [R] B,H,W 00000000	—	—	—	
000914_H	TPUTIE [R/W] B,H,W 00000000	—	—	—	
000918_H	TPUTMID [R] B,H,W 00000000 00000000 00000000 00000000				
$00091C_H$ to $00092C_H$	—	—	—	—	
000930_H	TPUTCN00 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000934_H	TPUTCN01 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000938_H	TPUTCN02 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
$00093C_H$	TPUTCN03 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000940_H	TPUTCN04 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000944_H	TPUTCN05 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000948_H	TPUTCN06 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
$00094C_H$	TPUTCN07 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000950_H	TPUTCN10 [R/W] B,H,W ---00000	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000954_H	TPUTCN11 [R/W] B,H,W ---00000	—	—	—	Time Protection Unit [S]
000958_H	TPUTCN12 [R/W] B,H,W ---00000	—	—	—	
$00095C_H$	TPUTCN13 [R/W] B,H,W ---00000	—	—	—	
000960_H	TPUTCN14 [R/W] B,H,W ---00000	—	—	—	
000964_H	TPUTCN15 [R/W] B,H,W ---00000	—	—	—	
000968_H	TPUTCN16 [R/W] B,H,W ---00000	—	—	—	
$00096C_H$	TPUTCN17 [R/W] B,H,W ---00000	—	—	—	
000970_H	TPUTCC0 [R] B,H,W -------- 00000000 00000000 00000000				
000974_H	TPUTCC1 [R] B,H,W -------- 00000000 00000000 00000000				
000978_H	TPUTCC2 [R] B,H,W -------- 00000000 00000000 00000000				
$00097C_H$	TPUTCC3 [R] B,H,W -------- 00000000 00000000 00000000				
000980_H	TPUTCC4 [R] B,H,W -------- 00000000 00000000 00000000				
000984_H	TPUTCC5 [R] B,H,W -------- 00000000 00000000 00000000				
000988_H	TPUTCC6 [R] B,H,W -------- 00000000 00000000 00000000				
$00098C_H$	TPUTCC7 [R] B,H,W -------- 00000000 00000000 00000000				
000990_H to $0009FC_H$	—	—	—	—	
$000A00_H$ to $000BEC_H$	—	—	—	—	Reserved
$000BF0_H$	HSCFR [R/W] B,H,W -------- ------00 00000000 00000000				OCDU
$000BF4_H$	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000BF8_H$	—	—	MBR [R/W] B,H,W 00------ XXXXXXXX		OCDU
$000BFC_H$	—	—	UER [W] B,H,W -------- -------X		
$000C00_H$	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				
$000C4C_H$	DDAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000C50_H$	DCCR5 [R/W] W 0----000 --00--00 00000000 0-000000				DMA Controller [S]
$000C54_H$	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				
$000C9C_H$	DDAR9 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CA0_H$	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		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000CA8_H$	DSAR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA Controller [S]
$000CAC_H$	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		
$000CE8_H$	DSAR14 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CEC_H$	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				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000D00_H$ 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 00000000	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	Data Direction Register
$000E0C_H$	DDR12 [R/W] B,H,W 00000000	DDR13 [R/W] B,H,W -0000000	DDR14 [R/W] B,H,W ---000--	DDR15 [R/W] B,H,W --000000	
$000E10_H$	—	—	—	—	
$000E14_H$	—	—	—	—	
$000E18_H$	DDR16 [R/W] B,H,W 00000000	DDR17 [R/W] B,H,W 00000000	DDR18 [R/W] B,H,W 00000000	DDR19 [R/W] B,H,W 00000000	
$000E1C_H$	—	—	—	—	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 00000000	Port Function Register
$000E24_H$	PFR04 [R/W] B,H,W 00000000	PFR05 [R/W] B,H,W 00000000	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 00000000	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 -0000000	PFR14 [R/W] B,H,W ---000--	PFR15 [R/W] B,H,W --000000	
$000E30_H$	—	—	—	—	
$000E34_H$	—	—	—	—	
$000E38_H$	PFR16 [R/W] B,H,W 00000000	PFR17 [R/W] B,H,W 00000000	PFR18 [R/W] B,H,W 00000000	PFR19 [R/W] B,H,W 00000000	
$000E3C_H$	—	—	—	—	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	Port 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 -XXXXXXX	PDDR14 [R] B,H,W ---XXX--	PDDR15 [R] B,H,W --XXXXXX	
$000E50_H$	—	—	—	—	Port Direct

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000E54_H$	—	—	—	—	Read Register
$000E58_H$	PDDR16 [R] B,H,W XXXXXXXX	PDDR17 [R] B,H,W XXXXXXXX	PDDR18 [R] B,H,W XXXXXXXX	PDDR19 [R] B,H,W XXXXXXXX	
$000E5C_H$	—	—	—	—	Reserved
$000E60_H$	EPFR00 [R/W] B,H,W 00000000	EPFR01 [R/W] B,H,W -0-0-000	EPFR02 [R/W] B,H,W ----0000	EPFR03 [R/W] B,H,W ---000-0	Extended Port Function Register
$000E64_H$	EPFR04 [R/W] B,H,W ----00-0	EPFR05 [R/W] B,H,W ----0000	EPFR06 [R/W] B,H,W ----000-	EPFR07 [R/W] B,H,W ---00000	
$000E68_H$	EPFR08 [R/W] B,H,W ---00000	EPFR09 [R/W] B,H,W -----00-	EPFR10 [R/W] B,H,W ----0000	EPFR11 [R/W] B,H,W ----0000	Extended Port Function Register
$000E6C_H$	EPFR12 [R/W] B,H,W ----0000	EPFR13 [R/W] B,H,W ------00	EPFR14 [R/W] B,H,W ------00	EPFR15 [R/W] B,H,W -----000	
$000E70_H$	—	—	—	—	
$000E74_H$	—	—	—	—	
$000E78_H$	—	—	EPFR26 [R/W] B,H,W 00000000	EPFR27 [R/W] B,H,W ---0----	
$000E7C_H$	EPFR28 [R/W] B,H,W --000-0-	EPFR29 [R/W] B,H,W 00000000	—	—	
$000E80_H$	—	EPFR33 [R/W] B,H,W -----00-	EPFR34 [R/W] B,H,W -----00-	EPFR35 [R/W] B,H,W ---00000	
$000E84_H$	EPFR36 [R/W] B,H,W ----000-	—	—	—	
$000E88_H$	—	—	EPFR42 [R/W] B,H,W ------00	EPFR43 [R/W] B,H,W 0--0000-	
$000E8C_H$	EPFR44 [R/W] B,H,W -00---0-	EPFR45 [R/W] B,H,W -0000000	—	—	
$000E90_H$	—	—	—	—	
$000E94_H$	—	—	—	—	
$000E98_H$	EPFR56 [R/W] B,H,W -----0-0	EPFR57 [R/W] B,H,W ----00-0	EPFR58 [R/W] B,H,W ----00-0	EPFR59 [R/W] B,H,W ----00-0	
$000E9C_H$	EPFR60 [R/W] B,H,W ----00-0	EPFR61 [R/W] B,H,W -----00-	EPFR62 [R/W] B,H,W -----00-	EPFR63 [R/W] B,H,W ---0000-	
$000EA0_H$	—	—	—	—	Reserved

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
to $000EBC_H$					
$000EC0_H$	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 -0000000	PPER14 [R/W] B,H,W ---000--	PPER15 [R/W] B,H,W --000000	
$000ED0_H$	—	—	—	—	
$000ED4_H$	—	—	—	—	
$000ED8_H$	PPER16 [R/W] B,H,W 00000000	PPER17 [R/W] B,H,W 00000000	PPER18 [R/W] B,H,W 00000000	PPER19 [R/W] B,H,W 00000000	
$000EDC_H$ to $000F3C_H$	—	—	—	—	Reserved
$000F40_H$	PORTEN [R/W] B,H,W -------0	—	—	—	Port Enable Register
$000F44_H$	KEYCDR [R/W] H 00000000 00000000		—	—	KeyCodeRegister
$000F48_H$ to $000F64_H$	—	—	—	—	Reserved
$000F68_H$	MSCY6 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 6,7 Cycle measurement data register 67
$000F6C_H$	MSCY7 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000F70_H$	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	
$000F88_H$	—	—	MSCH45 [R] B,H,W 00000000	MSCL45 [R/W] B,H,W ------00	Input Capture 4,5 32-bit ICU Cycle and pulse width measurement control 45
$000F8C_H$	—	—	MSCH67 [R]	MSCL67 [R/W]	Input Capture 6,7

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
			B,H,W 00000000	B,H,W ------00	32-bit ICU Cycle and pulse width measurement control 67
$000F90_H$	OCCP10 [R/W] W 00000000 00000000 00000000 00000000				Output Compare 10,11 32-bit OCU
$000F94_H$	OCCP11 [R/W] W 00000000 00000000 00000000 00000000				
$000F98_H$	—	—	OCSH1011 [R/W] B,H,W ---0--00	OCSL1011 [R/W] B,H,W 0000--00	
$000F9C_H$	—	—	—	OCLS1011 [R/W] B,H,W ----0000	OCU1011 Output level control register
$000FA0_H$	CPCLR5 [R/W] W 11111111 11111111 11111111 11111111				Free-run Timer 5 32-bit FRT
$000FA4_H$	TCDT5 [R/W] W 00000000 00000000 00000000 00000000				
$000FA8_H$	TCCSH5 [R/W]B,H,W 0-----00	TCCSL5 [R/W]B,H,W -1-00000	—	—	
$000FAC_H$ to $000FCC_H$	—	—	—	—	Reserved
$000FD0_H$	IPCP4 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 4,5 32-bit ICU
$000FD4_H$	IPCP5 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FD8_H$	—	—	LSYNS1 [R/W] B,H,W 00000000	ICS45 [R/W] B,H,W 00000000	
$000FDC_H$	IPCP6 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 6,7 32-bit ICU
$000FE0_H$	IPCP7 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FE4_H$	—	—	—	ICS67 [R/W] B,H,W 00000000	
$000FE8_H$	IPCP8 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 8,9 32-bit ICU
$000FEC_H$	IPCP9 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FF0_H$	—	—	—	ICS89 [R/W] B,H,W 00000000	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$000FF4_H$	MSCY8 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input Capture 8,9 32-bit ICU Cycle measurement data register 89
$000FF8_H$	MSCY9 [R] H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FFC_H$	—	—	MSCH89 [R] B,H,W 00000000	MSCL89 [R/W] B,H,W ------00	Cycle and pulse width measurement control 89
001000_H	SACR [R/W] B,H,W -------0	PICD [R/W] B,H,W ----0011	—	—	Clock Control
001004_H to $00112C_H$	—	—	—	—	Reserved
001130_H	—	—	—	CRCCR [R/W] B,H,W -0000000	CRC calculation unit
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 $0011FC_H$	—	—	—	—	Reserved
001200_H	TCGS [R/W] B,H,W ------00	—	—	TCGSE [R/W] B,H,W -----000	16-bit Free-run timer synchronous activation
001204_H	CPCLRB0/CPCLR0 [W] H,W 11111111 11111111		TCDT0 [R/W] H,W 00000000 00000000		16-bit Free-run Timer 0
001208_H	TCCS0 [R/W] B,H,W 00000000 01000000 ----0000 --------				
$00120C_H$	CPCLRB1/CPCLR1 [W] H,W 11111111 11111111		TCDT1 [R/W] H,W 00000000 00000000		16-bit Free-run Timer 1
001210_H	TCCS1 [R/W] B,H,W 00000000 01000000 ----0000 --------				
001214_H	CPCLRB2/CPCLR2 [W] H,W 11111111 11111111		TCDT2 [R/W] H,W 00000000 00000000		16-bit Free-run Timer 2
001218_H	TCCS2 [R/W] B,H,W 00000000 01000000 ----0000 --------				
$00121C_H$ to 001230_H	—	—	—	—	Reserved
001234_H	FRS0 [R/W] B,H,W -------- --00--00 --00--00 --00--00				16-bit Free-run timer selection
001238_H	—		FRS1 [R/W] B,H,W --00--00 --00--00		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$00123C_H$	FRS2 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				16-bit Free-run timer selection
001240_H	FRS3 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				
001244_H	FRS4 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				
001248_H	—	—	—	—	Reserved
$00124C_H$	OCCPB0/OCCP0 [R/W] H,W 00000000 00000000		OCCPB1/OCCP1 [R/W] H,W 00000000 00000000		16-bit Output compare 0/1
001250_H	OCS01 [R/W] B,H,W -110--00 00001100		—	OCMOD01 [R/W] B,H,W ------00	
001254_H	OCCPB2/OCCP2 [R/W] H,W 00000000 00000000		OCCPB3/OCCP3 [R/W] H,W 00000000 00000000		16-bit Output compare 2/3
001258_H	OCS23 [R/W] B,H,W -110--00 00001100		—	OCMOD23 [R/W] B,H,W ------00	
$00125C_H$	OCCPB4/OCCP4 [R/W] H,W 00000000 00000000		OCCPB5/OCCP5 [R/W] H,W 00000000 00000000		16-bit Output compare 4/5
001260_H	OCS45 [R/W] B,H,W -110--00 00001100		—	OCMOD45 [R/W] B,H,W ------00	
001264_H to 001278_H	—	—	—	—	Reserved
$00127C_H$	IPCP0 [R] H,W 00000000 00000000		IPCP1 [R] H,W 00000000 00000000		16-bit Input capture 0/1
001280_H	ICS01 [R/W] B,H,W ------00 00000000		—	LSYNS [R/W] B,H,W ----0000	
001284_H	IPCP2 [R] H,W 00000000 00000000		IPCP3 [R] H,W 00000000 00000000		16-bit Input capture 2/3
001288_H	ICS23 [R/W] B,H,W ------00 00000000		—	—	
$00128C_H$ to 001298_H	—	—	—	—	Reserved
$00129C_H$	—	—	—	—	Reserved
$0012A0_H$	TMRR0 [R/W] H,W 00000000 00000001		TMRR1 [R/W] H,W 00000000 00000001		Waveform generator 0/1/2
$0012A4_H$	TMRR2 [R/W] H,W 00000000 00000001		—	—	
$0012A8_H$	DTSCR0 [R/W] B,H,W 00000000	DTSCR1 [R/W] B,H,W 00000000	DTSCR2 [R/W] B,H,W 00000000	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0012AC_H$	—	DTIR0 [R/W] B,H,W 000000--	—	DTMNS0 [R/W] B,H,W 00---000	Waveform generator 0/1/2
$0012B0_H$	—	SIGCR10 [R/W] B,H,W 00000000	—	SIGCR20 [R/W] B,H,W 000000-1	
$0012B4_H$	PICS0 [R/W] B,H,W 000000-- -------- -------- --------				
$0012B8_H$ to $0012CC_H$	—	—	—	—	Reserved
$0012D0_H$	FRS5 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				16-bit Free-run timer selection A/D activation compare
$0012D4_H$	FRS6 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				
$0012D8_H$	FRS7 [R/W] B,H,W --00--00 --00--00 --00--00 --00--00				
$0012DC_H$ to $0012FC_H$	—	—	—	—	Reserved
001300_H	—				Reserved
001304_H	ADTSS0[R/W] B,H,W -------0	—	—	—	12-bit A/D converter 1/2 unit
001308_H	ADTSE0[R/W] B,H,W 00000000 00000000 00000000 00000000				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$00130C_H$	ADCOMP0/ADCOMPB0[R/W] H,W 00000000 00000000		ADCOMP1/ADCOMPB1[R/W] H,W 00000000 00000000		12-bit A/D converter 1/2 unit
001310_H	ADCOMP2/ADCOMPB2[R/W] H,W 00000000 00000000		ADCOMP3/ADCOMPB3[R/W] H,W 00000000 00000000		
001314_H	ADCOMP4/ADCOMPB4[R/W] H,W 00000000 00000000		ADCOMP5/ADCOMPB5[R/W] H,W 00000000 00000000		
001318_H	ADCOMP6/ADCOMPB6[R/W] H,W 00000000 00000000		ADCOMP7/ADCOMPB7[R/W] H,W 00000000 00000000		
$00131C_H$	ADCOMP8/ADCOMPB8[R/W] H,W 00000000 00000000		ADCOMP9/ADCOMPB9[R/W] H,W 00000000 00000000		
001320_H	ADCOMP10/ADCOMPB10[R/W] H,W 00000000 00000000		ADCOMP11/ADCOMPB11[R/W] H,W 00000000 00000000		
001324_H	ADCOMP12/ADCOMPB12[R/W] H,W 00000000 00000000		ADCOMP13/ADCOMPB13[R/W] H,W 00000000 00000000		
001328_H	ADCOMP14/ADCOMPB14[R/W] H,W 00000000 00000000		ADCOMP15/ADCOMPB15[R/W] H,W 00000000 00000000		
$00132C_H$	ADCOMP16/ADCOMPB16[R/W] H,W 00000000 00000000		ADCOMP17/ADCOMPB17[R/W] H,W 00000000 00000000		
001330_H	ADCOMP18/ADCOMPB18[R/W] H,W 00000000 00000000		ADCOMP19/ADCOMPB19[R/W] H,W 00000000 00000000		
001334_H	ADCOMP20/ADCOMPB20[R/W] H,W 00000000 00000000		ADCOMP21/ADCOMPB21[R/W] H,W 00000000 00000000		
001338_H	ADCOMP22/ADCOMPB22[R/W] H,W 00000000 00000000		ADCOMP23/ADCOMPB23[R/W] H,W 00000000 00000000		
$00133C_H$	ADCOMP24/ADCOMPB24[R/W] H,W 00000000 00000000		ADCOMP25/ADCOMPB25[R/W] H,W 00000000 00000000		
001340_H	ADCOMP26/ADCOMPB26[R/W] H,W 00000000 00000000		ADCOMP27/ADCOMPB27[R/W] H,W 00000000 00000000		
001344_H	ADCOMP28/ADCOMPB28[R/W] H,W 00000000 00000000		ADCOMP29/ADCOMPB29[R/W] H,W 00000000 00000000		
001348_H	ADCOMP30/ADCOMPB30[R/W] H,W 00000000 00000000		ADCOMP31/ADCOMPB31[R/W] H,W 00000000 00000000		
$00134C_H$	ADTCS0[R/W] B,H,W 00000000 0010----		ADTCS1[R/W] B,H,W 00000000 0010----		
001350_H	ADTCS2[R/W] B,H,W 00000000 0010----		ADTCS3[R/W] B,H,W 00000000 0010----		
001354_H	ADTCS4[R/W] B,H,W 00000000 0010----		ADTCS5[R/W] B,H,W 00000000 0010----		
001358_H	ADTCS6[R/W] B,H,W 00000000 0010----		ADTCS7[R/W] B,H,W 00000000 0010----		
$00135C_H$	ADTCS8[R/W] B,H,W 00000000 0010----		ADTCS9[R/W] B,H,W 00000000 0010----		
001360_H	ADTCS10[R/W] B,H,W 00000000 0010----		ADTCS11[R/W] B,H,W 00000000 0010----		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001364_H	ADTCS12[R/W] B,H,W 00000000 0010----		ADTCS13[R/W] B,H,W 00000000 0010----		12-bit A/D converter 1/2 unit
001368_H	ADTCS14[R/W] B,H,W 00000000 0010----		ADTCS15[R/W] B,H,W 00000000 0010----		
$00136C_H$	ADTCS16[R/W] B,H,W 00000000 0010----		ADTCS17[R/W] B,H,W 00000000 0010----		
001370_H	ADTCS18[R/W] B,H,W 00000000 0010----		ADTCS19[R/W] B,H,W 00000000 0010----		
001374_H	ADTCS20[R/W] B,H,W 00000000 0010----		ADTCS21[R/W] B,H,W 00000000 0010----		
001378_H	ADTCS22[R/W] B,H,W 00000000 0010----		ADTCS23[R/W] B,H,W 00000000 0010----		
$00137C_H$	ADTCS24[R/W] B,H,W 00000000 0010----		ADTCS25[R/W] B,H,W 00000000 0010----		
001380_H	ADTCS26[R/W] B,H,W 00000000 0010----		ADTCS27[R/W] B,H,W 00000000 0010----		
001384_H	ADTCS28[R/W] B,H,W 00000000 0010----		ADTCS29[R/W] B,H,W 00000000 0010----		
001388_H	ADTCS30[R/W] B,H,W 00000000 0010----		ADTCS31[R/W] B,H,W 00000000 0010----		
$00138C_H$	ADTCD0[R] B,H,W 10--0000 00000000		ADTCD1[R] B,H,W 10--0000 00000000		
001390_H	ADTCD2[R] B,H,W 10--0000 00000000		ADTCD3[R] B,H,W 10--0000 00000000		
001394_H	ADTCD4[R] B,H,W 10--0000 00000000		ADTCD5[R] B,H,W 10--0000 00000000		
001398_H	ADTCD6[R] B,H,W 10--0000 00000000		ADTCD7[R] B,H,W 10--0000 00000000		
$00139C_H$	ADTCD8[R] B,H,W 10--0000 00000000		ADTCD9[R] B,H,W 10--0000 00000000		
$0013A0_H$	ADTCD10[R] B,H,W 10--0000 00000000		ADTCD11[R] B,H,W 10--0000 00000000		
$0013A4_H$	ADTCD12[R] B,H,W 10--0000 00000000		ADTCD13[R] B,H,W 10--0000 00000000		
$0013A8_H$	ADTCD14[R] B,H,W 10--0000 00000000		ADTCD15[R] B,H,W 10--0000 00000000		
$0013AC_H$	ADTCD16[R] B,H,W 10--0000 00000000		ADTCD17[R] B,H,W 10--0000 00000000		
$0013B0_H$	ADTCD18[R] B,H,W 10--0000 00000000		ADTCD19[R] B,H,W 10--0000 00000000		
$0013B4_H$	ADTCD20[R] B,H,W 10--0000 00000000		ADTCD21[R] B,H,W 10--0000 00000000		
$0013B8_H$	ADTCD22[R] B,H,W 10--0000 00000000		ADTCD23[R] B,H,W 10--0000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0013BC_H$	ADTCD24[R] B,H,W 10--0000 00000000		ADTCD25[R] B,H,W 10--0000 00000000		12-bit A/D converter 1/2 unit
$0013C0_H$	ADTCD26[R] B,H,W 10--0000 00000000		ADTCD27[R] B,H,W 10--0000 00000000		
$0013C4_H$	ADTCD28[R] B,H,W 10--0000 00000000		ADTCD29[R] B,H,W 10--0000 00000000		
$0013C8_H$	ADTCD30[R] B,H,W 10--0000 00000000		ADTCD31[R] B,H,W 10--0000 00000000		
$0013CC_H$	ADTECS0[R/W] B,H,W -------0 ---00000		ADTECS1[R/W] B,H,W -------0 ---00000		
$0013D0_H$	ADTECS2[R/W] B,H,W -------0 ---00000		ADTECS3[R/W] B,H,W -------0 ---00000		
$0013D4_H$	ADTECS4[R/W] B,H,W -------0 ---00000		ADTECS5[R/W] B,H,W -------0 ---00000		
$0013D8_H$	ADTECS6[R/W] B,H,W -------0 ---00000		ADTECS7[R/W] B,H,W -------0 ---00000		
$0013DC_H$	ADTECS8[R/W] B,H,W -------0 ---00000		ADTECS9[R/W] B,H,W -------0 ---00000		
$0013E0_H$	ADTECS10[R/W] B,H,W -------0 ---00000		ADTECS11[R/W] B,H,W -------0 ---00000		
$0013E4_H$	ADTECS12[R/W] B,H,W -------0 ---00000		ADTECS13[R/W] B,H,W -------0 ---00000		
$0013E8_H$	ADTECS14[R/W] B,H,W -------0 ---00000		ADTECS15[R/W] B,H,W -------0 ---00000		
$0013EC_H$	ADTECS16[R/W] B,H,W -------0 ---00000		ADTECS17[R/W] B,H,W -------0 ---00000		
$0013F0_H$	ADTECS18[R/W] B,H,W -------0 ---00000		ADTECS19[R/W] B,H,W -------0 ---00000		
$0013F4_H$	ADTECS20[R/W] B,H,W -------0 ---00000		ADTECS21[R/W] B,H,W -------0 ---00000		
$0013F8_H$	ADTECS22[R/W] B,H,W -------0 ---00000		ADTECS23[R/W] B,H,W -------0 ---00000		
$0013FC_H$	ADTECS24[R/W] B,H,W -------0 ---00000		ADTECS25[R/W] B,H,W -------0 ---00000		
001400_H	ADTECS26[R/W] B,H,W -------0 ---00000		ADTECS27[R/W] B,H,W -------0 ---00000		
001404_H	ADTECS28[R/W] B,H,W -------0 ---00000		ADTECS29[R/W] B,H,W -------0 ---00000		
001408_H	ADTECS30[R/W] B,H,W -------0 ---00000		ADTECS31[R/W] B,H,W -------0 ---00000		
$00140C_H$	ADRCUT0[R/W] B,H,W ----0000 00000000		ADRCLT0[R/W] B,H,W ----0000 00000000		
001410_H	ADRCUT1[R/W] B,H,W ----0000 00000000		ADRCLT1[R/W] B,H,W ----0000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001414_H	ADRCUT2[R/W] B,H,W ----0000 00000000		ADRCLT2[R/W] B,H,W ----0000 00000000		12-bit A/D converter 1/2 unit
001418_H	ADRCUT3[R/W] B,H,W ----0000 00000000		ADRCLT3[R/W] B,H,W ----0000 00000000		
$00141C_H$	ADRCCS0[R/W] B,H,W 00000000	ADRCCS1[R/W] B,H,W 00000000	ADRCCS2[R/W] B,H,W 00000000	ADRCCS3[R/W] B,H,W 00000000	
001420_H	ADRCCS4[R/W] B,H,W 00000000	ADRCCS5[R/W] B,H,W 00000000	ADRCCS6[R/W] B,H,W 00000000	ADRCCS7[R/W] B,H,W 00000000	
001424_H	ADRCCS8[R/W] B,H,W 00000000	ADRCCS9[R/W] B,H,W 00000000	ADRCCS10[R/W] B,H,W 00000000	ADRCCS11[R/W] B,H,W 00000000	
001428_H	ADRCCS12[R/W] B,H,W 00000000	ADRCCS13[R/W] B,H,W 00000000	ADRCCS14[R/W] B,H,W 00000000	ADRCCS15[R/W] B,H,W 00000000	
$00142C_H$	ADRCCS16[R/W] B,H,W 00000000	ADRCCS17[R/W] B,H,W 00000000	ADRCCS18[R/W] B,H,W 00000000	ADRCCS19[R/W] B,H,W 00000000	
001430_H	ADRCCS20[R/W] B,H,W 00000000	ADRCCS21[R/W] B,H,W 00000000	ADRCCS22[R/W] B,H,W 00000000	ADRCCS23[R/W] B,H,W 00000000	
001434_H	ADRCCS24[R/W] B,H,W 00000000	ADRCCS25[R/W] B,H,W 00000000	ADRCCS26[R/W] B,H,W 00000000	ADRCCS27[R/W] B,H,W 00000000	
001438_H	ADRCCS28[R/W] B,H,W 00000000	ADRCCS29[R/W] B,H,W 00000000	ADRCCS30[R/W] B,H,W 00000000	ADRCCS31[R/W] B,H,W 00000000	
$00143C_H$	ADRCOT0[R] B,H,W 00000000 00000000 00000000 00000000				
001440_H	ADRCIF0[R,W] B,H,W 00000000 00000000 00000000 00000000				
001444_H	ADSCANS0[R/W] B,H,W 000-----	—	—	—	
001448_H	ADNCS0[R/W] B,H,W 0-000-00	ADNCS1[R/W] B,H,W 0-000-00	ADNCS2[R/W] B,H,W 0-000-00	ADNCS3[R/W] B,H,W 0-000-00	
$00144C_H$	ADNCS4[R/W] B,H,W 0-000-00	ADNCS5[R/W] B,H,W 0-000-00	ADNCS6[R/W] B,H,W 0-000-00	ADNCS7[R/W] B,H,W 0-000-00	
001450_H	ADNCS8[R/W] B,H,W 0-000-00	ADNCS9[R/W] B,H,W 0-000-00	ADNCS10[R/W] B,H,W 0-000-00	ADNCS11[R/W] B,H,W 0-000-00	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001454_H	ADNCS12[R/W] B,H,W 0-000-00	ADNCS13[R/W] B,H,W 0-000-00	ADNCS14[R/W] B,H,W 0-000-00	ADNCS15[R/W] B,H,W 0-000-00	12-bit A/D converter 1/2 unit
001458_H	ADPRTF0[R] B,H,W 00000000 00000000 00000000 00000000				
$00145C_H$	ADEOCF0[R] B,H,W 11111111 11111111 11111111 11111111				
001460_H	ADCS0[R] B,H,W 0------- --------		ADCH0[R] B,H,W ---00000	ADMD0[R/W] B,H,W 0---0000	
001464_H	ADSTPCS0[R/W] B,H,W 00000000	ADSTPCS1[R/W] B,H,W 00000000	ADSTPCS2[R/W] B,H,W 00000000	ADSTPCS3[R/W] B,H,W 00000000	
001468_H	ADSTPCS4[R/W] B,H,W 00000000	ADSTPCS5[R/W] B,H,W 00000000	ADSTPCS6[R/W] B,H,W 00000000	ADSTPCS7[R/W] B,H,W 00000000	
$00146C_H$	—				12-bit A/D converter 2/2 unit
001470_H	ADTSS1[R/W] B,H,W -------0	—	—	—	
001474_H	ADTSE1[R/W] B,H,W -------- -------- 00000000 00000000				
001478_H	ADCOMP32/ADCOMPB32[R/W] H,W 00000000 00000000		ADCOMP33/ADCOMPB33[R/W] H,W 00000000 00000000		
$00147C_H$	ADCOMP34/ADCOMPB34[R/W] H,W 00000000 00000000		ADCOMP35/ADCOMPB35[R/W] H,W 00000000 00000000		
001480_H	ADCOMP36/ADCOMPB36[R/W] H,W 00000000 00000000		ADCOMP37/ADCOMPB37[R/W] H,W 00000000 00000000		
001484_H	ADCOMP38/ADCOMPB38[R/W] H,W 00000000 00000000		ADCOMP39/ADCOMPB39[R/W] H,W 00000000 00000000		
001488_H	ADCOMP40/ADCOMPB40[R/W] H,W 00000000 00000000		ADCOMP41/ADCOMPB41[R/W] H,W 00000000 00000000		
$00148C_H$	ADCOMP42/ADCOMPB42[R/W] H,W 00000000 00000000		ADCOMP43/ADCOMPB43[R/W] H,W 00000000 00000000		
001490_H	ADCOMP44/ADCOMPB44[R/W] H,W 00000000 00000000		ADCOMP45/ADCOMPB45[R/W] H,W 00000000 00000000		
001494_H	ADCOMP46/ADCOMPB46[R/W] H,W 00000000 00000000		ADCOMP47/ADCOMPB47[R/W] H,W 00000000 00000000		
001498_H to $0014B4_H$	—	—	—	—	Reserved
$0014B8_H$	ADTCS32[R/W] B,H,W 00000000 0010----		ADTCS33[R/W] B,H,W 00000000 0010----		12-bit A/D converter 2/2 unit
$0014BC_H$	ADTCS34[R/W] B,H,W 00000000 0010----		ADTCS35[R/W] B,H,W 00000000 0010----		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0014C0_H$	ADTCS36[R/W] B,H,W 00000000 0010----		ADTCS37[R/W] B,H,W 00000000 0010----		12-bit A/D converter 2/2 unit
$0014C4_H$	ADTCS38[R/W] B,H,W 00000000 0010----		ADTCS39[R/W] B,H,W 00000000 0010----		
$0014C8_H$	ADTCS40[R/W] B,H,W 00000000 0010----		ADTCS41[R/W] B,H,W 00000000 0010----		
$0014CC_H$	ADTCS42[R/W] B,H,W 00000000 0010----		ADTCS43[R/W] B,H,W 00000000 0010----		
$0014D0_H$	ADTCS44[R/W] B,H,W 00000000 0010----		ADTCS45[R/W] B,H,W 00000000 0010----		
$0014D4_H$	ADTCS46[R/W] B,H,W 00000000 0010----		ADTCS47[R/W] B,H,W 00000000 0010----		
$0014D8_H$ to $0014F4_H$	—	—	—	—	Reserved
$0014F8_H$	ADTCD32[R] B,H,W 10--0000 00000000		ADTCD33[R] B,H,W 10--0000 00000000		12-bit A/D converter 2/2 unit
$0014FC_H$	ADTCD34[R] B,H,W 10--0000 00000000		ADTCD35[R] B,H,W 10--0000 00000000		
001500_H	ADTCD36[R] B,H,W 10--0000 00000000		ADTCD37[R] B,H,W 10--0000 00000000		
001504_H	ADTCD38[R] B,H,W 10--0000 00000000		ADTCD39[R] B,H,W 10--0000 00000000		
001508_H	ADTCD40[R] B,H,W 10--0000 00000000		ADTCD41[R] B,H,W 10--0000 00000000		
$00150C_H$	ADTCD42[R] B,H,W 10--0000 00000000		ADTCD43[R] B,H,W 10--0000 00000000		
001510_H	ADTCD44[R] B,H,W 10--0000 00000000		ADTCD45[R] B,H,W 10--0000 00000000		
001514_H	ADTCD46[R] B,H,W 10--0000 00000000		ADTCD47[R] B,H,W 10--0000 00000000		
001518_H to 001534_H	—	—	—	—	Reserved
001538_H	ADTECS32[R/W] B,H,W -------0 ----0000		ADTECS33[R/W] B,H,W -------0 ----0000		12-bit A/D converter 2/2 unit
$00153C_H$	ADTECS34[R/W] B,H,W -------0 ----0000		ADTECS35[R/W] B,H,W -------0 ----0000		
001540_H	ADTECS36[R/W] B,H,W -------0 ----0000		ADTECS37[R/W] B,H,W -------0 ----0000		
001544_H	ADTECS38[R/W] B,H,W -------0 ----0000		ADTECS39[R/W] B,H,W -------0 ----0000		
001548_H	ADTECS40[R/W] B,H,W -------0 ----0000		ADTECS41[R/W] B,H,W -------0 ----0000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$00154C_H$	ADTECS42[R/W] B,H,W -------0 ----0000		ADTECS43[R/W] B,H,W -------0 ----0000		12-bit A/D converter 2/2 unit
001550_H	ADTECS44[R/W] B,H,W -------0 ----0000		ADTECS45[R/W] B,H,W -------0 ----0000		
001554_H	ADTECS46[R/W] B,H,W -------0 ----0000		ADTECS47[R/W] B,H,W -------0 ----0000		
001558_H to 001574_H	—	—	—	—	Reserved
001578_H	ADRCUT4[R/W] B,H,W ----0000 00000000		ADRCLT4[R/W] B,H,W ----0000 00000000		12-bit A/D converter 2/2 unit
$00157C_H$	ADRCUT5[R/W] B,H,W ----0000 00000000		ADRCLT5[R/W] B,H,W ----0000 00000000		
001580_H	ADRCUT6[R/W] B,H,W ----0000 00000000		ADRCLT6[R/W] B,H,W ----0000 00000000		
001584_H	ADRCUT7[R/W] B,H,W ----0000 00000000		ADRCLT7[R/W] B,H,W ----0000 00000000		12-bit A/D converter 2/2 unit
001588_H	ADRCCS32[R/W] B,H,W 00000000	ADRCCS33[R/W] B,H,W 00000000	ADRCCS34[R/W] B,H,W 00000000	ADRCCS35[R/W] B,H,W 00000000	
$00158C_H$	ADRCCS36[R/W] B,H,W 00000000	ADRCCS37[R/W] B,H,W 00000000	ADRCCS38[R/W] B,H,W 00000000	ADRCCS39[R/W] B,H,W 00000000	
001590_H	ADRCCS40[R/W] B,H,W 00000000	ADRCCS41[R/W] B,H,W 00000000	ADRCCS42[R/W] B,H,W 00000000	ADRCCS43[R/W] B,H,W 00000000	
001594_H	ADRCCS44[R/W] B,H,W 00000000	ADRCCS45[R/W] B,H,W 00000000	ADRCCS46[R/W] B,H,W 00000000	ADRCCS47[R/W] B,H,W 00000000	
001598_H to $0015A4_H$	—	—	—	—	Reserved
$0015A8_H$	ADRCOT1 [R] B,H,W -------- -------- 00000000 00000000				12-bit A/D converter 2/2 unit
$0015AC_H$	ADRCIF1 [R,W] B,H,W -------- -------- 00000000 00000000				
$0015B0_H$	ADSCANS1 [R/W] B,H,W 000-----	—	—	—	
$0015B4_H$	ADNCS16 [R/W] B,H,W 0-000-00	ADNCS17 [R/W] B,H,W 0-000-00	ADNCS18 [R/W] B,H,W 0-000-00	ADNCS19 [R/W] B,H,W 0-000-00	
$0015B8_H$	ADNCS20 [R/W] B,H,W 0-000-00	ADNCS21 [R/W] B,H,W 0-000-00	ADNCS22 [R/W] B,H,W 0-000-00	ADNCS23 [R/W] B,H,W 0-000-00	
$0015BC_H$	—	—	—	—	
$0015C0_H$	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0015C4_H$	ADPRTF1 [R] B,H,W -------- -------- 00000000 00000000				12-bit A/D converter 2/2 unit
$0015C8_H$	ADEOCF1 [R] B,H,W -------- -------- 11111111 11111111				
$0015CC_H$	ADCS1 [R] B,H,W 0------- --------		ADCH1 [R] B,H,W ---00000	ADMD1 [R/W] B,H,W 0---0000	
$0015D0_H$	ADSTPCS8 [R/W] B,H,W 00000000	ADSTPCS9 [R/W] B,H,W 00000000	ADSTPCS10 [R/W] B,H,W 00000000	ADSTPCS11 [R/W] B,H,W 00000000	
$0015D4_H$ to $00174C_H$	—	—	—	—	Reserved
001750_H	SCR0/(IBCR0)[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000-00-0	SSR0[R/W] B,H,W 0-000011	ESCR0/(IBSR0)[R/W] B,H,W 00000000	Multi-UART0 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
001754_H	— /(RDR10/(TDR10))[R/W] B, H,W -------- -------- *3		RDR00/(TDR00)[R/W] B,H,W -------0 00000000 *1		
001758_H	SACSR0[R/W] B,H,W 0----000 00000000		STMR0[R] B,H,W 00000000 00000000		
$00175C_H$	STMCR0[R/W] B,H,W 00000000 00000000		— /(SCSCR0/SFUR0)[R/W] B,H,W -------- -------- *3 *4		
001760_H	— /(SCSTR30)/ (LAMSR0) [R/W] B,H,W -------- *3	— /(SCSTR20)/ (LAMCR0) [R/W] B,H,W -------- *3	— /(SCSTR10)/ (SFLR10) [R/W] B,H,W -------- *3	— /(SCSTR00)/ (SFLR00) [R/W] B,H,W -------- *3	
001764_H	—	— /(SCSFR20) [R/W] B,H,W -------- *3	— /(SCSFR10) [R/W] B,H,W -------- *3	— /(SCSFR00) [R/W] B,H,W -------- *3	
001768_H	—/(TBYTE30)/ (LAMESR0) [R/W] B,H,W -------- *3	—/(TBYTE20) / (LAMERT0) [R/W] B,H,W -------- *3	—/(TBYTE10)/ (LAMIER0) [R/W] B,H,W -------- *3	TBYTE00/ (LAMRID0)/ (LAMTID0) [R/W] B,H,W 00000000	
$00176C_H$	BGR0[R/W] H, W 00000000 00000000		— /(ISMK0) [R/W] B,H,W -------- *2	— /(ISBA0) [R/W] B,H,W -------- *2	
001770_H	FCR10[R/W] B,H,W ---00100	FCR00[R/W] B,H,W -0000000	FBYTE0[R/W] B,H,W 00000000 00000000		
001774_H	FTICR0[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001778_H	SCR1/(IBCR1) [R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000-00-0	SSR1[R/W] B,H,W 0-000011	ESCR1/(IBSR1)[R/W] B,H,W 00000000	Multi-UART1 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$00177C_H$	— /(RDR11/(TDR11))[R/W] B, H,W -------- -------- *3		RDR01/(TDR01)[R/W] B,H,W -------0 00000000 *1		
001780_H	SACSR1[R/W] B,H,W 0----000 00000000		STMR1[R] B,H,W 00000000 00000000		
001784_H	STMCR1[R/W] B,H,W 00000000 00000000		— /(SCSCR1/SFUR1)[R/W] B,H,W -------- -------- *3 *4		
001788_H	— /(SCSTR31)/ (LAMSR1) [R/W] B,H,W -------- *3	— /(SCSTR21)/ (LAMCR1) [R/W] B,H,W -------- *3	— /(SCSTR11)/ (SFLR11) [R/W] B,H,W -------- *3	— /(SCSTR01)/ (SFLR01) [R/W] B,H,W -------- *3	
$00178C_H$	—	— /(SCSFR21)[R/W] B,H,W -------- *3	— /(SCSFR11) [R/W] B,H,W -------- *3	— /(SCSFR01) [R/W] B,H,W -------- *3	
001790_H	—/(TBYTE31)/ (LAMESR1) [R/W] B,H,W -------- *3	—/(TBYTE21)/ (LAMERT1) [R/W] B,H,W -------- *3	—/(TBYTE11)/ (LAMIER1) [R/W] B,H,W -------- *3	TBYTE01/ (LAMRID1)/ (LAMTID1) [R/W] B,H,W 00000000	
001794_H	BGR1[R/W] H,W 00000000 00000000		— /(ISMK1)[R/W] B,H,W -------- *2	— /(ISBA1)[R/W] B,H,W -------- *2	
001798_H	FCR11[R/W] B,H,W ---00100	FCR01[R/W] B,H,W -0000000	FBYTE1[R/W] B,H,W 00000000 00000000		
$00179C_H$	FTICR1[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0017A0_H$	SCR2/(IBCR2)[R/W] B,H,W 0--00000	SMR2[R/W] B,H,W 000-00-0	SSR2[R/W] B,H,W 0-000011	ESCR2/(IBSR2)[R/W] B,H,W 00000000	Multi-UART2 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$0017A4_H$	— /(RDR12/(TDR12))[R/W] B, H,W -------- -------- *3		RDR02/(TDR02)[R/W] B,H,W -------0 00000000 *1		
$0017A8_H$	SACSR2[R/W] B,H,W 0----000 00000000		STMR2[R] B,H,W 00000000 00000000		
$0017AC_H$	STMCR2[R/W] B,H,W 00000000 00000000		— /(SCSCR2/SFUR2)[R/W] B,H,W -------- -------- *3 *4		
$0017B0_H$	— /(SCSTR32)/ (LAMSR2) [R/W] B,H,W -------- *3	— /(SCSTR22)/ (LAMCR2) [R/W] B,H,W -------- *3	— /(SCSTR12)/ (SFLR12) [R/W] B,H,W -------- *3	— /(SCSTR02)/ (SFLR02) [R/W] B,H,W -------- *3	
$0017B4_H$	—	— /(SCSFR22) [R/W] B,H,W -------- *3	— /(SCSFR12) [R/W] B,H,W -------- *3	— /(SCSFR02) [R/W] B,H,W -------- *3	
$0017B8_H$	—/(TBYTE32)/ (LAMESR2) [R/W] B,H,W -------- *3	—/(TBYTE22)/ (LAMERT2) [R/W] B,H,W -------- *3	—/(TBYTE12)/ (LAMIER2) [R/W] B,H,W -------- *3	TBYTE02/ (LAMRID2)/ (LAMTID2) [R/W] B,H,W 00000000	
$0017BC_H$	BGR2[R/W] H, W 00000000 00000000		— /(ISMK2)[R/W] B,H,W -------- *2	— /(ISBA2)[R/W] B,H,W -------- *2	
$0017C0_H$	FCR12[R/W] B,H,W ---00100	FCR02[R/W] B,H,W -0000000	FBYTE2[R/W] B,H,W 00000000 00000000		
$0017C4_H$	FTICR2[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0017C8_H$	SCR3/(IBCR3)[R/W] B,H,W 0--00000	SMR3[R/W] B,H,W 000-00-0	SSR3[R/W] B,H,W 0-000011	ESCR3/(IBSR3)[R/W] B,H,W 00000000	Multi-UART3 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$0017CC_H$	— /(RDR13/(TDR13))[R/W] B, H,W -------- -------- *3		RDR03/(TDR03)[R/W] B,H,W -------0 00000000 *1		
$0017D0_H$	SACSR3[R/W] B,H,W 0----000 00000000		STMR3[R] B,H,W 00000000 00000000		
$0017D4_H$	STMCR3[R/W] B,H,W 00000000 00000000		— /(SCSCR3/SFUR3)[R/W] B,H,W -------- -------- *3 *4		
$0017D8_H$	— /(SCSTR33)/ (LAMSR3) [R/W] B,H,W -------- *3	— /(SCSTR23)/ (LAMCR3) [R/W] B,H,W -------- *3	— /(SCSTR13)/ (SFLR13) [R/W] B,H,W -------- *3	— /(SCSTR03)/ (SFLR03) [R/W] B,H,W -------- *3	
$0017DC_H$	—	— /(SCSFR23) [R/W] B,H,W -------- *3	— /(SCSFR13) [R/W] B,H,W -------- *3	— /(SCSFR03) [R/W] B,H,W -------- *3	
$0017E0_H$	—/(TBYTE33)/ (LAMESR3) [R/W] B,H,W -------- *3	—/(TBYTE23)/ (LAMERT3) [R/W] B,H,W -------- *3	—/(TBYTE13)/ (LAMIER3) [R/W] B,H,W -------- *3	TBYTE03/ (LAMRID3)/ (LAMTID3) [R/W] B,H,W 00000000	
$0017E4_H$	BGR3[R/W] H, W 00000000 00000000		— /(ISMK3)[R/W] B,H,W -------- *2	— /(ISBA3)[R/W] B,H,W -------- *2	
$0017E8_H$	FCR13[R/W] B,H,W ---00100	FCR03[R/W] B,H,W -0000000	FBYTE3[R/W] B,H,W 00000000 00000000		
$0017EC_H$	FTICR3[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0017F0_H$	SCR4/(IBCR4) [R/W] B,H,W 0--00000	SMR4[R/W] B,H,W 000-00-0	SSR4[R/W] B,H,W 0-000011	ESCR4/(IBSR4)[R/W] B,H,W 00000000	Multi-UART4 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$0017F4_H$	— /(RDR14/(TDR14))[R/W] B, H,W -------- -------- *3		RDR04/(TDR04)[R/W] B,H,W -------0 00000000 *1		
$0017F8_H$	SACSR4[R/W] B,H,W 0----000 00000000		STMR4[R] B,H,W 00000000 00000000		
$0017FC_H$	STMCR4[R/W] B,H,W 00000000 00000000		— /(SCSCR4/SFUR4)[R/W] B,H,W -------- -------- *3 *4		
001800_H	— /(SCSTR34)/ (LAMSR4) [R/W] B,H,W -------- *3	— /(SCSTR24)/ (LAMCR4) [R/W] B,H,W -------- *3	— /(SCSTR14)/ (SFLR14) [R/W] B,H,W -------- *3	— /(SCSTR04)/ (SFLR04) [R/W] B,H,W -------- *3	
001804_H	—	— /(SCSFR24) [R/W] B,H,W -------- *3	— /(SCSFR14) [R/W] B,H,W -------- *3	— /(SCSFR04) [R/W] B,H,W -------- *3	
001808_H	—/(TBYTE34)/ (LAMESR4) [R/W] B,H,W -------- *3	—/(TBYTE24)/ (LAMERT4) [R/W] B,H,W -------- *3	—/(TBYTE14)/ (LAMIER4) [R/W] B,H,W -------- *3	TBYTE04/ (LAMRID4)/ (LAMTID4) [R/W] B,H,W 00000000	
$00180C_H$	BGR4[R/W] H, W 00000000 00000000		— /(ISMK4)[R/W] B,H,W -------- *2	— /(ISBA4)[R/W] B,H,W -------- *2	
001810_H	FCR14[R/W] B,H,W ---00100	FCR04[R/W] B,H,W -0000000	FBYTE4[R/W] B,H,W 00000000 00000000		
001814_H	FTICR4[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001818_H	SCR5/(IBCR5) [R/W] B,H,W 0--00000	SMR5[R/W] B,H,W 000-00-0	SSR5[R/W] B,H,W 0-000011	ESCR5/(IBSR5)[R/W] B,H,W 00000000	Multi-UART5 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$00181C_H$	— /(RDR15/(TDR15))[R/W] B, H,W -------- -------- *3		RDR05/(TDR05)[R/W] B,H,W -------0 00000000 *1		
001820_H	SACSR5[R/W] B,H,W 0----000 00000000		STMR5[R] B,H,W 00000000 00000000		
001824_H	STMCR5[R/W] B,H,W 00000000 00000000		— /(SCSCR5/SFUR5)[R/W] B,H,W -------- -------- *3 *4		
001828_H	— /(SCSTR35)/ (LAMSR5) [R/W] B,H,W -------- *3	— /(SCSTR25)/ (LAMCR5) [R/W] B,H,W -------- *3	— /(SCSTR15)/ (SFLR15) [R/W] B,H,W -------- *3	— /(SCSTR05)/ (SFLR05) [R/W] B,H,W -------- *3	
$00182C_H$	—	— /(SCSFR25) [R/W] B,H,W -------- *3	— /(SCSFR15) [R/W] B,H,W -------- *3	— /(SCSFR05) [R/W] B,H,W -------- *3	
001830_H	—/(TBYTE35)/ (LAMESR5) [R/W] B,H,W -------- *3	—/(TBYTE25)/ (LAMERT5) [R/W] B,H,W -------- *3	—/(TBYTE15)/ (LAMIER5) [R/W] B,H,W -------- *3	TBYTE05/ (LAMRID5)/ (LAMTID5) [R/W] B,H,W 00000000	
001834_H	BGR5[R/W] H, W 00000000 00000000		— /(ISMK5)[R/W] B,H,W -------- *2	— /(ISBA5)[R/W] B,H,W -------- *2	
001838_H	FCR15[R/W] B,H,W ---00100	FCR05[R/W] B,H,W -0000000	FBYTE5[R/W] B,H,W 00000000 00000000		
$00183C_H$	FTICR5[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001840_H	SCR6/(IBCR6) [R/W] B,H,W 0--00000	SMR6[R/W] B,H,W 000-00-0	SSR6[R/W] B,H,W 0-000011	ESCR6/(IBSR6)[R/W] B,H,W 00000000	Multi-UART6 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
001844_H	— /(RDR16/(TDR16))[R/W] B, H,W -------- -------- *3		RDR06/(TDR06)[R/W] B,H,W -------0 00000000 *1		
001848_H	SACSR6[R/W] B,H,W 0----000 00000000		STMR6[R] B,H,W 00000000 00000000		
$00184C_H$	STMCR6[R/W] B,H,W 00000000 00000000		— /(SCSCR6/SFUR6)[R/W] B,H,W -------- -------- *3 *4		
001850_H	— /(SCSTR36)/ (LAMSR6) [R/W] B,H,W -------- *3	— /(SCSTR26)/ (LAMCR6) [R/W] B,H,W -------- *3	— /(SCSTR16)/ (SFLR16) [R/W] B,H,W -------- *3	— /(SCSTR06)/ (SFLR06) [R/W] B,H,W -------- *3	
001854_H	—	— /(SCSFR26) [R/W] B,H,W -------- *3	— /(SCSFR16) [R/W] B,H,W -------- *3	— /(SCSFR06) [R/W] B,H,W -------- *3	
001858_H	—/(TBYTE36)/ (LAMESR6) [R/W] B,H,W -------- *3	—/(TBYTE26)/ (LAMERT6) [R/W] B,H,W -------- *3	—/(TBYTE16)/ (LAMIER6) [R/W] B,H,W -------- *3	TBYTE06/ (LAMRID6)/ (LAMTID6) [R/W] B,H,W 00000000	
$00185C_H$	BGR6[R/W] H, W 00000000 00000000		— /(ISMK6)[R/W] B,H,W -------- *2	— /(ISBA6)[R/W] B,H,W -------- *2	
001860_H	FCR16[R/W] B,H,W ---00100	FCR06[R/W] B,H,W -0000000	FBYTE6[R/W] B,H,W 00000000 00000000		
001864_H	FTICR6[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001868_H	SCR7/(IBCR7) [R/W] B,H,W 0--00000	SMR7[R/W] B,H,W 000-00-0	SSR7[R/W] B,H,W 0-000011	ESCR7/(IBSR7)[R/W] B,H,W 00000000	Multi-UART7 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$00186C_H$	— /(RDR17/(TDR17))[R/W] B, H,W -------- -------- *3		RDR07/(TDR07)[R/W] B,H,W -------0 00000000 *1		
001870_H	SACSR7[R/W] B,H,W 0----000 00000000		STMR7[R] B,H,W 00000000 00000000		
001874_H	STMCR7[R/W] B,H,W 00000000 00000000		— /(SCSCR7/SFUR7)[R/W] B,H,W -------- -------- *3 *4		
001878_H	— /(SCSTR37)/ (LAMSR7) [R/W] B,H,W -------- *3	— /(SCSTR27)/ (LAMCR7) [R/W] B,H,W -------- *3	— /(SCSTR17)/ (SFLR17) [R/W] B,H,W -------- *3	— /(SCSTR07)/ (SFLR07) [R/W] B,H,W -------- *3	
$00187C_H$	—	— /(SCSFR27) [R/W] B,H,W -------- *3	— /(SCSFR17) [R/W] B,H,W -------- *3	— /(SCSFR07) [R/W] B,H,W -------- *3	
001880_H	—/(TBYTE37)/ (LAMESR7) [R/W] B,H,W -------- *3	—/(TBYTE27)/ (LAMERT7) [R/W] B,H,W -------- *3	—/(TBYTE17)/ (LAMIER7) [R/W] B,H,W -------- *3	TBYTE07/ (LAMRID7)/ (LAMTID7) [R/W] B,H,W 00000000	
001884_H	BGR7[R/W] H, W 00000000 00000000		— /(ISMK7)[R/W] B,H,W -------- *2	— /(ISBA7)[R/W] B,H,W -------- *2	
001888_H	FCR17[R/W] B,H,W ---00100	FCR07[R/W] B,H,W -0000000	FBYTE7[R/W] B,H,W 00000000 00000000		
$00188C_H$	FTICR7[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001890_H	SCR8/(IBCR8) [R/W] B,H,W 0--00000	SMR8[R/W] B,H,W 000-00-0	SSR8[R/W] B,H,W 0-000011	ESCR8/(IBSR8)[R/W] B,H,W 00000000	Multi-UART8 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
001894_H	— /(RDR18/(TDR18))[R/W] B, H,W -------- -------- *3		RDR08/(TDR08)[R/W] B,H,W -------0 00000000 *1		
001898_H	SACSR8[R/W] B,H,W 0----000 00000000		STMR8[R] B,H,W 00000000 00000000		
$00189C_H$	STMCR8[R/W] B,H,W 00000000 00000000		— /(SCSCR8/SFUR8)[R/W] B,H,W -------- -------- *3 *4		
$0018A0_H$	— /(SCSTR38)/ (LAMSR8) [R/W] B,H,W -------- *3	— /(SCSTR28)/ (LAMCR8) [R/W] B,H,W -------- *3	— /(SCSTR18)/ (SFLR18) [R/W] B,H,W -------- *3	— /(SCSTR08)/ (SFLR08) [R/W] B,H,W -------- *3	
$0018A4_H$	—	— /(SCSFR28) [R/W] B,H,W -------- *3	— /(SCSFR18) [R/W] B,H,W -------- *3	— /(SCSFR08) [R/W] B,H,W -------- *3	
$0018A8_H$	—/(TBYTE38)/ (LAMESR8) [R/W] B,H,W -------- *3	—/(TBYTE28)/ (LAMERT8) [R/W] B,H,W -------- *3	—/(TBYTE18)/ (LAMIER8) [R/W] B,H,W -------- *3	TBYTE08/ (LAMRID8)/ (LAMTID8) [R/W] B,H,W 00000000	
$0018AC_H$	BGR8[R/W] H,W 00000000 00000000		— /(ISMK8)[R/W] B,H,W -------- *2	— /(ISBA8)[R/W] B,H,W -------- *2	
$0018B0_H$	FCR18[R/W] B,H,W ---00100	FCR08[R/W] B,H,W -0000000	FBYTE8[R/W] B,H,W 00000000 00000000		
$0018B4_H$	FTICR8[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0018B8_H$	SCR9/(IBCR9) [R/W] B,H,W 0--00000	SMR9[R/W] B,H,W 000-00-0	SSR9[R/W] B,H,W 0-000011	ESCR9/(IBSR9)[R/W] B,H,W 00000000	Multi-UART9 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$0018BC_H$	— /(RDR19/(TDR19))[R/W] B, H,W -------- -------- *3		RDR09/(TDR09)[R/W] B,H,W -------0 00000000 *1		
$0018C0_H$	SACSR9[R/W] B,H,W 0----000 00000000		STMR9[R] B,H,W 00000000 00000000		
$0018C4_H$	STMCR9[R/W] B,H,W 00000000 00000000		— /(SCSCR9/SFUR9)[R/W] B,H,W -------- -------- *3 *4		
$0018C8_H$	— /(SCSTR39)/ (LAMSR9) [R/W] B,H,W -------- *3	— /(SCSTR29)/ (LAMCR9) [R/W] B,H,W -------- *3	— /(SCSTR19)/ (SFLR19) [R/W] B,H,W -------- *3	— /(SCSTR09)/ (SFLR09) [R/W] B,H,W -------- *3	
$0018CC_H$	—	— /(SCSFR29) [R/W] B,H,W -------- *3	— /(SCSFR19) [R/W] B,H,W -------- *3	— /(SCSFR09) [R/W] B,H,W -------- *3	
$0018D0_H$	—/(TBYTE39)/ (LAMESR9) [R/W] B,H,W -------- *3	—/(TBYTE29)/ (LAMERT9) [R/W] B,H,W -------- *3	—/(TBYTE19)/ (LAMIER9) [R/W] B,H,W -------- *3	TBYTE09/ (LAMRID9)/ (LAMTID9) [R/W] B,H,W 00000000	
$0018D4_H$	BGR9[R/W] H, W 00000000 00000000		— /(ISMK9)[R/W] B,H,W -------- *2	— /(ISBA9)[R/W] B,H,W -------- *2	
$0018D8_H$	FCR19[R/W] B,H,W ---00100	FCR09[R/W] B,H,W -0000000	FBYTE9[R/W] B,H,W 00000000 00000000		
$0018DC_H$	FTICR9[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0018E0_H$	SCR10/(IBCR10) [R/W] B,H,W 0--00000	SMR10[R/W] B,H,W 000-00-0	SSR10[R/W] B,H,W 0-000011	ESCR10/(IBSR10) [R/W] B,H,W 00000000	Multi-UART10 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$0018E4_H$	— /(RDR110/(TDR110))[R/W] B, H,W -------- -------- *3		RDR010/(TDR010)[R/W] B,H,W -------0 00000000 *1		
$0018E8_H$	SACSR10[R/W] B,H,W 0----000 00000000		STMR10[R] B,H,W 00000000 00000000		
$0018EC_H$	STMCR10[R/W] B,H,W 00000000 00000000		— /(SCSCR10/SFUR10)[R/W] B,H,W -------- -------- *3 *4		
$0018F0_H$	— /(SCSTR310)/ (LAMSR10) [R/W] B,H,W -------- *3	— /(SCSTR210)/ (LAMCR10) [R/W] B,H,W -------- *3	— /(SCSTR110)/ (SFLR110)[R/W] B,H,W -------- *3	— /(SCSTR010)/ (SFLR010)[R/W] B,H,W -------- *3	
$0018F4_H$	—	— /(SCSFR210) [R/W] B,H,W -------- *3	— /(SCSFR110) [R/W] B,H,W -------- *3	— /(SCSFR010) [R/W] B,H,W -------- *3	
$0018F8_H$	—/(TBYTE310)/ (LAMESR10) [R/W] B,H,W -------- *3	—/(TBYTE210)/ (LAMERT10) [R/W] B,H,W -------- *3	—/(TBYTE110)/ (LAMIER10) [R/W] B,H,W -------- *3	TBYTE010/(LAMRID 10)/(LAMTID10) [R/W] B,H,W 00000000	
$0018FC_H$	BGR10[R/W] H, W 00000000 00000000		— /(ISMK10)[R/W] B,H,W -------- *2	— /(ISBA10)[R/W] B,H,W -------- *2	
001900_H	FCR110[R/W] B,H,W ---00100	FCR010[R/W] B,H,W -0000000	FBYTE10[R/W] B,H,W 00000000 00000000		
001904_H	FTICR10[R/W] B,H,W 00000000 00000000		—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001908_H	SCR11/(IBCR11) [R/W] B,H,W 0--00000	SMR11[R/W] B, H,W 000-00-0	SSR11[R/W] B,H,W 0-000011	ESCR11/(IBSR11) [R/W] B,H,W 00000000	Multi-UART11 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because I^2C mode is not set immediately after reset. *3: Reserved because CSIO mode is not set immediately after reset. *4: Reserved because LIN2.1 mode is not set immediately after reset.
$00190C_H$	— /(RDR111/(TDR111))[R/W] B, H,W -------- -------- *3		RDR011/(TDR011)[R/W] B,H,W -------0 00000000 *1		
001910_H	SACSR11[R/W] B,H,W 0----000 00000000		STMR11[R] B,H,W 00000000 00000000		
001914_H	STMCR11[R/W] B,H,W 00000000 00000000		— /(SCSCR11/SFUR11)[R/W] B,H,W -------- -------- *3 *4		
001918_H	— /(SCSTR311)/ (LAMSR11) [R/W] B,H,W -------- *3	— /(SCSTR211)/ (LAMCR11) [R/W] B,H,W -------- *3	— /(SCSTR111)/ (SFLR111)[R/W] B,H,W -------- *3	— /(SCSTR011)/ (SFLR011)[R/W] B,H,W -------- *3	
$00191C_H$	—	— /(SCSFR211) [R/W] B,H,W -------- *3	— /(SCSFR111) [R/W] B,H,W -------- *3	— /(SCSFR011) [R/W] B,H,W -------- *3	
001920_H	—/(TBYTE311)/ (LAMESR11) [R/W] B,H,W -------- *3	—/(TBYTE211)/ (LAMERT11) [R/W] B,H,W -------- *3	—/(TBYTE111)/ (LAMIER11) [R/W] B,H,W -------- *3	TBYTE011/(LAMRID 11)/(LAMTID11) [R/W] B,H,W 00000000	
001924_H	BGR11[R/W] H, W 00000000 00000000		— /(ISMK11)[R/W] B,H,W -------- *2	— /(ISBA11)[R/W] B,H,W -------- *2	
001928_H	FCR111[R/W] B,H,W ---00100	FCR011[R/W] B,H,W -0000000	FBYTE11[R/W] B,H,W 00000000 00000000		
$00192C_H$	FTICR11[R/W] B,H,W 00000000 00000000		—	—	
001930_H to $0019D8_H$	—	—	—	—	Reserved
$0019DC_H$	—	GATEC0 [R/W] B,H,W ------00	—	GATEC2 [R/W] B,H,W ------00	PPG GATE control
$0019E0_H$	—	GATEC4 [R/W] B,H,W ------00	—	—	
$0019E4_H$	—	—	—	—	Reserved
$0019E8_H$	GTRS0 [R/W] B,H,W -0000000 -0000000		GTRS1 [R/W] B,H,W -0000000 -0000000		PPG controller
$0019EC_H$	GTRS2 [R/W] B,H,W -0000000 -0000000		GTRS3 [R/W] B,H,W -0000000 -0000000		
$0019F0_H$	GTRS4 [R/W] B,H,W -0000000 -0000000		GTRS5 [R/W] B,H,W -0000000 -0000000		
$0019F4_H$	GTRS6 [R/W] B,H,W -0000000 -0000000		GTRS7 [R/W] B,H,W -0000000 -0000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0019F8_H$	GTRS8 [R/W] B,H,W -0000000 -0000000		GTRS9 [R/W] B,H,W -0000000 -0000000		PPG controller
$0019FC_H$	GTRS10 [R/W] B,H,W -0000000 -0000000		GTRS11 [R/W] B,H,W -0000000 -0000000		
$001A00_H$	GTRS12 [R/W] B,H,W -0000000 -0000000		GTRS13 [R/W] B,H,W -0000000 -0000000		
$001A04_H$	GTRS14 [R/W] B,H,W -0000000 -0000000		GTRS15 [R/W] B,H,W -0000000 -0000000		
$001A08_H$	GTRS16 [R/W] B,H,W -0000000 -0000000		GTRS17 [R/W] B,H,W -0000000 -0000000		
$001A0C_H$	GTRS18 [R/W] B,H,W -0000000 -0000000		GTRS19 [R/W] B,H,W -0000000 -0000000		
$001A10_H$	GTRS20 [R/W] B,H,W -0000000 -0000000		GTRS21 [R/W] B,H,W -0000000 -0000000		
$001A14_H$	GTRS22 [R/W] B,H,W -0000000 -0000000		GTRS23 [R/W] B,H,W -0000000 -0000000		
$001A18_H$ to $001A2C_H$	—	—	—	—	Reserved
$001A30_H$	—	—	—	—	Reserved
$001A34_H$	—	—	—	—	
$001A38_H$	GTREN0 [R/W] H,W 00000000 00000000		GTREN1 [R/W] H,W 00000000 00000000		PPG controller
$001A3C_H$	GTREN2 [R/W] H,W 00000000 00000000		—	—	
$001A40_H$	PCN0 [R/W] B,H,W 00000000 000000-0		PCSR0 [W] H,W XXXXXXXX XXXXXXXX		PPG0 * for communication
$001A44_H$	PDUT0 [W] H,W XXXXXXXX XXXXXXXX		PTMR0 [R] H,W 11111111 11111111		
$001A48_H$	PCN200 [R/W] B,H,W --000000 -----110		PSDR0 [R/W] H,W 00000000 00000000		
$001A4C_H$	PTPC0 [R/W] H,W 00000000 00000000		PCMDWD0 [R/W] B,H,W -------- ----0000		
$001A50_H$	PHCSR0 [W] H,W XXXXXXXX XXXXXXXX		PLCSR0 [W] H,W XXXXXXXX XXXXXXXX		
$001A54_H$	PHDUT0 [W] H,W XXXXXXXX XXXXXXXX		PLDUT0 [W] H,W XXXXXXXX XXXXXXXX		
$001A58_H$	PCMDDT0 [R/W] H,W 00000000 00000000		—	—	
$001A5C_H$	PCN1 [R/W] B,H,W 00000000 000000-0		PCSR1 [W] H,W XXXXXXXX XXXXXXXX		PPG1 * for communication
$001A60_H$	PDUT1 [W] H,W XXXXXXXX XXXXXXXX		PTMR1 [R] H,W 11111111 11111111		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001A64_H$	PCN201 [R/W] B,H,W --000000 -----110		PSDR1 [R/W] H,W 00000000 00000000		PPG1 * for communication
$001A68_H$	PTPC1 [R/W] H,W 00000000 00000000		PCMDWD1 [R/W] B,H,W -------- ----0000		
$001A6C_H$	PHCSR1 [W] H,W XXXXXXXX XXXXXXXX		PLCSR1 [W] H,W XXXXXXXX XXXXXXXX		
$001A70_H$	PHDUT1 [W] H,W XXXXXXXX XXXXXXXX		PLDUT1 [W] H,W XXXXXXXX XXXXXXXX		
$001A74_H$	PCMDDT1 [R/W] H,W 00000000 00000000		—	—	
$001A78_H$	PCN2 [R/W] B,H,W 00000000 000000-0		PCSR2 [W] H,W XXXXXXXX XXXXXXXX		PPG2 * for communication
$001A7C_H$	PDUT2 [W] H,W XXXXXXXX XXXXXXXX		PTMR2 [R] H,W 11111111 11111111		
$001A80_H$	PCN202 [R/W] B,H,W --000000 -----110		PSDR2 [R/W] H,W 00000000 00000000		
$001A84_H$	PTPC2 [R/W] H,W 00000000 00000000		PCMDWD2 [R/W] B,H,W -------- ----0000		
$001A88_H$	PHCSR2 [W] H,W XXXXXXXX XXXXXXXX		PLCSR2 [W] H,W XXXXXXXX XXXXXXXX		
$001A8C_H$	PHDUT2 [W] H,W XXXXXXXX XXXXXXXX		PLDUT2 [W] H,W XXXXXXXX XXXXXXXX		
$001A90_H$	PCMDDT2 [R/W] H,W 00000000 00000000		—	—	
$001A94_H$	PCN3 [R/W] B,H,W 00000000 000000-0		PCSR3 [W] H,W XXXXXXXX XXXXXXXX		PPG3 * for communication
$001A98_H$	PDUT3 [W] H,W XXXXXXXX XXXXXXXX		PTMR3 [R] H,W 11111111 11111111		
$001A9C_H$	PCN203 [R/W] B,H,W --000000 -----110		PSDR3 [R/W] H,W 00000000 00000000		
$001AA0_H$	PTPC3 [R/W] H,W 00000000 00000000		PCMDWD3 [R/W] B,H,W -------- ----0000		
$001AA4_H$	PHCSR3 [W] H,W XXXXXXXX XXXXXXXX		PLCSR3 [W] H,W XXXXXXXX XXXXXXXX		
$001AA8_H$	PHDUT3 [W] H,W XXXXXXXX XXXXXXXX		PLDUT3 [W] H,W XXXXXXXX XXXXXXXX		
$001AAC_H$	PCMDDT3 [R/W] H,W 00000000 00000000		—	—	
$001AB0_H$	PCN4 [R/W] B,H,W 00000000 000000-0		PCSR4 [W] H,W XXXXXXXX XXXXXXXX		PPG4
$001AB4_H$	PDUT4 [W] H,W XXXXXXXX XXXXXXXX		PTMR4 [R] H,W 11111111 11111111		
$001AB8_H$	PCN204 [R/W] B,H,W --000000 -----110		PSDR4 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001ABC_H$	PTPC4 [R/W] H,W 00000000 00000000		—	—	PPG4
$001AC0_H$	PCN5 [R/W] B,H,W 00000000 000000-0		PCSR5 [W] H,W XXXXXXXX XXXXXXXX		PPG5
$001AC4_H$	PDUT5 [W] H,W XXXXXXXX XXXXXXXX		PTMR5 [R] H,W 11111111 11111111		
$001AC8_H$	PCN205 [R/W] B,H,W --000000 -----110		PSDR5 [R/W] H,W 00000000 00000000		
$001ACC_H$	PTPC5 [R/W] H,W 00000000 00000000		—	—	
$001AD0_H$	PCN6 [R/W] B,H,W 00000000 000000-0		PCSR6 [W] H,W XXXXXXXX XXXXXXXX		PPG6
$001AD4_H$	PDUT6 [W] H,W XXXXXXXX XXXXXXXX		PTMR6 [R] H,W 11111111 11111111		
$001AD8_H$	PCN206 [R/W] B,H,W --000000 -----110		PSDR6 [R/W] H,W 00000000 00000000		
$001ADC_H$	PTPC6 [R/W] H,W 00000000 00000000		—	—	
$001AE0_H$	PCN7 [R/W] B,H,W 00000000 000000-0		PCSR7 [W] H,W XXXXXXXX XXXXXXXX		PPG7
$001AE4_H$	PDUT7 [W] H,W XXXXXXXX XXXXXXXX		PTMR7 [R] H,W 11111111 11111111		
$001AE8_H$	PCN207 [R/W] B,H,W --000000 -----110		PSDR7 [R/W] H,W 00000000 00000000		
$001AEC_H$	PTPC7 [R/W] H,W 00000000 00000000		—	—	
$001AF0_H$	PCN8 [R/W] B,H,W 00000000 000000-0		PCSR8 [W] H,W XXXXXXXX XXXXXXXX		PPG8
$001AF4_H$	PDUT8 [W] H,W XXXXXXXX XXXXXXXX		PTMR8 [R] H,W 11111111 11111111		
$001AF8_H$	PCN208 [R/W] B,H,W --000000 -----110		PSDR8 [R/W] H,W 00000000 00000000		
$001AFC_H$	PTPC8 [R/W] H,W 00000000 00000000		—	—	
$001B00_H$	PCN9 [R/W] B,H,W 00000000 000000-0		PCSR9 [W] H,W XXXXXXXX XXXXXXXX		PPG9
$001B04_H$	PDUT9 [W] H,W XXXXXXXX XXXXXXXX		PTMR9 [R] H,W 11111111 11111111		
$001B08_H$	PCN209 [R/W] B,H,W --000000 -----110		PSDR9 [R/W] H,W 00000000 00000000		
$001B0C_H$	PTPC9 [R/W] H,W 00000000 00000000		—	—	
$001B10_H$	PCN10 [R/W] B,H,W 00000000 000000-0		PCSR10 [W] H,W XXXXXXXX XXXXXXXX		PPG10

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001B14_H$	PDUT10 [W] H,W XXXXXXXX XXXXXXXX		PTMR10 [R] H,W 11111111 11111111		PPG10
$001B18_H$	PCN210 [R/W] B,H,W --000000 -----110		PSDR10 [R/W] H,W 00000000 00000000		
$001B1C_H$	PTPC10 [R/W] H,W 00000000 00000000		—	—	
$001B20_H$	PCN11 [R/W] B,H,W 00000000 000000-0		PCSR11 [W] H,W XXXXXXXX XXXXXXXX		PPG11
$001B24_H$	PDUT11 [W] H,W XXXXXXXX XXXXXXXX		PTMR11 [R] H,W 11111111 11111111		
$001B28_H$	PCN211 [R/W] B,H,W --000000 -----110		PSDR11 [R/W] H,W 00000000 00000000		
$001B2C_H$	PTPC11 [R/W] H,W 00000000 00000000		—	—	
$001B30_H$	PCN12 [R/W] B,H,W 00000000 000000-0		PCSR12 [W] H,W XXXXXXXX XXXXXXXX		PPG12
$001B34_H$	PDUT12 [W] H,W XXXXXXXX XXXXXXXX		PTMR12 [R] H,W 11111111 11111111		
$001B38_H$	PCN212 [R/W] B,H,W --000000 -----110		PSDR12 [R/W] H,W 00000000 00000000		
$001B3C_H$	PTPC12 [R/W] H,W 00000000 00000000		—	—	
$001B40_H$	PCN13 [R/W] B,H,W 00000000 000000-0		PCSR13 [W] H,W XXXXXXXX XXXXXXXX		PPG13
$001B44_H$	PDUT13 [W] H,W XXXXXXXX XXXXXXXX		PTMR13 [R] H,W 11111111 11111111		
$001B48_H$	PCN213 [R/W] B,H,W --000000 -----110		PSDR13 [R/W] H,W 00000000 00000000		
$001B4C_H$	PTPC13 [R/W] H,W 00000000 00000000		—	—	
$001B50_H$	PCN14 [R/W] B,H,W 00000000 000000-0		PCSR14 [W] H,W XXXXXXXX XXXXXXXX		PPG14
$001B54_H$	PDUT14 [W] H,W XXXXXXXX XXXXXXXX		PTMR14 [R] H,W 11111111 11111111		
$001B58_H$	PCN214 [R/W] B,H,W --000000 -----110		PSDR14 [R/W] H,W 00000000 00000000		
$001B5C_H$	PTPC14 [R/W] H,W 00000000 00000000		—	—	
$001B60_H$	PCN15 [R/W] B,H,W 00000000 000000-0		PCSR15 [W] H,W XXXXXXXX XXXXXXXX		PPG15
$001B64_H$	PDUT15 [W] H,W XXXXXXXX XXXXXXXX		PTMR15 [R] H,W 11111111 11111111		
$001B68_H$	PCN215 [R/W] B,H,W --000000 -----110		PSDR15 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001B6C$_H$	PTPC15 [R/W] H,W 00000000 00000000		—	—	PPG15
001B70$_H$	PCN16 [R/W] B,H,W 00000000 000000-0		PCSR16 [W] H,W XXXXXXXX XXXXXXXX		PPG16
001B74$_H$	PDUT16 [W] H,W XXXXXXXX XXXXXXXX		PTMR16 [R] H,W 11111111 11111111		
001B78$_H$	PCN216 [R/W] B,H,W --000000 -----110		PSDR16 [R/W] H,W 00000000 00000000		
001B7C$_H$	PTPC16 [R/W] H,W 00000000 00000000		—	—	
001B80$_H$	PCN17 [R/W] B,H,W 00000000 000000-0		PCSR17 [W] H,W XXXXXXXX XXXXXXXX		PPG17
001B84$_H$	PDUT17 [W] H,W XXXXXXXX XXXXXXXX		PTMR17 [R] H,W 11111111 11111111		
001B88$_H$	PCN217 [R/W] B,H,W --000000 -----110		PSDR17 [R/W] H,W 00000000 00000000		
001B8C$_H$	PTPC17 [R/W] H,W 00000000 00000000		—	—	
001B90$_H$	PCN18 [R/W] B,H,W 00000000 000000-0		PCSR18 [W] H,W XXXXXXXX XXXXXXXX		PPG18
001B94$_H$	PDUT18 [W] H,W XXXXXXXX XXXXXXXX		PTMR18 [R] H,W 11111111 11111111		
001B98$_H$	PCN218 [R/W] B,H,W --000000 -----110		PSDR18 [R/W] H,W 00000000 00000000		
001B9C$_H$	PTPC18 [R/W] H,W 00000000 00000000		—	—	
001BA0$_H$	PCN19 [R/W] B,H,W 00000000 000000-0		PCSR19 [W] H,W XXXXXXXX XXXXXXXX		PPG19
001BA4$_H$	PDUT19 [W] H,W XXXXXXXX XXXXXXXX		PTMR19 [R] H,W 11111111 11111111		
001BA8$_H$	PCN219 [R/W] B,H,W --000000 -----110		PSDR19 [R/W] H,W 00000000 00000000		
001BAC$_H$	PTPC19 [R/W] H,W 00000000 00000000		—	—	
001BB0$_H$	PCN20 [R/W] B,H,W 00000000 000000-0		PCSR20 [W] H,W XXXXXXXX XXXXXXXX		PPG20
001BB4$_H$	PDUT20 [W] H,W XXXXXXXX XXXXXXXX		PTMR20 [R] H,W 11111111 11111111		
001BB8$_H$	PCN220 [R/W] B,H,W --000000 -----110		PSDR20 [R/W] H,W 00000000 00000000		
001BBC$_H$	PTPC20 [R/W] H,W 00000000 00000000		—	—	
001BC0$_H$	PCN21 [R/W] B,H,W 00000000 000000-0		PCSR21 [W] H,W XXXXXXXX XXXXXXXX		PPG21

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001BC4_H$	PDUT21 [W] H,W XXXXXXXX XXXXXXXX		PTMR21 [R] H,W 11111111 11111111		PPG21
$001BC8_H$	PCN221 [R/W] B,H,W --000000 -----110		PSDR21 [R/W] H,W 00000000 00000000		
$001BCC_H$	PTPC21 [R/W] H,W 00000000 00000000		—	—	
$001BD0_H$	PCN22 [R/W] B,H,W 00000000 000000-0		PCSR22 [W] H,W XXXXXXXX XXXXXXXX		PPG22
$001BD4_H$	PDUT22 [W] H,W XXXXXXXX XXXXXXXX		PTMR22 [R] H,W 11111111 11111111		
$001BD8_H$	PCN222 [R/W] B,H,W --000000 -----110		PSDR22 [R/W] H,W 00000000 00000000		
$001BDC_H$	PTPC22 [R/W] H,W 00000000 00000000		—	—	
$001BE0_H$	PCN23 [R/W] B,H,W 00000000 000000-0		PCSR23 [W] H,W XXXXXXXX XXXXXXXX		PPG23
$001BE4_H$	PDUT23 [W] H,W XXXXXXXX XXXXXXXX		PTMR23 [R] H,W 11111111 11111111		
$001BE8_H$	PCN223 [R/W] B,H,W --000000 -----110		PSDR23 [R/W] H,W 00000000 00000000		
$001BEC_H$	PTPC23 [R/W] H,W 00000000 00000000		—	—	
$001BF0_H$	PCN24 [R/W] B,H,W 00000000 000000-0		PCSR24 [W] H,W XXXXXXXX XXXXXXXX		PPG24
$001BF4_H$	PDUT24 [W] H,W XXXXXXXX XXXXXXXX		PTMR24 [R] H,W 11111111 11111111		
$001BF8_H$	PCN224 [R/W] B,H,W --000000 -----110		PSDR24 [R/W] H,W 00000000 00000000		
$001BFC_H$	PTPC24 [R/W] H,W 00000000 00000000		—	—	
$001C00_H$	PCN25 [R/W] B,H,W 00000000 000000-0		PCSR25 [W] H,W XXXXXXXX XXXXXXXX		PPG25
$001C04_H$	PDUT25 [W] H,W XXXXXXXX XXXXXXXX		PTMR25 [R] H,W 11111111 11111111		
$001C08_H$	PCN225 [R/W] B,H,W --000000 -----110		PSDR25 [R/W] H,W 00000000 00000000		
$001C0C_H$	PTPC25 [R/W] H,W 00000000 00000000		—	—	
$001C10_H$	PCN26 [R/W] B,H,W 00000000 000000-0		PCSR26 [W] H,W XXXXXXXX XXXXXXXX		PPG26
$001C14_H$	PDUT26 [W] H,W XXXXXXXX XXXXXXXX		PTMR26 [R] H,W 11111111 11111111		
$001C18_H$	PCN226 [R/W] B,H,W --000000 -----110		PSDR26 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001C1C_H$	PTPC26 [R/W] H,W 00000000 00000000		—	—	PPG26
$001C20_H$	PCN27 [R/W] B,H,W 00000000 000000-0		PCSR27 [W] H,W XXXXXXXX XXXXXXXX		PPG27
$001C24_H$	PDUT27 [W] H,W XXXXXXXX XXXXXXXX		PTMR27 [R] H,W 11111111 11111111		
$001C28_H$	PCN227 [R/W] B,H,W --000000 -----110		PSDR27 [R/W] H,W 00000000 00000000		
$001C2C_H$	PTPC27 [R/W] H,W 00000000 00000000		—	—	
$001C30_H$	PCN28 [R/W] B,H,W 00000000 000000-0		PCSR28 [W] H,W XXXXXXXX XXXXXXXX		PPG28
$001C34_H$	PDUT28 [W] H,W XXXXXXXX XXXXXXXX		PTMR28 [R] H,W 11111111 11111111		
$001C38_H$	PCN228 [R/W] B,H,W --000000 -----110		PSDR28 [R/W] H,W 00000000 00000000		
$001C3C_H$	PTPC28 [R/W] H,W 00000000 00000000		—	—	
$001C40_H$	PCN29 [R/W] B,H,W 00000000 000000-0		PCSR29 [W] H,W XXXXXXXX XXXXXXXX		PPG29
$001C44_H$	PDUT29 [W] H,W XXXXXXXX XXXXXXXX		PTMR29 [R] H,W 11111111 11111111		
$001C48_H$	PCN229 [R/W] B,H,W --000000 -----110		PSDR29 [R/W] H,W 00000000 00000000		
$001C4C_H$	PTPC29 [R/W] H,W 00000000 00000000		—	—	
$001C50_H$	PCN30 [R/W] B,H,W 00000000 000000-0		PCSR30 [W] H,W XXXXXXXX XXXXXXXX		PPG30
$001C54_H$	PDUT30 [W] H,W XXXXXXXX XXXXXXXX		PTMR30 [R] H,W 11111111 11111111		
$001C58_H$	PCN230 [R/W] B,H,W --000000 -----110		PSDR30 [R/W] H,W 00000000 00000000		
$001C5C_H$	PTPC30 [R/W] H,W 00000000 00000000		—	—	
$001C60_H$	PCN31 [R/W] B,H,W 00000000 000000-0		PCSR31 [W] H,W XXXXXXXX XXXXXXXX		PPG31
$001C64_H$	PDUT31 [W] H,W XXXXXXXX XXXXXXXX		PTMR31 [R] H,W 11111111 11111111		
$001C68_H$	PCN231 [R/W] B,H,W --000000 -----110		PSDR31 [R/W] H,W 00000000 00000000		
$001C6C_H$	PTPC31 [R/W] H,W 00000000 00000000		—	—	
$001C70_H$	PCN32 [R/W] B,H,W 00000000 000000-0		PCSR32 [W] H,W XXXXXXXX XXXXXXXX		PPG32

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001C74_H$	PDUT32 [W] H,W XXXXXXXX XXXXXXXX		PTMR32 [R] H,W 11111111 11111111		PPG32
$001C78_H$	PCN232 [R/W] B,H,W --000000 -----110		PSDR32 [R/W] H,W 00000000 00000000		
$001C7C_H$	PTPC32 [R/W] H,W 00000000 00000000		—	—	
$001C80_H$	PCN33 [R/W] B,H,W 00000000 000000-0		PCSR33 [W] H,W XXXXXXXX XXXXXXXX		PPG33
$001C84_H$	PDUT33 [W] H,W XXXXXXXX XXXXXXXX		PTMR33 [R] H,W 11111111 11111111		
$001C88_H$	PCN233 [R/W] B,H,W --000000 -----110		PSDR33 [R/W] H,W 00000000 00000000		
$001C8C_H$	PTPC33 [R/W] H,W 00000000 00000000		—	—	
$001C90_H$	PCN34 [R/W] B,H,W 00000000 000000-0		PCSR34 [W] H,W XXXXXXXX XXXXXXXX		PPG34
$001C94_H$	PDUT34 [W] H,W XXXXXXXX XXXXXXXX		PTMR34 [R] H,W 11111111 11111111		
$001C98_H$	PCN234 [R/W] B,H,W --000000 -----110		PSDR34 [R/W] H,W 00000000 00000000		
$001C9C_H$	PTPC34 [R/W] H,W 00000000 00000000		—	—	
$001CA0_H$	PCN35 [R/W] B,H,W 00000000 000000-0		PCSR35 [W] H,W XXXXXXXX XXXXXXXX		PPG35
$001CA4_H$	PDUT35 [W] H,W XXXXXXXX XXXXXXXX		PTMR35 [R] H,W 11111111 11111111		
$001CA8_H$	PCN235 [R/W] B,H,W --000000 -----110		PSDR35 [R/W] H,W 00000000 00000000		
$001CAC_H$	PTPC35 [R/W] H,W 00000000 00000000		—	—	
$001CB0_H$	PCN36 [R/W] B,H,W 00000000 000000-0		PCSR36 [W] H,W XXXXXXXX XXXXXXXX		PPG36
$001CB4_H$	PDUT36 [W] H,W XXXXXXXX XXXXXXXX		PTMR36 [R] H,W 11111111 11111111		
$001CB8_H$	PCN236 [R/W] B,H,W --000000 -----110		PSDR36 [R/W] H,W 00000000 00000000		
$001CBC_H$	PTPC36 [R/W] H,W 00000000 00000000		—	—	
$001CC0_H$	PCN37 [R/W] B,H,W 00000000 000000-0		PCSR37 [W] H,W XXXXXXXX XXXXXXXX		PPG37
$001CC4_H$	PDUT37 [W] H,W XXXXXXXX XXXXXXXX		PTMR37 [R] H,W 11111111 11111111		
$001CC8_H$	PCN237 [R/W] B,H,W --000000 -----110		PSDR37 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001CCC_H$	PTPC37 [R/W] H,W 00000000 00000000		—	—	PPG37
$001CD0_H$	PCN38 [R/W] B,H,W 00000000 000000-0		PCSR38 [W] H,W XXXXXXXX XXXXXXXX		PPG38
$001CD4_H$	PDUT38 [W] H,W XXXXXXXX XXXXXXXX		PTMR38 [R] H,W 11111111 11111111		
$001CD8_H$	PCN238 [R/W] B,H,W --000000 -----110		PSDR38 [R/W] H,W 00000000 00000000		
$001CDC_H$	PTPC38 [R/W] H,W 00000000 00000000		—	—	
$001CE0_H$	PCN39 [R/W] B,H,W 00000000 000000-0		PCSR39 [W] H,W XXXXXXXX XXXXXXXX		PPG39
$001CE4_H$	PDUT39 [W] H,W XXXXXXXX XXXXXXXX		PTMR39 [R] H,W 11111111 11111111		
$001CE8_H$	PCN239 [R/W] B,H,W --000000 -----110		PSDR39 [R/W] H,W 00000000 00000000		
$001CEC_H$	PTPC39 [R/W] H,W 00000000 00000000		—	—	
$001CF0_H$	PCN40 [R/W] B,H,W 00000000 000000-0		PCSR40 [W] H,W XXXXXXXX XXXXXXXX		PPG40
$001CF4_H$	PDUT40 [W] H,W XXXXXXXX XXXXXXXX		PTMR40 [R] H,W 11111111 11111111		
$001CF8_H$	PCN240 [R/W] B,H,W --000000 -----110		PSDR40 [R/W] H,W 00000000 00000000		
$001CFC_H$	PTPC40 [R/W] H,W 00000000 00000000		—	—	
$001D00_H$	PCN41 [R/W] B,H,W 00000000 000000-0		PCSR41 [W] H,W XXXXXXXX XXXXXXXX		PPG41
$001D04_H$	PDUT41 [W] H,W XXXXXXXX XXXXXXXX		PTMR41 [R] H,W 11111111 11111111		
$001D08_H$	PCN241 [R/W] B,H,W --000000 -----110		PSDR41 [R/W] H,W 00000000 00000000		
$001D0C_H$	PTPC41 [R/W] H,W 00000000 00000000		—	—	
$001D10_H$	PCN42 [R/W] B,H,W 00000000 000000-0		PCSR42 [W] H,W XXXXXXXX XXXXXXXX		PPG42
$001D14_H$	PDUT42 [W] H,W XXXXXXXX XXXXXXXX		PTMR42 [R] H,W 11111111 11111111		
$001D18_H$	PCN242 [R/W] B,H,W --000000 -----110		PSDR42 [R/W] H,W 00000000 00000000		
$001D1C_H$	PTPC42 [R/W] H,W 00000000 00000000		—	—	
$001D20_H$	PCN43 [R/W] B,H,W 00000000 000000-0		PCSR43 [W] H,W XXXXXXXX XXXXXXXX		PPG43

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$001D24_H$	PDUT43 [W] H,W XXXXXXXX XXXXXXXX		PTMR43 [R] H,W 11111111 11111111		PPG43
$001D28_H$	PCN243 [R/W] B,H,W --000000 -----110		PSDR43 [R/W] H,W 00000000 00000000		
$001D2C_H$	PTPC43 [R/W] H,W 00000000 00000000		—	—	
$001D30_H$	PCN44 [R/W] B,H,W 00000000 000000-0		PCSR44 [W] H,W XXXXXXXX XXXXXXXX		PPG44
$001D34_H$	PDUT44 [W] H,W XXXXXXXX XXXXXXXX		PTMR44 [R] H,W 11111111 11111111		
$001D38_H$	PCN244 [R/W] B,H,W --000000 -----110		PSDR44 [R/W] H,W 00000000 00000000		
$001D3C_H$	PTPC44 [R/W] H,W 00000000 00000000		—	—	
$001D40_H$	PCN45 [R/W] B,H,W 00000000 000000-0		PCSR45 [W] H,W XXXXXXXX XXXXXXXX		PPG45
$001D44_H$	PDUT45 [W] H,W XXXXXXXX XXXXXXXX		PTMR45 [R] H,W 11111111 11111111		
$001D48_H$	PCN245 [R/W] B,H,W --000000 -----110		PSDR45 [R/W] H,W 00000000 00000000		
$001D4C_H$	PTPC45 [R/W] H,W 00000000 00000000		—	—	
$001D50_H$	PCN46 [R/W] B,H,W 00000000 000000-0		PCSR46 [W] H,W XXXXXXXX XXXXXXXX		PPG46
$001D54_H$	PDUT46 [W] H,W XXXXXXXX XXXXXXXX		PTMR46 [R] H,W 11111111 11111111		
$001D58_H$	PCN246 [R/W] B,H,W --000000 -----110		PSDR46 [R/W] H,W 00000000 00000000		
$001D5C_H$	PTPC46 [R/W] H,W 00000000 00000000		—	—	
$001D60_H$	PCN47 [R/W] B,H,W 00000000 000000-0		PCSR47 [W] H,W XXXXXXXX XXXXXXXX		PPG47
$001D64_H$	PDUT47 [W] H,W XXXXXXXX XXXXXXXX		PTMR47 [R] H,W 11111111 11111111		
$001D68_H$	PCN247 [R/W] B,H,W --000000 -----110		PSDR47 [R/W] H,W 00000000 00000000		
$001D6C_H$	PTPC47 [R/W] H,W 00000000 00000000		—	—	
$001D70_H$ to $001FFC_H$	—	—	—	—	Reserved
002000_H	CTRLR0 [R/W] B,H,W -------- 000-0001		STATR0 [R/W] B,H,W -------- 00000000		CAN0 (128msb)

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002004_H	ERRCNT0 [R] B,H,W 00000000 00000000		BTR0 [R/W] B,H,W -0100011 00000001		CAN0 (128msb)
002008_H	INTR0 [R] B,H,W 00000000 00000000		TESTR0 [R/W] B,H,W -------- X00000--		
$00200C_H$	BRPER0 [R/W] B,H,W -------- ----0000		—	—	
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	—	—	—	—	
$00202C_H$	—	—	—	—	
002030_H, 002034_H	Reserved(IF1 data mirror)				
002038_H	—	—	—	—	
$00203C_H$	—	—	—	—	
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		
002054_H	IF2DTB10 [R/W] B,H,W 00000000 00000000		IF2DTB20 [R/W] B,H,W 00000000 00000000		
002058_H	—	—	—	—	
$00205C_H$	—	—	—	—	
002060_H, 002064_H	Reserved(IF2 data mirror)				
002068_H to $00207C_H$	—				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002080_H	TREQR20 [R] B,H,W 00000000 00000000		TREQR10 [R] B,H,W 00000000 00000000		CAN0 (128msb)
002084_H	TREQR40 [R] B,H,W 00000000 00000000		TREQR30 [R] B,H,W 00000000 00000000		
002088_H	TREQR60 [R] B,H,W 00000000 00000000		TREQR50 [R] B,H,W 00000000 00000000		
$00208C_H$	TREQR80 [R] B,H,W 00000000 00000000		TREQR70 [R] B,H,W 00000000 00000000		
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	NEWDT60 [R] B,H,W 00000000 00000000		NEWDT50 [R] B,H,W 00000000 00000000		
$00209C_H$	NEWDT80 [R] B,H,W 00000000 00000000		NEWDT70 [R] B,H,W 00000000 00000000		
$0020A0_H$	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		
$0020A8_H$	INTPND60 [R] B,H,W 00000000 00000000		INTPND50 [R] B,H,W 00000000 00000000		
$0020AC_H$	INTPND80 [R] B,H,W 00000000 00000000		INTPND70 [R] B,H,W 00000000 00000000		
$0020B0_H$	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		
$0020B8_H$	MSGVAL60 [R] B,H,W 00000000 00000000		MSGVAL50 [R] B,H,W 00000000 00000000		
$0020BC_H$	MSGVAL80 [R] B,H,W 00000000 00000000		MSGVAL70 [R] B,H,W 00000000 00000000		
$0020C0_H$ to $0020FC_H$	—				
002100_H	CTRLR1 [R/W] B,H,W -------- 000-0001		STATR1 [R/W] B,H,W -------- 00000000		CAN1 (64msb)
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		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002114_H	IF1MSK21 [R/W] B,H,W 11-11111 11111111		IF1MSK11 [R/W] B,H,W 11111111 11111111		CAN1 (64msb)
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	—	—	—	—	
$00212C_H$	—	—	—	—	
002130_H, 002134_H	Reserved (IF1 data mirror)				
002138_H	—	—	—	—	
$00213C_H$	—	—	—	—	
002140_H	IF2CREQ1 [R/W] B,H,W 0------- 00000001		IF2CMSK1 [R/W] B,H,W -------- 00000000		
002144_H	IF2MSK21 [R/W] B,H,W 11-11111 11111111		IF2MSK11 [R/W] B,H,W 11111111 11111111		
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	—	—	—	—	
$00215C_H$	—	—	—	—	
002160_H, 002164_H	Reserved (IF2 data mirror)				
002168_H to $00217C_H$	—				
002180_H	TREQR21 [R] B,H,W 00000000 00000000		TREQR11 [R] B,H,W 00000000 00000000		
002184_H	TREQR41 [R] B,H,W 00000000 00000000		TREQR31 [R] B,H,W 00000000 00000000		
002188_H	—	—	—	—	
$00218C_H$	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002190_H	NEWDT21 [R] B,H,W 00000000 00000000		NEWDT11 [R] B,H,W 00000000 00000000		CAN1 (64msb)
002194_H	NEWDT41 [R] B,H,W 00000000 00000000		NEWDT31 [R] B,H,W 00000000 00000000		
002198_H	—	—	—	—	
$00219C_H$	—	—	—	—	
$0021A0_H$	INTPND21 [R] B,H,W 00000000 00000000		INTPND11 [R] B,H,W 00000000 00000000		
$0021A4_H$	INTPND41 [R] B,H,W 00000000 00000000		INTPND31 [R] B,H,W 00000000 00000000		
$0021A8_H$	—	—	—	—	
$0021AC_H$	—	—	—	—	
$0021B0_H$	MSGVAL21 [R] B,H,W 00000000 00000000		MSGVAL11 [R] B,H,W 00000000 00000000		
$0021B4_H$	MSGVAL41 [R] B,H,W 00000000 00000000		MSGVAL31 [R] B,H,W 00000000 00000000		
$0021B8_H$	—	—	—	—	
$0021BC_H$	—	—	—	—	
$0021C0_H$ to $0021FC_H$	—				
002200_H	CTRLR2 [R/W] B,H,W -------- 000-0001		STATR2 [R/W] B,H,W -------- 00000000		CAN2 (64msb)
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	—	—	—	—	
$00222C_H$	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002230_H, 002234_H	Reserved (IF1 data mirror)				CAN2 (64msb)
002238_H	—	—	—	—	
$00223C_H$	—	—	—	—	
002240_H	IF2CREQ2 [R/W] B,H,W 0------- 00000001		IF2CMSK2 [R/W] B,H,W -------- 00000000		
002244_H	IF2MSK22 [R/W] B,H,W 11-11111 11111111		IF2MSK12 [R/W] B,H,W 11111111 11111111		
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	—	—	—	—	
$00225C_H$	—	—	—	—	
002260_H, 002264_H	Reserved (IF2 data mirror)				
002268_H to $00227C_H$	—				
002280_H	TREQR22 [R] B,H,W 00000000 00000000		TREQR12 [R] B,H,W 00000000 00000000		
002284_H	TREQR42 [R] B,H,W 00000000 00000000		TREQR32 [R] B,H,W 00000000 00000000		
002288_H	—	—	—	—	
$00228C_H$	—	—	—	—	
002290_H	NEWDT22 [R] B,H,W 00000000 00000000		NEWDT12 [R] B,H,W 00000000 00000000		
002294_H	NEWDT42 [R] B,H,W 00000000 00000000		NEWDT32 [R] B,H,W 00000000 00000000		
002298_H	—	—	—	—	
$00229C_H$	—	—	—	—	
$0022A0_H$	INTPND22 [R] B,H,W 00000000 00000000		INTPND12 [R] B,H,W 00000000 00000000		
$0022A4_H$	INTPND42 [R] B,H,W 00000000 00000000		INTPND32 [R] B,H,W 00000000 00000000		
$0022A8_H$	—	—	—	—	
$0022AC_H$	—	—	—	—	
$0022B0_H$	MSGVAL22 [R] B,H,W 00000000 00000000		MSGVAL12 [R] B,H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$0022B4_H$	MSGVAL42 [R] B,H,W 00000000 00000000		MSGVAL32 [R] B,H,W 00000000 00000000		CAN2 (64msb)
$0022B8_H$	—	—	—	—	
$0022BC_H$	—	—	—	—	
$0022C0_H$ to $0022FC_H$	—				
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 -0000000 00000000		DEEARX [R] B,H,W -0000000 00000000		XBS RAM ECC control
002404_H	EECSRX [R/W] B,H,W ----00--	—	EFEARX [R/W] B,H,W -0000000 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
003004_H	EECSRA [R/W] B,H,W ----00--	—	EFEARA [R/W] B,H,W -----000 00000000		
003008_H	—	EFECRA [R/W] B,H,W -------0 00000000 00000000			
$00300C_H$	TEAR0X[R] B,H,W 000----- -------- -0000000 00000000				RAM/ diagnosis XBS RAM
003010_H	TEAR1X[R] B,H,W 000----- -------- -0000000 00000000				
003014_H	TEAR2X[R] B,H,W 000----- -------- -0000000 00000000				
003018_H	TAEARX [R/W] B,H,W -1111111 11111111		TASARX [R/W] B,H,W -0000000 00000000		
$00301C_H$	TFECRX [R/W] B,H,W ----0000	TICRX [R/W] B,H,W ----0000	TTCRX [R/W] B,H,W ------00 00001100		
003020_H	TSRCRX [W] B,H,W 0-------	—	—	TKCCRX [R/W] B,H,W 00----00	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003024_H to $00302C_H$	—				Reserved
003030_H	TEAR0A[R] B,H,W 000----- -------- -----000 00000000				RAM/ diagnosis Backup RAM
003034_H	TEAR1A[R] B,H,W 000----- -------- -----000 00000000				
003038_H	TEAR2A[R] B,H,W 000----- -------- -----000 00000000				
$00303C_H$	TAEARA[R/W] B,H,W -----111 11111111		TASARA[R/W] B,H,W -----000 00000000		
003040_H	TFECRA [R/W] B,H,W ----0000	TICRA [R/W] B,H,W ----0000	TTCRA [R/W] B,H,W ------00 00001100		RAM/ diagnosis Backup RAM
003044_H	TSRCRA [R/W] B,H,W 0-------	—	—	TKCCRA [R/W] B,H,W 00----00	
003048_H to $0030FC_H$	—				Reserved
003100_H	BUSDIGSR0[R/W] H,W 00000000 0-----00		BUSDIGSR1[R/W] H,W 00000000 0-----00		BUS diagnosis
003104_H	BUSDIGSR2[R/W] H,W 00000000 0-----00		BUSTSTR0[R/W] H,W 00--0000 00000000		
003108_H	BUSADR0 [R] W 00000000 00000000 00000000 00000000				
$00310C_H$	BUSADR1 [R] W 00000000 00000000 00000000 00000000				
003110_H	BUSADR2 [R] W 00000000 00000000 00000000 00000000				
003114_H	—	—	BUSDIGSR3[R/W] H,W 00000000 0-----00		BUS diagnosis
003118_H	BUSDIGSR4[R/W] H,W 00000000 0-----00		BUSTSTR1[R/W] H,W 00--000- 00000000		
$00311C_H$	—	—	—	—	
003120_H	BUSADR3 [R] W 00000000 00000000 00000000 00000000				
003124_H	BUSADR4 [R] W 00000000 00000000 00000000 00000000				
003128_H to $003FFC_H$	—				Reserved
004000_H to $005FFC_H$	Backup-RAM				Backup RAM area
006000_H to $00EFFC_H$	—	—	—	—	Reserved

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
$00F000_H$ to $00FEFC_H$	—	—	—	—	Reserved [S]
$00FF00_H$	DSUCR [R/W] B,H,W -------- -------0		—	—	OCDU [S]
$00FF04_H$ to $00FF0C_H$	—				Reserved [S]
$00FF10_H$	PCSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FF14_H$	PSSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FF18_H$ 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]: This is a system register. If read/write is performed to these registers in user mode, the illegal instruction exception (data access error) is generated.

B. List of Interrupt Vector

List of Interrupt Vector is shown.

This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.

Table B-1 Interrupt Vector MB91F52xB (64pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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	15	0F	$15(F_H)$ Fixed	$3C0_H$	$000FFFC0_H$	-
Error generation at internal bus diagnosis						
XBS RAM double-bit error detection						
Backup RAM double-bit error detection						
TPU violation						
External interrupt 0-7	16	10	ICR00	$3BC_H$	$000FFFBC_H$	0
External interrupt 8-15	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*[8]
External low-voltage detection interrupt						
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*[2]
Reload timer 3/6/7	19	13	ICR03	$3B0_H$	$000FFFB0_H$	3*[2]
Multi-function serial interface ch.0 (reception completed)	20	14	ICR04	$3AC_H$	$000FFFAC_H$	4*[1]
Multi-function serial interface ch.0 (status)						
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*[1]
-	22	16	ICR06	$3A4_H$	$000FFFA4_H$	-*[7]

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
-	23	17	ICR07	$3A0_H$	$000FFFA0_H$	-*7
-	24	18	ICR08	$39C_H$	$000FFF9C_H$	-*7
-	25	19	ICR09	398_H	$000FFF98_H$	-*7
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*1
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*1
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*1
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*1
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
-	38	26	ICR22	364_H	$000FFF64_H$	-*7
16-bit free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360_H	$000FFF60_H$	23

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
PPG1/10/11/20/30/31 16-bit free-run timer 1 (0 detection) / (compare clear)	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*3
PPG2/3/12/13/23/43 16-bit free-run timer 2 (0 detection) / (compare clear)	41	29	ICR25	358_H	$000FFF58_H$	25*3
PPG4/24/35	42	2A	ICR26	354_H	$000FFF54_H$	26*3
PPG7/16/17/27/37	43	2B	ICR27	350_H	$000FFF50_H$	27*3
PPG19	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*3
16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29
Main timer Sub timer PLL timer 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*1 *4
A/D converter0/1/7/10/11/14/15/16/ 17/22/27/28/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*6
16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*1
Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*1

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/34/35/37/38/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41
32-bit OCU7/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
-	60	3C	ICR44	$30C_H$	$000FFF0C_H$	-*7
- -	61	3D	ICR45	308_H	$000FFF08_H$	-
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.

*1: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.

*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.

*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.

*4: The clock calibration unit does not support the DMA transfer by the interrupt.

*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.

*7: There is no resource corresponding to interrupt level.

*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

Table B-2 Interrupt Vector MB91F52xD (80pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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 Error generation at internal bus diagnosis XBS RAM double-bit error detection Backup RAM double-bit error detection TPU violation	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 External low-voltage detection interrupt	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*[8]
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*[2]
Reload timer 3/6/7	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*[1]
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*[1]
-	22	16	ICR06	$3A4_H$	$000FFFA4_H$	-*[7]
-	23	17	ICR07	$3A0_H$	$000FFFA0_H$	-*[7]
Multi-function serial interface ch.2 (reception completed) Multi-function serial interface ch.2 (status)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8*[1]
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9*[1]

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*[1]
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*[1]
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*[1]
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*[1]
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
-	38	26	ICR22	364_H	$000FFF64_H$	-*[7]
16-bit free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360_H	$000FFF60_H$	23
PPG1/10/11/20/30/31	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*[3]
16-bit free-run timer 1 (0 detection) / (compare clear)						
PPG2/3/12/13/23/43	41	29	ICR25	358_H	$000FFF58_H$	25*[3]
16-bit free-run timer 2 (0 detection) / (compare clear)						
PPG4/5/15/24/35	42	2A	ICR26	354_H	$000FFF54_H$	26*[3]
PPG7/16/17/26/27/37	43	2B	ICR27	350_H	$000FFF50_H$	27*[3]
PPG8/18/19/29	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*[3]

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29
Main timer Sub timer PLL timer 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*1 *4
A/D converter 0/1/7/10/11/12/14/15/16/ 17/19/22/26/27/28/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*6
32-bit free-run timer 5 16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35*6
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*1
Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*1
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit OCU7/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
-	60	3C	ICR44	$30C_H$	$000FFF0C_H$	-*7
Base timer 1 IRQ0	61	3D	ICR45	308_H	$000FFF08_H$	45
Base timer 1 IRQ1						
-						
-						
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.
*1: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.
*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.
*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.
*4: The clock calibration unit does not support the DMA transfer by the interrupt.
*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.
*7: There is no resource corresponding to interrupt level.
*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

Table B-3 Interrupt Vector MB91F52xF (100pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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 Error generation at internal bus diagnosis XBS RAM double-bit error detection Backup RAM double-bit error detection TPU violation	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 External low-voltage detection interrupt	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*[8]
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*[2]
Reload timer 2/3/6/7	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*[1]
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*[1]
Multi-function serial interface ch.1 (reception completed) Multi-function serial interface ch.1 (status)	22	16	ICR06	$3A4_H$	$000FFFA4_H$	6*[1]
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	$3A0_H$	$000FFFA0_H$	7*[1]
Multi-function serial interface ch.2 (reception completed) Multi-function serial interface ch.2 (status)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8*[1]

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9*[1]
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*[1]
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*[1]
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*[1]
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*[1]
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
Multi-function serial interface ch.7 (reception completed)	38	26	ICR22	364_H	$000FFF64_H$	22*[1]
Multi-function serial interface ch.7 (status)						

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
16-bit free-run timer 0 (0 detection) / (compare clear) Multi-function serial interface ch.7 (transmission completed)	39	27	ICR23	360_H	$000FFF60_H$	23
PPG1/10/11/20/21/30/31 16-bit free-run timer 1 (0 detection) / (compare clear)	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*3
PPG2/3/12/13/23/32/43 16-bit free-run timer 2 (0 detection) / (compare clear)	41	29	ICR25	358_H	$000FFF58_H$	25*3
PPG4/5/14/15/24/25/35/44	42	2A	ICR26	354_H	$000FFF54_H$	26*3
PPG6/7/16/17/26/27/37	43	2B	ICR27	350_H	$000FFF50_H$	27*3
PPG8/9/18/19/28/29	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*3
Multi-function serial interface ch.8 (reception completed) Multi-function serial interface ch.8 (status) 16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29*1
Main timer Sub timer PLL timer Multi-function serial interface ch.8 (transmission completed) 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*1 *4
A/D converter0/1/7/9/10/11/12/13/14/15/16 17/18/19/22/23/26/27/28/29/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*6
32-bit free-run timer 3/5 16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35*6
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*1

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit ICU7 (fetching /measurement) Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*[1]
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41
32-bit OCU6/7/10/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
- -	60	3C	ICR44	$30C_H$	$000FFF0C_H$	44
Base timer 1 IRQ0 Base timer 1 IRQ1 - -	61	3D	ICR45	308_H	$000FFF08_H$	45
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.
*1: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.
*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.
*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.
*4: The clock calibration unit does not support the DMA transfer by the interrupt.
*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.
*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

Table B-4 Interrupt Vector MB91F52xJ (120pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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 Error generation at internal bus diagnosis XBS RAM double-bit error detection Backup RAM double-bit error detection TPU violation	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 External low-voltage detection interrupt	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*8
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*2
Reload timer 2/3/6/7	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*1
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*1
Multi-function serial interface ch.1 (reception completed) Multi-function serial interface ch.1 (status)	22	16	ICR06	$3A4_H$	$000FFFA4_H$	6*1
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	$3A0_H$	$000FFFA0_H$	7*1
Multi-function serial interface ch.2 (reception completed) Multi-function serial interface ch.2 (status)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8*1

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9*[1]
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*[1]
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*[1]
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*[1]
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*[1]
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
Multi-function serial interface ch.7 (reception completed)	38	26	ICR22	364_H	$000FFF64_H$	22*[1]
Multi-function serial interface ch.7 (status)						
16-bit free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360_H	$000FFF60_H$	23
Multi-function serial interface ch.7 (transmission completed)						

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
PPG0/1/10/11/20/21/30/31 16-bit free-run timer 1 (0 detection) / (compare clear)	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*[3]
PPG2/3/12/13/22/23/32/33/43 16-bit free-run timer 2 (0 detection) / (compare clear)	41	29	ICR25	358_H	$000FFF58_H$	25*[3]
PPG4/5/14/15/24/25/35/44	42	2A	ICR26	354_H	$000FFF54_H$	26*[3]
PPG6/7/16/17/26/27/37	43	2B	ICR27	350_H	$000FFF50_H$	27*[3]
PPG8/9/18/19/28/29	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*[3]
Multi-function serial interface ch.8 (reception completed) Multi-function serial interface ch.8 (status) 16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29*[1]
Main timer Sub timer PLL timer Multi-function serial interface ch.8 (transmission completed) 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*[1] *[4]
A/D converter 0/1/7/9/10/11/12/13/14/15/16 17/18/19/20/21/22/23/24/25/26/27/28/29/30/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*[6]
32-bit free-run timer 3/5 16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35*[6]
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*[1]
32-bit ICU7 (fetching /measurement) Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*[1]
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41
32-bit OCU6/7/10/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
Base timer 0 IRQ0 Base timer 0 IRQ1	60	3C	ICR44	$30C_H$	$000FFF0C_H$	44
Base timer 1 IRQ0 Base timer 1 IRQ1 - -	61	3D	ICR45	308_H	$000FFF08_H$	45
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.

*1: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.

*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.

*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.

*4: The clock calibration unit does not support the DMA transfer by the interrupt.

*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.

*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

Table B-5 Interrupt Vector MB91F52xK (144pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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 Error generation at internal bus diagnosis XBS RAM double-bit error detection Backup RAM double-bit error detection TPU violation	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 External low-voltage detection interrupt	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*[8]
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*[2]
Reload timer 2/3/6/7	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*[1]
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*[1]
Multi-function serial interface ch.1 (reception completed) Multi-function serial interface ch.1 (status)	22	16	ICR06	$3A4_H$	$000FFFA4_H$	6*[1]
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	$3A0_H$	$000FFFA0_H$	7*[1]
Multi-function serial interface ch.2 (reception completed) Multi-function serial interface ch.2 (status)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8*[1]

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9*[1]
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*[1]
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*[1]
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*[1]
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*[1]
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
Multi-function serial interface ch.7 (reception completed)	38	26	ICR22	364_H	$000FFF64_H$	22*[1]
Multi-function serial interface ch.7 (status)						
16-bit free-run timer 0 ("0" detection) / (compare clear)	39	27	ICR23	360_H	$000FFF60_H$	23
Multi-function serial interface ch.7 (transmission completed)						

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
PPG0/1/10/11/20/21/30/31/40/41 16-bit free-run timer 1 ("0" detection) / (compare clear)	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*[3]
PPG2/3/12/13/22/23/32/33/43 16-bit free-run timer 2 ("0" detection) / (compare clear)	41	29	ICR25	358_H	$000FFF58_H$	25*[3]
PPG4/5/14/15/24/25/34/35/44	42	2A	ICR26	354_H	$000FFF54_H$	26*[3]
PPG6/7/16/17/26/27/36/37	43	2B	ICR27	350_H	$000FFF50_H$	27*[3]
PPG8/9/18/19/28/29/38/39	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*[3]
Multi-function serial interface ch.8 (reception completed) Multi-function serial interface ch.8 (status) 16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29*[1]
Main timer Sub timer PLL timer Multi-function serial interface ch.8 (transmission completed) 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*[1] *[4]
A/D converter 0/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16 17/18/19/20/21/22/23/24/25/26/27/28/29/30/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*[6]
32-bit free-run timer 3/5 16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35*[6]
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*[1]
32-bit ICU7 (fetching /measurement) Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*[1]
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41
32-bit OCU6/7/10/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
Base timer 0 IRQ0 Base timer 0 IRQ1	60	3C	ICR44	$30C_H$	$000FFF0C_H$	44
Base timer 1 IRQ0 Base timer 1 IRQ1 - -	61	3D	ICR45	308_H	$000FFF08_H$	45
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.

*1:: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.

*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.

*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.

*4: The clock calibration unit does not support the DMA transfer by the interrupt.

*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.

*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

Table B-6 Interrupt Vector MB91F52xL (176pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Reset	0	0	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	1	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	2	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	3	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	4	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	5	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	6	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	7	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	8	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	9	-	$3D8_H$	$000FFFD8_H$	-
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 Error generation at internal bus diagnosis XBS RAM double-bit error detection Backup RAM double-bit error detection TPU violation	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 External low-voltage detection interrupt	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1*8
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2*2
Reload timer 2/3/6/7	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*1
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5*1
Multi-function serial interface ch.1 (reception completed) Multi-function serial interface ch.1 (status)	22	16	ICR06	$3A4_H$	$000FFFA4_H$	6*1
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	$3A0_H$	$000FFFA0_H$	7*1
Multi-function serial interface ch.2 (reception completed) Multi-function serial interface ch.2 (status)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8*1

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9*[1]
Multi-function serial interface ch.3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10*[1]
Multi-function serial interface ch.3 (status)						
Multi-function serial interface ch.3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
Multi-function serial interface ch.4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12*[1]
Multi-function serial interface ch.4 (status)						
Multi-function serial interface ch.4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
Multi-function serial interface ch.5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14*[1]
Multi-function serial interface ch.5 (status)						
Multi-function serial interface ch.5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
Multi-function serial interface ch.6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16*[1]
Multi-function serial interface ch.6 (status)						
Multi-function serial interface ch.6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
RAM diagnosis completed						
RAM initialization completed						
Error generation at RAM diagnosis						
Backup RAM diagnosis completed						
Backup RAM initialization completed						
Error generation at Backup RAM diagnosis						
CAN2	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Up/down counter 0						
Up/down counter 1						
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
Multi-function serial interface ch.7 (reception completed)	38	26	ICR22	364_H	$000FFF64_H$	22*[1]
Multi-function serial interface ch.7 (status)						
16-bit free-run timer 0 ("0" detection) / (compare clear)	39	27	ICR23	360_H	$000FFF60_H$	23
Multi-function serial interface ch.7 (transmission completed)						

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
PPG0/1/10/11/20/21/30/31/40/41 16-bit free-run timer 1 ("0" detection) / (compare clear)	40	28	ICR24	$35C_H$	$000FFF5C_H$	24*[3]
PPG2/3/12/13/22/23/32/33/42/43 16-bit free-run timer 2 ("0" detection) / (compare clear)	41	29	ICR25	358_H	$000FFF58_H$	25*[3]
PPG4/5/14/15/24/25/34/35/44/45	42	2A	ICR26	354_H	$000FFF54_H$	26*[3]
PPG6/7/16/17/26/27/36/37/46/47	43	2B	ICR27	350_H	$000FFF50_H$	27*[3]
PPG8/9/18/19/28/29/38/39	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28*[3]
Multi-function serial interface ch.8 (reception completed) Multi-function serial interface ch.8 (status) 16-bit ICU 0 (fetching) / 16-bit ICU 1 (fetching)	45	2D	ICR29	348_H	$000FFF48_H$	29*[1]
Main timer Sub timer PLL timer Multi-function serial interface ch.8 (transmission completed) 16-bit ICU 2 (fetching) /16-bit ICU 3 (fetching)	46	2E	ICR30	344_H	$000FFF44_H$	30
Clock calibration unit (Sub oscillation) Multi-function serial interface ch.9 (reception completed) Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31*[1] *[4]
A/D converter 0/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16 17/18/19/20/21/22/23/24/25/26/27/28/29/30/31	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration unit (CR oscillation) Multi-function serial interface ch.9 (transmission completed) 16-bit OCU 0 (match) / 16-bit OCU 1 (match)	49	31	ICR33	338_H	$000FFF38_H$	33
32-bit free-run timer 4 16-bit OCU 2 (match) / 16-bit OCU 3 (match)	50	32	ICR34	334_H	$000FFF34_H$	34*[6]
32-bit free-run timer 3/5 16-bit OCU 4 (match) / 16-bit OCU 5 (match)	51	33	ICR35	330_H	$000FFF30_H$	35*[6]
32-bit ICU6 (fetching /measurement) Multi-function serial interface ch.10 (reception completed) Multi-function serial interface ch.10 (status)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36*[1]
32-bit ICU7 (fetching /measurement) Multi-function serial interface ch.10 (transmission completed)	53	35	ICR37	328_H	$000FFF28_H$	37

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexa-decimal				
32-bit ICU8 (fetching /measurement) Multi-function serial interface ch.11 (reception completed) Multi-function serial interface ch.11 (status)	54	36	ICR38	324_H	$000FFF24_H$	38*[1]
32-bit ICU9 (fetching /measurement) WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320_H	$000FFF20_H$	39
32-bit ICU4 (fetching /measurement) Multi-function serial interface ch.11 (transmission completed)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
32-bit ICU5 (fetching /measurement) A/D converter 32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47	57	39	ICR41	318_H	$000FFF18_H$	41
32-bit OCU6/7/10/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
32-bit OCU8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
Base timer 0 IRQ0 Base timer 0 IRQ1	60	3C	ICR44	$30C_H$	$000FFF0C_H$	44
Base timer 1 IRQ0 Base timer 1 IRQ1 - -	61	3D	ICR45	308_H	$000FFF08_H$	45
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.)	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	$000FFEF4_H$ | $000FFC00_H$	-

*: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.
*1: The status of the multi-function serial interface does not support the DMA transfer by the I^2C reception.
*2: The reload timer ch.4 to ch.7 does not support the DMA transfer by the interrupt.
*3: The PPG ch.24 to ch.47 does not support the DMA transfer by the interrupt.
*4: The clock calibration unit does not support the DMA transfer by the interrupt.
*6: The 32-bit free-run timer ch.3 to ch.5 does not support the DMA transfer by the interrupt.
*8: It does not support the DMA transfer by the external low-voltage detection interrupt.

C. Pins Statuses in State of CPU

Pin statuses in state of CPU are shown.

Table C-1 : Pin Statuses

Pin No.(QFP176)	Pin No.(QFP144)	Pin No.(QFP120)	Pin No.(QFP100)	Pin No.(QFP80)	Pin No.(QFP64)	Pin name	Pin function	Input/level Output/level Circuit type	Pin status after reset factor generation and after reset releasing										Pin status in the sleep and standby mode						Influence by setting of PORTEN
									Reset factor1(*1) [MB91F52xxB/MB91F52xxC/ MB91F52xxE] (*1 Power-on reset, external reset NMI enable) [MB91F52xxD] (*1 Power-on reset, external reset)		Reset factor2(*2) (*2 Low-voltage detect (external power supply voltage))				Reset factor3(*3) [MB91F52xxB/MB91F52xxC/MB91F52xxE] External reset NMI disabled				Internal reset factor4(*4)	Internal reset factor5(*14)	Sleep mode		Stop mode		Watch mode		
									External factor generate	External factor after releases	External factor generate		External factor After releases		External factor generate		External factor After releases									
									Internal reset issuance in progress	Wdt internal reset is after release (oscillation wait is in progress)	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Wdt internal reset is after release (oscillation wait is in progress)	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Wdt internal reset is after release (oscillation wait is in progress)	After external reset releasing	Wdt internal reset is after release (oscillation wait is in progress)	0LVL1:0LVL0=0x	0LVL1:0LVL0=1x	0LVL1:0LVL0=0x	0LVL1:0LVL0=1x	0LVL1:0LVL0=0x	0LVL1:0LVL0=1x	
1	1	1	1	1	1	1	VSS	GND																			
						2	P015 D29 TRG0_0	General-purpose I/O port External bus data bit29 I/O PPG trigger0 input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained(*7)	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						3	P016 D30 TRG1_0	General-purpose I/O port External bus data bit30 I/O PPG trigger1 input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained(*7)	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						4	P170 PPG36_1	General-purpose I/O port PPG ch36 output(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained(*7)	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						5	P017 D31 TRG2_0	General-purpose I/O port External bus data bit31 I/O PPG trigger2 input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained(*7)	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						6	P171 PPG37_1	General-purpose I/O port PPG ch37 output(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
2	2	2	2	2	5	7	P020 ASX SIN3_1 TRG3_0 TIN0_2 RTC6_1	General-purpose I/O port External bus address strobe output Multi-function serial ch3 serial data input(1) PPG trigger3 input(0) Reload-timer ch0 eventinput(2) Waveform generator_ch5 output pin(1)	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						8	P021 CS0X SOT3_1 TRG4_1 TRG4_0	General-purpose I/O port External bus chip select0 output Multi-function serial ch3 serial data output(1) PPG trigger5 input(1) PPG trigger4 input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						9	P022 CS1X SCK3_1 TRG7_1 TRG5_0	General-purpose I/O port External bus chip select1 output Multi-function serial ch3 clock I/O(1) PPG trigger7 input(1) PPG trigger5 input(0)	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
						10	P023 RDX SCS3_1 PPG22_0 TIN0_0	General-purpose I/O port External bus read strobe output Serial chip select3 output(1) PPG ch22 output(0) Reload-timer ch0 eventinput(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
3	3	3	3	6	9	11	P024 WR0X SIN4_1 PPG24_0 TIN1_0 RTC4_1 INT15_0	General-purpose I/O port External bus write strobe0 output Multi-function serial ch4 serial data input(1) PPG ch24 output(0) Reload-timer ch1 eventinput(0) Waveform generator ch4 output pin(1) INT15 ext.interrupt input	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		4	7	10	12	P025 WR1X SOT4_1 PPG25_0 TIN2_0	General-purpose I/O port External bus write strobe1 output Multi-function serial ch4 serial data output(1) PPG ch25 output Reload-timer ch2 eventinput(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					13	P172 PPG38_1	General-purpose I/O port PPG ch38 output(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
4	5	8	11	14	P026 A00 SCK4_1 PPG26_0 TIN3_0	General-purpose I/O port External bus address bit0 output(0) Multi-function serial ch4 clock I/O(1) PPG ch26 output(0) Reload-timer ch3 eventinput(0)	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
4	5	6	9	12	15	P027 A01 SCS40_1 PPG27_0 TOT0_0 RTC0_1	General-purpose I/O port External bus address bit1 output(0) Serial chip select40 output(1) PPG ch27 output(0) Reload-timer ch0 output(0) Waveform generator_ch3 output pin(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					16	P173 PPG39_1	General-purpose I/O port PPG ch39 output(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		7	10	13	17	P030 A02 SCS41_1 PPG28_0 TOT1_0	General-purpose I/O port External bus address bit2 output(0) Serial chip select41 output(1) PPG ch28 output(0) Reload-timer ch1 output(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
	6	8	11	14	18	P031 A03 SCS42_1 PPG29_0 TOT2_0	General-purpose I/O port External bus address bit3 output(0) Serial chip select42 output(1) PPG ch29 output(0) Reload-timer ch2 output(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
5	7	9	12	15	19	P032 A04 SCS43_1 PPG30_0 TOT3_0 RTC0_1	General-purpose I/O port External bus address bit4 output(0) Serial chip select43 output(1) PPG ch30 output(0) Reload-timer ch3 output(0) Waveform generator_ch2 output pin(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
6	8	10	13	16	20	P033 A05 PPG31_0 ICU0_3 TIN4_0 RTC6_1 SCK3_2	General-purpose I/O port External bus address bit5 output(0) PPG ch31 output(0) Input capture ch3 input(3) Reload-timer ch4 eventinput(0) Waveform generator ch1 output pin(1) Multi-function serial ch3 clock I/O(2)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
7	9	11	14	17	21	P034 A06 OCU11_1 ICU2_3 TIN5_0 RTC0_1 SOT3_2	General-purpose I/O port External bus address bit6 output(0) Output compare ch11 output(1) Input capture ch2 input(3) Reload-timer ch5 eventinput(0) Waveform generator ch0 output pin(1) Multi-function serial ch3 serial data output(2)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		12	15	18	22	P150 SOT8_0/SDA8 OCU10_1 TRG8_0 ICU8_3 TIN8_0	General-purpose I/O port Multi-function serial ch8 serial data output(0) / I2C bus serial data I/O Output compare ch10 output(1) PPG trigger8 input(0) Input capture ch1 input(3) Reload-timer ch8 eventinput(0)	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
8	10	13	16	19	23	P151 SCK8_0/SCL8 OCU9_1 TRG7_0 ICU0_3 TIN7_0 ZIN0_2 DTTI_1	General-purpose I/O port Multi-function serial ch8 clock I/O(0) / I2C bus serial clock I/O Output compare ch9 output(1) PPG trigger7 input(0) Input capture ch0 input(3) Reload-timer ch7 eventinput(0) UFD counter ch0 ZIN input(2) Waveform generator ch0-ch5 input	input CMOS hysteresis output CMOS F	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
9	11	14	17	20	24	P035 A07 SIN8_0 OCU8_1 TOT4_0 AIN0_0 INT11_0	General-purpose I/O port External bus address bit7 output Multi-function serial ch8 serial data input(0) Output compare ch8 output(1) Reload-timer ch4 output(0) UFD counter ch0 AIN input(0) INT11 ext.interrupt input(0)	5V トレラント入力 input CMOS hysteresis output CMOS	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
10	12	15	18	21	25	P036 A08 SCS8_0 OCU7_1 TOT5_0 BIN0_0	General-purpose I/O port External bus address bit8 output(0) Serial chip select 8 output(0) Output compare ch7 output(1) Reload-timer ch5 output(0) UFD counter ch0 BIN input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		16	19	22	26	P037 A09 OCU6_1 TOT6_0 ZIN0_0	General-purpose I/O port External bus address bit9 output(0) Output compare ch6 output(1) Reload-timer ch6 output(0) UFD counter ch0 ZIN input(0)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					27	P174 TRG8_1	General-purpose I/O port PPG trigger8 input(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					28	P175 TRG9_1	General-purpose I/O port PPG trigger9 input(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
11	13	17	20	23	29	P040 A10 PPG23_1 TOT7_0 AIN1_0 SIN0_1	General-purpose I/O port External bus address bit10 output(0) PPG ch23 output(1) Reload-timer ch7 output(0) UFD counter ch1 AIN input(0) Multi-function serial ch0 serial data input(1)	input automotive output CMOS A	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
12	14	18	21	24	30	P041 A11 SIN9_0 ICU9_1 BIN1_0 INT12_0	General-purpose I/O port External bus address bit11 output(0) Multi-function serial ch9 serial data input(0) Input capture ch9 input(1) UFD counter ch1 BIN input(0) INT12 ext.interrupt input(0)	5V トレラント入力 input CMOS hysteresis output CMOS I	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes

Pin No (SMP/V)	Pin No (SMP/H)	Pin No (LQFP/V)	Pin No (LQFP/H)	Pin No (LQFP/N)	Pin No (LQFP/S)	Pin name	Pin function	Input/level Output/level Circuit type	Pin status after reset factor generation and after reset releasing												Pin status in the sleep and standby mode						Influence by setting of PDSTOP
									Reset factor1(*1) [MB91FS2xxB/MB91FS2xxC/ MB91FS2xxE] (*1 Power-on reset, external reset NMI enable) [MB91FS2xxD] (*1 Power-on reset, external reset)		Reset factor2(*2) (*2 Low-voltage detect (external power supply voltage))			Reset factor3(*3) [MB91FS2xxB/MB91FS2xxC/MB91FS2xxE] External reset NMI disabled			Internal reset factor4(*4)	Internal reset factor5(*14)	Sleep mode		Stop mode		Watch mode		
									External factor generate	External factor after releases	External factor generate		External factor After releases	External factor generate		External factor After releases										
									Internal reset issuance in progress	Within internal reset(no after releases (oscillation stabilization wait (t=prescribe))	Before internal reset issuance	Internal reset issuance in progress	Within internal reset(no after releases (oscillation stabilization wait (t=prescribe))	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Within internal reset(no after releases (oscillation stabilization wait (t=prescribe))	After internal reset releasing	Within internal reset(no after releases (oscillation stabilization wait (t=prescribe))	After internal reset releasing	SLV11-0[1:0]=00x	SLV11-0[1:0]=01x	SLV11-0[1:0]=00x	SLV11-0[1:0]=01x	SLV11-0[1:0]=00x	SLV11-0[1:0]=01x	
13	15	19	22	25	31	P042 AI12 SOT9_0 AN47 ICU8_1 TRG0_1 SIN4_0	General-purpose I/O port External bus address bit12 output Multi-function serial ch9 serial data output(0) ADC AN47 input Input capture ch8 input(1) PPG trigger0 input(1) UFD counter ch1 ZIN input(0)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		20	23	26	32	P043 AI13 ICU7_1 TRG1_1	General-purpose I/O port External bus address bit13 output(0) Input capture ch7 input(1) PPG trigger1 input(1)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
	16	21	24	27	33	P044 AI14 SCS9_0 ICU6_1 TRG2_1	General-purpose I/O port External bus address bit14 output(0) Serial chip select9 output(0) Input capture ch6 input(1) PPG trigger2 input(1)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
14	17	22	25	28	34	P045 AI15 SCK9_0 AN46 ICU5_1 TRG3_1 TOT1_2	General-purpose I/O port External bus address bit15 output(0) Multi-function serial ch9 clock I/O(0) ADC AN46 input Input capture ch5 input(1) PPG trigger3 input(1) Reload-timer ch1 output(2)	input: CMOS hysteresis output: CMOS G	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		26	29	35		P046 AI16 ICU4_1 TRG4_1	General-purpose I/O port External bus address bit16 output(0) Input capture ch4 input(1) PPG trigger4 input(1)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					35	P178 TRG10_0	General-purpose I/O port PPG trigger10 input(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
15	18	23	27	30	36	P047 AI17 AN45 TRG8_0 TIN3_2 SOT0_1	General-purpose I/O port External bus address bit17 output(0) ADC AN45 input PPG trigger8 input(0) Reload-timer ch3 eventinput(2) Multi-function serial ch0 serial data output(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					38	P177 TRG11_0	General-purpose I/O port PPG trigger11 input(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		28	31	39		P050 AI18 TRG5_1 PPG33_0	General-purpose I/O port External bus address bit18 output PPG trigger5 input(1) PPG ch33 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
				32	40	P051 AI19 TRG9_0	General-purpose I/O port External bus address bit19 output PPG trigger9 input(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
				33	41	P052 AI20 PPG34_0 INT14_0	General-purpose I/O port External bus address bit20 output PPG ch34 output(0) INT14 ext.interrupt input(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
16	19	24	29	34	42	P053 AI21 AN44 PPG35_0 INT14_1 SCK0_1	General-purpose I/O port External bus address bit21 output ADC AN44 input PPG ch35 output INT14 ext.interrupt input(1) Multi-function serial ch0 clock I/O(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					43	P054 SYSCLK PPG36_0	General-purpose I/O port External bus system clock output PPG ch36 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
17	20	25	30	36	44	VCC	5.0V power supply	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	21	26	31	37	45	VSS	GND	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	22	27	32	38	46	P055 CS2X SIN10_0 AN43 PPG37_0 TIN4_1	General-purpose I/O port External bus chip select2 output Multi-function serial ch10 serial data input(0) ADC AN43 input PPG ch37 output(0) Reload-timer ch4 eventinput(1)	input: CMOS hysteresis output: CMOS G	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					47	P180 PPG40_0	General-purpose I/O port PPG ch40 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					48	P181 PPG41_0	General-purpose I/O port PPG ch41 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
				33	39	49	P056 CS3X SCS8_0 PPG0_1 ICU0_1 TIN5_1 TOT0_2	General-purpose I/O port External bus chip select3 output Input capture ch8 input(0) PPG ch0 output(1) Input capture ch0 input(1) Reload-timer ch5 eventinput(1) Waveform generator ch0-ch5 input(2)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
20	23	28	33	39	47	AVCC1	Analog power supply for ADC1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	24	29	35	41	51	P057 RDY SCK10_1 AN42 ICU8_0 TIN0_2 PPG1_1 ICU1_1 TIN6_1	General-purpose I/O port External bus ready input Multi-function serial ch10 clock I/O(1) ADC AN42 input Input capture ch8 input(0) PPG trigger0 input(2) PPG ch1 output(1) Input capture ch1 input(1) Reload-timer ch6 eventinput(1)	input: CMOS hysteresis output: CMOS G	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
20	25	30	36	42	52	AVRH1	A/D converter upper limit reference voltage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	26	31	37	43	53	AVSS1/AVRL1	A/D/DA converter GND A/D converter lower limit reference voltage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
				44	54	P142 SCK10_0/SCL1 0 PPG38_0 TIN7_1	General-purpose I/O port Multi-function serial ch10 clock I/O(0) / I2C bus serial clock I/O PPG ch38 output(0) Reload-timer ch7 eventinput(1)	input: CMOS hysteresis output: CMOS F	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
				45	55	P143 SOT10_0/SDA1 0 PPG39_0 TOT4_1	General-purpose I/O port Multi-function serial ch10 serial data output(0) / I2C bus serial data I/O PPG ch39 output(0) Reload-timer ch4 output(1)	input: CMOS hysteresis output: CMOS F	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					56	P182 PPG42_0	General-purpose I/O port PPG ch42 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
			32	38	46	P060 SCS10_0 PPG2_1 ICU2_1 TOT5_1 INT13_0	General-purpose I/O port Serial chip select10 output(0) PPG ch2 output(1) Input capture ch2 input(1) Reload-timer ch5 output(1) INT13 ext.interrupt input(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
22	27	33	39	47	57	P061 SOT10_1 AN41 ICU6_0 IRQ3_1 ICU5_1 TOT6_1 INT13_1	General-purpose I/O port Multi-function serial ch10 serial data output(1) ADC AN41 input Input capture ch6 input(0) PPG ch3 output(1) Input capture ch3 input(1) Reload-timer ch6 output(1) INT13 ext.interrupt input(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
23	28	34	40	48	58	P062 SCS10_1 SCS40_0 AN40 PPG4_1 FRCK0_0 TOT7_1 ZIN1_1	General-purpose I/O port Serial chip select10 output(1) Serial chip select40 output(0) ADC AN40 input PPG ch4 output(1) Free-run timer0 clock input(0) Reload-timer ch7 output(1) UFD counter ch1 ZIN input(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
	29	35	41	49	60	P063 SCS41_0 AN39 PPG5_1 FRCK1_0 BIN1_1	General-purpose I/O port Serial chip select41 output(0) ADC AN39 input PPG ch5 output(1) Free-run timer1 clock input(0) UFD counter ch1 BIN input(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					61	P183 PPG43_0	General-purpose I/O port PPG ch43 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
24	30	36	42	50	62	P064 SCS42_0 AN38 FRCK2_0 AIN1_1 PPG43_1	General-purpose I/O port Serial chip select42 output(0) ADC AN38 input Free-run timer2 clock input(0) UFD counter ch1 AIN input(1) PPG ch43 output(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		37	43	51	63	P065 SCS43_0 FRCK3_0 ZIN0_1 PPG44_1	General-purpose I/O port Serial chip select43 output(0) Free-run timer3 clock input(0) UFD counter ch0 ZIN input(1) PPG ch44 output(1)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					64	P184 PPG44_0	General-purpose I/O port PPG ch44 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					65	P185 PPG45_0	General-purpose I/O port PPG ch45 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
25	31	38	44	52	66	P066 SOT4_2 SCS3_0 AN37 FRCK4_0 BIN0_1	General-purpose I/O port Multi-function serial ch4 serial data output(2) Serial chip select3 output(0) ADC AN37 input Free-run timer4 clock input(0) UFD counter ch0 BIN input(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
	32	39	45	53	67	P067 AN36 FRCK5_0 AIN0_1	General-purpose I/O port ADC AN36 input Free-run timer5 clock input(0) UFD counter ch0 AIN input(1)	input: automotive output: CMOS B	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
		40	46	54	68	P070 ICU0_2	General-purpose I/O port Input capture ch0 input(2)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
26	33	41	47	55	69	P071 SCK4_2 AN35 ICU1_2 MCNCLK	General-purpose I/O port Multi-function serial ch4 clock I/O(2) ADC AN35 input Input capture ch1 input(2) clock monitor output pin	input: CMOS hysteresis output: CMOS G	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
27	34	42	48	56	70	P072 SIN4_0 AN34 ICU2_2 INT5_0	General-purpose I/O port Multi-function serial ch4 serial data input(0) ADC AN34 input Input capture ch2 input(2) INT5 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS G	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
	35	43	49	57	71	P073 SOT4_0/SDA4 AN33 ICU3_2	General-purpose I/O port Multi-function serial ch4 serial data output(0) / I2C bus serial data I/O ADC AN33 input Input capture ch3 input(2)	input: I2C hysteresis output: CMOS D	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					72	P186 PPG46_0	General-purpose I/O port PPG ch46 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	Hi-Z/input blocked(*5)	Last status retained	Hi-Z/input blocked(*5)	yes
					73	P187 PPG47_0	General-purpose I/O port PPG ch47 output(0)	input: automotive output: CMOS A	Hi-Z/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	Hi-Z/input blocked	Hi-Z/input blocked	Status before entering shutdown is

Pin No (QFP100)	Pin No (QFP112)	Pin No (LQFP120)	Pin No (LQFP144)	Pin No (LQFP176)	Pin name	Pin function	Input/level Output/level Circuit type	Pin status after reset factor generation and after reset releasing												Pin status in the sleep and standby mode								RB Error
								Reset factor:1(*1) ([MB91F52xxB/MB91F52xxC/ MB91F52xxE] (*1 Power-on reset, external reset mode enable) ([MB91F52xxD] (*1 Power-on reset, external reset)		Reset factor:2(*2) (*2 Low-voltage detect (external power supply voltage))				Reset factor:3(*3) ([MB91F52xxB/MB91F52xxC/MB91F52xxE] External reset NMI disabled				Internal reset factor:4(*4)	Internal reset factor:5(*1-4)	Sleep mode		Stop mode		Watch mode		
								External factor generate	External factor after releases	External factor generate		External factor After Releases		External factor generate		External factor After Releases											
								Internal reset is occurs in progress	Write internal reset(0) After releases (oscillation stabilization wait is progress)	Before internal reset occurrence	Internal reset is occurs in progress	Before internal reset occurrence	Write internal reset(0) After releases (oscillation stabilization wait is progress)	Before internal reset occurrence	Internal reset is occurs in progress	Before internal reset occurrence	Write internal reset(0) After releases (oscillation stabilization wait is progress)	After internal reset releasing	After internal reset releasing	SLVL1 ([3:2]=2'b00)	SLVL0 ([1:0]=2'b00)	SLVL1 ([3:2]=2'b00)	SLVL0 ([1:0]=2'b00)	SLVL1 ([3:2]=2'b00)	SLVL0 ([1:0]=2'b00)	
		50	58	74	P074 SCK4_0/SCL4	General-purpose I/O port Multi-function serial ch4 clock I/O(0) / I2C bus serial clock I/O	input: I2C hysteresis output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		51	59	75	P075 SIN3_0 INT4_0	General-purpose I/O port Multi-function serial ch3 serial data input(0) INT4 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		52	60	76	P076 SOT3_0/SDA3	General-purpose I/O port Multi-function serial ch3 serial data output(0) / I2C bus serial data I/O	input: I2C hysteresis output: CMOS E	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		53	61	77	P077 SCK3_0/SCL3	General-purpose I/O port Multi-function serial ch3 clock I/O(0) / I2C bus serial clock I/O	input: I2C hysteresis output: CMOS E	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		44	54	62	78	P152 SCS3_0	General-purpose I/O port Serial chip select53 output(0)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
28	36	45	55	63	79	P153 SCK5_0/SCL5 ANI2 FRCK4_1 INT4_1	General-purpose I/O port Multi-function serial ch5 clock I/O(0) / I2C bus serial clock I/O ADC AIN2 input Free-run timer1 clock input(1) INT4 ext.interrupt input(1)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		54	62	80	P080 SCS52_0 PPG0_0	General-purpose I/O port Serial chip select52 output(0) PPG ch0 output(0)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
29	37	46	56	65	81	P081 SOT5_0/SDA5 AN8 PPG1_0	General-purpose I/O port Multi-function serial ch5 serial data output(0) / I2C bus serial data I/O ADC AN8 input PPG ch1 output(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
30	38	47	57	66	82	P082 SIN5_0 AN9 PPG2_0	General-purpose I/O port Multi-function serial ch5 serial data input(0) ADC AN9 input PPG ch2 output(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		67	83	97	P083 SCS50_0 AN2 PPG3_0	General-purpose I/O port Serial chip select50 output(0) ADC AN2 input PPG ch3 output(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		68	84	98	P084 SCS51_0 AN3 PPG4_0	General-purpose I/O port Serial chip select51 output(0) ADC AN3 input PPG ch4 output(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		69	85	99	P085 PPG5_0	General-purpose I/O port PPG ch5 output(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		48	58	70	86	P086 DAO1 PPG6_0	General-purpose I/O port DAC AN1 output PPG ch6 output(0)	input: automotive output: CMOS C	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
31	39	49	59	71	87	P087 DAO0 PPG7_0 INT8_0	General-purpose I/O port DAC AN0 output PPG ch7 output(0) INT8 ext.interrupt input(0)	input: automotive output: CMOS C	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
32	40	50	60	72	88	VCC	+5.0v power supply	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
33	41	51	61	73	89	VSS	GND	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
					90	P190 TIN0_1	General-purpose I/O port Reload-timer ch0 eventinput(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					91	P191 TIN1_1	General-purpose I/O port Reload-timer ch1 eventinput(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				74	92	P050 AN4 IC1I_0 TIN2_1	General-purpose I/O port ADC AN4 input Input capture ch0 input(0) Reload-timer ch2 eventinput(1)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				75	93	P051 AN5 PPG41_1 IC1I_0 TIN3_1	General-purpose I/O port ADC AN5 input PPG ch41 output(1) Input capture ch1 input(0) Reload-timer ch3 eventinput(1)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				76	94	P052 AN6 PPG40_1 IC1I_2 TOT0_1	General-purpose I/O port ADC AN6 input PPG ch40 output(1) Input capture ch2 input(0) Reload-timer ch0 output(1)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				95	P192 PPG24_1 TOT1_1	General-purpose I/O port PPG ch24 output(1) Reload-timer ch1 output(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
34	42	52	62	77	96	P093 TIN1_1 SIN11_0 AN7 ICU4_2 PPG16_1 ICU3_0 TOT2_1	General-purpose I/O port CAN transmission data0 output(1) Multi-function serial ch11 serial data input(0) ADC AN7 input Input capture ch4 input(2) PPG ch16 output(1) Input capture ch3 input(0) Reload-timer ch2 output(1)	5V tolerant input input: CMOS hysteresis output: CMOS J	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				78	97	P094 ANI ICU4_0 TOT3_1	General-purpose I/O port ADC AN8 input Input capture ch4 input(0) Reload-timer ch3 output(1)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		53	63	79	98	P095 TX0(128) SCS11_0 AN8	General-purpose I/O port CAN transmission data0 output Serial chip select11 output(0) ADC AN8 input	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
35	43	54	64	80	99	P096 RX0(128) SOT11_0/SDA11 A11 AN10 INT0_0	General-purpose I/O port CAN reception data0 input Multi-function serial ch11 serial data output(0) / I2C bus serial data input ADC AN10 input INT0 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
36	44	55	65	81	100	P097 SCK11_0/SC L11 AN11 ICU5_0 PPG17_1	General-purpose I/O port Multi-function serial ch11 clock I/O(0) / I2C bus serial clock input ADC AN11 input Input capture ch5 input(0) PPG ch17 output(1)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
37	45	56	66	82	101	AVSS0/AVRL0	GND for ADC/DAC A/D converter lower limit reference voltage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	46	57	67	83	102	AVRH0	A/D converter upper limit reference voltage	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	47	58	68	84	103	AVCC0	Analog power supply for ADC0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	48	59	69	85	104	P100 SCK7_0/SCL7 AN12 PPG8_0	General-purpose I/O port Multi-function serial ch7 clock I/O(0) / I2C bus serial clock I/O ADC AN12 input PPG ch8 output(0)	input: CMOS hysteresis output: CMOS D	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		60	70	86	105	P101 SOT7_0/SDA7 AN13 PPG9_0	General-purpose I/O port Multi-function serial ch7 serial data output(0) / I2C bus serial data I/O ADC AN13 input PPG ch9 output(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
40	49	61	71	87	106	P102 SIN7_0 AN14 PPG10_0 INT10_0	General-purpose I/O port Multi-function serial ch7 serial data input(0) ADC AN14 input PPG ch10 output(0) INT10 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
41	50	62	72	88	107	P103 SCS73_0 AN15 PPG11_0	General-purpose I/O port Serial chip select73 output(0) ADC AN15 input PPG ch11 output(0)	large current port 12mA input: automotive output: CMOS H	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
42	51	63	73	89	108	P104 SCS72_0 AN16 PPG12_0	General-purpose I/O port Serial chip select72 output(0) ADC AN16 input PPG ch12 output(0)	large current port 12mA input: automotive output: CMOS H	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
43	52	64	74	90	109	P105 SCS71_0 AN17 PPG13_0	General-purpose I/O port Serial chip select71 output(0) ADC AN17 input PPG ch13 output(0)	large current port 12mA input: automotive output: CMOS H	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		65	75	91	110	P106 SCS70_0 AN18 PPG14_0	General-purpose I/O port Serial chip select70 output(0) ADC AN18 input PPG ch14 output(0)	large current port 12mA input: automotive output: CMOS H	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		53	66	76	92	111	P107 AN19 PPG15_0	General-purpose I/O port ADC AN19 input PPG ch15 output(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					112	P193 PPG25_1	General-purpose I/O port PPG ch25 output(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		77	93	113	P154 AN20	General-purpose I/O port ADC AN20 input	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		78	94	114	P155 AN21	General-purpose I/O port ADC AN21 input	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
44	54	67	79	95	115	NMIX	No mask interrupt input	input: CMOS hysteresis M	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	
45	55	68	80	96	116	P110 TX1(64) SCS63_0 AN22	General-purpose I/O port CAN transmission data1 output Serial chip select63 output(0) ADC AN22 input	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		69	81	97	117	P111 RX1(64) SCS62_0 AN23 INT1_0	General-purpose I/O port CAN reception data1 input Serial chip select62 output(0) ADC AN23 input INT1 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
			82	98	118	P112 AN24 PPG18_0 RTO0_0	General-purpose I/O port ADC AN24 input PPG ch18 output(0) Waveform generator ch0 output pin(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
			83	99	119	P113 AN25 PPG17_0 RTO1_0	General-purpose I/O port ADC AN25 input PPG ch17 output(0) Waveform generator ch1 output pin(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					120	P194 FRCK5_1 PPG26_1	General-purpose I/O port Free-run timer5 clock input(1) PPG ch26 output(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					121	P195 FRCK4_1 PPG27_1	General-purpose I/O port Free-run timer4 clock input(1) PPG ch27 output(1)	input: automotive output: CMOS A	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		56	70	84	100	P114 SCS61_0 AN26 PPG18_0 RTO2_0	General-purpose I/O port Serial chip select61 output(0) ADC AN26 input PPG ch18 output(0) Waveform generator ch2 output pin(0)	input: automotive output: CMOS B	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
46	57	71	85	101	123	P115 RX1_1 SCK9_0/SDA6 AN27 PPG19_0 RTO3_0 INT1_1	General-purpose I/O port CAN reception data1 input(1) Multi-function serial ch6 serial data output(0) / I2C bus serial data I/O ADC AN27 input PPG ch19 output(0) Waveform generator ch3 output pin(0) INT1 ext.interrupt input(1)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ input enabled	HiZ input blocked	HiZ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes

Pin No (64-Pin)	Pin No (80-Pin)	Pin No (100-Pin)	Pin No (128-Pin)	Pin No (144-Pin)	Pin No (176-Pin)	Pin name	Pin function	Input/level Output/level Circuit type	Pin status after reset factor generation and after reset releasing												Pin status in the sleep and standby mode						Reference to analog characteristics
									Reset factor1(*1) [MB91FS2xxB/MB91FS2xxC/ MB91FS2xxE] (*1 Power-on reset, external reset NMI enable) [MB91FS2xxD] (*1 Power-on reset, external reset)		Reset factor2(*2) (*2 Low-voltage detect (external power supply voltage))				Reset factor3(*3) [MB91FS2xxB/MB91FS2xxC/MB91FS2xxE] External reset NMI disabled				Internal reset factor4(*4)	Internal reset factor5(*14)	Sleep mode		Stop mode		Watch mode		
									External factor generate	External factor after releases	External factor generate		External factor After releases		External factor generate		External factor After releases										
									Internal reset issuance in progress	Within internal reset to after (oscillation stabilization wait completion)	Before external reset is source	Internal reset issuance in progress	Before external reset is source	Within internal reset to after releases (oscillation stabilization wait completion)	Before external reset is source	Internal reset issuance in progress	Before external reset is source	Within internal reset to after releases (oscillation stabilization wait completion)	Within internal reset to after (oscillation stabilization wait completion)	After internal reset releasing	SLVI.EL=[]=0x30x	SLVI.EL=[]=0x31x	SLVI.EL=[]=0x30x	SLVI.EL=[]=0x31x	SLVI.EL=[]=0x30x	SLVI.EL=[]=0x31x	
47	58	72	86	102	124	P116 SCK6_0/SCL6 AN26 PPG20_0 RTO4_0	General-purpose I/O port Multi-function serial ch6 clock I/O(0) / I2C bus serial clock input ADC AN26 input PPG ch20 output(0) Waveform generator ch4 output pin(0)	input: CMOS hysteresis output: CMOS G	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		73	87	103	125	P117 SCS60_0 AN29 PPG21_0 RTO5_0	General-purpose I/O port Serial chip select60 output(0) ADC AN29 input PPG ch21 output(0) Waveform generator ch5 output pin(0)	input: automotive output: CMOS B	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					126	P196 FRCK3_1 PPG28_1	General-purpose I/O port Free-run timer3 clock input(1) PPG ch28 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		88	104	127		P120 AN30 OCU6_0 PPG22_0 INT9_0	General-purpose I/O port ADC AN30 input Output compare ch6 output(0) PPG ch22 output(0) INT9 ext.interrupt input(0)	input: automotive output: CMOS B	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				105	128	P121 OCU7_0 PPG23_0	General-purpose I/O port Output compare ch7 output(0) PPG ch23 output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
48	59	74	89	106	129	P122 SIN6_0 AN31 OCU8_0 INT9_1	General-purpose I/O port Multi-function serial ch6 serial data input(0) ADC AN31 input Output compare ch8 output(0) INT9 ext.interrupt input(1)	5V tolerant input input: CMOS hysteresis output: CMOS J	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					130	P197 PPG29_1	General-purpose I/O port PPG ch29 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				107	131	P123 OCU9_0	General-purpose I/O port Output compare ch9 output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
	60	75	90	108	132	GND	GND	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	61	76	91	109	133	VCC	+5.0v power supply	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
49	62	77	92	110	134	DEBUGIF	Debugger(OCD) DEBUGIF I/O	Nch open drain input: TTL output: 25mA(NOD) L	HiZ/input enabled	input enabled	input disabled (*11)	input disabled	input enabled (*12)	input disabled	input enabled (*12)	input disabled (*11)	input disabled	input enabled (*12)	input disabled (*11)	input enabled (*12)	input disabled (*11)	input enabled (*12)	input disabled (*11)	input disabled(*11)	input disabled(*11)	input disabled(*11)	input disabled(*11)	input disabled(*11)	input disabled(*11)	-
					135	P160 PPG30_1	General-purpose I/O port PPG ch30 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					136	P161 PPG31_1	General-purpose I/O port PPG ch31 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				111	137	P124 OCU10_0	General-purpose I/O port Output compare ch10 output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		93	112	138		P125 OCU11_0	General-purpose I/O port Output compare ch11 output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
50	63	78	94	113	139	P126 SIN0_0 INT8_0	General-purpose I/O port Multi-function serial ch0 serial data input(0) INT8 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS F	HiZ/input enabled		Hi-Z/ input enabled	Hi-Z/input enabled	Hi-Z/input enabled	HiZ/input enabled	Hi-Z/ input enabled	Hi-Z/input enabled	Hi-Z/ input enabled	Hi-Z/ input enabled	HiZ/input enabled	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	no
	64	79	95	114	140	P127 SOT0_0	General-purpose I/O port Multi-function serial ch0 serial data output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		80	96	115	141	P130 SCK0_0	General-purpose I/O port Multi-function serial ch0 clock I/O(0)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					142	P162 TRG5_2	General-purpose I/O port PPG trigger5 input(2)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					143	P163 TRG6_2	General-purpose I/O port PPG trigger6 input(2)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
51	65	81	97	116	144	MD0	Mode pin 0	input: CMOS hysteresis K	input enabled		input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	-
52	66	82	98	117	145	MD1	Mode pin 1	input: CMOS hysteresis K	input enabled		input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	-
53	67	83	99	118	146	X0	Main clock oscilation input	N	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
54	68	84	100	119	147	X1	Main clock oscilation ouput	N	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
55	69	85	101	120	148	VSS	GND	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
56	70	86	102	121	149	P135 OTS_0	General-purpose I/O port Waveform generator ch0-ch5 input(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
						X1A (*13)	Subclock oscilation output	O	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	71	87	103	122	150	P136	General-purpose I/O port	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
						X0A (*13)	Subclock oscilation input	G	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	72	88	104	123	151	RSTX	External reset input	input: CMOS hysteresis M	input enabled		input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	input enabled	-
				124	152	P131 ADTG0_0	General-purpose I/O port ADC external trigger0 input(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		105	125	153		P132 SCS1_0 ADTG1_0	General-purpose I/O port Serial chip select1 output(0) ADC external trigger1 input(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		89	106	126	154	P133 TX2(64)	General-purpose I/O port CAN transmission data2 output	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		90	107	127	155	P134 RX2(64) SCS1_1 ICU7_0 INT7_0	General-purpose I/O port CAN reception data2 input Serial chip select1 output(1) Input capture ch7 input(0) INT7 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		91	108	128	156	P144 SCK1_1	General-purpose I/O port Multi-function serial ch1 clock I/O(1)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
59	73	92	109	129	157	VSS	GND	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	74	93	110	130	158	C	External capacitor connect output	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		94	111	131	159	P000 SI16 SIN3_0 TOA0_1 INT2_0	General-purpose I/O port External bus data bit16 I/O(0) Multi-function serial ch1 serial data input(0) Base timer ch0 TIOA I/O(1) INT2 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		75	95	112	132	P001 SI17 SOT1_0 TOA1_1	General-purpose I/O port External bus data bit17 I/O Multi-function serial ch1 serial data output(0) Base timer ch1 TIOA I/O(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		113	133	161		P002 SI18 SCK1_0 TIOB0_1	General-purpose I/O port External bus data bit18 I/O Multi-function serial ch1 clock I/O(0) Base timer ch0 TIOB I/O(1)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
		76	96	114	160	P003 SI19 SIN2_0 TIOB1_1 INT3_0	General-purpose I/O port External bus data bit19 I/O Multi-function serial ch2 serial data input(0) Base timer ch1 TIOB I/O(1) INT3 ext.interrupt input(0)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				135	161	P004 SO20 SOT2_0	General-purpose I/O port External bus data bit20 I/O(0) Multi-function serial ch2 serial data output(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				164	162	P164 PPG32_1	General-purpose I/O port PPG ch32 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
61	77	97	115	136	165	P005 SI21 SCK2_0 ADTG2_1 INT7_1	General-purpose I/O port External bus data bit21 I/O(0) Multi-function serial ch2 clock I/O(0) ADC external trigger input(1) INT7 ext.interrupt input(1)	input: CMOS hysteresis output: CMOS F	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					166	P165 PPG33_1	General-purpose I/O port PPG ch33 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
62	78	98	116	137	167	P006 SI22 SCS2_0 ADTG3_1 INT2_1 (TX2(64))	General-purpose I/O port External bus data bit22 I/O(0) Serial chip select2 output(0) ADC external trigger input(1) INT2 ext.interrupt input(1) (CAN transmission data2 output, only 64pin or 80pin)	input: automotive output: CMOS A	HiZ/input enabled		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input enabled	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input enabled	HiZ/input enabled	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	no
			117	138	168	P007 SI23	General-purpose I/O port External bus data bit23 I/O	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					169	P166 PPG34_1	General-purpose I/O port PPG ch34 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
			118	139	170	P010 SI24	General-purpose I/O port External bus data bit24 I/O	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
63	79	99	119	140	171	P011 WOT SI25 SOT2_1 TIOA0_2 INT3_1	General-purpose I/O port RTC output External bus data bit25 I/O Multi-function serial ch2 serial data output(1) Base timer ch0 TIOA I/O(0) INT3 ext.interrupt input(1)	input: automotive output: CMOS A	HiZ/input enabled		Hi-Z/ input enabled	HiZ/ input blocked	Hi-Z/ input enabled	HiZ/input enabled	Hi-Z/ input enabled	HiZ/ input blocked	Hi-Z/ input enabled	HiZ/input enabled	HiZ/input enabled	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	no
				141	172	P012 SI26 TIOB0_2	General-purpose I/O port External bus data bit26 I/O Base timer ch0 TIOB I/O(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
					173	P167 PPG35_1	General-purpose I/O port PPG ch35 output(1)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				142	174	P013 SI27 TIOA1_0	General-purpose I/O port External bus data bit27 I/O Base timer ch1 TIOA I/O(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
				143	175	P014 SI28 TIOB1_0	General-purpose I/O port External bus data bit28 I/O Base timer ch1 TIOB I/O(0)	input: automotive output: CMOS A	HiZ/input blocked		Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	Hi-Z/ input enabled	Hi-Z/ input blocked	Hi-Z/ input enabled	HiZ/input blocked	HiZ/input blocked	Status before entering shutdown is retained.	Last status retained(*7)	Last status retained	Last status retained	HiZ/input blocked(*5)	Last status retained	HiZ/input blocked(*5)	yes
64	80	100	120	144	176	VCC	+5.0v power supply	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(*1) Factor: [MB91FS2xxB/MB91FS2xxC/MB91FS2xxE] power-on reset, low-voltage detect (internal low-voltage detection), external reset (NMI enable) .
[MB91FS2xxD] power-on reset, low-voltage detect (internal low-voltage detection), external reset.
(*2) Factor: low-voltage detection (external power supply voltage).
(*3) Factor: [MB91FS2xxB/MB91FS2xxC/MB91FS2xxE] external reset (NMI disabled)
(*4) Factor: software reset, soft/hard wear watchdog reset, return form stand-by where the I/O state is maintained until returning from the standby mode (include time out)
(*5) At ext.interrupt is enable, input blocked is disabled. (setting to EINR, EPFR bit for corespond to ext.interrupt channel, only inflense settting of PORTEN pin, clearing GPORTEN)
(*7) Operation is continued according to the peripheral function.
(*11) Emulation mode: input enabled, free-run mode: input disabled
(*12) Only INIT level reset is issue input enabled, other input disabled
(*13) Single clock device is P135/P136, dual clock device is XLA/X0A.
(*14) Factor: return from stand-by (power supply blocked) where the I/O state is maintained until the I/O maintain cancellation request has been cleared.

D. Major Changes

Major Changes are shown.

Major Changes

Spansion Publication Number: MB91F526_MN705-00010

Page	Section	Change Results
Revision 4.0		
i	Preface	The preface should be modified as follows: (Error) Thank you for your continued use of Fujitsu semiconductor products. Read this manual and "Data Sheet" thoroughly before using products in the MB91520 series. (Correct) Thank you for your continued use of Spansion semiconductor products. Read this manual and "MB91520 Series Data Sheet" thoroughly before using the products in the MB91520 series.
5	1.2.2	A sentence in the clock generation description (equipped with SSCG function) in "2.2 Peripheral Functions" should be modified as follows: (Error) · Clock generation (equipped with SSCG function) · Main oscillation (4MHz to 16MHz) · Sub oscillation (32kHz to 100kHz) or no sub oscillation · PLL multiplication rate : 1 to 20 times (Correct) · Clock generation (equipped with SSCG function) · Main oscillation (4MHz to 16MHz) · Sub oscillation (32kHz) or no sub oscillation · PLL multiplication rate : 1 to 20 times · Equipped with a 100kHz CR oscillator

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>7</td><td>1.2.2</td><td>The description in the base timer in "2.2. Peripheral Functions" should be modified as follows:

(Error)
· Base timer : Max. 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.

(Correct)
· Base timer : Max. 2 channels
· 16-bit timer
· The timer mode is selected from PWM/PPG/PWC/reload.
· As for the PWC function and the reload timer function, a pair of 16-bit timers can be used as one 32-bit timer in the cascaded mode.</td></tr>
<tr><td>8</td><td>1.3</td><td>A sentence in the multi-function serial description in the Table 3-1 Product Line-up (64 pin) in "3. Product Line-up" should be modified as follows:

(Error)
<table><tr><td>Multi-Function Serial</td><td>8 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>8 channels*1</td></tr></table></td></tr>
<tr><td>9</td><td>1.3</td><td>The following sentence should be added under Table 3-1 Product Line-up (64 pin) in "3. Product Line-up".

(Correct)
*1: Only channel 5, channel 6 and channel 11 support the I^{2}C (standard mode).</td></tr>
<tr><td>10</td><td>1.3</td><td>A sentence in the multi-function serial description in Table 3-2 Product Line-up (80 pin) in "3. Product Line-up" should be modified as follows:

(Error)
<table><tr><td>Multi-Function Serial</td><td>9 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>9 channels*1</td></tr></table></td></tr>
<tr><td>11</td><td>1.3</td><td>The following sentence should be added under Table 3-2 Product Line-up (80 pin) in "3. Product Line-up":

(Correct)
*1: Only channel 5, channel 6 and channel 11 support the I^{2}C (standard mode).</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>12</td><td>1.3</td><td>A sentence in the multi-function serial description in Table 3-3 Product Line-up (100 pin) in "3. Product Line-up" should be modified as follows:

(Error)
<table><tr><td>Multi-Function Serial</td><td>12 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>12 channels*1</td></tr></table></td></tr>
<tr><td>13</td><td>1.3</td><td>The following sentence should be added under Table 3-3 Product Line-up (100 pin) in "3. Product Line-up":

(Correct)
*1: Only channel 5, channel 6, channel 7, channel 8 and channel 11 support the I^{2}C (standard mode).</td></tr>
<tr><td>14</td><td>1.3</td><td>A sentence in the multi-function serial description in Table 3-4 Product Line-up (120 pin) in 3. Product Line-up should be modified as follows:

(Error)
<table><tr><td>Multi-Function Serial</td><td>12 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>12 channels*1</td></tr></table></td></tr>
<tr><td>14</td><td>1.3</td><td>The following sentences should be added under Table 3-4 Product Line-up (120 pin) in "3. Product Line-up":

(Correct)
*1: Only channel 3 and channel 4 support the I^{2}C (high-speed mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8 and channel 11 support the I^{2}C (standard mode).</td></tr>
<tr><td>15</td><td>1.3</td><td>A sentence in the multi-function serial description in Table 3-5 Product Line-up (144 pin) in "3. Product Line-up" should be modified as follows:

(Error)
<table><tr><td>Multi-Function Serial</td><td>12 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>12 channels*1</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>16</td><td>1.3</td><td>The following sentences should be added under Table 3-5 Product Line-up (144 pin) in "3. Product Line-up":

(Correct)
*1: Only channel 3 and channel 4 support the I^{2}C (high-speed mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8, channel 10 and channel 11 support the I^{2}C (standard mode).</td></tr>
<tr><td>17</td><td>1.3</td><td>A sentence in the multi-function serial description in Table 3-6 Product Line-up (176 pin) in "3. Product Line-up" should be modified as follows:

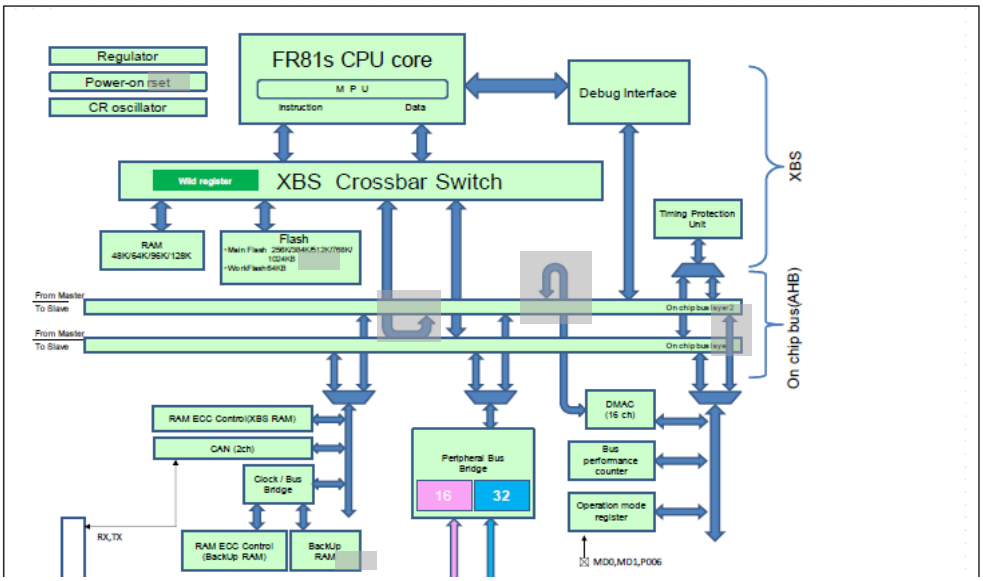
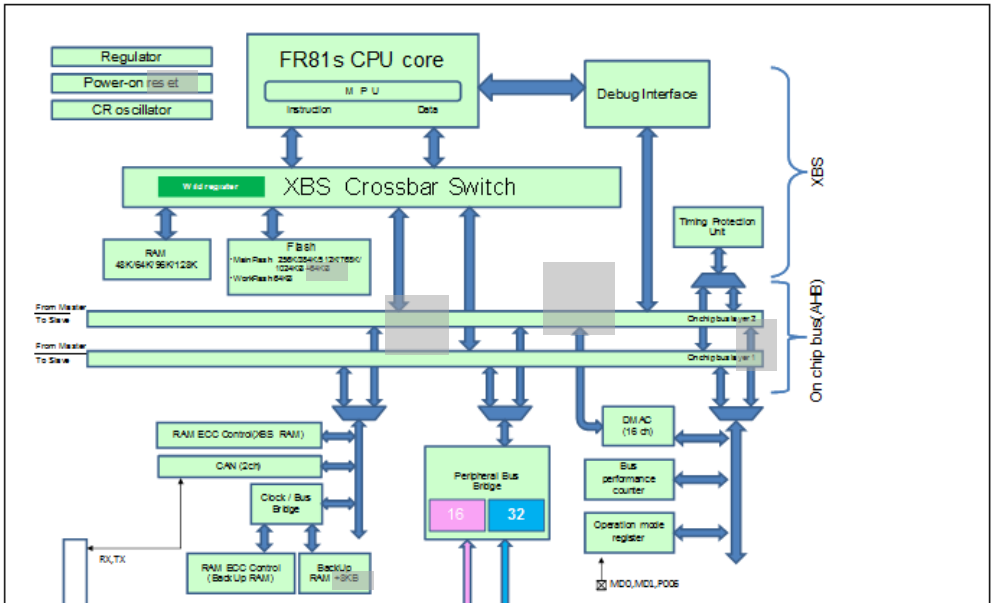
(Error)
<table><tr><td>Multi-Function Serial</td><td>12 channels</td></tr></table>
(Correct)
<table><tr><td>Multi-Function Serial</td><td>12 channels*1</td></tr></table></td></tr>
<tr><td>18</td><td>1.3</td><td>The following sentences should be added under Table 3-6 Product Line-up (176 pin) in "3. Product Line-up":

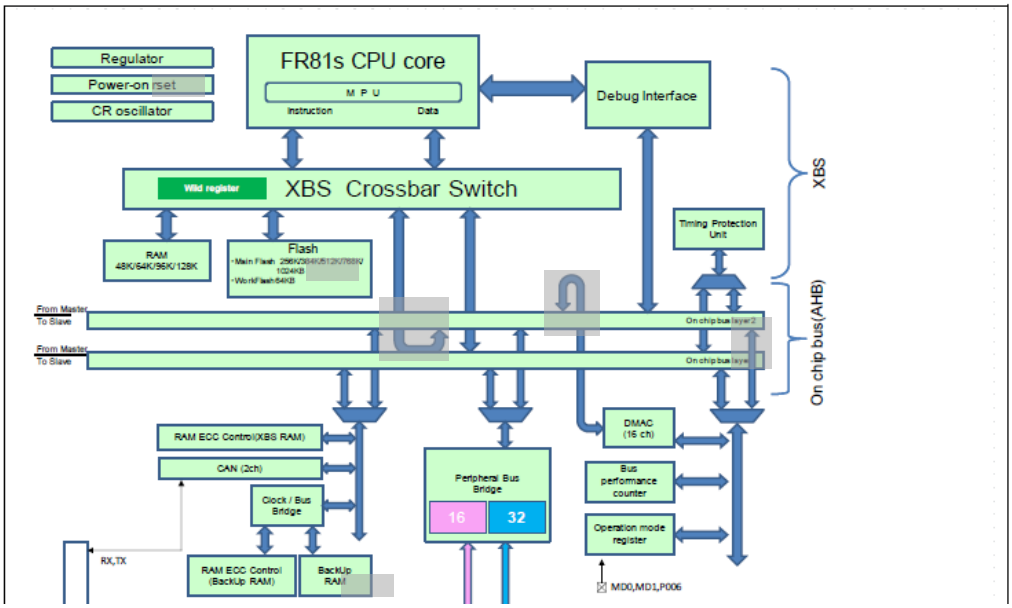
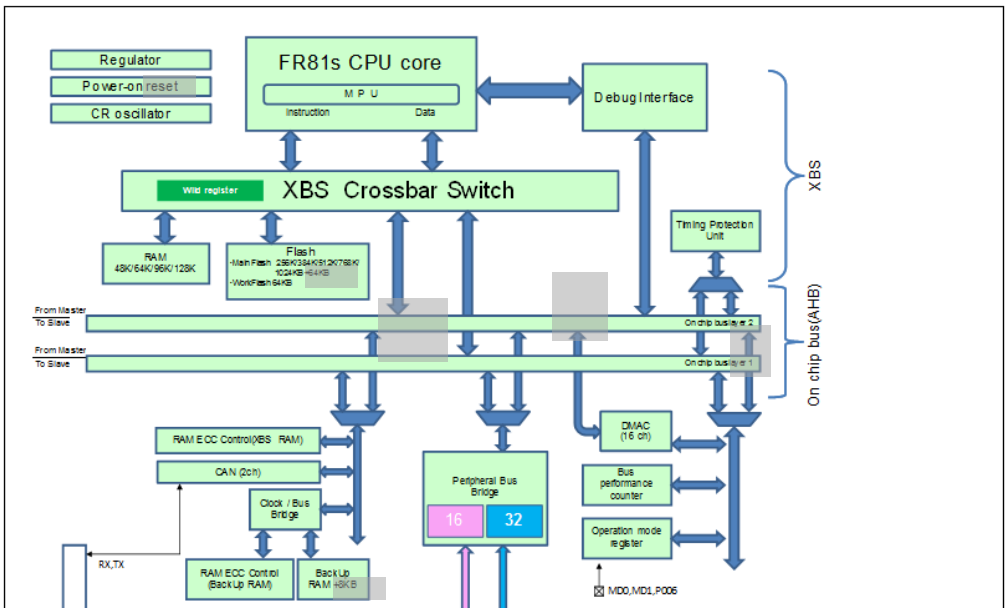
(Correct)
*1: Only channel 3 and channel 4 support the I^{2}C (high-speed mode/standard mode).
Only channel 5, channel 6, channel 7, channel 8, channel 10 and channel 11 support the I^{2}C (standard mode).</td></tr>
<tr><td>19</td><td>1.4</td><td>The description of the Clock in Table 4-1 Function overview in "4. Function overview" should be modified as follows:

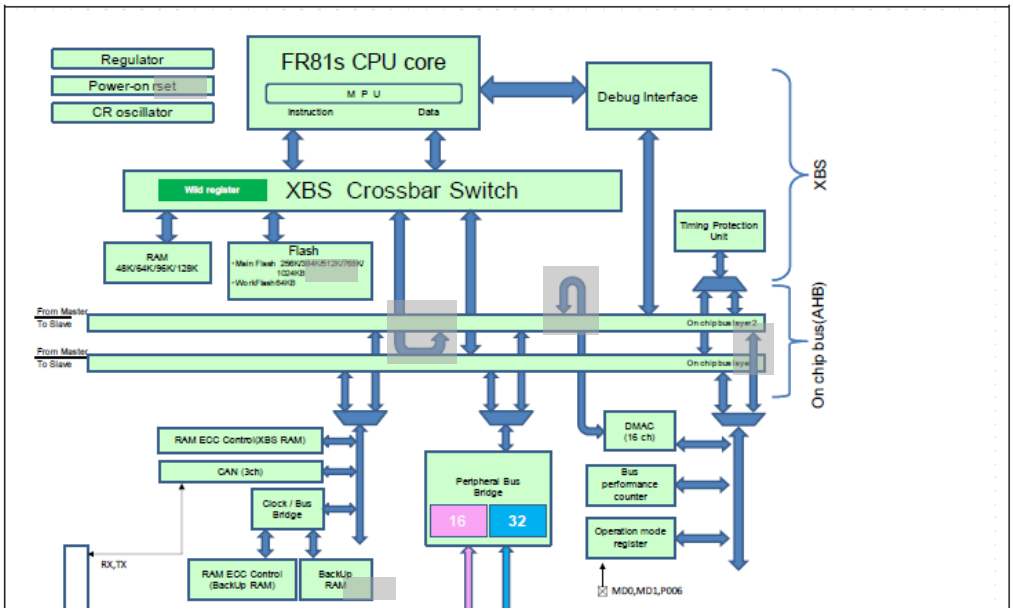
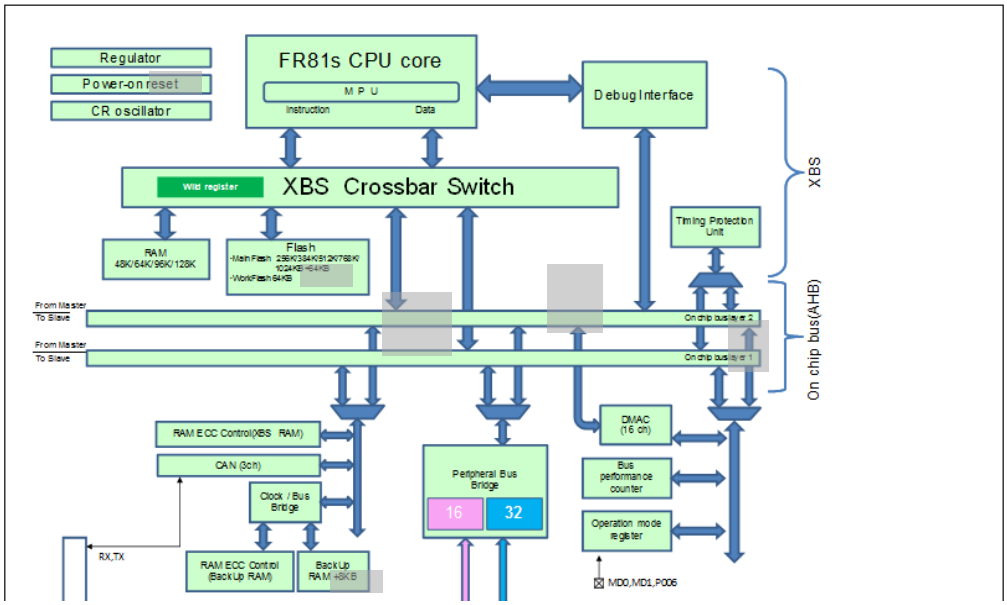
(Error)
Sub oscillation : 32kHz or 100kHz or None

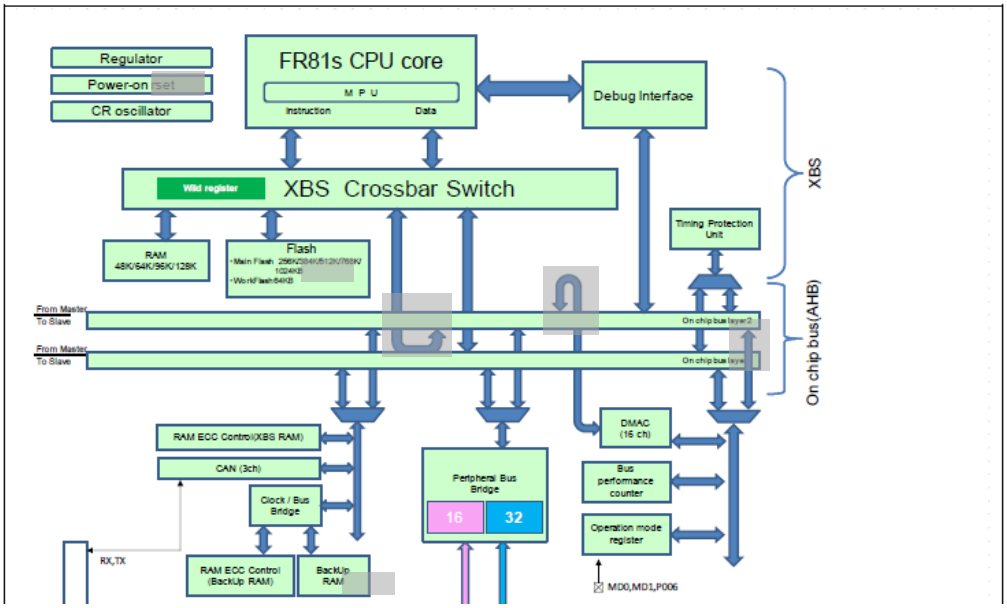
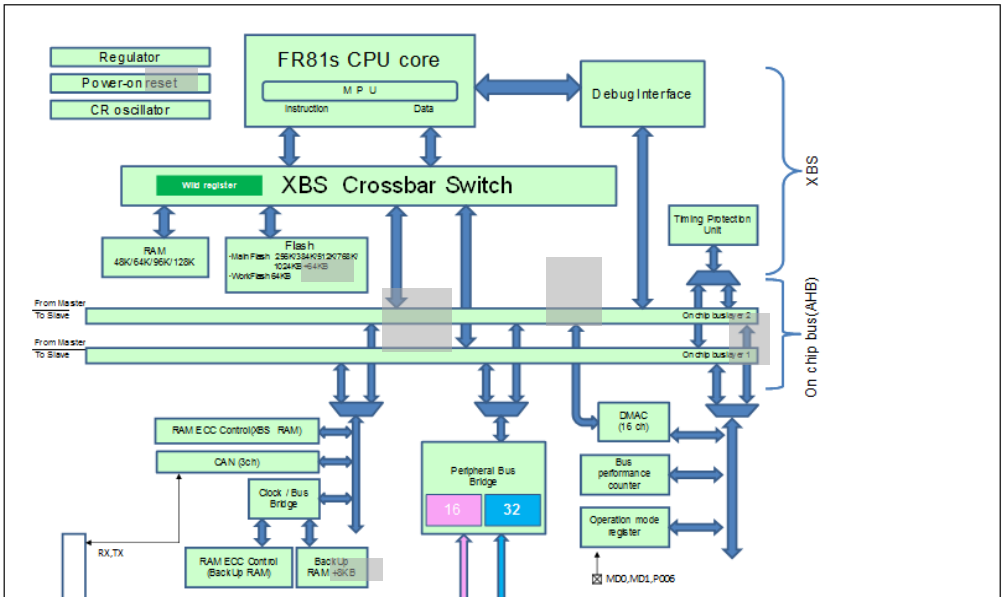
(Correct)
Sub oscillation : 32kHz or None</td></tr>
</table>

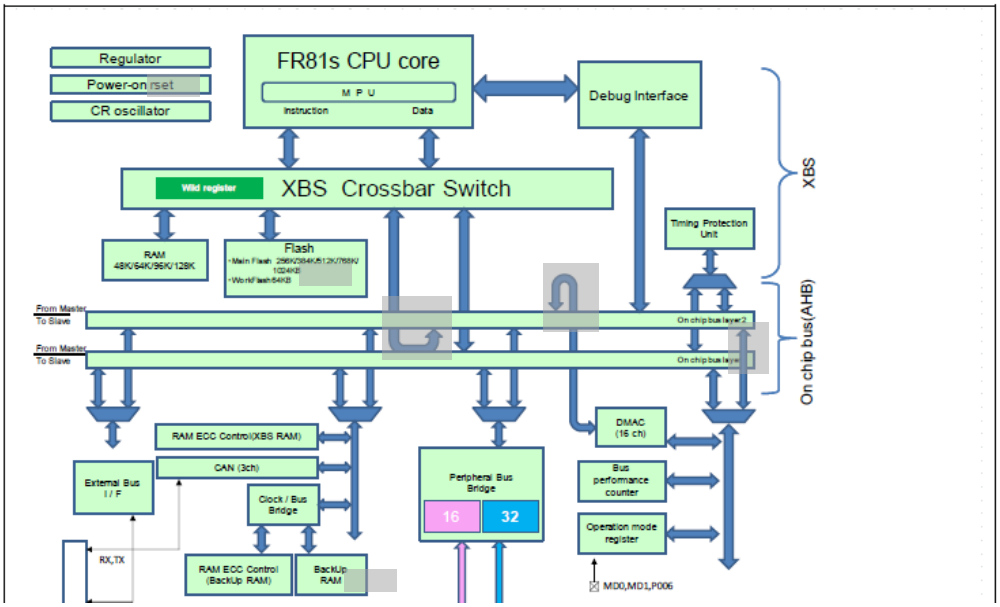
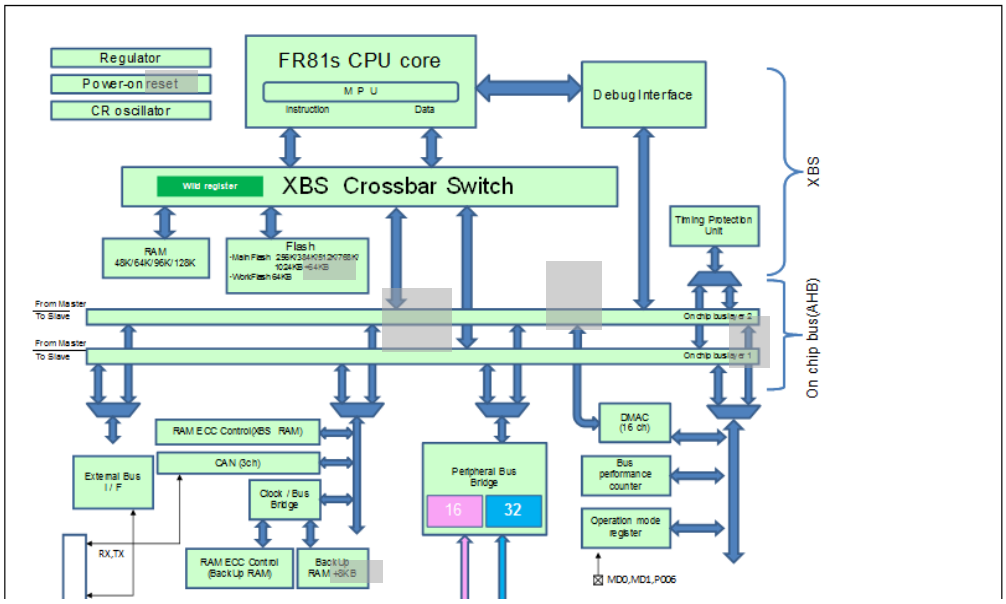
Page	Section	Change Results
22	1.4	The description of the I/O relocation in Table 4-1 Function overview in "4. Function overview" should be modified as follows: (Error) Multi-function serial (Max. 2 divergences for ch.0 to ch.2, ch.10; Max. 3 divergences for ch.3 and ch.4) (Correct) Multi-function serial (Max. 2 divergences for ch.0 to ch.2, ch.10; Max. 3 divergences for ch.3 and ch.4) Note that the I^2C cannot be relocated.

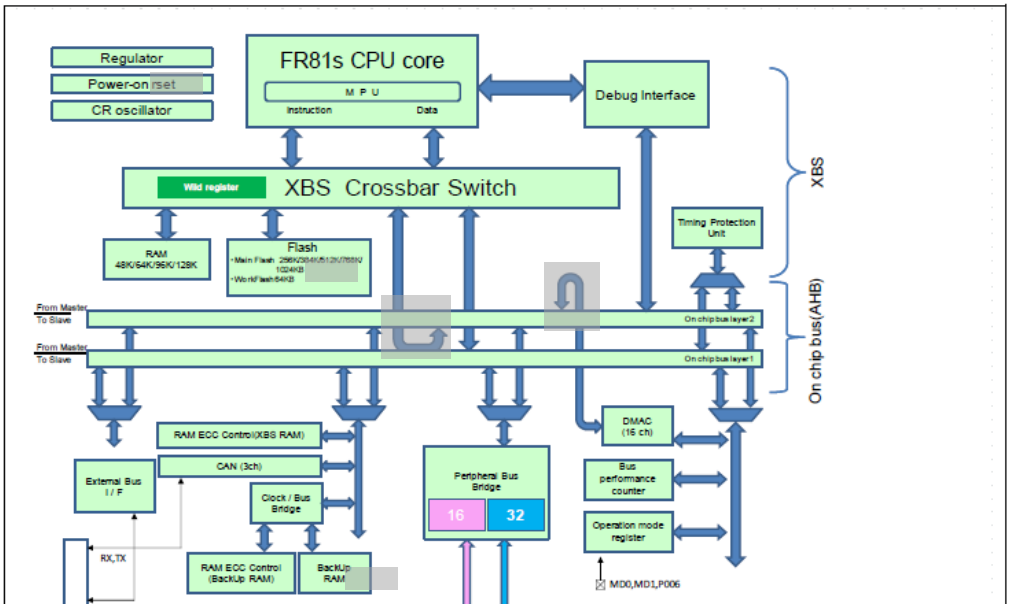
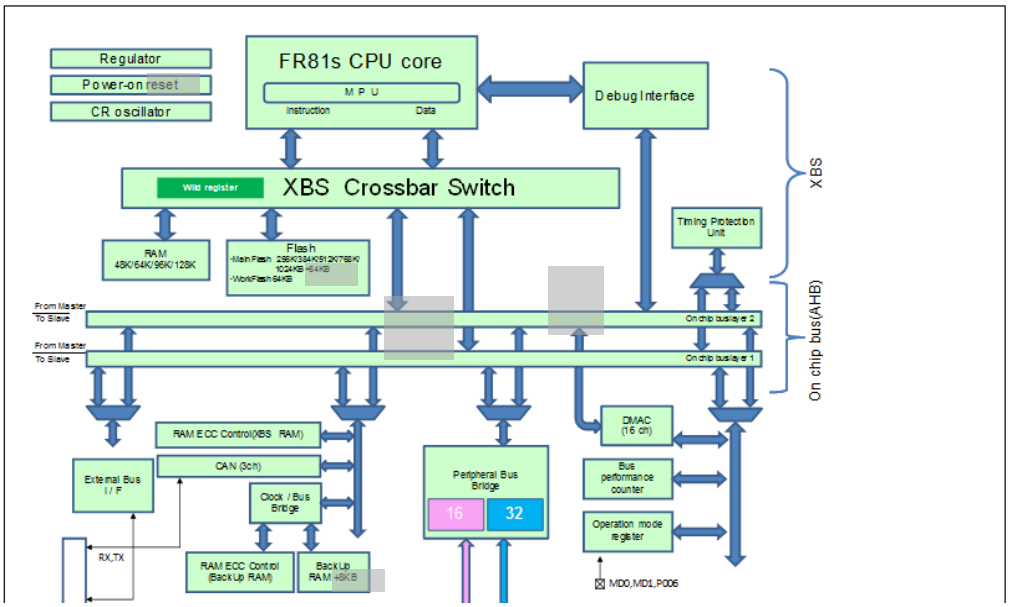
Page	Section	Change Results
23	1.5	Figure 5-1 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

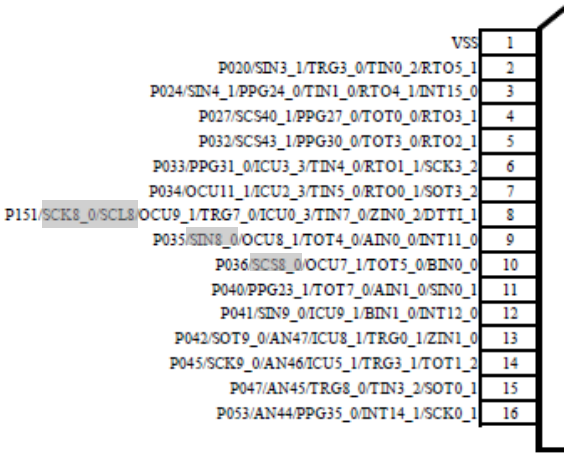
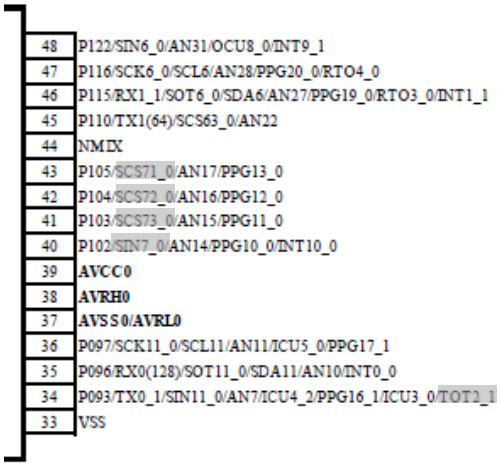
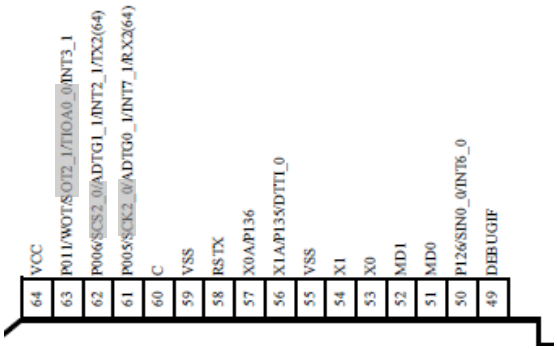
Page	Section	Change Results
24	1.5	Figure 5-2 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
25	1.5	Figure 5-3 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

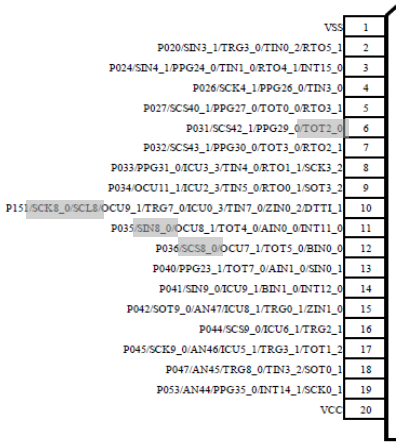
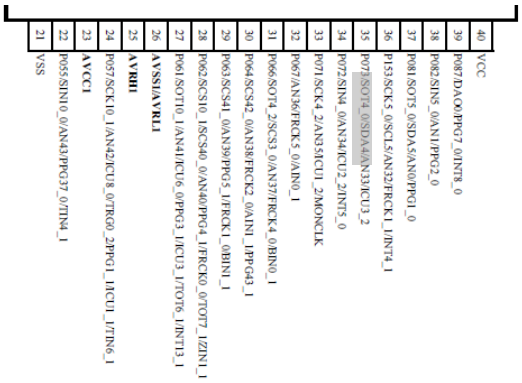
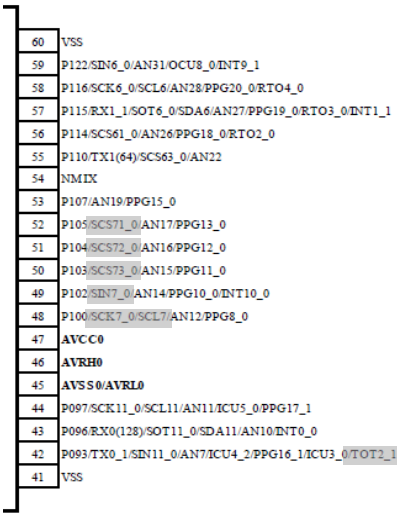
Page	Section	Change Results
26	1.5	Figure 5-4 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

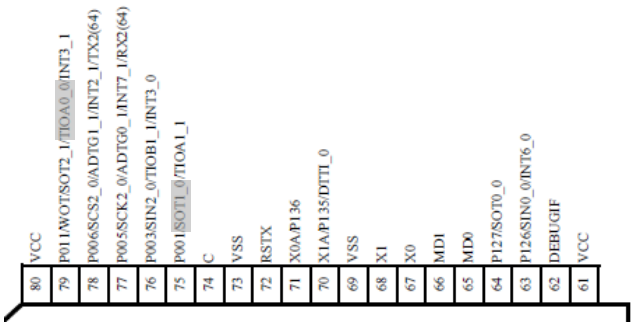
Page	Section	Change Results
27	1.5	Figure 5-5 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

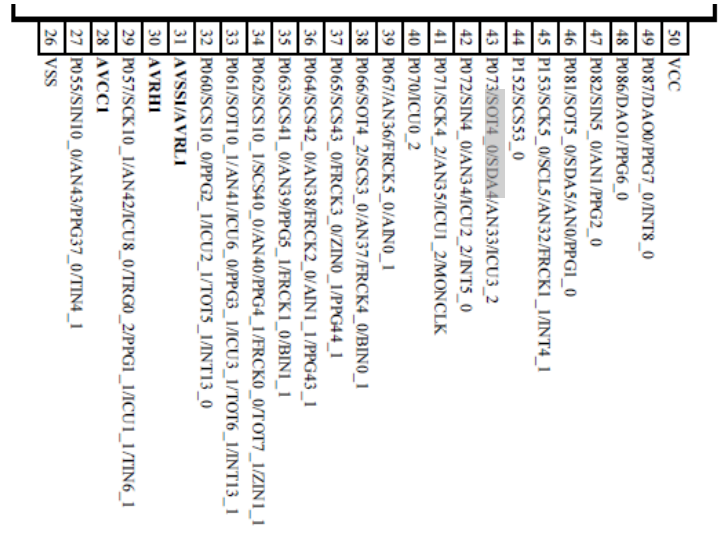
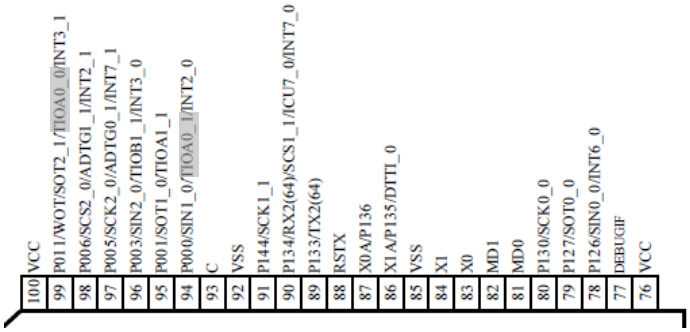
Page	Section	Change Results
28	1.5	Figure 5-6 in "5. Block Diagram" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
31	1.7	Signals indicated by the shading below should be deleted in Figure 7-1 in "7. Pin Assignment". - Left side  - Right side  - Top 

Page	Section	Change Results
31	1.7	The following note should be added on the bottom left of Figure 7-1 in "7. Pin Assignment". (Correct) * In a single clock product, pin 56 and pin 57 are the general-purpose ports.

Page	Section	Change Results
32	1.7	Signals indicated by the shading below should be deleted in Figure 7-2 in "7. Pin Assignment". - Left side  - Bottom  - Right side 

Page	Section	Change Results
32	1.7	(Continued) - Top 
32	1.7	The following note should be added on the bottom left of Figure 7-2 in "7. Pin Assignment". (Correct) * In a single clock product, pin 71 and pin 72 are the general-purpose ports.

Page	Section	Change Results
33	1.7	Signals indicated by the shading below should be deleted in Figure 7-3 in "7. Pin Assignment". - Bottom  - Top 
33	1.7	The following note should be added on the bottom left of Figure 7-3 in "7. Pin Assignment". (Correct) * In a single clock product, pin 86 and pin 87 are the general-purpose ports.
34	1.7	The following note should be added on the bottom left of Figure 7-4 in "7. Pin Assignment". (Correct) * In a single clock product, pin 102 and pin 103 are the general-purpose ports.

Page	Section	Change Results
35	1.7	The following note should be added on the bottom left of Figure 7-5 in "7. Pin Assignment". (Correct) * In a single clock product, pin 121 and pin 122 are the general-purpose ports.
36	1.7	The following note should be added on the bottom left of Figure 7-6 in "7. Pin Assignment". (Correct) * In a single clock product, pin 149 and pin 150 are the general-purpose ports.

Page	Section	Change Results
44	1.9	Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified. (Error)

Pin Number						Pin Name
64	80	100	120	64	176	
-	-	-	-	2	2	P015 D29 TRG0_0
-	-	-	-	3	3	P016 D30 TRG1_0
-	-	-	-	-	4	P170 PPG36_1
-	-	-	-	4	5	P017 D31 TRG2_0
-	-	-	-	-	6	P171 PPG37_1
2	2	2	2	5	7	P020 ASX SIN3_1 TRG3_0 TIN0_2 RTO5_1
-	-	-	3	6	8	P021 CS0X SOT3_1 TRG6_1 TRG4_0
-	-	-	4	7	9	P022 CS1X SCK3_1 TRG7_1 TRG5_0
-	-	-	5	8	10	P023 RDX SCS3_1 PPG32_0 TIN0_0
3	3	3	6	9	11	P024 WR0X SIN4_1 PPG24_0 TIN1_0 RTO4_1 INT15_0
-	-	4	7	10	12	P025 WR1X SOT4_1 PPG25_0 TIN2_0

Page	Section	Change Results
44	1.9	(Continued) (Correct)

Pin Number						Pin Name
64	80	100	120	64	176	
-	-	-	-	2	2	P015 D29 TRG0_0
-	-	-	-	3	3	P016 D30 TRG1_0
-	-	-	-	-	4	P170 PPG36_1
-	-	-	-	4	5	P017 D31 TRG2_0
-	-	-	-	-	6	P171 PPG37_1
2 *1	2 *1	2 *1	2 *1	5	7	P020 ASX *2, *3, *4, *5 SIN3_1 TRG3_0 TIN0_2 RTO5_1
-	-	-	3 *1	6	8	P021 CS0X *5 SOT3_1 TRG6_1 TRG4_0
-	-	-	4 *1	7	9	P022 CS1X *5 SCK3_1 TRG7_1 TRG5_0
-	-	-	5 *1	8	10	P023 RDX *5 SCS3_1 PPG32_0 TIN0_0
3 *1	3 *1	3 *1	6 *1	9	11	P024 WR0X *2, *3, *4, *5 SIN4_1 PPG24_0 TIN1_0 RTO4_1 INT15_0
-	-	4 *1	7 *1	10	12	P025 WR1X *4, *5 SOT4_1 PPG25_0 TIN2_0

Page	Section	Change Results
45	1.9	Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified. (Error)

Pin Number						Pin Name
64	80	100	120	64	176	
-	4	5	8	11	14	P026 A00 SCK4_1 PPG26_0 TIN3_0
4	5	6	9	12	15	P027 A01 SCS40_1 PPG27_0 TOT0_0 RTO3_1
-	-	-	-	-	16	P173 PPG39_1
-	-	7	10	13	17	P030 A02 SCS41_1 PPG28_0 TOT1_0
-	6	8	11	14	18	P031 A03 SCS42_1 PPG29_0 TOT2_0 *2
5	7	9	12	15	19	P032 A04 SCS43_1 PPG30_0 TOT3_0 RTO2_1
6	8	10	13	16	20	P033 A05 PPG31_0 ICU3_3 TIN4_0 RTO1_1 SCK3_2
7	9	11	14	17	21	P034 A06 OCU11_1 ICU2_3 TIN5_0 RTO0_1 SOT3_2

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>45</td><td>1.9</td><td>(Continued)
(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>-</td><td>4 *1</td><td>5 *1</td><td>8 *1</td><td>11</td><td>14</td><td>P026
A00 *3, *4, *5
SCK4_1
PPG26_0
TIN3_0</td></tr>
<tr><td>4 *1</td><td>5 *1</td><td>6 *1</td><td>9 *1</td><td>12</td><td>15</td><td>P027
A01 *2, *3, *4, *5
SCS40_1
PPG27_0
TOT0_0
RTO3_1</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>16</td><td>P173
PPG39_1</td></tr>
<tr><td>-</td><td>-</td><td>7 *1</td><td>10 *1</td><td>13</td><td>17</td><td>P030
A02 *4, *5
SCS41_1
PPG28_0
TOT1_0</td></tr>
<tr><td>-</td><td>6 *1</td><td>8 *1</td><td>11 *1</td><td>14</td><td>18</td><td>P031
A03 *3, *4, *5
SCS42_1
PPG29_0
TOT2_0 *3</td></tr>
<tr><td>5 *1</td><td>7 *1</td><td>9 *1</td><td>12 *1</td><td>15</td><td>19</td><td>P032
A04 *2, *3, *4, *5
SCS43_1
PPG30_0
TOT3_0
RTO2_1</td></tr>
<tr><td>6 *1</td><td>8 *1</td><td>10 *1</td><td>13 *1</td><td>16</td><td>20</td><td>P033
A05 *2, *3, *4, *5
PPG31_0
ICU3_3
TIN4_0
RTO1_1
SCK3_2</td></tr>
<tr><td>7 *1</td><td>9 *1</td><td>11 *1</td><td>14 *1</td><td>17</td><td>21</td><td>P034
A06 *2, *3, *4, *5
OCU11_1
ICU2_3
TIN5_0
RTO0_1
SOT3_2</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
46	1.9	Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified. (Error)

Pin Number						Pin Name
64	80	100	120	64	176	
8	10	13	16	19	23	P151 SCK8_0/SCL8 OCU9_1 TRG7_0 ICU0_3 TIN7_0 ZIN0_2 DTTI_1
9	11	14	17	20	24	P035 A07 SIN8_0 OCU8_1 TOT4_0 AIN0_0 INT11_0
10	12	15	18	21	25	P036 A08 SCS8_0 OCU7_1 TOT5_0 BIN0_0
-	-	16	19	22	26	P037 A09 OCU6_1 TOT6_0 ZIN0_0
-	-	-	-	-	27	P174 TRG8_1
-	-	-	-	-	28	P175 TRG9_1
11	13	17	20	23	29	P040 A10 PPG23_1 TOT7_0 AIN1_0 SIN0_1
12	14	18	21	24	30	P041 A11 SIN9_0 ICU9_1 BIN1_0 INT12_0
13	15	19	22	25	31	P042 A12 SOT9_0 AN47 ICU8_1 TRG0_1 ZIN1_0
-	-	20	23	26	32	P043 A13 ICU7_1 TRG1_1

Page	Section	Change Results
46	1.9	(Continued) (Correct)

Pin Number						Pin Name
64	80	100	120	64	176	
8 *1	10 *1	13	16	19	23	P151 SCK8_0/SCL8 *2, *3 OCU9_1 TRG7_0 ICU0_3 TIN7_0 ZIN0_2 DTTI_1
9 *1	11 *1	14 *1	17 *1	20	24	P035 A07 *2, *3, *4, *5 SIN8_0 *2, *3 OCU8_1 TOT4_0 AIN0_0 INT11_0
10 *1	12 *1	15 *1	18 *1	21	25	P036 A08 *2, *3, *4, *5 SCS8_0 *2, *3 OCU7_1 TOT5_0 BIN0_0
-	-	16 *1	19 *1	22	26	P037 A09 *4, *5 OCU6_1 TOT6_0 ZIN0_0
-	-	-	-	-	27	P174 TRG8_1
-	-	-	-	-	28	P175 TRG9_1
11 *1	13 *1	17 *1	20 *1	23	29	P040 A10 *2, *3, *4, *5 PPG23_1 TOT7_0 AIN1_0 SIN0_1
12 *1	14 *1	18 *1	21 *1	24	30	P041 A11 *2, *3, *4, *5 SIN9_0 ICU9_1 BIN1_0 INT12_0
13 *1	15 *1	19 *1	22 *1	25	31	P042 A12 *2, *3, *4, *5 SOT9_0 AN47 ICU8_1 TRG0_1 ZIN1_0
-	-	20 *1	23 *1	26	32	P043 A13 *4, *5 ICU7_1 TRG1_1

Page	Section	Change Results
47	1.9	Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified. (Error)

Pin Number						Pin Name
64	80	100	120	64	176	
-	16	21	24	27	33	P044 A14 SCS9_0 ICU6_1 TRG2_1
14	17	22	25	28	34	P045 A15 SCK9_0 AN46 ICU5_1 TRG3_1 TOT1_2
-	-	-	26	29	35	P046 A16 ICU4_1 TRG4_1
-	-	-	-	-	36	P176 TRG10_0
15	18	23	27	30	37	P047 A17 AN45 TRG8_0 TIN3_2 SOT0_1
-	-	-	-	-	38	P177 TRG11_0
-	-	-	28	31	39	P050 A18 TRG5_1 PPG33_0
-	-	-	-	32	40	P051 A19 TRG9_0
-	-	-	-	33	41	P052 A20 PPG34_0 INT14_0
16	19	24	29	34	42	P053 A21 AN44 PPG35_0 INT14_1 SCK0_1
-	-	-	-	35	43	P054 SYSCLK PPG36_0
17	22	27	32	38	46	P055 CS2X SIN10_0 AN43 PPG37_0 TIN4_1

Page	Section	Change Results
47	1.9	(Continued) (Correct)

Pin Number						Pin Name
64	80	100	120	64	176	
-	16[*1]	21[*1]	24[*1]	27	33	P044 A14[*3, *4, *5] SCS9_0 ICU6_1 TRG2_1
14[*1]	17[*1]	22[*1]	25[*1]	28	34	P045 A15[*2, *3, *4, *5] SCK9_0 AN46 ICU5_1 TRG3_1 TOT1_2
-	-	-	26[*1]	29	35	P046 A16[*5] ICU4_1 TRG4_1
-	-	-	-	-	36	P176 TRG10_0
15[*1]	18[*1]	23[*1]	27[*1]	30	37	P047 A17[*2, *3, *4, *5] AN45 TRG8_0 TIN3_2 SOT0_1
-	-	-	-	-	38	P177 TRG11_0
-	-	-	28[*1]	31	39	P050 A18[*5] TRG5_1 PPG33_0
-	-	-	-	32	40	P051 A19 TRG9_0
-	-	-	-	33	41	P052 A20 PPG34_0 INT14_0
16[*1]	19[*1]	24[*1]	29[*1]	34	42	P053 A21[*2, *3, *4, *5] AN44 PPG35_0 INT14_1 SCK0_1
-	-	-	-	35	43	P054 SYSCLK PPG36_0
17[*1]	22[*1]	27[*1]	32[*1]	38	46	P055 CS2X[*2, *3, *4, *5] SIN10_0 AN43 PPG37_0 TIN4_1

Page	Section	Change Results
47 to 48	1.9	"9. Explanation of Pin Functions" should be modified as follows: (Error) General-purpose I/O port External Bus chip select 2 output pin(0) Multi-function serial ch.10 serial data input pin(0) ADC analog 43 input pin PPG ch.37 output pin(0) Reload timer ch.4 event input pin(1) General-purpose I/O port External Bus chip select 3 output pin(0) Input capture ch.9 input pin(0) PPG ch.0 output pin(1) Input capture ch.0 input pin(1) Reload timer ch.5 event input pin(1) Waveform generator ch.0 to ch.5 input pin(2) (Correct) General-purpose I/O port External Bus chip select 2 output pin Multi-function serial ch.10 serial data input pin(0) ADC analog 43 input pin PPG ch.37 output pin(0) Reload timer ch.4 event input pin(1) General-purpose I/O port External Bus chip select 3 output pin Input capture ch.9 input pin(0) PPG ch.0 output pin(1) Input capture ch.0 input pin(1) Reload timer ch.5 event input pin(1) Waveform generator ch.0 to ch.5 input pin(2)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>48</td><td>1.9</td><td>
Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>48</td><td>P181
PPG41_0</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>33</td><td>39</td><td>49</td><td>P056
CS3X
ICU9_0
PPG0_1
ICU0_1
TIN5_1
DTTI_2</td></tr>
<tr><td>19</td><td>24</td><td>29</td><td>35</td><td>41</td><td>51</td><td>P057
RDY
SCK10_1
AN42
ICU8_0
TRG0_2
PPG1_1
ICU1_1
TIN6_1</td></tr>
</table>

(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>48</td><td>P181
PPG41_0</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>33*1</td><td>39</td><td>49</td><td>P056
CS3X*5
ICU9_0
PPG0_1
ICU0_1
TIN5_1
DTTI_2</td></tr>
<tr><td>19*1</td><td>24*1</td><td>29*1</td><td>35*1</td><td>41</td><td>51</td><td>P057
RDY*2, *3, *4, *5
SCK10_1
AN42
ICU8_0
TRG0_2
PPG1_1
ICU1_1
TIN6_1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>49</td><td>1.9</td><td>Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th><th rowspan="2">Polarity</th><th rowspan="2">I/O circuit form</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>35</td><td>43</td><td>49</td><td>57</td><td>71</td><td>P073
SOT4_0/SDA4
AN33
ICU3_2</td><td>-
-
-
-</td><td>E</td></tr>
</table>
(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th><th rowspan="2">Polarity</th><th rowspan="2">I/O circuit form</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>35*3</td><td>43*4</td><td>49</td><td>57</td><td>71</td><td>P073
SOT4_0/SDA4*3, *4
AN33
ICU3_2</td><td>-
-
-
-</td><td>D</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>51</td><td>1.9</td><td>Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>34</td><td>42</td><td>52</td><td>62</td><td>77</td><td>96</td><td>P093
TX0_1
SIN11_0
AN7
ICU4_2
PPG16_1
ICU3_0
TOT2_1</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>48</td><td>59</td><td>69</td><td>85</td><td>104</td><td>P100
SCK7_0/SCL7
AN12
PPG8_0</td></tr>
</table></td></tr>
<tr><td>51</td><td>1.9</td><td>(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>34*1</td><td>42*1</td><td>52</td><td>62</td><td>77</td><td>96</td><td>P093
TX0_1
SIN11_0
AN7
ICU4_2
PPG16_1
ICU3_0
TOT2_1*2, *3</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>48*1</td><td>59</td><td>69</td><td>85</td><td>104</td><td>P100
SCK7_0/SCL7*3
AN12
PPG8_0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>52</td><td>1.9</td><td>Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>40</td><td>49</td><td>61</td><td>71</td><td>87</td><td>106</td><td>P102
SIN7_0
AN14
PPG10_0
INT10_0</td></tr>
<tr><td>41</td><td>50</td><td>62</td><td>72</td><td>88</td><td>107</td><td>P103
SCS73_0
AN15
PPG11_0</td></tr>
<tr><td>42</td><td>51</td><td>63</td><td>73</td><td>89</td><td>108</td><td>P104
SCS72_0
AN16
PPG12_0</td></tr>
<tr><td>43</td><td>52</td><td>64</td><td>74</td><td>90</td><td>109</td><td>P105
SCS71_0
AN17
PPG13_0</td></tr>
</table></td></tr>
<tr><td>52</td><td>1.9</td><td>(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>40 *1</td><td>49 *1</td><td>61</td><td>71</td><td>87</td><td>106</td><td>P102
SIN7_0 *2, *3
AN14
PPG10_0
INT10_0</td></tr>
<tr><td>41 *1</td><td>50 *1</td><td>62</td><td>72</td><td>88</td><td>107</td><td>P103
SCS73_0 *2, *3
AN15
PPG11_0</td></tr>
<tr><td>42 *1</td><td>51 *1</td><td>63</td><td>73</td><td>89</td><td>108</td><td>P104
SCS72_0 *2, *3
AN16
PPG12_0</td></tr>
<tr><td>43 *1</td><td>52 *1</td><td>64</td><td>74</td><td>90</td><td>109</td><td>P105
SCS71_0 *2, *3
AN17
PPG13_0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>54</td><td>1.9</td><td>Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>-</td><td>94</td><td>111</td><td>131</td><td>159</td><td>P000
D16
SIN1_0
TIOA0_1
INT2_0</td></tr>
<tr><td>-</td><td>75</td><td>95</td><td>112</td><td>132</td><td>160</td><td>P001
D17
SOT1_0
TIOA1_1</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>113</td><td>133</td><td>161</td><td>P002
D18
SCK1_0
TIOB0_1</td></tr>
</table></td></tr>
<tr><td>54</td><td>1.9</td><td>(Correct)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>-</td><td>-</td><td>94 *1</td><td>111 *1</td><td>131</td><td>159</td><td>P000
D16 *4, *5
SIN1_0
TIOA0_1 *4
INT2_0</td></tr>
<tr><td>-</td><td>75 *1</td><td>95 *1</td><td>112 *1</td><td>132</td><td>160</td><td>P001
D17 *3, *4, *5
SOT1_0 *3
TIOA1_1</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>113 *1</td><td>133</td><td>161</td><td>P002
D18 *5
SCK1_0
TIOB0_1</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>55</td><td>1.9</td><td>Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions" should be modified.

(Error)
<table>
<tr><th colspan="6">Pin Number</th><th rowspan="2">Pin Name</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>-</td><td>76</td><td>96</td><td>114</td><td>134</td><td>162</td><td>P003
D19
SIN2_0
TIOB1_1
INT3_0</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>135</td><td>163</td><td>P004
D20
SOT2_0</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>164</td><td>P164
PPG32_1</td></tr>
<tr><td>61</td><td>77</td><td>97</td><td>115</td><td>136</td><td>165</td><td>P005
D21
SCK2_0 *1
ADTG0_1
INT7_1
RX2(64)</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>166</td><td>P165
PPG33_1</td></tr>
<tr><td>62</td><td>78</td><td>98</td><td>116</td><td>137</td><td>167</td><td>P006
D22
SCS2_0 *1
ADTG1_1
INT2_1
TX2(64)</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>117</td><td>138</td><td>168</td><td>P007
D23</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>169</td><td>P166
PPG34_1</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>118</td><td>139</td><td>170</td><td>P010
D24</td></tr>
<tr><td>63</td><td>79</td><td>99</td><td>119</td><td>140</td><td>171</td><td>P011
WOT
D25
SOT2_1
TIOA0_0
INT3_1</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
55	1.9	(Continued) (Correct)

Pin Number						Pin Name
64	80	100	120	64	176	
-	76[*1]	96[*1]	114[*1]	134	162	P003 D19[*3, *4, *5] SIN2_0 TIOB1_1 INT3_0
-	-	-	-	135	163	P004 D20 SOT2_0
-	-	-	-	-	164	P164 PPG32_1
61[*1]	77[*1]	97[*1]	115[*1]	136[*1]	165[*1]	P005 D21[*2, *3, *4, *5] SCK2_0[*2] ADTG0_1 INT7_1 RX2(64)[*4, *5, *6, *7]
-	-	-	-	-	166	P165 PPG33_1
62[*1]	78[*1]	98[*1]	116[*1]	137[*1]	167[*1]	P006 D22[*2, *3, *4, *5] SCS2_0[*2] ADTG1_1 INT2_1 TX2(64)[*4, *5, *6, *7]
-	-	-	117[*1]	138	168	P007 D23[*5]
-	-	-	-	-	169	P166 PPG34_1
-	-	-	118[*1]	139	170	P010 D24[*5]
63[*1]	79[*1]	99[*1]	119[*1]	140	171	P011 WOT D25[*2, *3, *4, *5] SOT2_1[*2] TIOA0_0[*2, *3, *4] INT3_1

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>56</td><td>1.9</td><td>The following sentences should be added under Table 9-1 List of Pin Functions in "9. Explanation of Pin Functions":

(Correct)
*1: There is a restriction of pin functions. See "Pin Name" of this table.
*2: not supported in 64pin
*3: not supported in 80pin
*4: not supported in 100pin
*5: not supported in 120pin
*6: not supported in 144pin
*7: not supported in 176pin</td></tr>
<tr><td>59</td><td>1.9.1</td><td>Table 9-6 Pins of the Multi-function Serial Interface (ch.0 to ch.11) in "9.1. Pins of Each Function" should be modified.
(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.1 serial data output pin(0)</td><td>SOT1_0</td><td>-</td><td>-</td><td>75</td><td>95</td><td>112</td><td>132</td><td>160</td></tr>
<tr><td>MFS ch.1 serial data input pin(0)</td><td>SIN1_0</td><td>No</td><td>-</td><td>-</td><td>94</td><td>111</td><td>131</td><td>159</td></tr>
<tr><td>MFS ch.2 clock I/O pin(0)</td><td>SCK2_0</td><td>No</td><td>61</td><td>77</td><td>97</td><td>115</td><td>136</td><td>165</td></tr>
<tr><td>MFS ch.2 serial data output pin(0)</td><td>SOT2_0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>135</td><td>163</td></tr>
<tr><td>MFS ch.2 serial data output pin(1)</td><td>SOT2_1</td><td>-</td><td>63</td><td>79</td><td>99</td><td>119</td><td>140</td><td>171</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.1 serial data output pin(0)</td><td>SOT1_0</td><td>-</td><td>-</td><td>-</td><td>95</td><td>112</td><td>132</td><td>160</td></tr>
<tr><td>MFS ch.1 serial data input pin(0)</td><td>SIN1_0</td><td>No</td><td>-</td><td>-</td><td>94</td><td>111</td><td>131</td><td>159</td></tr>
<tr><td>MFS ch.2 clock I/O pin(0)</td><td>SCK2_0</td><td>No</td><td>-</td><td>77</td><td>97</td><td>115</td><td>136</td><td>165</td></tr>
<tr><td>MFS ch.2 serial data output pin(0)</td><td>SOT2_0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>135</td><td>163</td></tr>
<tr><td>MFS ch.2 serial data output pin(1)</td><td>SOT2_1</td><td>-</td><td>-</td><td>79</td><td>99</td><td>119</td><td>140</td><td>171</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>59</td><td>1.9.1</td><td>Table 9-6 Pins of the Multifunction Serial Interface (ch.0 to ch.11) in "9.1. Pins of Each Function" should be modified.
(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.4 serial data output pin(0) / I^{2}C bus serial data I/O pin</td><td>SOT4_0/ SDA4</td><td>-</td><td>-</td><td>35</td><td>43</td><td>49</td><td>57</td><td>71</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.4 serial data output pin(0) / I^{2}C bus serial data I/O pin</td><td>SOT4_0/ SDA4</td><td>-</td><td>-</td><td>-</td><td>-</td><td>49</td><td>57</td><td>71</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>60</td><td>1.9.1</td><td>Table 9-6 Pins of the Multifunction Serial Interface (ch.0 to ch.11) in "9.1. Pins of Each Function" should be modified.
(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.7 clock I/O pin(0)/
I2C bus serial clock I/O pin</td><td>SCK7_0/
SCL7</td><td>No</td><td>-</td><td>48</td><td>59</td><td>69</td><td>85</td><td>104</td></tr>
<tr><td>MFS ch.7 serial data output pin(0)/
I2C bus serial data I/O pin</td><td>SOT7_0/
SDA7</td><td>-</td><td>-</td><td>-</td><td>60</td><td>70</td><td>86</td><td>105</td></tr>
<tr><td>MFS ch.7 serial data input pin(0)</td><td>SIN7_0</td><td>No</td><td>40</td><td>49</td><td>61</td><td>71</td><td>87</td><td>106</td></tr>
<tr><td>MFS ch.8 clock I/O pin(0)/
I2C bus serial clock I/O pin</td><td>SCK8_0/
SCL8</td><td></td><td>8</td><td>10</td><td>13</td><td>16</td><td>19</td><td>23</td></tr>
<tr><td>MFS ch.8 serial data output pin(0)/
I2C bus serial data I/O pin</td><td>SOT8_0/
SDA8</td><td>-</td><td>-</td><td>-</td><td>12</td><td>15</td><td>18</td><td>22</td></tr>
<tr><td>MFS ch.8 serial data input pin(0)</td><td>SIN8_0</td><td>No</td><td>9</td><td>11</td><td>14</td><td>17</td><td>20</td><td>24</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.2 serial chip select I/O pin(0)</td><td>SCS2_0</td><td>No</td><td>62</td><td>78</td><td>98</td><td>116</td><td>137</td><td>167</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.7 clock I/O pin(0)/
I2C bus serial clock I/O pin</td><td>SCK7_0/
SCL7</td><td>No</td><td>-</td><td>-</td><td>59</td><td>69</td><td>85</td><td>104</td></tr>
<tr><td>MFS ch.7 serial data output pin(0)/
I2C bus serial data I/O pin</td><td>SOT7_0/
SDA7</td><td>-</td><td>-</td><td>-</td><td>60</td><td>70</td><td>86</td><td>105</td></tr>
<tr><td>MFS ch.7 serial data input pin(0)</td><td>SIN7_0</td><td>No</td><td>-</td><td>-</td><td>61</td><td>71</td><td>87</td><td>106</td></tr>
<tr><td>MFS ch.8 clock I/O pin(0)/
I2C bus serial clock I/O pin</td><td>SCK8_0/
SCL8</td><td>No</td><td>-</td><td>-</td><td>13</td><td>16</td><td>19</td><td>23</td></tr>
<tr><td>MFS ch.8 serial data output pin(0)/
I2C bus serial data I/O pin</td><td>SOT8_0/
SDA8</td><td>-</td><td>-</td><td>-</td><td>12</td><td>15</td><td>18</td><td>22</td></tr>
<tr><td>MFS ch.8 serial data input pin(0)</td><td>SIN8_0</td><td>No</td><td>-</td><td>-</td><td>14</td><td>17</td><td>20</td><td>24</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.2 serial chip select I/O pin(0)</td><td>SCS2_0</td><td>No</td><td>-</td><td>78</td><td>98</td><td>116</td><td>137</td><td>167</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>60
to
61</td><td>1.9.1</td><td>Table 9-6 Pins of the Multifunction Serial Interface (ch.0 to ch.11) in "9.1. Pins of Each Function" should be modified.

(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.71 serial chip select output pin(0)</td><td>SCS71_0</td><td>-</td><td>43</td><td>52</td><td>64</td><td>74</td><td>90</td><td>109</td></tr>
<tr><td>MFS ch.72 serial chip select output pin(0)</td><td>SCS72_0</td><td>-</td><td>42</td><td>51</td><td>63</td><td>73</td><td>89</td><td>108</td></tr>
<tr><td>MFS ch.73 serial chip select output pin(0)</td><td>SCS73_0</td><td>-</td><td>41</td><td>50</td><td>62</td><td>72</td><td>88</td><td>107</td></tr>
<tr><td>MFS ch.8 serial chip select I/O pin(0)</td><td>SCS8_0</td><td>No</td><td>10</td><td>12</td><td>15</td><td>18</td><td>21</td><td>25</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MFS ch.71 serial chip select output pin(0)</td><td>SCS71_0</td><td>-</td><td>-</td><td>-</td><td>64</td><td>74</td><td>90</td><td>109</td></tr>
<tr><td>MFS ch.72 serial chip select output pin(0)</td><td>SCS72_0</td><td>-</td><td>-</td><td>-</td><td>63</td><td>73</td><td>89</td><td>108</td></tr>
<tr><td>MFS ch.73 serial chip select output pin(0)</td><td>SCS73_0</td><td>-</td><td>-</td><td>-</td><td>62</td><td>72</td><td>88</td><td>107</td></tr>
<tr><td>MFS ch.8 serial chip select I/O pin(0)</td><td>SCS8_0</td><td>No</td><td>-</td><td>-</td><td>15</td><td>18</td><td>21</td><td>25</td></tr>
</table></td></tr>
<tr><td>65</td><td>1.9.1</td><td>Table 9-13 Pins of Base Timer (ch.0, ch.1) in "9.1. Pins of Each Function" should be modified.

(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>Base timer ch.0 TIOA output pin(0)</td><td>TIOA0_0</td><td>-</td><td>63</td><td>79</td><td>99</td><td>119</td><td>140</td><td>171</td></tr>
<tr><td>Base timer ch.0 TIOA output pin(1)</td><td>TIOA0_1</td><td>-</td><td>-</td><td>-</td><td>94</td><td>111</td><td>131</td><td>159</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td>Base timer ch.0 TIOA output pin(0)</td><td>TIOA0_0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>119</td><td>140</td><td>171</td></tr>
<tr><td>Base timer ch.0 TIOA output pin(1)</td><td>TIOA0_1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>111</td><td>131</td><td>159</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>65</td><td>1.9.1</td><td>Table 9-14 Pins of Reload Timer (ch.0 to ch.7) in "9.1. Pins of Each Function" should be modified.

(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Reload timer ch.2 output pin(0)</td><td>TOT2_0</td><td>-</td><td>-</td><td>6</td><td>8</td><td>11</td><td>14</td><td>18</td></tr>
<tr><td>Reload timer ch.2 output pin(1)</td><td>TOT2_1</td><td>-</td><td>34</td><td>42</td><td>52</td><td>62</td><td>77</td><td>96</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Reload timer ch.3 event input pin(2)</td><td>TIN3_2</td><td>Yes</td><td>15</td><td>18</td><td>-</td><td>27</td><td>30</td><td>37</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Reload timer ch.2 output pin(0)</td><td>TOT2_0</td><td>-</td><td>-</td><td>-</td><td>8</td><td>11</td><td>14</td><td>18</td></tr>
<tr><td>Reload timer ch.2 output pin(1)</td><td>TOT2_1</td><td>-</td><td>-</td><td>-</td><td>52</td><td>62</td><td>77</td><td>96</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Reload timer ch.3 event input pin(2)</td><td>TIN3_2</td><td>Yes</td><td>15</td><td>18</td><td>23</td><td>27</td><td>30</td><td>37</td></tr>
</table></td></tr>
<tr><td>67</td><td>1.9.1</td><td>A sentence should be added as follows in Table 9-16 Pins of Waveform Generator (ch.0 to ch.5) in "9.1. Pins of Each Function".

(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>Waveform generator ch.0-ch.5 input pin(0)</td><td>DTTI_0</td><td>Yes</td><td>56</td><td>70</td><td>86</td><td>102</td><td>121</td><td>149</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td>Waveform generator ch.0-ch.5 input pin(0) (unavailable with the sub oscillation)</td><td>DTTI_0</td><td>Yes</td><td>56</td><td>70</td><td>86</td><td>102</td><td>121</td><td>149</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>69</td><td>1.9.1</td><td>A sentence should be added as follows in Table 9-18 Pins of Port Function (General-Purpose I/O) in "9.1. Pins of Each Function".

(Error)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>64</th><th>176</th></tr>
<tr><td>General-purpose I/O port</td><td>P135</td><td>Yes</td><td>56</td><td>70</td><td>86</td><td>102</td><td>121</td><td>149</td></tr>
<tr><td>General-purpose I/O port</td><td>P136</td><td>Yes</td><td>57</td><td>71</td><td>87</td><td>103</td><td>122</td><td>150</td></tr>
</table>

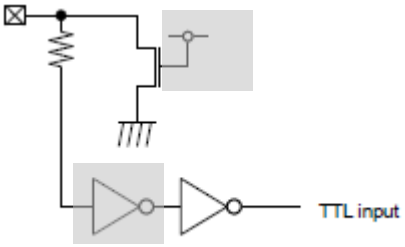
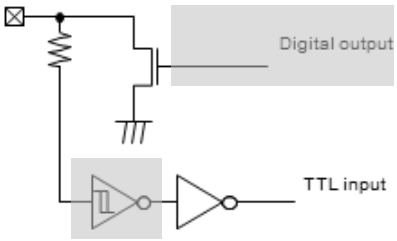
(Correct)
<table>
<tr><th rowspan="2">Function</th><th rowspan="2">Pin Name</th><th rowspan="2">Noise Filter</th><th colspan="6">Pin Number</th></tr>
<tr><th>64</th><th>80</th><th>100</th><th>120</th><th>144</th><th>176</th></tr>
<tr><td>General-purpose I/O port
(without sub oscillation only)</td><td>P135</td><td>Yes</td><td>56</td><td>70</td><td>86</td><td>102</td><td>121</td><td>149</td></tr>
<tr><td>General-purpose I/O port
(without sub oscillation only)</td><td>P136</td><td>Yes</td><td>57</td><td>71</td><td>87</td><td>103</td><td>122</td><td>150</td></tr>
</table>
</td></tr>
<tr><td>72</td><td>1.10</td><td>Remarks for Type D in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows:

(Error)
- I^2C Analog input,General-purpose I/O port
- Output 3mA
- Pull-up resistor control 50kΩ
- I^2C Schmitt input

(Correct)
- I^2C Analog input, General-purpose I/O port
- Output 3mA
- Pull-up resistor control 50kΩ
- I^2C hysteresis input</td></tr>
</table>

Page	Section	Change Results
73	1.10	Remarks for Type E in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - I^2C ,General-purpose I/O port - Output 3mA - Pull-up resistor control 50kΩ - I^2C Schmitt input (Correct) - I^2C, General-purpose I/O port - Output 3mA - Pull-up resistor control 50kΩ - I^2C hysteresis input
73	1.10	Remarks for Type F in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - Schmitt input (Correct) - General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - CMOS hysteresis input
73	1.10	Remarks for Type G in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - Analog input,General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - Schmitt input (Correct) - Analog input, General-purpose I/O port - Output 4mA - Pull-up resistor control 50kΩ - CMOS hysteresis input

Page	Section	Change Results
74	1.10	Remarks for Type I in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - 3V pad power supply (5V tolerant),General-purpose I/O port - Output 4mA - Schmitt input (Correct) - General-purpose I/O port (5V tolerant) - Output 4mA - CMOS hysteresis input
74	1.10	Remarks for Type J in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - 3V pad power supply (5V tolerant),Analog input,General-purpose I/O port - Output 4mA - Schmitt input (Correct) - Analog input, General-purpose I/O port (5V tolerant) - Output 4mA - CMOS hysteresis input
74	1.10	Remarks for Type K in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - Mode I/O - Schmitt input (Correct) - Mode I/O - CMOS hysteresis input

Page	Section	Change Results
74	1.10	Remarks for Type L in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - Open-drain I/O - Output 25mA (NOD) - TTL input (Correct) - Open-drain I/O - Output 25mA (Nch open-drain) - TTL input
74	1.10	The circuit for Type L in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error)  (Correct) 
74	1.10	Remarks for Type M in Figure 10-1 I/O Circuit Types in "10. I/O Circuit Types" should be modified as follows: (Error) - Schmitt input - Pull-up resistor 50kΩ (5V cont) (Correct) - CMOS hysteresis input - Pull-up resistor 50kΩ

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>102</td><td>3.7</td><td>A sentence in "7. Addressing" should be modified as follows:

(Error)
The CPU manages the address space in 32 bytes. Specify a value of 32-bit for the address on the address space to access from the CPU. Figure 7-1 shows the address space.

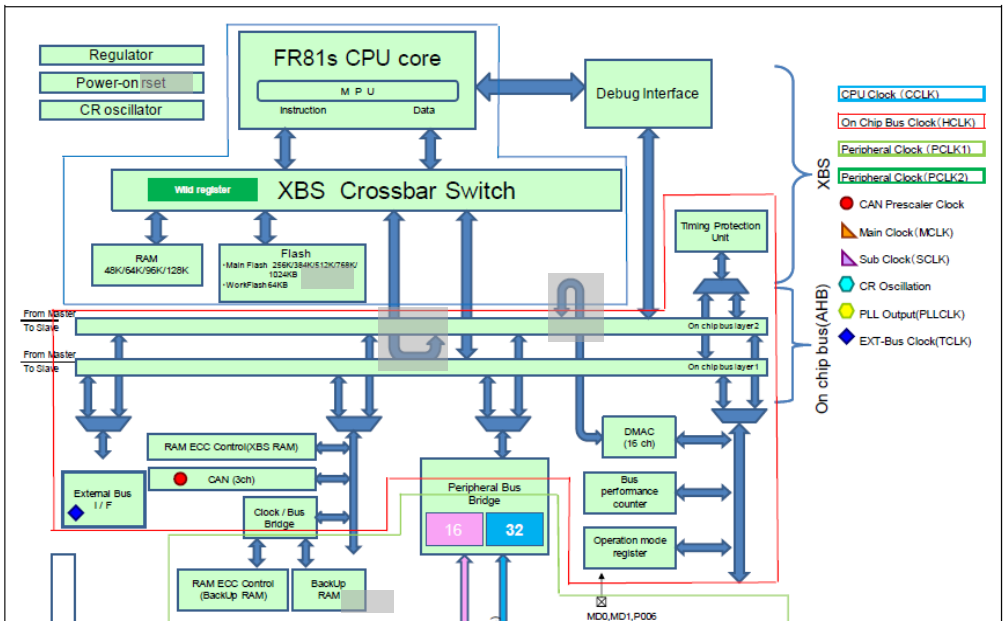
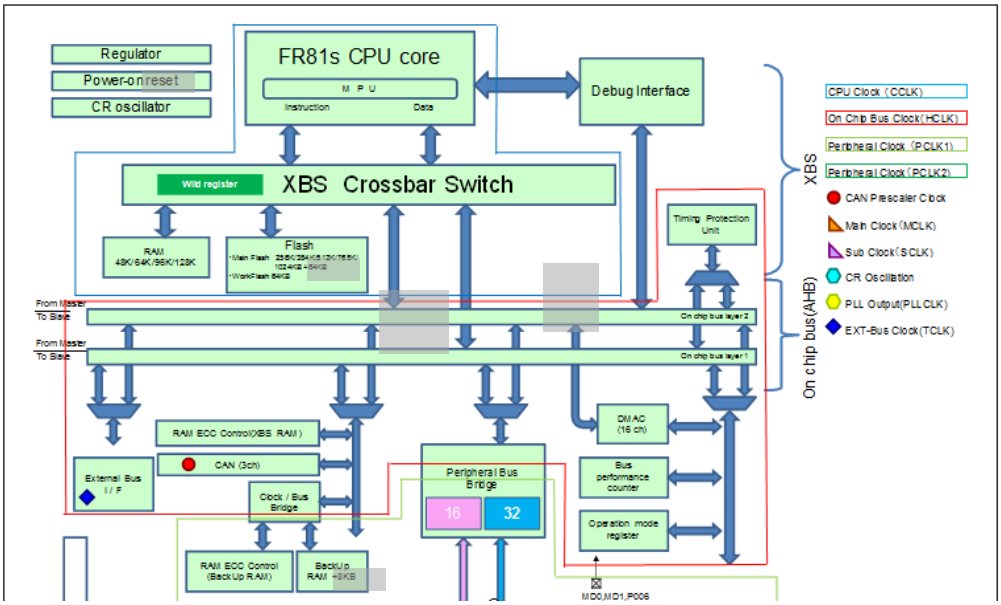
(Correct)
The CPU manages the address space by bytes. Specify a value of 32-bit for the address on the address space to access from the CPU. Figure 7-1 shows the address space.</td></tr>
<tr><td>104</td><td>3.8.1</td><td>The content of R13 in the configuration and initial values of general-purpose registers in Figure 8-1 Initial Values of General-purpose Registers, Dedicated Registers, and Floating Point Registers in "8.1 General-purpose Registers, Dedicated Registers, and Floating Point Registers" should be modified as follows:

(Error)
R13 | AG | XXXX | XXXX_H
R14 | FP | XXXX | XXXX_H
R15 | SP | 0000 | 0000_H

(Correct)
R13 | AC | XXXX | XXXX_H
R14 | FP | XXXX | XXXX_H
R15 | SP | 0000 | 0000_H</td></tr>
<tr><td>152</td><td>4.5.3.2</td><td>MD1 in Table 5-3 Setting Pins in Serial Writer Mode in "5.3.2. Serial Writer Mode" should be modified as follows:

(Error)
52 | 66 | 82 | 98 | 117 | 145 | MD1 | VDD (Pull-up) | Hys. | -

(Correct)
52 | 66 | 82 | 98 | 117 | 145 | MD1 | VCC (Pull-up) | Hys. | -</td></tr>
</table>

Page	Section	Change Results
162	5.3	Figure 3-8 in "3. Configuration" should be modified as follows: (Error)  (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>163</td><td>5.3</td><td>The following table should be added in "3. Configuration":

(Correct)
Table 3-1 List of functions that use PCLK1/PCLK2
<table>
<tr><th>Functions that use PCLK1</th><th>Functions that use PCLK2</th></tr>
<tr><td>RAM ECC Control (BackUp RAM)</td><td>CAN prescaler</td></tr>
<tr><td>BackUp RAM</td><td>RTC/WDT1 Calibration</td></tr>
<tr><td>Watchdog timer(SW and HW)</td><td>I/O port setting</td></tr>
<tr><td>DMA transfer request generate/clear</td><td>32bit Free-run timer (3ch)</td></tr>
<tr><td>Interrupt request batch read</td><td>32bit Input capture (6ch)</td></tr>
<tr><td>Clock control (divide control)</td><td>32bit Output compare (6ch)</td></tr>
<tr><td>Reset control register</td><td>Base timer (2ch)</td></tr>
<tr><td>Low-power consumption setting register</td><td>U/D counter (2ch)</td></tr>
<tr><td>Delay interrupt</td><td>Reload timer (8ch)</td></tr>
<tr><td>Interrupt controller</td><td>8bit DA converter (2ch)</td></tr>
<tr><td>External interrupt input (16ch)</td><td>Clock monitor</td></tr>
<tr><td>Real time clock</td><td>CRC</td></tr>
<tr><td>Clock supervisor</td><td>Wave generator (6ch)</td></tr>
<tr><td>NMI</td><td>16bit Free-run timer (3ch)</td></tr>
<tr><td>Low-voltage detection
(External power supply low-voltage detection)</td><td>16bit Input capture (4ch)</td></tr>
<tr><td>Low-voltage detection
(Internal power supply low-voltage detection)</td><td>16bit Output compare (6ch)</td></tr>
<tr><td>Clock control
(Clock setting, Main timer, Sub timer, PLL timer)</td><td>12bit AD converter (32ch + 16ch)</td></tr>
<tr><td>-</td><td>Multi-function serial interface (12ch)</td></tr>
<tr><td>-</td><td>PPG (48ch)</td></tr>
</table></td></tr>
<tr><td>174
to
175</td><td>5.4.6</td><td>The description in [bit7] MTIF in "4.6. Main Timer Control Register : MTMCR (Main clock TiMer Control Register)" should be modified as follows:

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

(Correct)
If a set factor and a clear factor occur at the same time, the set factor will take precedence.
An internal reset is issued at the return from standby mode (power-shutdown), and the main timer interrupt flag is not set.</td></tr>
</table>

Page	Section	Change Results
176	5.4.7	The description in [bit7] STIF in "4.7. Sub Timer Control Register : STMCR (Sub clock TiMer Control Register)" should be modified as follows: (Error) If a set factor and a clear factor occur at the same time, the set factor will take precedence. (Correct) If a set factor and a clear factor occur at the same time, the set factor will take precedence. An internal reset is issued at the return from standby mode (power-shutdown), and the sub timer interrupt flag is not set.
184	5.4.11	The description in "4.11. PLL/SSCG Clock Selection Register : CCPSSELR (CCtl Pll/Sscg clock SELect Register)" should be modified as follows: (Error) This register selects the clock source supplied to system. (Correct) This register selects which to use, PLL or SSCG.
208	5.5.1.2	A sentence in "5.1.2. Sub Clock (SBCLK)" should be modified as follows: (Error) · When the clock is used as a port because the clock is used for sub oscillation and port (metal option). (Correct) · When the clock is used as a port because the clock is used for sub oscillation and port (without sub oscillation).
211	5.5.1.3	Note in "5.1.3. PLL/SSCG Clock (PLLSSCLK)" should be modified as follows: (Error) · In debug operation (E_DBCR.PLOCK =1), PLL cannot stop because always supplying the PLL clock is required for MDI communication. (Correct) · In debug operation, PLL cannot stop because always supplying the PLL clock is required for MDI communication.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>212</td><td>5.5.1.4</td><td>The following sentence should be added to the note regarding the table "Clock Control PLL Clock Frequency" in "5.1.4. Limitations when PLL/SSCG Clock is used".

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

(Correct)
Note:
Set PLLCR or CCPSDIVR and CCPLLFBR so as not to exceed frequency (max).
The frequency (max) is 80 MHz.</td></tr>
<tr><td>212</td><td>5.5.1.4</td><td>The table "Microcontroller Unit Clock Control SSCG Clock Frequency" in "5.1.4. Limitations when PLL/SSCG Clock is used" should be modified as follows:

(Error)
<table>
<tr><th>Frequency (max)</th><th>FCTLR: FAW</th><th>CCPSSELR: PCSEL</th><th>CCSSCCR0: SSEN</th><th>CCSSCCR0: SMODE</th><th>CCSSCCR1: RATESEL</th><th>Remarks</th></tr>
<tr><td>72MHz</td><td>00</td><td>1</td><td>1</td><td>0</td><td>000 to 110</td><td>Down Spread</td></tr>
<tr><td>72MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>000</td><td>Center Spread (0.5%)</td></tr>
<tr><td>72MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>010</td><td>Center Spread (1%)</td></tr>
<tr><td>72MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>011</td><td>Center Spread (2%)</td></tr>
<tr><td>71MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>100</td><td>Center Spread (3%)</td></tr>
<tr><td>71MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>101</td><td>Center Spread (4%)</td></tr>
<tr><td>70MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>110</td><td>Center Spread (5%)</td></tr>
<tr><td>80MHz</td><td>01</td><td>1</td><td>0</td><td>0/1</td><td>000 to 110</td><td>Spread 0%</td></tr>
<tr><td>80MHz</td><td>00</td><td>1</td><td>0</td><td>0/1</td><td>000 to 110</td><td>Spread 0%</td></tr>
</table>
(Correct)
<table>
<tr><th>Frequency (max)</th><th>FCTLR: FAW</th><th>CCPSSELR: PCSEL</th><th>CCSSCCR0: SSEN</th><th>CCSSCCR0: SMODE</th><th>CCSSCCR1: RATESEL</th><th>Remarks</th></tr>
<tr><td>72 MHz</td><td>00</td><td>1</td><td>1</td><td>0</td><td>000 to 110</td><td>Down Spread</td></tr>
<tr><td>72 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>000</td><td>Center Spread (0.5%)</td></tr>
<tr><td>72 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>010</td><td>Center Spread (1%)</td></tr>
<tr><td>72 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>011</td><td>Center Spread (2%)</td></tr>
<tr><td>71 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>100</td><td>Center Spread (3%)</td></tr>
<tr><td>71 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>101</td><td>Center Spread (4%)</td></tr>
<tr><td>70 MHz</td><td>00</td><td>1</td><td>1</td><td>1</td><td>110</td><td>Center Spread (5%)</td></tr>
<tr><td>80 MHz</td><td>01</td><td>1</td><td>0</td><td>0/1</td><td>000 to 110</td><td>Spread 0%</td></tr>
<tr><td>72 MHz</td><td>00</td><td>1</td><td>0</td><td>0/1</td><td>000 to 110</td><td>Spread 0%</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>212</td><td>5.5.1.4</td><td>The following sentence is added to the note of the table "Microcontroller Unit Clock Control SSCG Clock Frequency" in "5.1.4. Limitations when PLL/SSCG Clock is used".

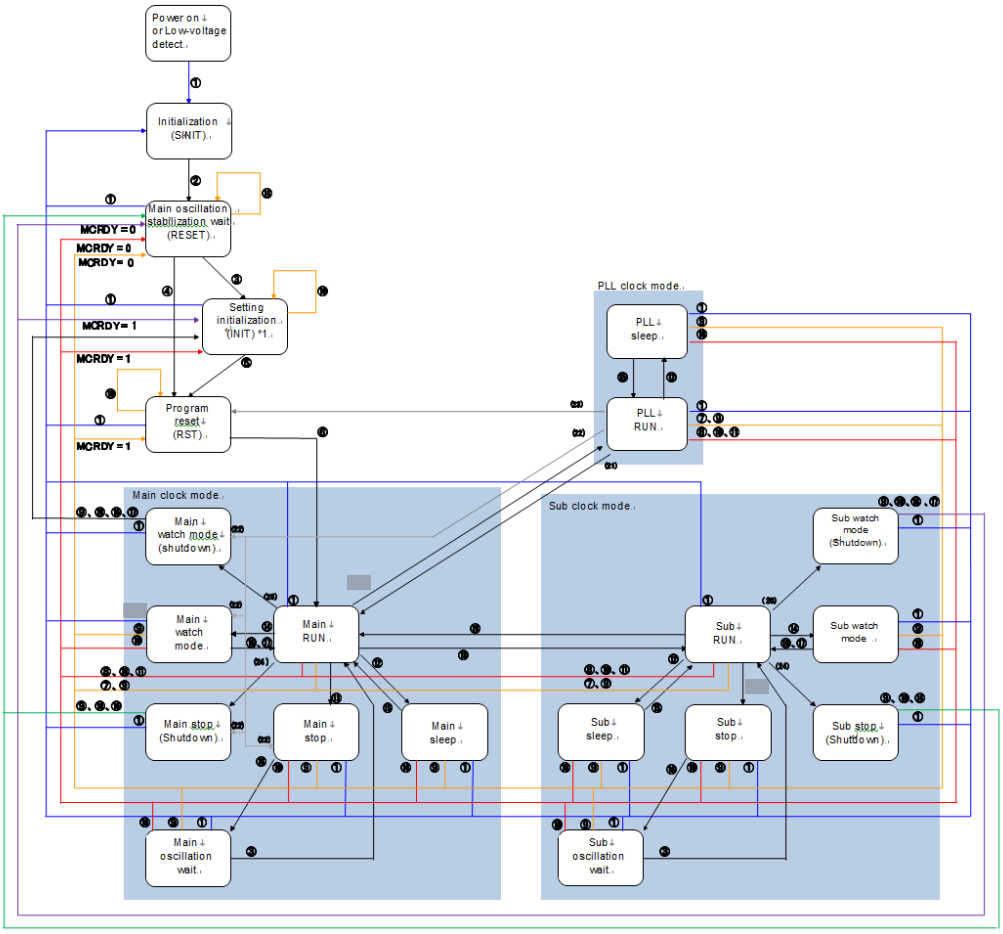
(Error)
Note:
Set CCPSDIVR, CCSSFBR0 and CCSSFBR1 so as not to exceed frequency (max).

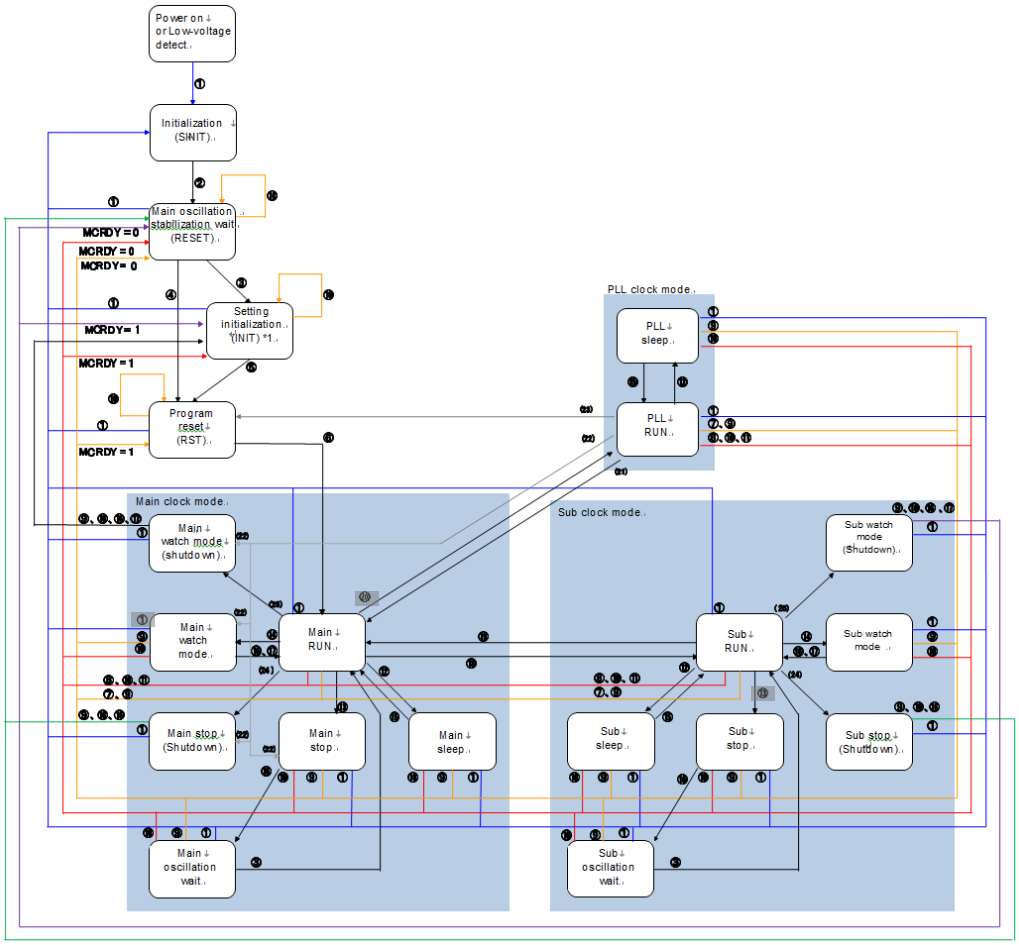
(Correct)
Note:
Set CCPSDIVR, CCSSFBR0 and CCSSFBR1 so as not to exceed frequency (max).
The frequency (max) is 80 MHz.</td></tr>
<tr><td>212</td><td>5.5.1.4</td><td>The descriptions of SSSCR1:RATESEL[2:0] and SDIVCR0:NDIV[5:0] should be deleted from the table indicating the relations between the modulation rate and division ratio when SSCG is used in "5.1.4. Limitations when PLL/SSCG clock is used".

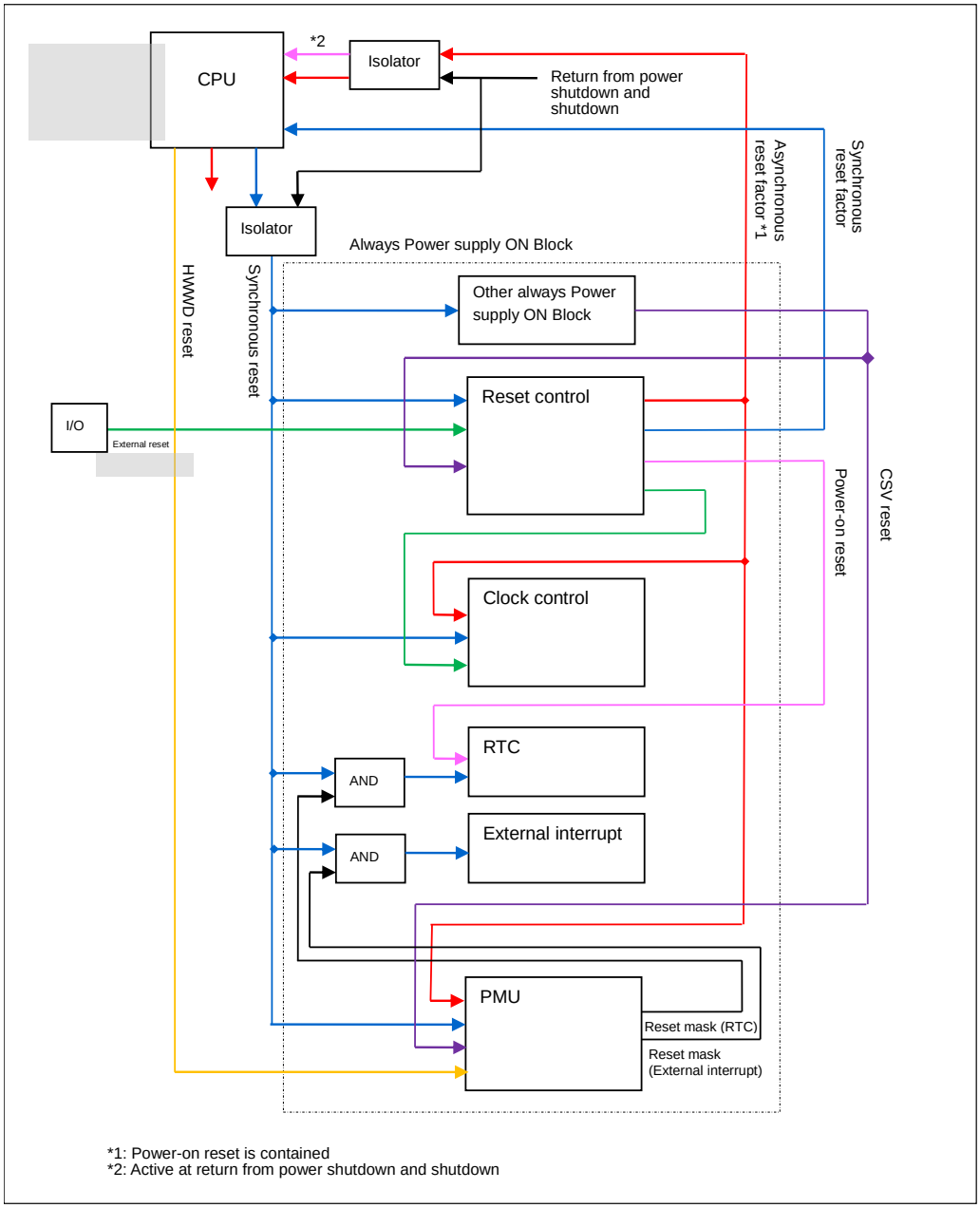
(Error)
<table>
<tr><td colspan="2">CCSSCCR1:RATESEL[2:0]
SSSCR1:RATESEL[2:0]</td><td colspan="3">CCSSFBR0:NDIV[5:0]
SDIVCR0:NDIV[5:0]</td></tr>
<tr><td>Modulation rate</td><td>Set value</td><td>Range of division ratio</td><td>Set value lower limit</td><td>Set value upper limit</td></tr>
</table>
(Correct)
<table>
<tr><td colspan="2">CCSSCCR1:RATESEL[2:0]</td><td colspan="3">CCSSFBR0:NDIV[5:0]</td></tr>
<tr><td>Modulation rate</td><td>Set value</td><td>Range of division ratio</td><td>Set value lower limit</td><td>Set value upper limit</td></tr>
</table></td></tr>
</table>

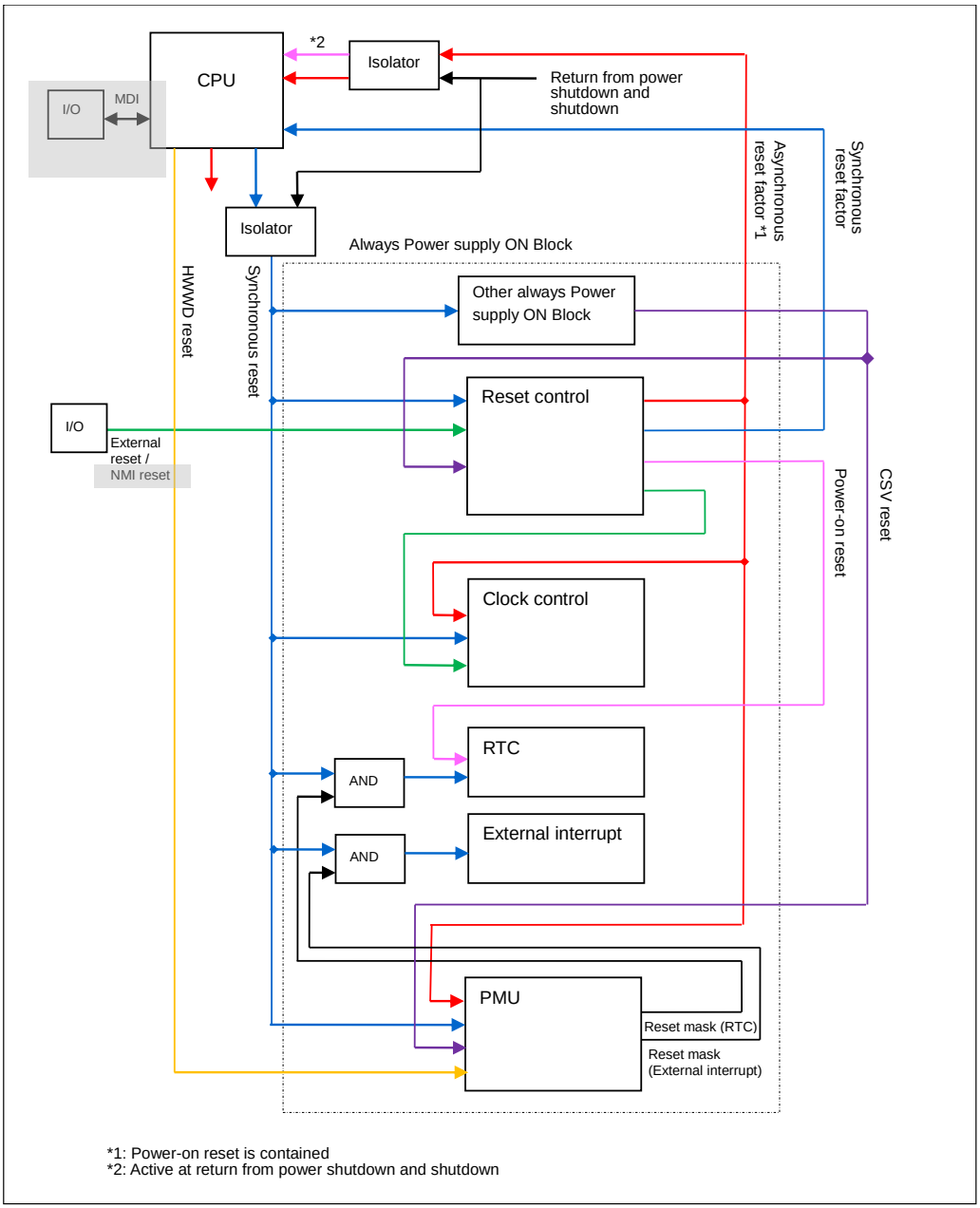
Page	Section	Change Results
236	5.5.7	The description in "5.7. Operations during MDI communications" should be modified as follows: (Error) Moreover, during MDI high speed communication (E_DBCR.PLOCK=1), the main oscillation is controlled so that the PLL reference clock is supplied even if CSELR.PCEN is cleared. The value of the register related to PLL is maintained and not updated. However, when software sets PLLCR.PCEN=0, the value of the register related to PLL can be freely updated (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. When you change the setting value in the debug, monitor the value of E_DBCR.PLOCK and rewrite the register related to PLL in the status of E_DBCR.PLOCK =0. (Correct) Moreover, during MDI high speed communication, the main oscillation is controlled so that the PLL reference clock is supplied even if CSELR.PCEN is cleared. The value of the register related to PLL is maintained and not updated. However, when software sets PLLCR.PCEN=0, the value of the register related to PLL can be freely updated (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.
237	5.5.8	The description in "5.8. About PMU clock (PMUCLK)" should be modified as follows: (Error) (3) G-divider must be set so that PMU clock frequency become 1/4 of the peripheral clock frequency (PCLK1). (Correct) (3) G-divider must be set so that PMU clock frequency becomes 1/4 or less of the peripheral clock frequency (PCLK1).

Page	Section	Change Results
238	5.5.8	The description in "5.8. About PMU clock (PMUCLK)" should be modified as follows: (Error) (2) Set F-divider so that the PMU clock frequency 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. (Correct) (2) Set F-divider so that the PMU clock frequency becomes 32 kHz or less. The PMU clock is used to control the power switch. Specify a frequency of 32 kHz or less for reasons of stabilization at times of rising pressure such as when the power supply is input etc.
238	5.5.8	The description in "5.8. About PMU clock (PMUCLK)" should be modified as follows: (Error) (3) Set G-divider so that PMU clock frequency becomes 1/4 of the peripheral clock frequency (PCLK1). Clock transfer between peripheral clock (PCLK1) and PMU clock (PMUCLK) needs 4 PMU clock cycles. When the source clock of peripheral clock(PCLK1) is sub oscillation clock (CMONR.CKM="10"), set the CCPMUCR1.GDIV register so that the frequency of peripheral clock(PCLK1) is quadruple (or higher) the frequency of PMU clock. (Correct) (3) Set G-divider so that PMU clock frequency becomes 1/4 or less of the peripheral clock frequency (PCLK1). Clock transfer between peripheral clock (PCLK1) and PMU clock (PMUCLK) needs 4 PMU clock cycles. When the source clock of the peripheral clock (PCLK1) is the sub oscillation clock (CMONR.CKM="10"), set the CCPMUCR1.GDIV register so that the frequency of the peripheral clock (PCLK1) is quadruple or higher the frequency of the PMU clock.

Page	Section	Change Results
242	6.2	Figure 2-1 Diagram of Device State Transitions should be modified as follows: (Error) 

Page	Section	Change Results
242	6.2	(Continued) (Correct) 
244	6.2.2	The following should be added to ■ Stop State in "2.2. Explanation of Each State". (Error) Transits to the oscillation stabilization wait RUN state by NMI interrupt. (Correct) Transits to the oscillation stabilization wait RUN state by NMI interrupt or the external interrupt.

Page	Section	Change Results
252	7.3	Figure 3-1 Configuration Diagram of Reset in "3. Configuration" should be modified as follows: (Error) 

Page	Section	Change Results
252	7.3	(Continued) (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>256</td><td>7.4.1</td><td>The following description should be added to the table in [bit6] ERST (External ReSeT) in "4.1. Reset Source Register : RSTRR (ReSeT Result Register)".

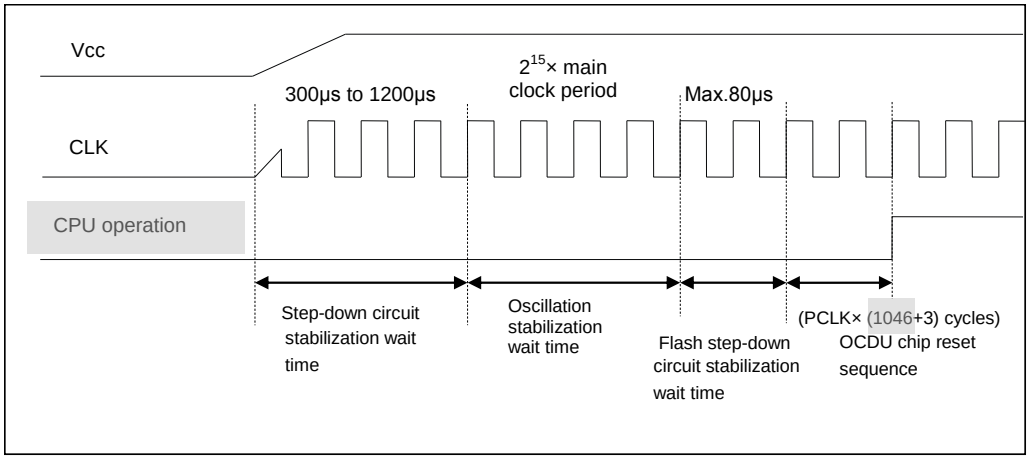
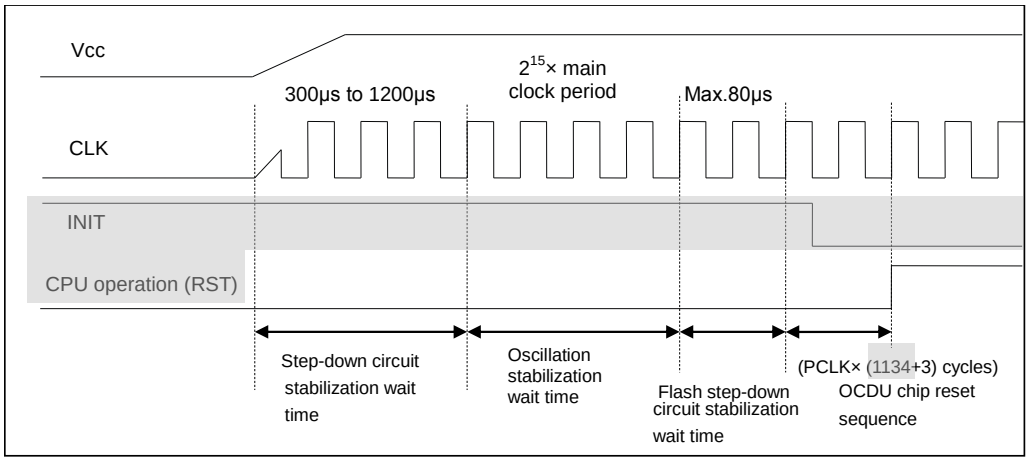
(Error)
<table><tr><td>ERST</td><td>RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset or simultaneous assert of RSTX and NMIX external pins</td></tr><tr><td>0</td><td>Undetected</td></tr><tr><td>1</td><td>Detected</td></tr></table>
(Correct)
<table><tr><td>ERST</td><td>RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset, low-voltage detection (external low-voltage detection) or simultaneous assertion of RSTX and NMIX external pins</td></tr><tr><td>0</td><td>Undetected</td></tr><tr><td>1</td><td>Detected</td></tr></table></td></tr>
<tr><td>260</td><td>7.4.4</td><td>The description in [bit7] PMUST in "4.4. PMU Status Register : PMUSTR (Power Management Unit Status register)" should be modified as follows:

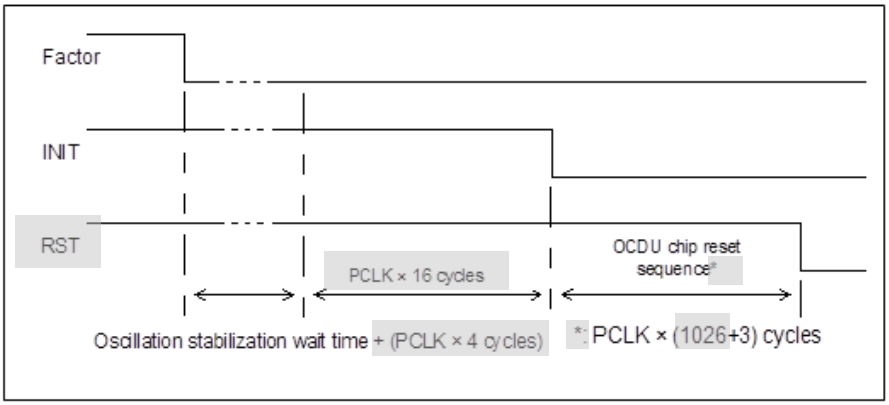
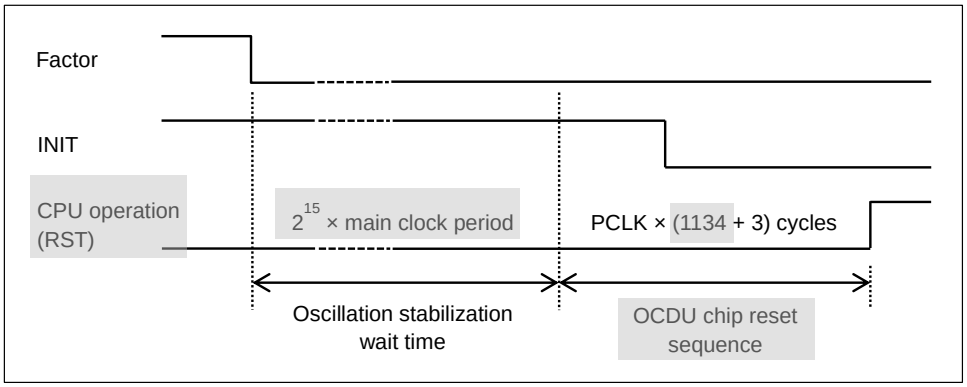
(Error)
This bit is cleared by writing "0". Writing "1" to this bit is ignored.

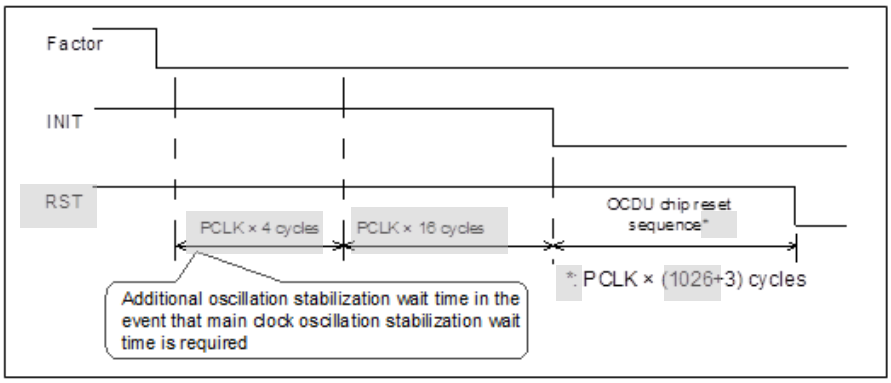
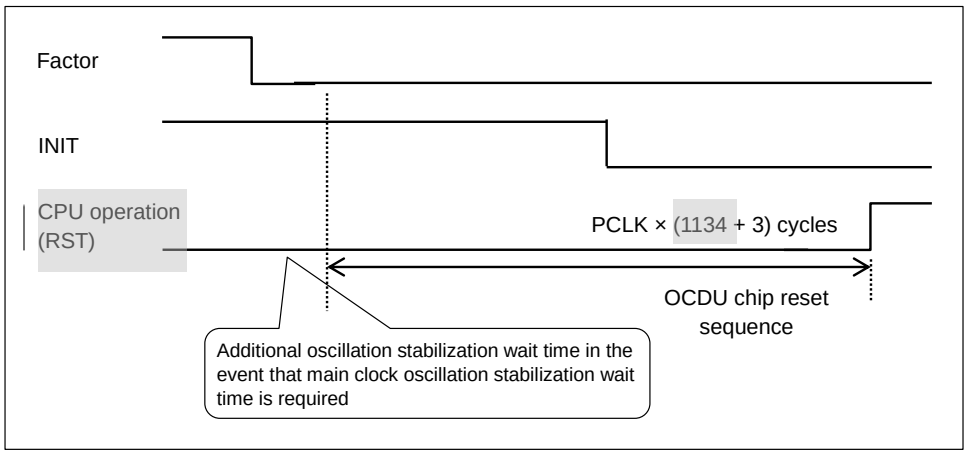
(Correct)
This bit is cleared by writing "0". Writing "1" to this bit is ignored.
This bit is initialized only by power-on reset, low-voltage detection reset, and simultaneous assertion of RSTX and NMIX. So, check other reset factors before checking the recovery from shutdown using this bit.</td></tr>
<tr><td>260</td><td>7.4.4</td><td>The description in [bit1] PONR_F in "4.4. PMU Status Register : PMUSTR (Power Management Unit Status register)" should be modified as follows:

(Error)
This bit is a power-on reset detection flag.

(Correct)
This bit is a power-on reset or internal low-voltage detection reset detection flag.</td></tr>
</table>

Page	Section	Change Results
282	7.5.4.1	Figure 5-1 Oscillation Stabilization Wait Time for Power-on Reset should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
283	7.5.4.1	Figure 5-2 Super Initialize Reset (SINIT) Sequence should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
284	7.5.4.2	Figure 5-3 Initialize Reset (INIT) Sequence should be modified as follows: (Error)  (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>309</td><td>8.4.2</td><td>Items in the table in [bit3 to bit0] in "4.2 DMA Channel Control Register 0 to 15 : DCCR0 to 15" should be modified.

(Error)
<table><tr><td>BLK[3:0]</td><td>Block size</td></tr><tr><td>0000</td><td>Once (initial value)</td></tr><tr><td>0001</td><td>Twice</td></tr></table>
(Correct)
<table><tr><td>BLK[3:0]</td><td>Transfer count</td></tr><tr><td>0000</td><td>Once (initial value)</td></tr><tr><td>0001</td><td>Twice</td></tr></table></td></tr>
<tr><td>313</td><td>8.4.5</td><td>The description in [bit31 to bit0] in "4.5. DMA Transfer Source Register 0 to 15 : DSAR0 to 15 (DMA Source Address Register 0 to 15)" should be modified as follows:

(Error)
If the DMA transfer request source has a peripheral interrupt (DCCR.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)".

(Correct)
If the DMA transfer request source has a peripheral interrupt (DCCRx.RS[1:0]=01), at least either the transfer source address (DSAR) or the transfer destination address (DDAR) must be within the address range of peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus. For details, see "■ Setting the ST Bit (Transfer source type) and DT Bit (Transfer destination type)".</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>328</td><td>8.5.1.3</td><td>The rightmost column should be deleted from Table 5-2 Relationship between Transfer Request Detection Conditions and Transfer Mode in "5.1.3. Operations".

(Error)
<table>
<tr><td></td><td>Block transfer</td><td>Burst transfer</td><td>-</td></tr>
<tr><td>Request by software</td><td>Set the DCCRx.CE bit to "1".</td><td>Set the DCCRx.CE bit to "1".</td><td>-</td></tr>
<tr><td>Request by interrupt</td><td>Edge detection</td><td>Edge detection</td><td>-</td></tr>
</table>
(Correct)
<table>
<tr><td></td><td>Block transfer</td><td>Burst transfer</td></tr>
<tr><td>Request by software</td><td>Set the DCCRx.CE bit to "1".</td><td>Set the DCCRx.CE bit to "1".</td></tr>
<tr><td>Request by interrupt</td><td>Edge detection</td><td>Edge detection</td></tr>
</table></td></tr>
<tr><td>352</td><td>9.4.5</td><td>The table in [bit2 to bit0] in "4.5. DMA Request Clear Register 5 : ICSEL5 (Interrupt Clear SELect register 5)" should be modified as follows:

(Error)
<table>
<tr><th>SG_RX_SEL1[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>001</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>010</td><td>16-bit free-run timer 0 zero detection</td></tr>
<tr><td>011</td><td>16-bit free-run timer 0 compare clear</td></tr>
<tr><td>100</td><td>Multi-function serial ch.7 transmission completion</td></tr>
<tr><td>101 to 111</td><td>Reserved(Does not clear any)</td></tr>
</table>
(Correct)
<table>
<tr><th>SG_RX_SEL1[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>001</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>010</td><td>16-bit free-run timer 0 zero detection</td></tr>
<tr><td>011</td><td>16-bit free-run timer 0 compare clear</td></tr>
<tr><td>100</td><td>Multi-function serial ch.7 transmission completion</td></tr>
<tr><td>101 to 111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>353</td><td>9.4.6</td><td>The table in [bit3 to bit0] in "4.6. DMA Request Clear Register 6 : ICSEL6 (Interrupt Clear SELect register 6)" should be modified as follows:

(Error)
<table>
<tr><th>PPGSEL0[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>PPG0</td></tr>
<tr><td>0001</td><td>PPG1</td></tr>
<tr><td>0010</td><td>PPG10</td></tr>
<tr><td>0011</td><td>PPG11</td></tr>
<tr><td>0100</td><td>PPG20</td></tr>
<tr><td>0101</td><td>PPG21</td></tr>
<tr><td>0110</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>0111</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>1000</td><td>16-bit free-run timer 1 zero detection</td></tr>
<tr><td>1001</td><td>16-bit free-run timer 1 compare clear</td></tr>
<tr><td>1010 to 1111</td><td>Reserved(Does not clear any)</td></tr>
</table>
(Correct)
<table>
<tr><th>PPGSEL0[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>PPG0</td></tr>
<tr><td>0001</td><td>PPG1</td></tr>
<tr><td>0010</td><td>PPG10</td></tr>
<tr><td>0011</td><td>PPG11</td></tr>
<tr><td>0100</td><td>PPG20</td></tr>
<tr><td>0101</td><td>PPG21</td></tr>
<tr><td>0110</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>0111</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>1000</td><td>16-bit free-run timer 1 zero detection</td></tr>
<tr><td>1001</td><td>16-bit free-run timer 1 compare clear</td></tr>
<tr><td>1010 to 1111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>354</td><td>9.4.7</td><td>The table in [bit3 to bit0] in "4.7. DMA Request Clear Register 7 : ICSEL7 (Interrupt Clear SELect register 7)" should be modified as follows:

(Error)
<table>
<tr><th>PPGSEL1[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>PPG2</td></tr>
<tr><td>0001</td><td>PPG3</td></tr>
<tr><td>0010</td><td>PPG12</td></tr>
<tr><td>0011</td><td>PPG13</td></tr>
<tr><td>0100</td><td>PPG22</td></tr>
<tr><td>0101</td><td>PPG23</td></tr>
<tr><td>0110</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>0111</td><td>Reserved(Does not clear any)</td></tr>
<tr><td>1000</td><td>16-bit free-run timer 2 zero detection</td></tr>
<tr><td>1001</td><td>16-bit free-run timer 2 compare clear</td></tr>
<tr><td>1010 to 1111</td><td>Reserved(Does not clear any)</td></tr>
</table>
(Correct)
<table>
<tr><th>PPGSEL1[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>PPG2</td></tr>
<tr><td>0001</td><td>PPG3</td></tr>
<tr><td>0010</td><td>PPG12</td></tr>
<tr><td>0011</td><td>PPG13</td></tr>
<tr><td>0100</td><td>PPG22</td></tr>
<tr><td>0101</td><td>PPG23</td></tr>
<tr><td>0110</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>0111</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>1000</td><td>16-bit free-run timer 2 zero detection</td></tr>
<tr><td>1001</td><td>16-bit free-run timer 2 compare clear</td></tr>
<tr><td>1010 to 1111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>358</td><td>9.4.11</td><td>The table in [bit2 to bit0] in "4.11. DMA Request Clear Register 11 : ICSEL11 (Interrupt Clear SELect register 11)" should be modified as follows:

<table>
<tr><th>PMSTSEL[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Main timer</td></tr>
<tr><td>001</td><td>Sub timer</td></tr>
<tr><td>010</td><td>PLL timer</td></tr>
<tr><td>011</td><td>Multi-function serial ch.8 transmission completion</td></tr>
<tr><td>100</td><td>16bit ICU2</td></tr>
<tr><td>101</td><td>16bit ICU3</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>PMSTSEL[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Main timer</td></tr>
<tr><td>001</td><td>Sub timer</td></tr>
<tr><td>010</td><td>PLL timer</td></tr>
<tr><td>011</td><td>Multi-function serial ch.8 transmission completion</td></tr>
<tr><td>100</td><td>16-bit ICU2</td></tr>
<tr><td>101</td><td>16-bit ICU3</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
<tr><td>359</td><td>9.4.12</td><td>The table in [bit1 to bit0] in "4.12. DMA Request Clear Register 13 : ICSEL13 (Interrupt Clear SELect register 13)" should be modified as follows:

<table>
<tr><th>ICUSEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>01</td><td>32bit ICU CH6</td></tr>
<tr><td>10</td><td>Multi-function serial ch.10 reception completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>ICUSEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>01</td><td>32-bit ICU ch.6</td></tr>
<tr><td>10</td><td>Multi-function serial ch.10 reception completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>360</td><td>9.4.13</td><td>The table in [bit1 to bit0] in "4.13. DMA Request Clear Register 14 : ICSEL14 (Interrupt Clear SELect register 14)" should be modified as follows:

<table>
<tr><th>ICUSEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>01</td><td>32bit ICU CH7</td></tr>
<tr><td>10</td><td>Multi-function serial ch.10 transmission completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>ICUSEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>01</td><td>32-bit ICU ch.7</td></tr>
<tr><td>10</td><td>Multi-function serial ch.10 transmission completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
<tr><td>361</td><td>9.4.14</td><td>The table in [bit1 to bit0] in "4.14. DMA Request Clear Register 15 : ICSEL15 (Interrupt Clear SELect register 15)" should be modified as follows:

<table>
<tr><th>ICUSEL2[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>01</td><td>32bit ICU CH8</td></tr>
<tr><td>10</td><td>Multi-function serial ch.11 reception completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>ICUSEL2[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>01</td><td>32-bit ICU ch.8</td></tr>
<tr><td>10</td><td>Multi-function serial ch.11 reception completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>362</td><td>9.4.15</td><td>The table in [bit3 to bit0] in "4.15. DMA Request Clear Register 16 : ICSEL16 (Interrupt Clear SELect register 16)" should be modified as follows:

(Error)
<table>
<tr><th>ICUSEL3[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>0001</td><td>32bit ICU CH9</td></tr>
<tr><td>0010</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0011</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0100</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0101</td><td>WG dead timer reload 0</td></tr>
<tr><td>0110</td><td>WG dead timer reload 1</td></tr>
<tr><td>0111</td><td>WG dead timer reload 2</td></tr>
<tr><td>1000</td><td>WG DTTI0</td></tr>
<tr><td>1001 to 1111</td><td>Any clearness is not selected</td></tr>
</table>
(Correct)
<table>
<tr><th>ICUSEL3[3:0]</th><th>Clear target</th></tr>
<tr><td>0000</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>0001</td><td>32-bit ICU ch.9</td></tr>
<tr><td>0010</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0011</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0100</td><td>WG dead timer underflow 0</td></tr>
<tr><td>0101</td><td>WG dead timer reload 0</td></tr>
<tr><td>0110</td><td>WG dead timer reload 1</td></tr>
<tr><td>0111</td><td>WG dead timer reload 2</td></tr>
<tr><td>1000</td><td>WG DTTI0</td></tr>
<tr><td>1001 to 1111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>363</td><td>9.4.16</td><td>The table in [bit1 to bit0] in "4.16. DMA Request Clear Register 17 : ICSEL17 (Interrupt Clear SELect register 17)" should be modified as follows:

<table>
<tr><th>ICUSEL4[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>32bit ICU CH4</td></tr>
<tr><td>01</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>10</td><td>Multi-function serial ch.11 transmission completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>ICUSEL4[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>32-bit ICU ch.4</td></tr>
<tr><td>01</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>10</td><td>Multi-function serial ch.11 transmission completion</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>364</td><td>9.4.17</td><td>The table in [bit4 to bit0] in "4.17. DMA Request Clear Register 18 : ICSEL18 (Interrupt Clear SELect register 18)" should be modified as follows:

(Error)
<table>
<tr><th>ICUSEL5[4:0]</th><th>Clear target</th></tr>
<tr><td>00000</td><td>32bit ICU CH5</td></tr>
<tr><td>00001</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>00010</td><td>A/D converter ch.32</td></tr>
<tr><td>00011</td><td>A/D converter ch.33</td></tr>
<tr><td>00100</td><td>A/D converter ch.34</td></tr>
<tr><td>00101</td><td>A/D converter ch.35</td></tr>
<tr><td>00110</td><td>A/D converter ch.36</td></tr>
<tr><td>00111</td><td>A/D converter ch.37</td></tr>
<tr><td>01000</td><td>A/D converter ch.38</td></tr>
<tr><td>01001</td><td>A/D converter ch.39</td></tr>
<tr><td>01010</td><td>A/D converter ch.40</td></tr>
<tr><td>01011</td><td>A/D converter ch.41</td></tr>
<tr><td>01100</td><td>A/D converter ch.42</td></tr>
<tr><td>01101</td><td>A/D converter ch.43</td></tr>
<tr><td>01110</td><td>A/D converter ch.44</td></tr>
<tr><td>01111</td><td>A/D converter ch.45</td></tr>
<tr><td>10000</td><td>A/D converter ch.46</td></tr>
<tr><td>10001</td><td>A/D converter ch.47</td></tr>
<tr><td>10010 to 11111</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>ICUSEL5[4:0]</th><th>Clear target</th></tr>
<tr><td>00000</td><td>32-bit ICU ch.5</td></tr>
<tr><td>00001</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>00010</td><td>A/D converter ch.32</td></tr>
<tr><td>00011</td><td>A/D converter ch.33</td></tr>
<tr><td>00100</td><td>A/D converter ch.34</td></tr>
<tr><td>00101</td><td>A/D converter ch.35</td></tr>
<tr><td>00110</td><td>A/D converter ch.36</td></tr>
<tr><td>00111</td><td>A/D converter ch.37</td></tr>
<tr><td>01000</td><td>A/D converter ch.38</td></tr>
<tr><td>01001</td><td>A/D converter ch.39</td></tr>
<tr><td>01010</td><td>A/D converter ch.40</td></tr>
<tr><td>01011</td><td>A/D converter ch.41</td></tr>
<tr><td>01100</td><td>A/D converter ch.42</td></tr>
<tr><td>01101</td><td>A/D converter ch.43</td></tr>
<tr><td>01110</td><td>A/D converter ch.44</td></tr>
<tr><td>01111</td><td>A/D converter ch.45</td></tr>
<tr><td>10000</td><td>A/D converter ch.46</td></tr>
<tr><td>10001</td><td>A/D converter ch.47</td></tr>
<tr><td>10010 to 11111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>365</td><td>9.4.18</td><td>The table in [bit2 to bit0] in "4.18. DMA Request Clear Register 19 : ICSEL19 (Interrupt Clear SELect register 19)" should be modified as follows:

<table>
<tr><th>OCUSEL0[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>001</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>010</td><td>32bit OCU6</td></tr>
<tr><td>011</td><td>32bit OCU7</td></tr>
<tr><td>100</td><td>32bit OCU10</td></tr>
<tr><td>101</td><td>32bit OCU11</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>OCUSEL0[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>001</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>010</td><td>32-bit OCU6</td></tr>
<tr><td>011</td><td>32-bit OCU7</td></tr>
<tr><td>100</td><td>32-bit OCU10</td></tr>
<tr><td>101</td><td>32-bit OCU11</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>366</td><td>9.4.19</td><td>The table in [bit2 to bit0] in "4.19. DMA Request Clear Register 20 : ICSEL20 (Interrupt Clear SELect register 20)" should be modified as follows:

<table>
<tr><th>OCUSEL1[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>001</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>010</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>011</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>100</td><td>32bit OCU8</td></tr>
<tr><td>101</td><td>32bit OCU9</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>OCUSEL1[2:0]</th><th>Clear target</th></tr>
<tr><td>000</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>001</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>010</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>011</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>100</td><td>32-bit OCU8</td></tr>
<tr><td>101</td><td>32-bit OCU9</td></tr>
<tr><td>110</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>111</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
<tr><td>367</td><td>9.4.20</td><td>The table in [bit1 to bit0] in "4.20. DMA Request Clear Register 21 : ICSEL21 (Interrupt Clear SELect register 21)" should be modified as follows:

<table>
<tr><th>BT_SG_SEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Base Timer0 IRQ0</td></tr>
<tr><td>01</td><td>Base Timer0 IRQ1</td></tr>
<tr><td>10</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>BT_SG_SEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Base Timer0 IRQ0</td></tr>
<tr><td>01</td><td>Base Timer0 IRQ1</td></tr>
<tr><td>10</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>368</td><td>9.4.21</td><td>The table in [bit1 to bit0] in "4.21 DMA Request Clear Register 22 : ICSEL22 (Interrupt Clear SELect register 22)" should be modified as follows:

<table>
<tr><th>BT_SG_SEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Base Timer1 IRQ0</td></tr>
<tr><td>01</td><td>Base Timer1 IRQ1</td></tr>
<tr><td>10</td><td>Reserved (Does not clear any)</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>BT_SG_SEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Base Timer1 IRQ0</td></tr>
<tr><td>01</td><td>Base Timer1 IRQ1</td></tr>
<tr><td>10</td><td>Reserved (Does not clear any interrupt)</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
<tr><td>369</td><td>9.4.22</td><td>The table in [bit1 to bit] in "4.22 DMA Request Clear Register 23 : ICSEL23 (Interrupt Clear SELect register 23)" should be modified as follows:

<table>
<tr><th>MFS_SEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Multi-function serial ch.8 reception completion</td></tr>
<tr><td>01</td><td>16bit ICU0</td></tr>
<tr><td>10</td><td>16bit ICU1</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>MFS_SEL0[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Multi-function serial ch.8 reception completion</td></tr>
<tr><td>01</td><td>16-bit ICU0</td></tr>
<tr><td>10</td><td>16-bit ICU1</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>370</td><td>9.4.23</td><td>The table in [bit1 to bit0] in "4.23 DMA Request Clear Register 24 : ICSEL24 (Interrupt Clear SELect register 24)" should be modified as follows:

<table>
<tr><th>MFS_SEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Multi-function serial ch.9 transmission completion</td></tr>
<tr><td>01</td><td>16bit OCU0</td></tr>
<tr><td>10</td><td>16bit OCU1</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any)</td></tr>
</table>

(Correct)
<table>
<tr><th>MFS_SEL1[1:0]</th><th>Clear target</th></tr>
<tr><td>00</td><td>Multi-function serial ch.9 transmission completion</td></tr>
<tr><td>01</td><td>16-bit OCU0</td></tr>
<tr><td>10</td><td>16-bit OCU1</td></tr>
<tr><td>11</td><td>Reserved (Does not clear any interrupt)</td></tr>
</table>
</td></tr>
<tr><td>397</td><td>11.4</td><td>A comment should be added to the register function with the address: 0x0F40 PORTEN in "4. Registers".

(Error)
<table>
<tr><td>0x0F40</td><td>PORTEN</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Port input enable register</td></tr>
</table>

(Correct)
<table>
<tr><td>0x0F40</td><td>PORTEN</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Port input enable register
(Key code target registers)</td></tr>
</table>
</td></tr>
<tr><td>398</td><td>11.4.1</td><td>The description in [bit7 to bit0] P (Port) in "4.1. Port Data Register 00 to 19 : PDR00 to 19 (Port Data Register 00 to 19)" should be modified as follows:

(Error)
PDR0.P[7:0] is for external pins P007 to P000
PDR1.P[7:0] is for external pins P017 to P010
PDR2.P[7:0] is for external pins P027 to P020

(Correct)
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</td></tr>
</table>

Page	Section	Change Results
398	11.4.1	The description in "4.1. Port Data Register 00 to 19 : PDR00 to 19 (Port Data Register 00 to 19)" should be modified as follows: (Error) For details of which port is missing, see "1.16 Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW" (Correct) For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER: OVERVIEW".
399	11.4.2	The description in [bit7 to bit0] P (Port) in "4.2. Data Direction Register 00 to 19 : DDR00 to 19 (Data Direction Register 00 to 9)" should be modified as follows: (Error) DDR0.P[7:0] is for external pins P007 to P000 DDR1.P[7:0] is for external pins P017 to P010 DDR2.P[7:0] is for external pins P027 to P020 (Correct) 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
399	11.4.2	The description in "4.2. Data Direction Register 00 to 19 : DDR00 to 19 (Data Direction Register 00 to 19)" should be modified as follows: (Error) For details of which port is missing, see "1.16 Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". (Correct) For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER: OVERVIEW".

Page	Section	Change Results
400	11.4.3	The description in [bit7 to bit0] P (Port) in "4.3. Port Function Register 00 to 19 : PFR00 to 19 (Port Function Register 00 to 19)" should be modified as follows: (Error) PFR0.P[7:0] is for external pins P007 to P000 PFR1.P[7:0] is for external pins P017 to P010 PFR2.P[7:0] is for external pins P027 to P020 (Correct) 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
400	11.4.3	The description in "4.3. Port Function Register 00 to 19 : PFR00 to 19 (Port Function Register 00 to 19)" should be modified as follows: (Error) For details of which port is missing, see "1.16 Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". (Correct) For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER: OVERVIEW".
401	11.4.4	The description in [bit7 to bit0] P (Port) in "4.4. Input Data Direct Register 00 to 19 : PDDR00 to 19 (Port Data Direct Register 00 to 19)" should be modified as follows: (Error) PDDR0.P[7:0] is for external pins P007 to P000 PDDR1.P[7:0] is for external pins P017 to P010 PDDR2.P[7:0] is for external pins P027 to P020 (Correct) 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

Page	Section	Change Results
401	11.4.4	The description in "4.4. Input Data Direct Register 00 to 19 : PDDR00 to 19 (Port Data Direct Register 00 to 19)" should be modified as follows: (Error) For details of which port is missing, see "1.16 Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". (Correct) For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER: OVERVIEW".
402	11.4.5	The description in [bit7 to bit0] P (Port) in "4.5. Port Pull-up/down Enable Register 00 to 19 : PPER00 to 19 (Port Pull-up/down Enable Register 00 to 19)" should be modified as follows: (Error) PPER0.P[7:0] is for external pins P007 to P000 PPER1.P[7:0] is for external pins P017 to P010 PPER2.P[7:0] is for external pins P027 to P020 (Correct) 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
402	11.4.5	The description in "4.5. Port Pull-up/down Enable Register 00 to 19 : PPER00 to 19 (Port Pull-up/down Enable Register 00 to 19)" should be modified as follows: (Error) For details of which port is missing, see "1.16 Port Function (General-Purpose I/O) Pins" in "CHAPTER:OVERVIEW". (Correct) For details of which port is missing, see "Port Function (General-Purpose I/O) Pins" in "CHAPTER: OVERVIEW".

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>421</td><td>11.4.6.7</td><td>
The table in TIAnE[1:0](n=0, 1) in "4.6.7. Extended Port Function Register 26 : EPFR26" should be modified as follows:

(Error)
TIAnE[1:0] (n=0, 1) : Base Timer TIOA0 output pin select
<table>
<tr><th>TIAnE[1:0] (n=0, 1)</th><th>Operation</th></tr>
<tr><td>00</td><td>Base timer TIOAn_0 output disabled Input from the base timer TIOA1_0 (Initial value)</td></tr>
<tr><td>01</td><td>Base timer TIOAn_0 output enabled Input from the base timer TIOA1_0 (Initial value)</td></tr>
<tr><td>1x</td><td>Base timer TIOAn_1 output enabled Input from the base timer TIOA1_1 (Initial value)</td></tr>
</table>
(Correct)
TIA0E[1:0]: Base Timer TIOA0 output pin select
<table>
<tr><th>TIA0E[1:0]</th><th>Operation</th></tr>
<tr><td>00</td><td>Base timers TIOA0_0, TIOA0_1 output disabled (Initial value)</td></tr>
<tr><td>01</td><td>Base timer TIOA0_0 output enabled</td></tr>
<tr><td>1x</td><td>Base timer TIOA0_1 output enabled</td></tr>
</table>
TIA1E[1:0]: Base Timer TIOA1 output/input pin select
<table>
<tr><th>TIA1E[1:0]</th><th>Operation</th></tr>
<tr><td>00</td><td>Base timers TIOA1_0, TIOA1_1 output disabled, Input from the base timer TIOA1_0 (Initial value)</td></tr>
<tr><td>01</td><td>Base timer TIOA1_0 output enabled, Input from the base timer TIOA1_0</td></tr>
<tr><td>1x</td><td>Base timer TIOA1_1 output enabled, Input from the base timer TIOA1_1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>432</td><td>11.4.6.16</td><td>The content of DTTI0E operation in the table of ■ EPFR85 : Address 01CD_H (Access: Byte, Half-word, Word) in "4.6.16. Extended Port Function Register 84, 85 : EPFR84,85(Extended Port Function Register84, 85)" should be modified as follows:

(Error)
<table>
<tr><th>DTTI0E</th><th>Operation</th></tr>
<tr><td>00</td><td>Input from the DTTI0_0 pin (Initial value)</td></tr>
<tr><td>01</td><td>Input from the DTTI0_1 pin</td></tr>
<tr><td>1x</td><td>Input from the DTTI0_2 pin</td></tr>
</table>
(Correct)
<table>
<tr><th>DTTI0E</th><th>Operation</th></tr>
<tr><td>00</td><td>Input from the DTTI_0 pin (Initial value)</td></tr>
<tr><td>01</td><td>Input from the DTTI_1 pin</td></tr>
<tr><td>1x</td><td>Input from the DTTI_2 pin</td></tr>
</table></td></tr>
<tr><td>438</td><td>11.5</td><td>The following section should be added to the chapter field in the explanation of "5. Operation".

(Correct)
5.11. Input blocked when specific peripheral functions are used</td></tr>
<tr><td>451</td><td>11.5.4</td><td>The content written in the note in "5.4. Noise Filter" should be modified as follows:

(Error)
Please refer to the terminal list of each function of the chapter of Basic Information for details.

(Correct)
Please refer to "Pins of Each Function" in the chapter of "OVERVIEW" for details.</td></tr>
<tr><td>452</td><td>11.5.5</td><td>The following description should be added to "5.5. Input blocked by GPORTEN".

(Correct)
When the state of a pin to be the input blocked state is read during the input blocked by GPORTEN, "0" is always read out.</td></tr>
</table>

Page	Section	Change Results
453	11.5.6	The description in "5.6. Notes on Pins with the A/D Converter Function " should be modified as follows: (Error) When using a pin with the A/D converter function to perform a different function, set the relevant bit of the A/D converter analog input enable register (ADER) to "Analog input disable" in advance. (Correct) When using a pin with the A/D converter function to perform a different function (digital port, peripheral function), set the relevant bit of the A/D converter analog input enable register (ADER) to "Analog input disable" in advance. In this case an A/D conversion should not be done on this analog input, because the digital inputs of this port pin are fixed at "0" during A/D conversion.
455	11.5.8	The content of "5.8. Key Code Register Function Settings" in "5. Operation" should be modified to add the following information. (Error) ·The DDR, PFR, EPFR, PPER, ADER, and DACR are the target key code registers. (Correct) · The DDR, PFR, EPFR, PPER, PORTEN, ADER, and DACR are the target key code registers.
458	11.5.11	Section "5.11. Input blocked when specific peripheral functions are used" should be added and the following descriptions should be added in the next page of "5.10. Notes on switching the I/O port function". (Correct) 5.11. Input blocked when specific peripheral functions are used A note regarding blocked input when specific peripheral functions are used is shown below: When a pin is used as the A/D function and the state of the pin is read, "0" is always read.

Page	Section	Change Results
475	13.4	The table of external pins in "4. Registers" should be modified as follows: (Error)

Channel	Base_addr	External Pins
		INT
0	0x0550	INT0_0
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
6	0x0550	INT6_0
7	0x0550	INT7_0 / INT7_1
8	0x0554	INT8_0
9	0x0554	INT9_0 / INT9_1
10	0x0554	INT10_0
11	0x0554	INT11_0
12	0x0554	INT12_0
13	0x0554	INT13_0 / INT13_1
14	0x0554	INT14_0 / INT14_1
15	0x0554	INT15_0

Page	Section	Change Results
475	13.4	(Continued) (Correct) ■ List of External Pins

Channel	Base_addr	External Pins (INT)				
		MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK, MB91F52xL
0	0x0550	INT0_0	INT0_0	INT0_0	INT0_0	INT0_0
1	0x0550	INT1_1	INT1_1	INT1_0/ INT1_1	INT1_0/ INT1_1	INT1_0/ INT1_1
2	0x0550	INT2_1	INT2_1	INT2_0/ INT2_1	INT2_0/ INT2_1	INT2_0/ INT2_1
3	0x0550	INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1	INT3_0/ INT3_1
4	0x0550	INT4_1	INT4_1	INT4_1	INT4_0/ INT4_1	INT4_0/ INT4_1
5	0x0550	INT5_0	INT5_0	INT5_0	INT5_0	INT5_0
6	0x0550	INT6_0	INT6_0	INT6_0	INT6_0	INT6_0
7	0x0550	INT7_1	INT7_1	INT7_0/ INT7_1	INT7_0/ INT7_1	INT7_0/ INT7_1
8	0x0554	INT8_0	INT8_0	INT8_0	INT8_0	INT8_0
9	0x0554	INT9_1	INT9_1	INT9_1	INT9_0/ INT9_1	INT9_0/ INT9_1
10	0x0554	INT10_0	INT10_0	INT10_0	INT10_0	INT10_0
11	0x0554	INT11_0	INT11_0	INT11_0	INT11_0	INT11_0
12	0x0554	INT12_0	INT12_0	INT12_0	INT12_0	INT12_0
13	0x0554	INT13_1	INT13_1	INT13_0/ INT13_1	INT13_0/ INT13_1	INT13_0/ INT13_1
14	0x0554	INT14_1	INT14_1	INT14_1	INT14_1	INT14_0/ INT14_1
15	0x0554	INT15_0	INT15_0	INT15_0	INT15_0	INT15_0

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>480</td><td>13.5</td><td>The content in 2. Transition to standby mode on Figure 5-2 Operation of External Interrupt in "5. Operation" should be modified as follows:

(Error)
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.

(Correct)
Channels not to be used should be moved to disable state before letting them go into the standby mode. External pins enter an input blocked state at standby mode, but external pins of external interrupt enabled channels enter an input enabled state.</td></tr>
<tr><td>490</td><td>14.5</td><td>The following sentence is added to the end of ■ Recovering From Stop Mode in "5. Operation".

(Error)
*: The watch mode and the watch mode (power-shutdown) are similarly controlled.

(Correct)
Note:
・The watch mode and the watch mode (power-shutdown) are similarly controlled.
・An internal reset is issued at the return from standby mode (power-shutdown), and no NMI request is accepted.</td></tr>
<tr><td>535</td><td>16.4.29</td><td>The following description should be added in "4.29. Interrupt Request Batch Read Register 15 upper-order : IRPR15H (Interrupt Request Peripheral Read register 15H)".

(Error)
<table><tr><th>Read value of each bit</th><th>Meaning</th></tr><tr><td>0</td><td>No interrupt request has been issued.</td></tr><tr><td>1</td><td>An interrupt request has been issued.</td></tr></table>
(Correct)
<table><tr><th>Read value of each bit</th><th>Meaning</th></tr><tr><td>0</td><td>No interrupt request has been issued.</td></tr><tr><td>1</td><td>An interrupt request has been issued.</td></tr></table>
An internal reset is issued at the return from standby mode (power-shutdown), no NMI request can be maintained.</td></tr>
</table>

Page	Section	Change Results
540	17.1	The following description is added to "1. Overview". (Correct) The numbers of available external output pins are shown below: MB91F52xB (64pin) : 21 MB91F52xD (80pin) : 27 MB91F52xF (100pin) : 34 MB91F52xJ (120pin) : 38 MB91F52xK (144pin) : 44 MB91F52xL (176pin) : 48

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>546</td><td>17.4</td><td>
The following list of external pins is added to "4. Registers".

(Correct)
■ List of External Pins
<table>
<tr><th rowspan="2">Channel</th><th colspan="6">External Pins (PPG Output)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD</th><th>MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK</th><th>MB91F52xL</th></tr>
<tr><td>0</td><td>No</td><td>No</td><td>No</td><td>PPG0_1</td><td>PPG0_0/
PPG0_1</td><td>PPG0_0/
PPG0_1</td></tr>
<tr><td>1</td><td>PPG1_0/
PPG1_1</td><td>PPG1_0/
PPG1_1</td><td>PPG1_0/
PPG1_1</td><td>PPG1_0/
PPG1_1</td><td>PPG1_0/
PPG1_1</td><td>PPG1_0/
PPG1_1</td></tr>
<tr><td>2</td><td>PPG2_0</td><td>PPG2_0</td><td>PPG2_0/
PPG2_1</td><td>PPG2_0/
PPG2_1</td><td>PPG2_0/
PPG2_1</td><td>PPG2_0/
PPG2_1</td></tr>
<tr><td>3</td><td>PPG3_1</td><td>PPG3_1</td><td>PPG3_1</td><td>PPG3_1</td><td>PPG3_0/
PPG3_1</td><td>PPG3_0/
PPG3_1</td></tr>
<tr><td>4</td><td>PPG4_1</td><td>PPG4_1</td><td>PPG4_1</td><td>PPG4_1</td><td>PPG4_0/
PPG4_1</td><td>PPG4_0/
PPG4_1</td></tr>
<tr><td>5</td><td>No</td><td>PPG5_1</td><td>PPG5_1</td><td>PPG5_1</td><td>PPG5_0/
PPG5_1</td><td>PPG5_0/
PPG5_1</td></tr>
<tr><td>6</td><td>No</td><td>No</td><td>PPG6_0</td><td>PPG6_0</td><td>PPG6_0</td><td>PPG6_0</td></tr>
<tr><td>7</td><td>PPG7_0</td><td>PPG7_0</td><td>PPG7_0</td><td>PPG7_0</td><td>PPG7_0</td><td>PPG7_0</td></tr>
<tr><td>8</td><td>No</td><td>PPG8_0</td><td>PPG8_0</td><td>PPG8_0</td><td>PPG8_0</td><td>PPG8_0</td></tr>
<tr><td>9</td><td>No</td><td>No</td><td>PPG9_0</td><td>PPG9_0</td><td>PPG9_0</td><td>PPG9_0</td></tr>
<tr><td>10</td><td>PPG10_0</td><td>PPG10_0</td><td>PPG10_0</td><td>PPG10_0</td><td>PPG10_0</td><td>PPG10_0</td></tr>
<tr><td>11</td><td>PPG11_0</td><td>PPG11_0</td><td>PPG11_0</td><td>PPG11_0</td><td>PPG11_0</td><td>PPG11_0</td></tr>
<tr><td>12</td><td>PPG12_0</td><td>PPG12_0</td><td>PPG12_0</td><td>PPG12_0</td><td>PPG12_0</td><td>PPG12_0</td></tr>
<tr><td>13</td><td>PPG13_0</td><td>PPG13_0</td><td>PPG13_0</td><td>PPG13_0</td><td>PPG13_0</td><td>PPG13_0</td></tr>
<tr><td>14</td><td>No</td><td>No</td><td>PPG14_0</td><td>PPG14_0</td><td>PPG14_0</td><td>PPG14_0</td></tr>
<tr><td>15</td><td>No</td><td>PPG15_0</td><td>PPG15_0</td><td>PPG15_0</td><td>PPG15_0</td><td>PPG15_0</td></tr>
<tr><td>16</td><td>PPG16_1</td><td>PPG16_1</td><td>PPG16_1</td><td>PPG16_0/
PPG16_1</td><td>PPG16_0/
PPG16_1</td><td>PPG16_0/
PPG16_1</td></tr>
<tr><td>17</td><td>PPG17_1</td><td>PPG17_1</td><td>PPG17_1</td><td>PPG17_0/
PPG17_1</td><td>PPG17_0/
PPG17_1</td><td>PPG17_0/
PPG17_1</td></tr>
<tr><td>18</td><td>No</td><td>PPG18_0</td><td>PPG18_0</td><td>PPG18_0</td><td>PPG18_0</td><td>PPG18_0</td></tr>
<tr><td>19</td><td>PPG19_0</td><td>PPG19_0</td><td>PPG19_0</td><td>PPG19_0</td><td>PPG19_0</td><td>PPG19_0</td></tr>
<tr><td>20</td><td>PPG20_0</td><td>PPG20_0</td><td>PPG20_0</td><td>PPG20_0</td><td>PPG20_0</td><td>PPG20_0</td></tr>
<tr><td>21</td><td>No</td><td>No</td><td>PPG21_0</td><td>PPG21_0</td><td>PPG21_0</td><td>PPG21_0</td></tr>
<tr><td>22</td><td>No</td><td>No</td><td>No</td><td>PPG22_0</td><td>PPG22_0</td><td>PPG22_0</td></tr>
<tr><td>23</td><td>PPG23_1</td><td>PPG23_1</td><td>PPG23_1</td><td>PPG23_1</td><td>PPG23_0/
PPG23_1</td><td>PPG23_0/
PPG23_1</td></tr>
<tr><td>24</td><td>PPG24_0</td><td>PPG24_0</td><td>PPG24_0</td><td>PPG24_0</td><td>PPG24_0</td><td>PPG24_0/
PPG24_1</td></tr>
<tr><td>25</td><td>No</td><td>No</td><td>PPG25_0</td><td>PPG25_0</td><td>PPG25_0</td><td>PPG25_0/
PPG25_1</td></tr>
<tr><td>26</td><td>No</td><td>PPG26_0</td><td>PPG26_0</td><td>PPG26_0</td><td>PPG26_0</td><td>PPG26_0/
PPG26_1</td></tr>
<tr><td>27</td><td>PPG27_0</td><td>PPG27_0</td><td>PPG27_0</td><td>PPG27_0</td><td>PPG27_0</td><td>PPG27_0/
PPG27_1</td></tr>
<tr><td>28</td><td>No</td><td>No</td><td>PPG28_0</td><td>PPG28_0</td><td>PPG28_0</td><td>PPG28_0/
PPG28_1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
546	17.4	(Continued)

Channel	External Pins (PPG Output)					
	MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK	MB91F52xL
29	No	PPG29_0	PPG29_0	PPG29_0	PPG29_0	PPG29_0/ PPG29_1
30	PPG30_0	PPG30_0	PPG30_0	PPG30_0	PPG30_0	PPG30_0/ PPG30_1
31	PPG31_0	PPG31_0	PPG31_0	PPG31_0	PPG31_0	PPG31_0/ PPG31_1
32	No	No	No	PPG32_0	PPG32_0	PPG32_0/ PPG32_1
33	No	No	No	PPG33_0	PPG33_0	PPG33_0/ PPG33_1
34	No	No	No	No	PPG34_0	PPG34_0/ PPG34_1
35	PPG35_0	PPG35_0	PPG35_0	PPG35_0	PPG35_0	PPG35_0/ PPG35_1
36	No	No	No	No	PPG36_0	PPG36_0/ PPG36_1
37	PPG37_0	PPG37_0	PPG37_0	PPG37_0	PPG37_0	PPG37_0/ PPG37_1
38	No	No	No	No	PPG38_0	PPG38_0/ PPG38_1
39	No	No	No	No	PPG39_0	PPG39_0/ PPG39_1
40	No	No	No	No	PPG40_1	PPG40_0/ PPG40_1
41	No	No	No	No	PPG41_1	PPG41_0/ PPG41_1
42	No	No	No	No	No	PPG42_0
43	PPG43_1	PPG43_1	PPG43_1	PPG43_1	PPG43_1	PPG43_0/ PPG43_1
44	No	No	PPG44_1	PPG44_1	PPG44_1	PPG44_0/ PPG44_1
45	No	No	No	No	No	PPG45_0
46	No	No	No	No	No	PPG46_0
47	No	No	No	No	No	PPG47_0

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>548</td><td>17.4</td><td>(Continued)
<table>
<tr><th rowspan="2">Channel</th><th colspan="6">External Pins (PPG Output)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD</th><th>MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK</th><th>MB91F52xL</th></tr>
<tr><td>0</td><td>TRG0_1/
TRG0_2</td><td>TRG0_1/
TRG0_2</td><td>TRG0_1/
TRG0_2</td><td>TRG0_1/
TRG0_2</td><td>TRG0_0/
TRG0_1/
TRG0_2</td><td>TRG0_0/
TRG0_1/
TRG0_2</td></tr>
<tr><td>1</td><td>No</td><td>No</td><td>TRG1_1</td><td>TRG1_1</td><td>TRG1_0/
TRG1_1</td><td>TRG1_0/
TRG1_1</td></tr>
<tr><td>2</td><td>No</td><td>TRG2_1</td><td>TRG2_1</td><td>TRG2_1</td><td>TRG2_0/
TRG2_1</td><td>TRG2_0/
TRG2_1</td></tr>
<tr><td>3</td><td>TRG3_0/
TRG3_1</td><td>TRG3_0/
TRG3_1</td><td>TRG3_0/
TRG3_1</td><td>TRG3_0/
TRG3_1</td><td>TRG3_0/
TRG3_1</td><td>TRG3_0/
TRG3_1</td></tr>
<tr><td>4</td><td>No</td><td>No</td><td>No</td><td>TRG4_0/
TRG4_1</td><td>TRG4_0/
TRG4_1</td><td>TRG4_0/
TRG4_1</td></tr>
<tr><td>5</td><td>No</td><td>No</td><td>No</td><td>TRG5_0/
TRG5_1</td><td>TRG5_0/
TRG5_1</td><td>TRG5_0/
TRG5_1/
TRG5_2</td></tr>
<tr><td>6</td><td>No</td><td>No</td><td>TRG6_0</td><td>TRG6_0/
TRG6_1</td><td>TRG6_0/
TRG6_1</td><td>TRG6_0/
TRG6_1/
TRG6_2</td></tr>
<tr><td>7</td><td>TRG7_0</td><td>TRG7_0</td><td>TRG7_0</td><td>TRG7_0/
TRG7_1</td><td>TRG7_0/
TRG7_1</td><td>TRG7_0/
TRG7_1</td></tr>
<tr><td>8</td><td>TRG8_0</td><td>TRG8_0</td><td>TRG8_0</td><td>TRG8_0</td><td>TRG8_0</td><td>TRG8_0/
TRG8_1</td></tr>
<tr><td>9</td><td>No</td><td>No</td><td>No</td><td>No</td><td>TRG9_0</td><td>TRG9_0/
TRG9_1</td></tr>
<tr><td>10</td><td>No</td><td>No</td><td>No</td><td>No</td><td>No</td><td>TRG10_0</td></tr>
<tr><td>11</td><td>No</td><td>No</td><td>No</td><td>No</td><td>No</td><td>TRG11_0</td></tr>
</table></td></tr>
<tr><td>567
to
572</td><td>17.4.8
to
17.4.13</td><td>The notes in sections from "4.8. PPG Communication Mode High Format Cycle Setting Register : PHCSR0 to PHCSR3" to "4.13. PPG Communication Mode Data Bit Length Setting Register : PCMDWD0 to PCMDWD3" should be deleted.

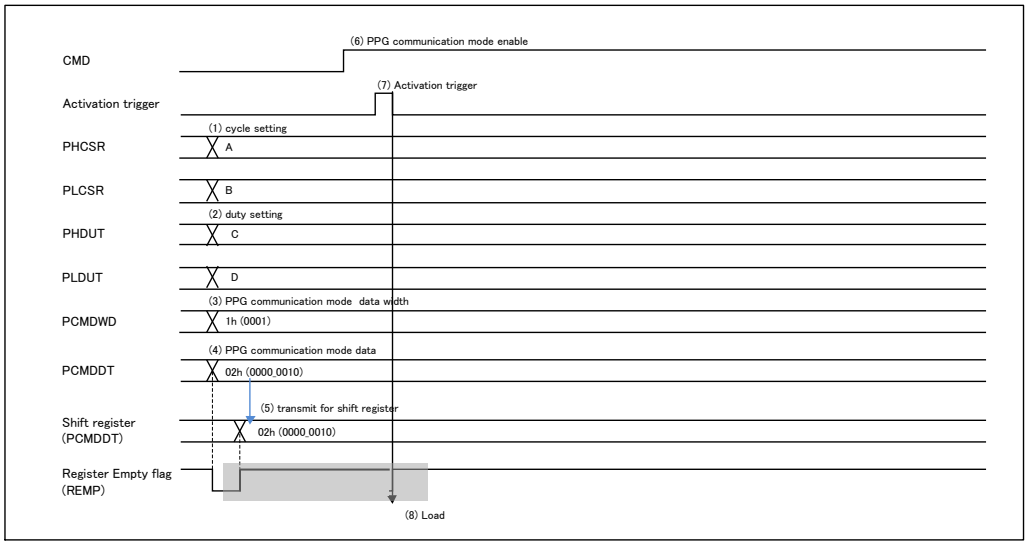
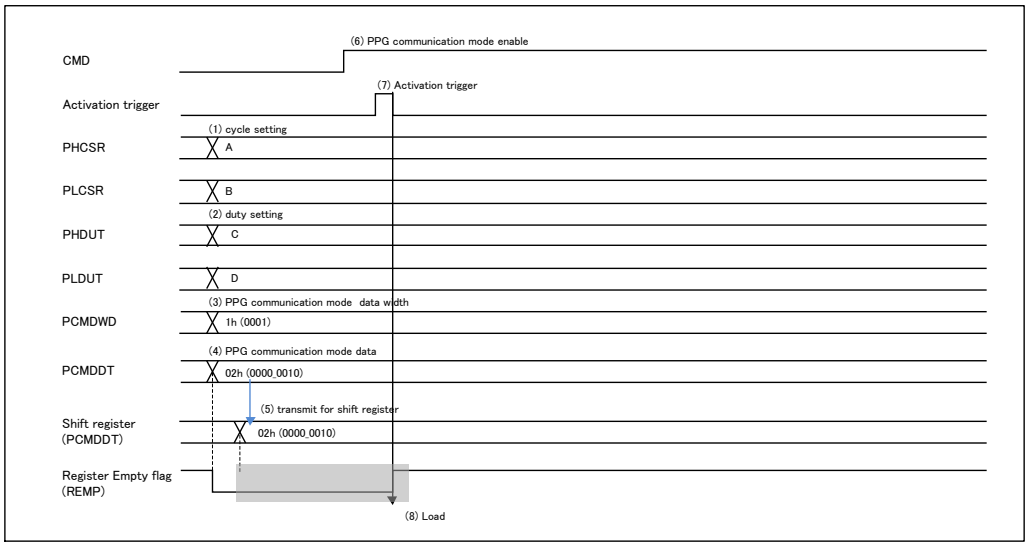
(Error)
Note: In PPG4 to PPG47, the communication function is not built into. The read value of this bit is always "0". This bit must always be written to "0".

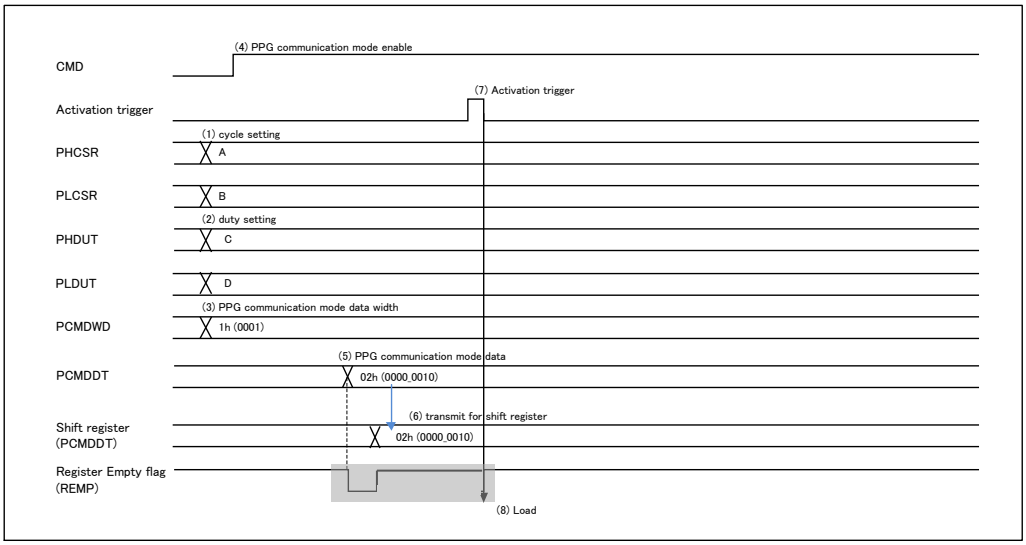
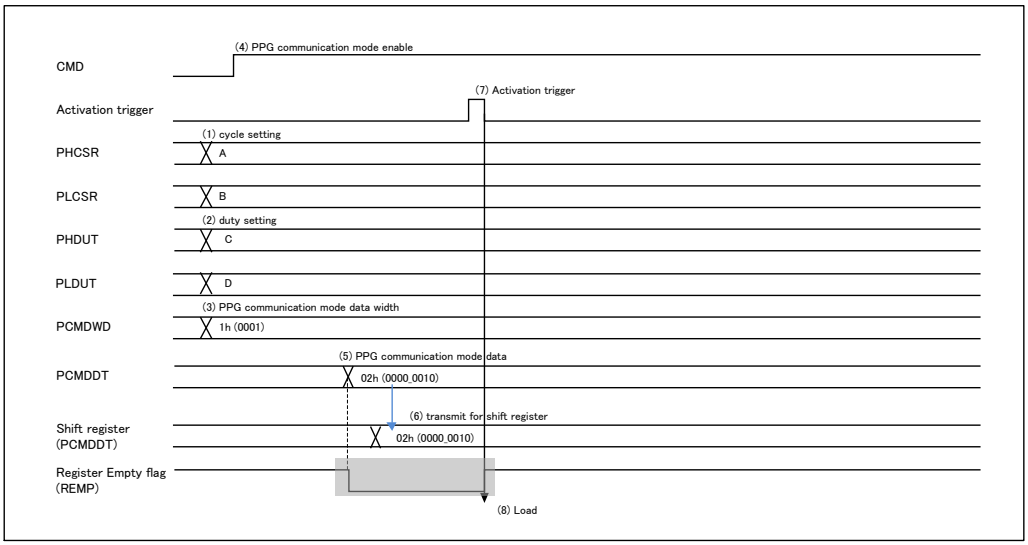
(Correct)
Deleted</td></tr>
</table>

Page	Section	Change Results
589	17.5.6	The following descriptions should be added at the end of Figure 5-7 PPG Counter Operation by GATE Function in "5.6. GATE Operation". (Error) If the one-shot pulse operation (PCN:MDSE= "1") is set when PPG operation is selected (GATEC:STGR) by GATE function, continuous pulses are output from the PPG output pin in the same way as the PWM operation. (Correct) If the one-shot pulse operation (PCN:MDSE= "1") is set when PPG operation is selected (GATEC:STGR) by GATE function, continuous pulses are output from the PPG output pin in the same way as the PWM operation. Note: · If the GATE signal has changed from "1" to "0" (when EDGE="0") during the PPG operation, the PPG down counter value (PTMG) will be maintained and the PPG output will be changed to "L" and will be stopped. · When the GATE function is enabled (STGR = "1") and the restart enable register is valid (RTRG = "1"), restart operation does not start even when another activation trigger is input.

Page	Section	Change Results
594	17.5.9	The description in "5.9. PPG Communication Mode Operation" should be added/deleted as follows: (Error) Valid registers: Software trigger (STRG), Count clock selection (CKS1, CKS0), PPG output mask selection (PGMS), Trigger input selection (EGS1, EGS0), Interrupt request enable (IREN), Interrupt factor selection (IRS1, IRS0), PPG output polarity selection (OSEL), PPG timer (PTMR), GATE function control (GATEC), PPG communication mode data reading selection (CMDSEL), PPG communication mode enable (CMD), PPG communication data register Empty flag (REMP), PPG communication data shift register Empty flag (SREMP), Interrupt factor selection2 (IRS2), High format cycle setting (PHCSR), Low format cycle setting (PLCSR), High format duty setting (PHDUT), Low format duty setting (PLDUT), Communication mode data setting (PCMDDT), Communication mode data bit length setting (PCMDWD) Invalid registers: Timer operation enable (CNTE), Mode selection (MDSE), Restart enable (RTRG), PPG output waveform selection (OWFS), Interrupt request flag (IRQF), PPG cycle setting (PCSR), PPG duty setting (PDUT), Timing Point Capture enable (TPC), Start Delay enable (STRD), Start Delay value setting (PSDR), Timing Point Capture value setting (PTPC)

Page	Section	Change Results
594	17.5.9	(Continued) (Correct) Valid registers: Software trigger (STRG), Count clock selection (CKS1, CKS0), PPG output mask selection (PGMS), Trigger input selection (EGS1, EGS0), Interrupt request enable (IREN), Interrupt factor selection (IRS1, IRS0), Interrupt request flag (IRQF) (*1), PPG output polarity selection (OSEL), PPG timer (PTMR), GATE function control (GATEC), Low format pulse polarity selection (LFPR), high format pulse polarity selection (HFPR), PPG communication mode data reading selection (CMDSEL), PPG communication mode enable (CMD), PPG communication data register Empty flag (REMP) (*2), PPG communication data shift register Empty flag (SREMP) (*2), Interrupt factor selection2 (IRS2), High format cycle setting (PHCSR), Low format cycle setting (PLCSR), High format duty setting (PHDUT), Low format duty setting (PLDUT), Communication mode data setting (PCMDDT), Communication mode data bit length setting (PCMDWD) Invalid registers: Timer operation enable (CNTE), Mode selection (MDSE), Restart enable (RTRG), PPG output waveform selection (OWFS), PPG cycle setting (PCSR), PPG duty setting (PDUT), Timer Point Capture enable (TPC), Start Delay enable (STRD), Start Delay value setting (PSDR), Timing Point Capture value setting (PTPC) (*1) IRS[2:0]=000b to 100b of the interrupt selection cannot be set during PPG communication; however, the registers are enabled. (*2) Cannot be set because this is a read only register.

Page	Section	Change Results
595	17.5.10	Figure 5-10 Example of PPG Communication Mode Operation (Activation Operation Case 1) should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
595	17.5.10	Figure 5-11 Example of PPG Communication Mode Operation (Activation Operation Case 2) should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
605	17.6	The description in 3. in ■ PPG Operation Activation in "6. Notes" should be modified as follows: (Error) 3. The PPG activation with an internal trigger is the same operation as the PPG activation using a software trigger or an external trigger (TRG pin). (Correct) 3. The PPG activation with a PSTR pin (PPG activation trigger) as a trigger is the same operation as the PPG activation using a software trigger or an external trigger (TRG pin).
606	17.6	The description in ■ GATE Function in "6. Notes" should be modified as follows: (Error) ■ GATE Function 1. Set the GATE function control register (GATEC) before PPG activation. 2. Do not change the GATE selection bit (STGR) and the polarity selection bit (EDGE) of the GATE function control register (GATEC) during the PPG operation 3. It takes 4 clocks in internal clock until the PPG output is stopped after GATE signal is negated. (Correct) ■ GATE Function 1. Set the GATE function control register (GATEC) before PPG activation. Do not change the GATE selection bit (STGR) and the polarity selection bit (EDGE) of the GATE function control register (GATEC) during the PPG operation. 2. It takes 4 clocks in internal clock until the PPG output is stopped after GATE signal is negated. 3. When the GATE function is valid (STGR = "1"), the restart enable (RTRG) is valid. The restart operation does not start even when other trigger is input. 4. If the GATE signal is changed from "1" to "0" (when EDG = "0") during the PPG operation, the PPG down counter value will be maintained and the PPG output will be changed to "L" and will be stopped. Thereafter, to start the PPG, set the GATE signal from "0" to "1" to reload the cycle and the duty values.

Page	Section	Change Results
608	17.6	Valid registers in 2. PPG communication of ■ PPG Communication Mode Function in "6. Notes" should be modified as follows: (Error) 2. In the PPG communication mode, the following registers are valid or invalid. Valid registers: Software trigger (STRG) Count clock selection (CKS1, CKS0) PPG output mask selection (PGMS) Trigger input selection (EGS1, EGS0) Interrupt request enable (IREN) Interrupt factor selection (IRS1, IRS0) Interrupt request flag (IRQF)*1 PPG output enable (OE) PPG output polarity selection (OSEL) PPG timer (PTMR) GATE function control (GATEC) PPG communication mode data reading selection (CMDSEL) PPG communication mode enable (CMD) Timing Point Capture interrupt (IRS2) PPG communication data register Empty flag (REMP)*2 PPG communication data shift register Empty flag (SREMP)*2 High format cycle setting (PHCSR) Low format cycle setting (PLCSR) High format duty setting (PHDUT) Low format duty setting (PLDUT) Communication mode data setting (PCMDDT) Communication mode data bit length setting (PCMDWD)

Page	Section	Change Results
608	17.6	(Continued) (Correct) 2. In the PPG communication mode, the following registers are valid or invalid. Valid registers: Software trigger (STRG) Count clock selection (CKS1, CKS0) PPG output mask selection (PGMS) Trigger input selection (EGS1, EGS0) Interrupt request enable (IREN) Interrupt factor selection (IRS1, IRS0) Interrupt request flag (IRQF) (*1) PPG output polarity selection (OSEL) PPG timer (PTMR) GATE function control (GATEC) PPG communication mode data reading selection (CMDSEL) PPG communication mode enable (CMD) Timing Point Capture interrupt (IRS2) PPG communication data register Empty flag (REMP) (*2) PPG communication data shift register Empty flag (SREMP) (*2) High format cycle setting (PHCSR) Low format cycle setting (PLCSR) High format duty setting (PHDUT) Low format duty setting (PLDUT) Communication mode data setting (PCMDDT) Communication mode data bit length setting (PCMDWD) Low format pulse polarity selection (LFPR) High format pulse polarity selection (HFPR)
618	18.4.1	The register name in "4.1. Watchdog Control Register 0 : WDTCR0 (WatchDog Timer Configuration Register 0)" should be modified as follows: (Error) 4.1. Watchdog Control Register 0 : WDTCR0 (WatchDog Timer Configuration Register 0) (Correct) 4.1. Watchdog Timer 0 Control Register: WDTCR0 (WatchDog Timer 0 Configuration Register)
638	19.1	The following description should be added to "1. Overview". (Correct) The base timer function is not provided in MB91F52xB(64pin). Only ch.1 is available in MB91F52xD (80pin) and MB91F52xF (100pin).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>645</td><td>19.4</td><td>Table 4-1 Table of Base Addresses (Base_addr) and External Pins in "4. Registers" should be changed as follows:

(Error)
<table>
<tr><th>Channel number</th><th>Base address</th><th>External pin</th></tr>
<tr><td>0</td><td>0x0080</td><td rowspan="2">TIOA0, TIOA1, TIOB0, and TIOB1 are assigned based on the BTSEL01 register setting.</td></tr>
<tr><td>1</td><td>0x0090</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Channel number</th><th rowspan="2">Base address</th><th colspan="4">External pin *</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD
MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK
MB91F52xL</th></tr>
<tr><td>0</td><td>0x0080</td><td rowspan="2">No</td><td>No</td><td>TIOA0_0/TIOA0_1,
TIOB0_1</td><td>TIOA0_0/TIOA0_1,
TIOB0_0/TIOB0_1</td></tr>
<tr><td>1</td><td>0x0090</td><td>TIOA1_1,
TIOB1_1</td><td>TIOA1_1,
TIOB1_1</td><td>TIOA1_0/TIOA1_1,
TIOB1_0/TIOB1_1</td></tr>
</table>
* : TIOA0, TIOA1, TIOB0, and TIOB1 are assigned according to the BTSEL01 register setting, but the setting without external pins is disabled.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>649</td><td>19.4.1.2</td><td>■ BTxTMCR : Address Base_addr + 02_H (Access: Half-word) in "4.1.2. Timer Control Registers 0, 1 : BTxTMCR (Base Timer 0/1 TiMer Control Register)" should be modified as follows:

(Error)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>bit13</td><td>bit12</td><td>bit11</td><td>bit10</td><td>bit9</td><td>bit8</td></tr>
<tr><td></td><td>Reserved</td><td colspan="3">CKS[2:0]</td><td>[PWM - PPG]
RTGEN
[Others]
Reserved</td><td>[PWM - PPG]
PMSK
[PWC]
EGS[2]
[Others]
Reserved</td><td colspan="2">EGS[1:0]</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W
R0,WX *3</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W
R0,WX *1</td><td>R/W
R0,WX *1</td><td>R/W</td><td>R/W</td></tr>
<tr><td></td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td>[Reload timer
-
PWC]
T32
[Others]
Reserved</td><td colspan="3">FMD[2:0]</td><td>[Reload timer
- PWM
- PPG]
OSEL
[Others]
Reserved</td><td>MDSE</td><td>CTEN</td><td>STRG</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W
R0,WX *1
R0,WX *2</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W
R/W0 *1</td><td>R/W</td><td>R,W</td><td>R0,W
R0,W</td></tr>
</table>
(Correct)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>bit13</td><td>bit12</td><td>bit11</td><td>bit10</td><td>bit9</td><td>bit8</td></tr>
<tr><td></td><td>Reserved</td><td colspan="3">CKS[2:0]</td><td>[PWM - PPG]
RTGEN
[Others]
Reserved</td><td>[PWM - PPG]
PMSK
[PWC]
EGS[2]
[Others]
Reserved</td><td colspan="2">EGS[1:0]</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W0
R0,W0 *3</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W
R0,WX *1</td><td>R/W
R0,WX *1</td><td>R/W</td><td>R/W</td></tr>
<tr><td></td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td>[Reload timer
-
PWC]
T32
[Others]
Reserved</td><td colspan="3">FMD[2:0]</td><td>[Reload timer
- PWM
- PPG]
OSEL
[Others]
Reserved</td><td>MDSE</td><td>CTEN</td><td>STRG</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W
R0,W0 *1
R0,W0 *2</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W
R/W0 *1</td><td>R/W</td><td>R,W</td><td>R0,W
R0,W0 *1</td></tr>
</table>
</td></tr>
</table>

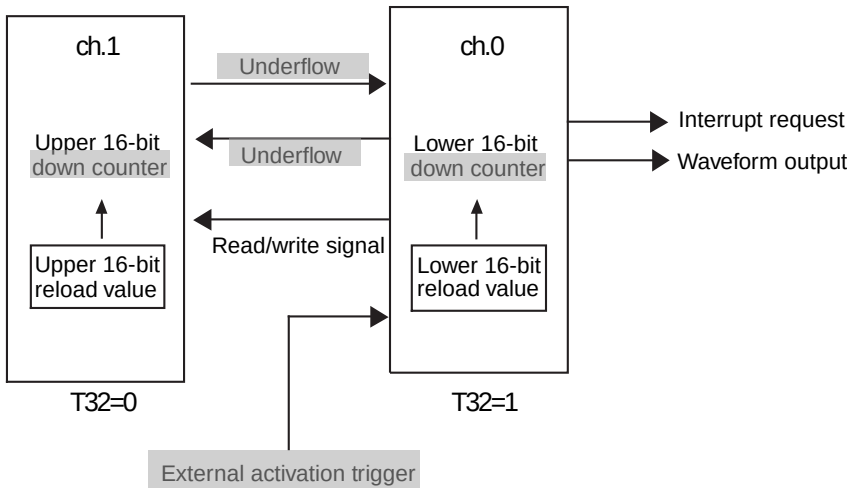
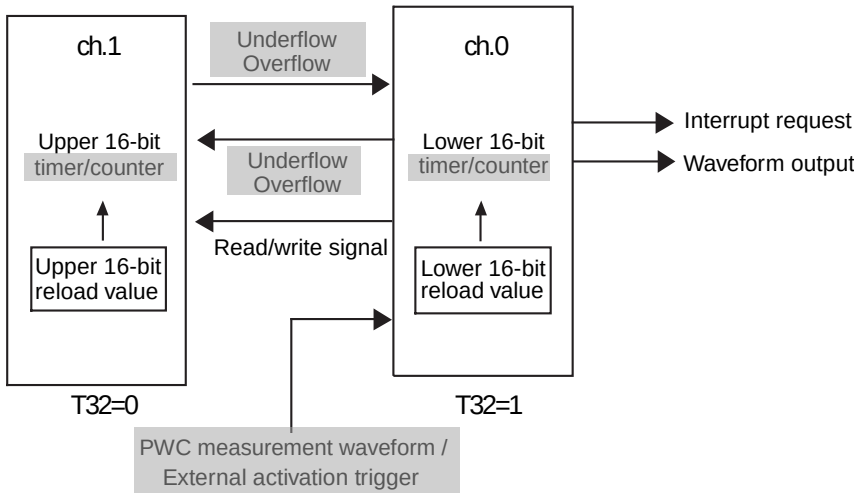
Page	Section	Change Results
650	19.4.1.2	The description in [bit15] Reserved in "4.1.2. Timer Control Registers 0, 1 : BTxTMCR (Base Timer 0/1 TiMer Control Register)" should be modified as follows: (Error) Writing to this bit does not affect the operation. (Correct) Write 0 to this bit.
654	19.4.1.2	A description should be added in [bit0] STRG in "4.1.2. Timer Control Registers 0, 1 : BTxTMCR (Base Timer 0/1 TiMer Control Register)". (Error) Functions as a trigger for timer activation, etc. (Correct) Functions as a trigger for timer activation, etc. The read value at PWC is "0". Write "0" in this bit at PWC.
655	19.4.1.3	The notes in "4.1.3. I/O Selection Register : BTSEL01 (Base Timer SELect register ch.0 and ch.1)" should be modified as follows: (Error) · These registers must be accessed in 8-bit mode. · These registers will not be initialized even if reset mode is set (writing of BTxTMCR:FMD = 000). (Correct) · These registers must be accessed in 8-bit mode. · Rewrite this register after setting the FMD2 to FMD0 bits of the base timer x the timer control register (BTxTMCR) to the base timer reset mode (FMD2 to FMD0 = 000).

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>659</td><td>19.4.2.2</td><td>The initial value in "4.2.2. Cycle Setting Registers 0, 1 : BTxPCSR (Base Timer 0/1 Pulse Counter Start Register)" should be modified as follows:

(Error)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>- - -</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td colspan="6">D[15:0]</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>- - -</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R/W</td><td>- - -</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>
(Correct)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>- - -</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td colspan="6">D[15:0]</td></tr>
<tr><td>Initial value</td><td>X</td><td>X</td><td>- - -</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R/W</td><td>- - -</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table></td></tr>
<tr><td>667</td><td>19.4.4.2</td><td>The initial value in "4.4.2. L Width Setting Registers 0, 1: BTxPRLL (Base Timer 0/1 Pulse Length of "L" register)" should be modified as follows:

(Error)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>- - -</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td colspan="6">D[15:0]</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>- - -</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R/W</td><td>- - -</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>
(Correct)
<table>
<tr><td></td><td>bit15</td><td>bit14</td><td>- - -</td><td>bit2</td><td>bit1</td><td>bit0</td></tr>
<tr><td></td><td colspan="6">D[15:0]</td></tr>
<tr><td>Initial value</td><td>X</td><td>X</td><td>- - -</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R/W</td><td>- - -</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
668	19.4.4.3	The initial value in "4.4.3. H Width Setting Registers 0, 1: BTxPRLH (Base Timer 0/1 Pulse Length of "H" register)" should be modified as follows: (Error) 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 (Correct) 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

Page	Section	Change Results
691	19.5.4.4	Figure 5-11 Configurations in 32-bit Timer Mode in "5.4.4. 32-bit Timer Mode Operation" should be modified as follows: (Error)  (Correct) 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>732</td><td>20.1</td><td>The following description should be added in "1. Overview".

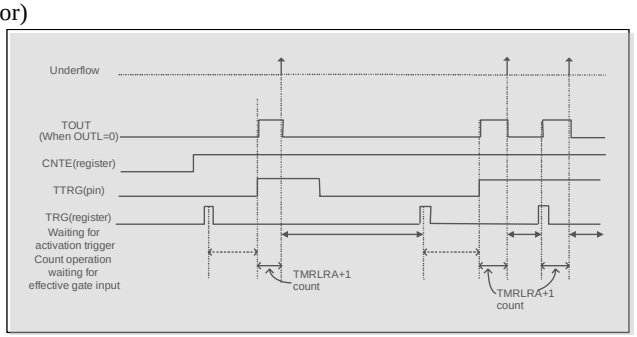
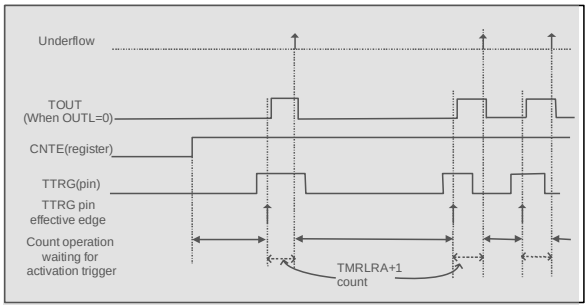
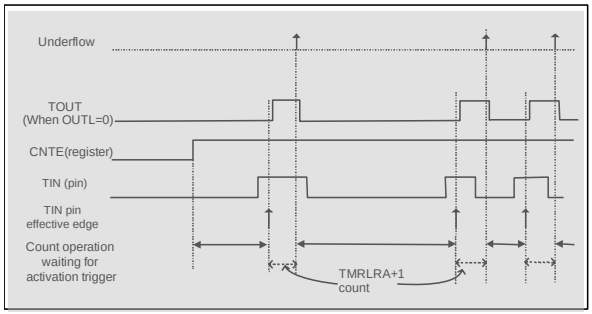
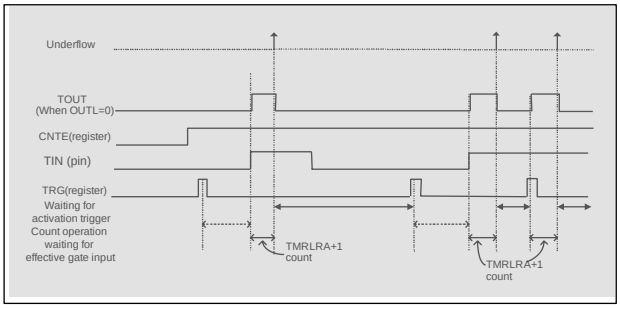
(Correct)
The numbers of available channels are shown below.
MB91F52xB (64pin) : 7
MB91F52xD (80pin) : 7
MB91F52xF (100pin) : 8
MB91F52xJ (120pin) : 8
MB91F52xK (144pin) : 8
MB91F52xL (176pin) : 8</td></tr>
<tr><td>735</td><td>20.4</td><td>Table 4-1 Table of Base Address (Base_addr), External Pins in "4. Registers" should be modified as follows:

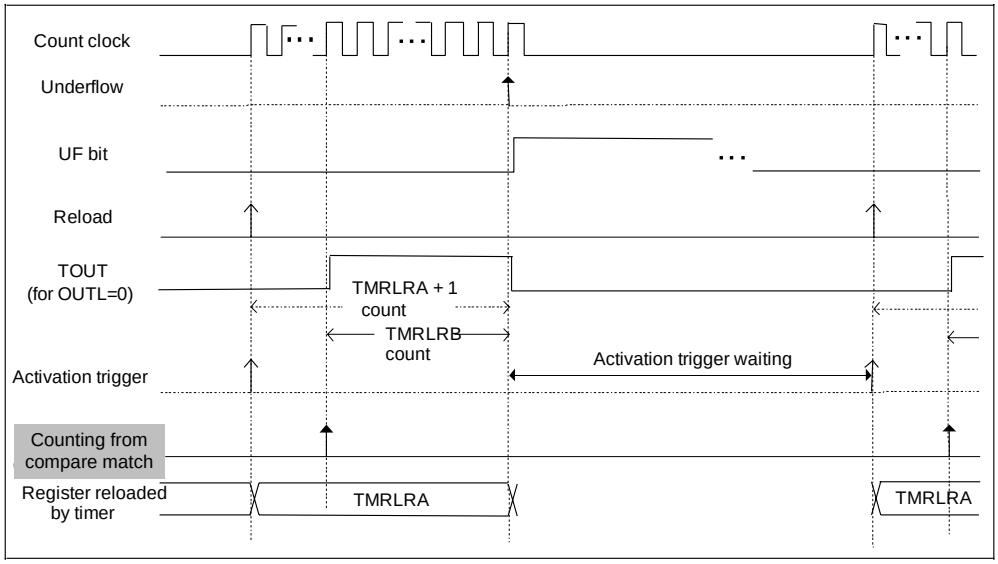
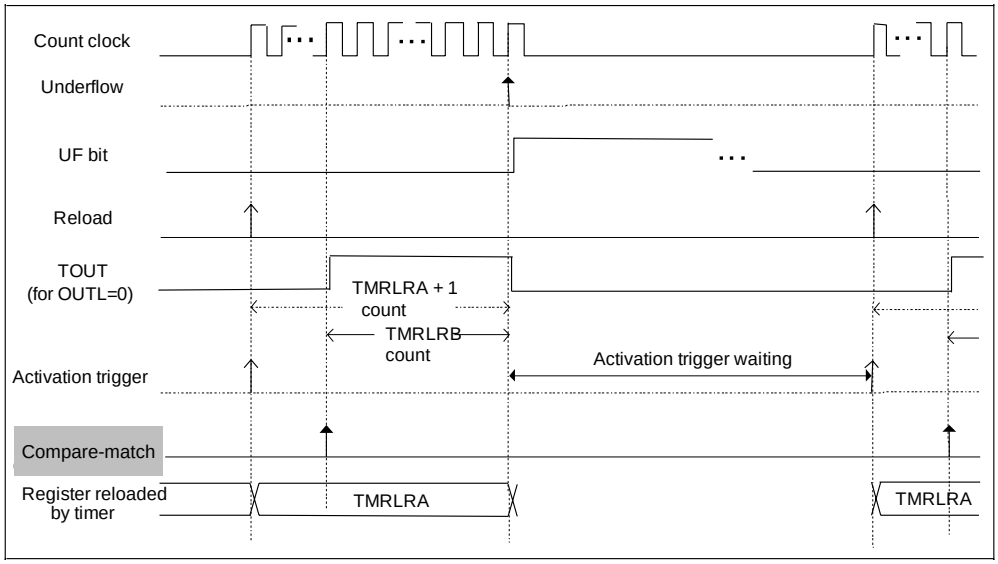
(Error)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th colspan="2">External pin</th></tr>
<tr><th>TOUT</th><th>TIN</th></tr>
<tr><td>0</td><td>0x0060</td><td>TOT0_0/TOT0_1</td><td>TIN0_0/TIN0_1/TIN0_2</td></tr>
<tr><td>1</td><td>0x0100</td><td>TOT1_0/TOT1_1/TOT1_2</td><td>TIN1_0/TIN1_1</td></tr>
<tr><td>2</td><td>0x0108</td><td>TOT2_0/TOT2_1</td><td>TIN2_0/TIN2_1</td></tr>
<tr><td>3</td><td>0x0110</td><td>TOT3_0/TOT3_1</td><td>TIN3_0/TIN3_1/TIN3_2</td></tr>
<tr><td>4</td><td>0x01D8</td><td>TOT4_0/TOT4_1</td><td>TIN4_0/TIN4_1</td></tr>
<tr><td>5</td><td>0x01F0</td><td>TOT5_0/TOT5_1</td><td>TIN5_0/TIN5_1</td></tr>
<tr><td>6</td><td>0x01F8</td><td>TOT6_0/TOT6_1</td><td>TIN6_0/TIN6_1</td></tr>
<tr><td>7</td><td>0x0068</td><td>TOT7_0/TOT7_1</td><td>TIN7_0/TIN7_1</td></tr>
</table></td></tr>
</table>

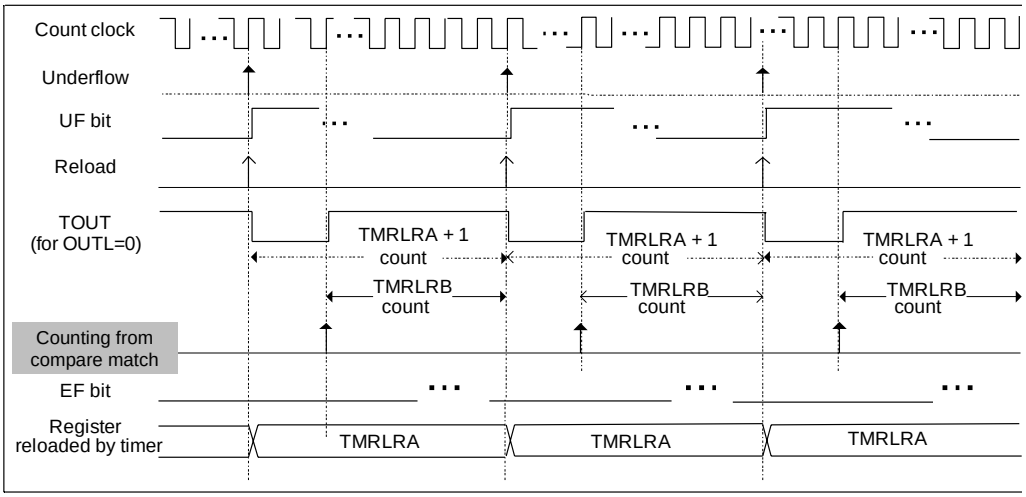
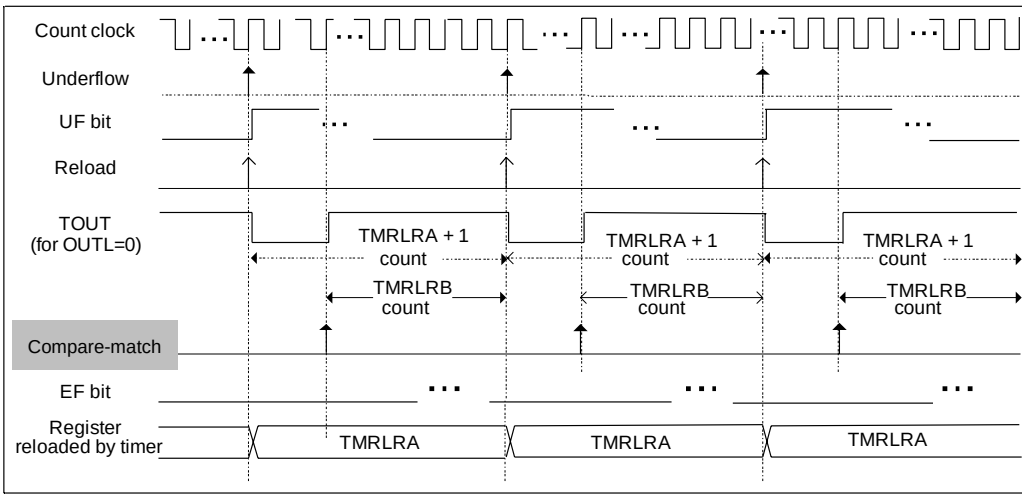
Page	Section	Change Results
735	20.4	(Continued) (Correct)

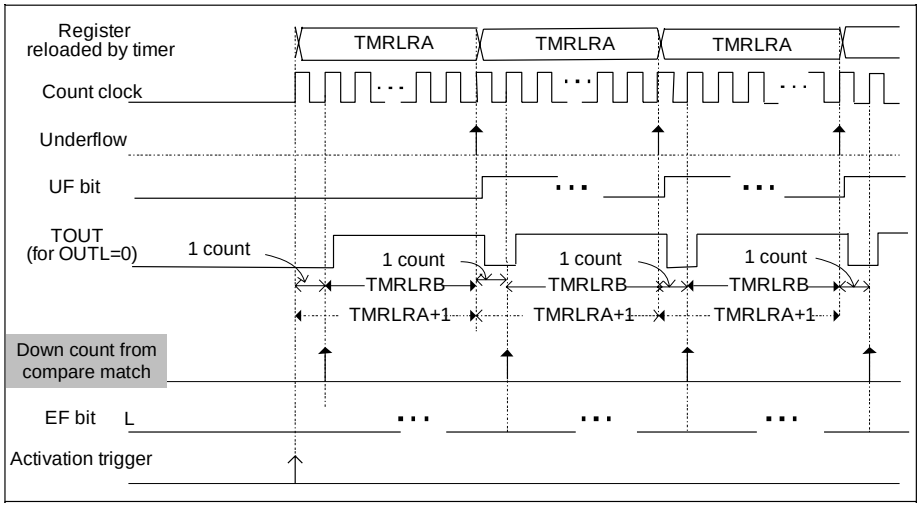
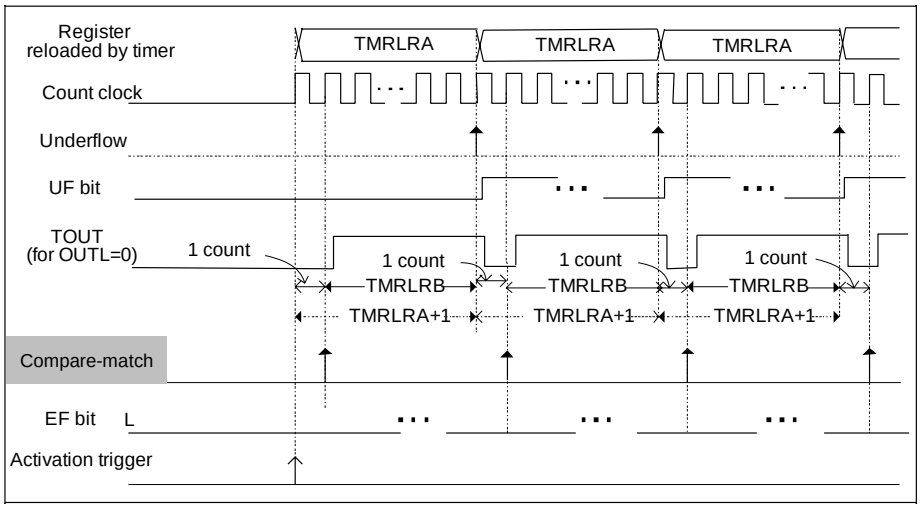
Channel	Base_addr	External pin (TOUT output, TIN input)					
		MB91F52xB	MB91F52xD	MB91F52xF	MB91F52xJ	MB91F52xK	MB91F52xL
0	0x0060	TOT0_0	TOT0_0	TOT0_0	TOT0_0	TOT0_0/ TOT0_1	TOT0_0/ TOT0_1
		TIN0_2	TIN0_2	TIN0_2	TIN0_0/ TIN0_2	TIN0_0/ TIN0_2	TIN0_0/ TIN0_1/ TIN0_2
1	0x0100	TOT1_2	TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_2	TOT1_0/ TOT1_1/ TOT1_2
		TIN1_0	TIN1_0	TIN1_0	TIN1_0	TIN1_0	TIN1_0/ TIN1_1
2	0x0108	none	none	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1	TOT2_0/ TOT2_1
				TIN2_0	TIN2_0	TIN2_0/ TIN2_1	TIN2_0/ TIN2_1
3	0x0110	TOT3_0	TOT3_0	TOT3_0	TOT3_0	TOT3_0/ TOT3_1	TOT3_0/ TOT3_1
		TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_2	TIN3_0/ TIN3_1/ TIN3_2	TIN3_0/ TIN3_1/ TIN3_2
4	0x01D8	TOT4_0	TOT4_0	TOT4_0	TOT4_0	TOT4_0/ TOT4_1	TOT4_0/ TOT4_1
		TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1	TIN4_0/ TIN4_1
5	0x01F0	TOT5_0	TOT5_0	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1	TOT5_0/ TOT5_1
		TIN5_0/	TIN5_0/	TIN5_0/	TIN5_0/ TIN5_1	TIN5_0/ TIN5_1	TIN5_0/ TIN5_1
6	0x01F8	TOT6_1	TOT6_1	TOT6_0/ TOT6/1	TOT6_0/ TOT6/1	TOT6_0/ TOT6/1	TOT6_0/ TOT6/1
		TIN6_1	TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1	TIN6_0/ TIN6_1
7	0x0068	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1	TOT7_0/ TOT7_1
		TIN7_0	TIN7_0	TIN7_0	TIN7_0	TIN7_0/ TIN7_1	TIN7_0/ TIN7_1

Page	Section	Change Results
737	20.4.1	The description in [bit13, bit12] in "4.1. Control Status Register: TMCSR (TiMer Control and Status Register)" should be modified as follows: (Error) [Interval timer mode, trigger input (bit8:GATE bit = "0")] (Correct) [Interval timer mode, trigger input (bit8: GATE = "0")]
742	20.4.3	The description in "4.3. 16-bit Timer Reload Register A, 16-bit Timer Reload Register B: TMRLRA, TMRLRB (16bit TiMer ReLoad Register A/B)" should be modified as follows: (Error) This register sets the count initial value and other items. (Correct) TMRLRA sets the count initial value. TMRLRB applies different functions according to the operation mode.

Page	Section	Change Results
763	20.5.3.1	The timing charts of "Single one-shot timer (GATE="0": When the trigger input and rising edge trigger are selected)" and "Single one-shot timer (GATE="1": gate input, TRGM: H input interval count)" in Figure 5-7 in "5.3.1. Single One-shot Operation" should be modified as follows: (Error)  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) (Correct)  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)

Page	Section	Change Results
771	20.5.3.5	The timing chart for "· Sets TMRLRB < TMRLRA" in Figure 5-13 Compare One-shot Operation (1 / 2) in "5.3.5. Compare One-shot Operation" should be modified as follows: (Error)  Compare one-shot function (TMRLRB < TMRLRA) (Correct)  Compare one-shot function (TMRLRB < TMRLRA)

Page	Section	Change Results
774	20.5.3.6	The timing chart for "· Sets TMRLRB < TMRLRA" in Figure 5-16 Compare Reload Operation (1 / 2) in "5.3.6. Compare Reload Operation" should be modified as follows: (Error)  Compare reload function (TMRLRB < TMRLRA) trigger input (Correct)  Compare reload function (TMRLRB < TMRLRA) trigger input

Page	Section	Change Results
775	20.5.3.6	The timing chart for "· Sets TMRLRB = TMRLRA" in Figure 5-17 Compare Reload Operation (2 / 2) in "5.3.6. Compare Reload Operation" should be modified as follows: (Error)  Compare reload function (TMRLRB = TMRLRA) trigger input (Correct)  Compare reload function (TMRLRB = TMRLRA) trigger input

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>776</td><td>20.5.3.7</td><td>The following content in the description in "5.3.7. Capture Mode" should be modified.

(Error)
When you use TIN input as the gate input (when bit8:GATE= "1" of the TMCSR register), generate a retrigger by bit0:TRG of the TMCSR register.
In a mode other than trigger, a capture will not be performed at a retrigger. The EF interrupt will also not be generated.
The timer operation and the TOUT output will be the same for the single one-shot feature and the single reload feature.

(Correct)
When you use TIN input as the gate input (when bit8:GATE= "1" of the TMCSR register), generate a retrigger by bit0:TRG of the TMCSR register.
In a mode other than capture, capture will not be performed upon retrigger. The EF interrupt will also not be generated.
The timer operation and the TOUT output will be the same for the single one-shot feature and the single reload feature.</td></tr>
<tr><td>795</td><td>20.6.5</td><td>"6.5. PWC" should be modified as follows:

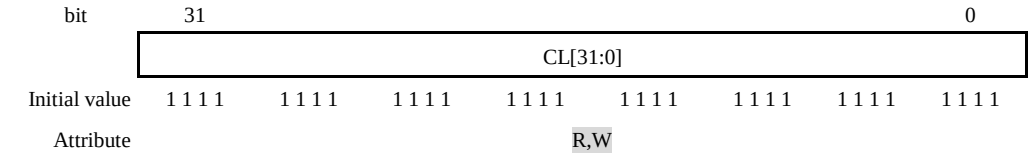
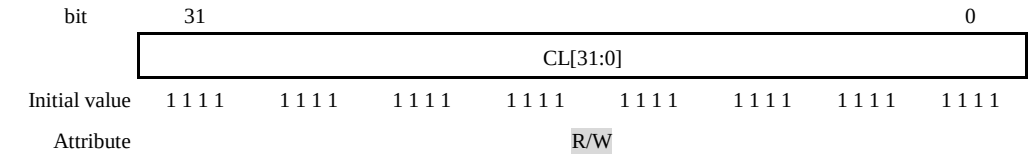
(Error)
<table>
<tr><th colspan="11">TMCSR</th><th>TMRLRA</th><th>TMRLRB</th></tr>
<tr><th>MOD [1:0]</th><th>TRGM [1:0]</th><th>CSL [2:0]</th><th>GATE</th><th>EF</th><th>OUTL</th><th>RELD</th><th>INTE</th><th>UF</th><th>CNTE</th><th>TRG</th><td rowspan="2">(A)</td><td rowspan="2">(B)</td></tr>
<tr><td>11</td><td>*1</td><td>*2</td><td>0</td><td>−</td><td>*3</td><td>*4</td><td>*5</td><td>−</td><td>1</td><td>S</td></tr>
</table>
(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

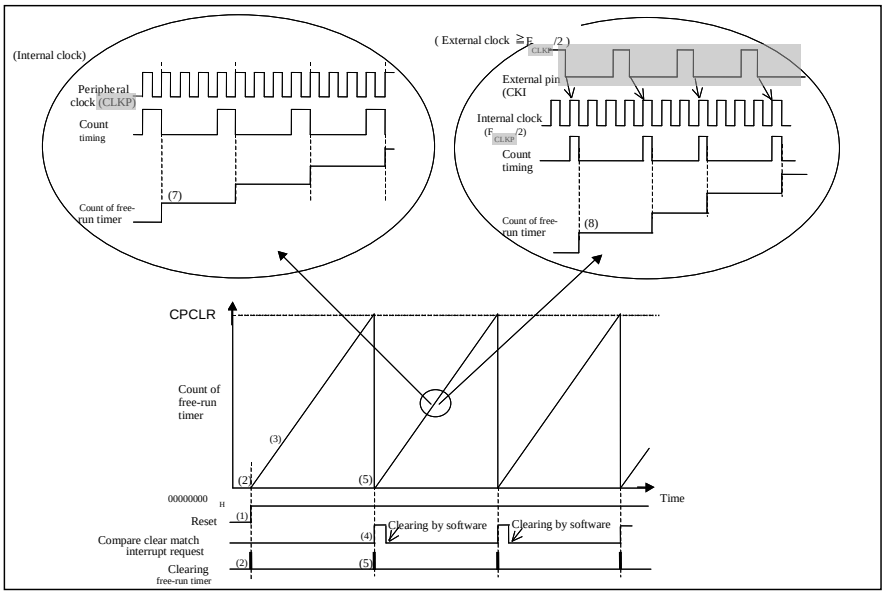
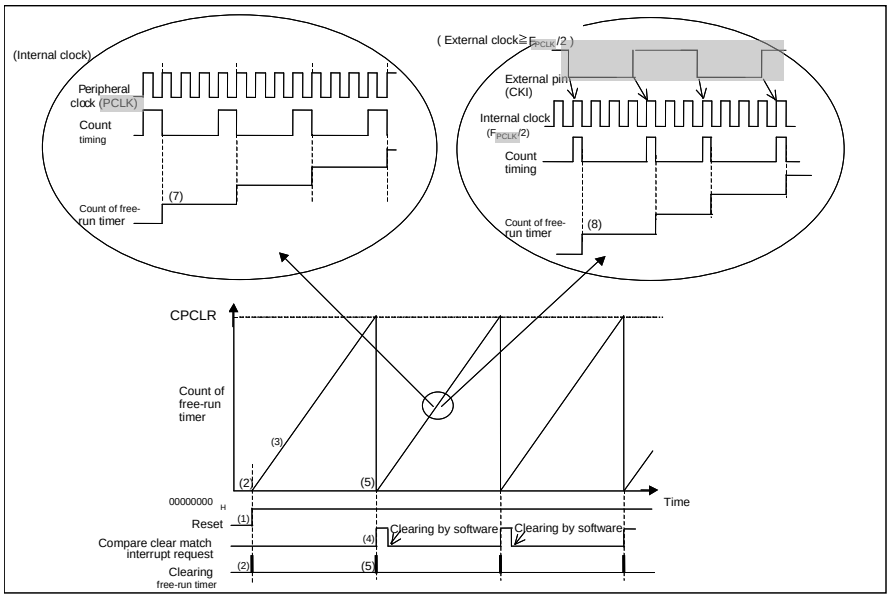
(Correct)
<table>
<tr><th colspan="11">TMCSR</th><th>TMRLRA</th><th>TMRLRB</th></tr>
<tr><th>MOD [1:0]</th><th>TRGM [1:0]</th><th>CSL [2:0]</th><th>GATE</th><th>EF</th><th>OUTL</th><th>RELD</th><th>INTE</th><th>UF</th><th>CNTE</th><th>TRG</th><td rowspan="2">(A)</td><td rowspan="2">(B)</td></tr>
<tr><td>11</td><td>*1</td><td>*2</td><td>0</td><td>−</td><td>*3</td><td>*4</td><td>*5</td><td>−</td><td>1</td><td>S</td></tr>
</table>
(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)
(B): Count value when a trigger occurs during the count operation
S :Use at timer activation
-:Does not influence operation</td></tr>
</table>

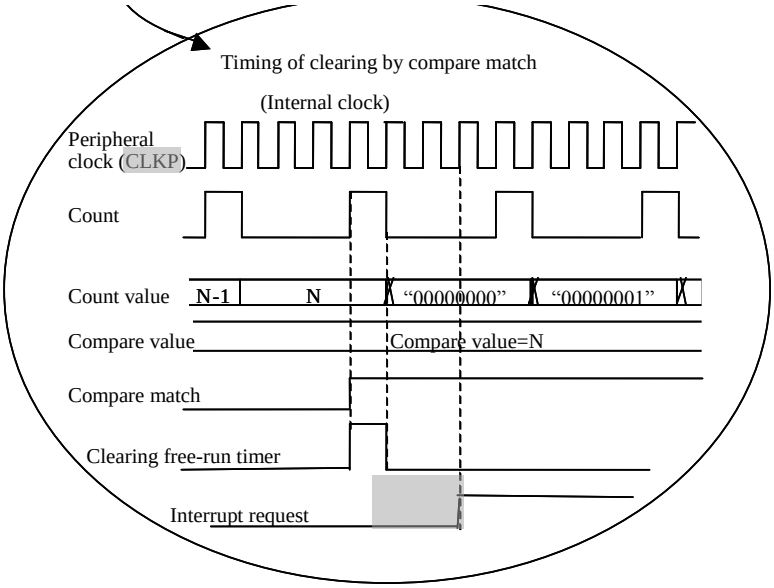
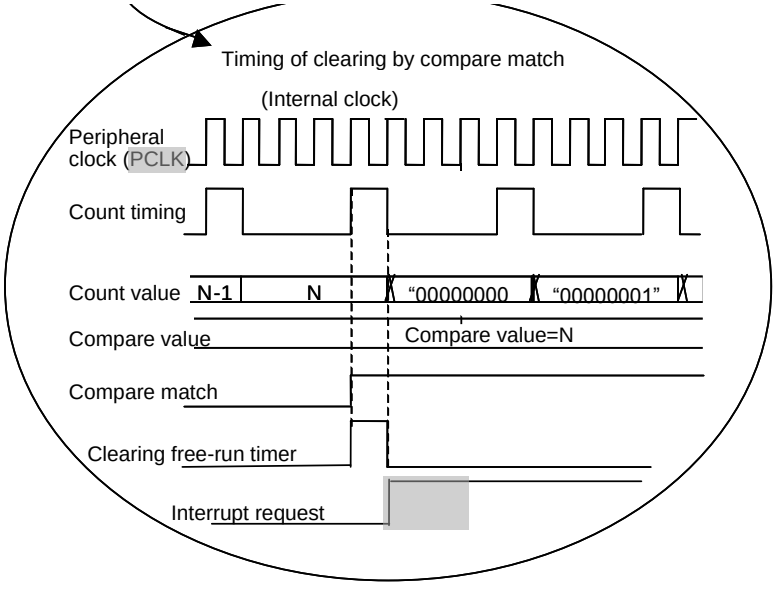
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>798</td><td>21.1</td><td>The following description should be added in "1. Overview".

(Correct)
The numbers of channels available from external clocks are shown below.
MB91F52xB (64pin) : 1
MB91F52xD (80pin) : 2
MB91F52xF (100pin) : 3
MB91F52xJ (120pin) : 3
MB91F52xK (144pin) : 3
MB91F52xL (176pin) : 3</td></tr>
<tr><td>805</td><td>21.4</td><td>Table 4-1 Base Address (Base_Addr) and External Pin Table in "4. Registers" should be modified as follows:

(Error)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th>External pin</th></tr>
<tr><th>FRCK</th></tr>
<tr><td>3</td><td>0x0240</td><td>FRCK3_0/FRCK3_1</td></tr>
<tr><td>4</td><td>0x024C</td><td>FRCK4_0/FRCK4_1</td></tr>
<tr><td>5</td><td>0x0FA0</td><td>FRCK5_0/FRCK5_1</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th colspan="4">External pin (FRCK)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD</th><th>MB91F52xF
MB91F52xJ
MB91F52xK</th><th>MB91F52xL</th></tr>
<tr><td>3</td><td>0x0240</td><td>none</td><td>none</td><td>FRCK3_0</td><td>FRCK3_0/FRCK3_1</td></tr>
<tr><td>4</td><td>0x024C</td><td>FRCK4_0</td><td>FRCK4_0</td><td>FRCK4_0</td><td>FRCK4_0/FRCK4_1</td></tr>
<tr><td>5</td><td>0x0FA0</td><td>none</td><td>FRCK5_0</td><td>FRCK5_0</td><td>FRCK5_0/FRCK5_1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
811	21.4.1.3	The attributes in "4.1.3. Compare Clear Register: CPCLR" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
822	21.5.1.1	The figures in "5.1.1. Count Operation" should be modified as follows: (Error)  (Correct) 

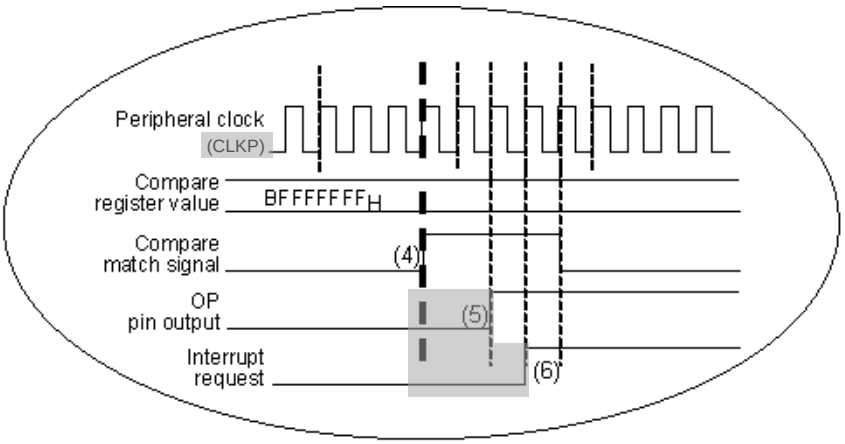
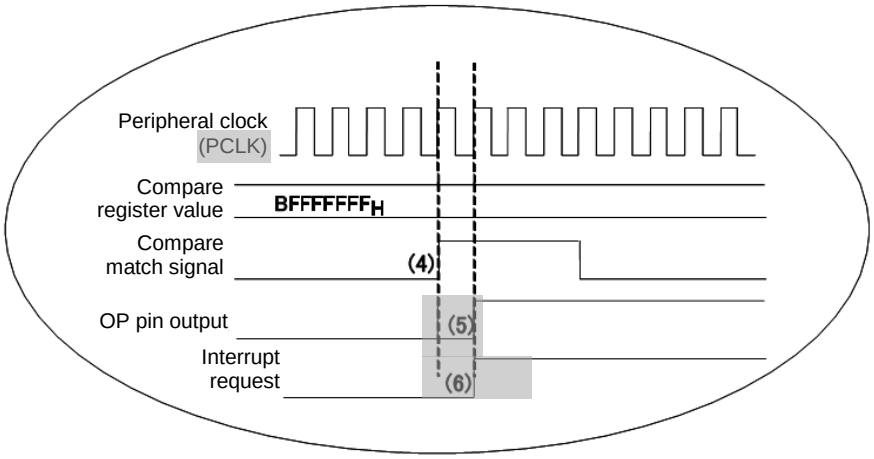
Page	Section	Change Results
825	21.5.1.4	The figures in "5.1.4. Each Clear Operations of the Free-run Timer" should be modified as follows: (Error)  (Correct) 

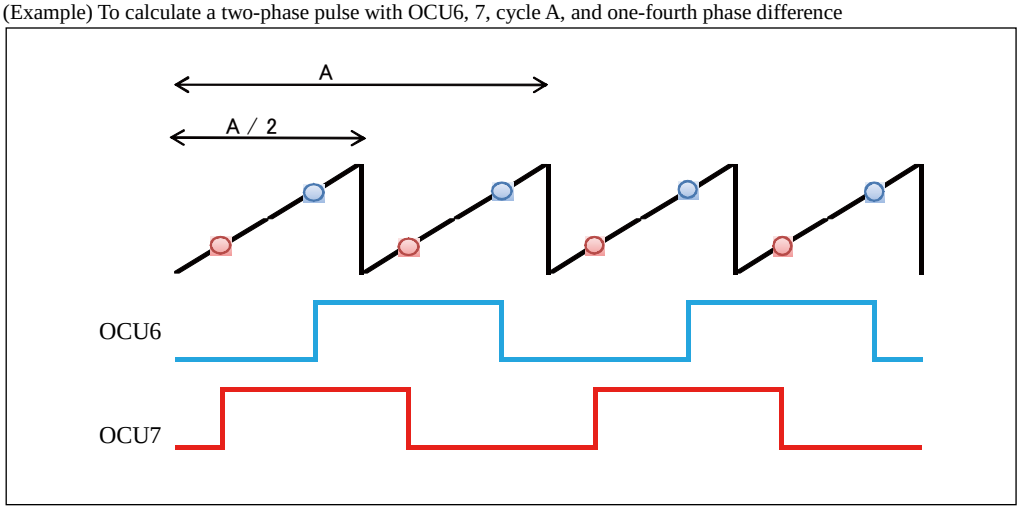
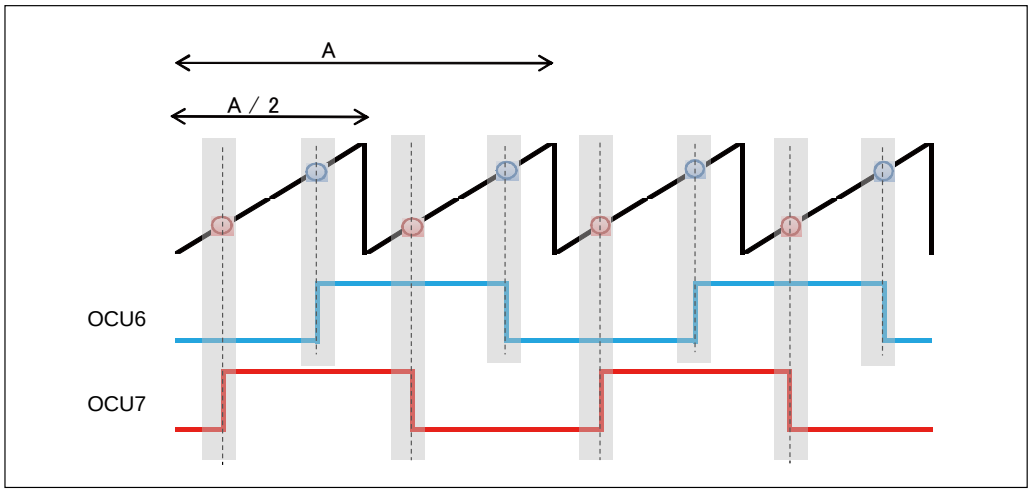
Page	Section	Change Results
838	21.9	The following content of the description in ● Clear Timing of the Free-run Timer in "9. Notes" should be modified as follows: (Error) · A software clear (TCCSL.SCLR=1) clears the counter as soon as the clear request is generated. However, in the case of compare match, the counter is cleared in the same timing as the counting up. (Correct) · A software clear (TCCSL.SCLR=1) clears the counter in the following cycle when a clear request is generated. However, in the case of compare match, the counter is cleared with the same timing as counting up.
840	22.1	The following description should be added in "1. Overview". (Correct) The numbers of available external output pins are shown below. MB91F52xB (64pin) : 4 MB91F52xD (80pin) : 4 MB91F52xF (100pin) : 6 MB91F52xJ (120pin) : 6 MB91F52xK (144pin) : 6 MB91F52xL (176pin) : 6

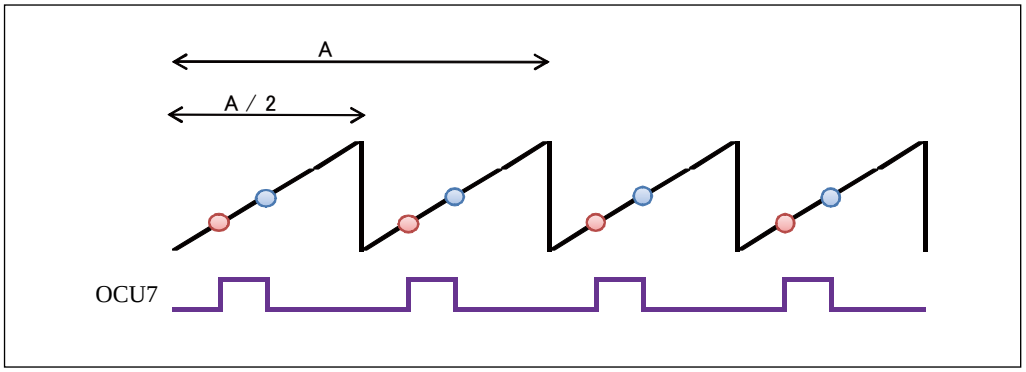
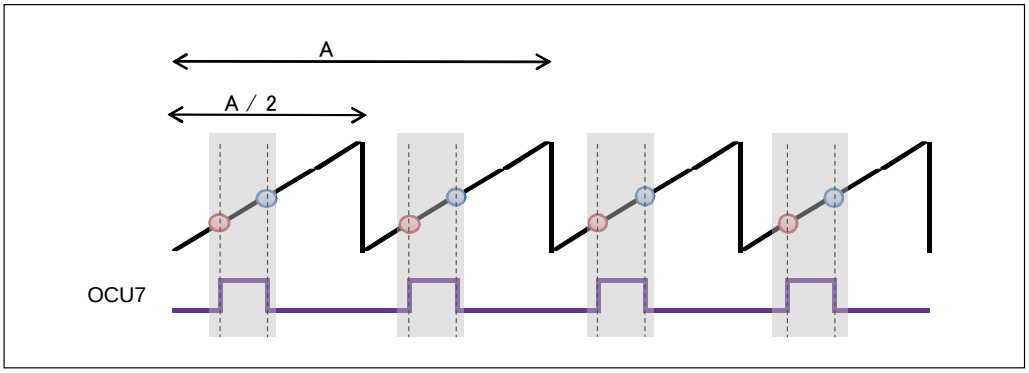
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>844</td><td>22.4</td><td>
Table 4-1 List of Base_addr and External Pins in "4. Registers" should be modified as follows:

(Error)
Table 4-1 List of Base_addr and External Pins
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th>External pin</th></tr>
<tr><th>OCU</th></tr>
<tr><td>6</td><td>0x0120</td><td>OCU6_0/OCU6_1</td></tr>
<tr><td>7</td><td>0x0120</td><td>OCU7_0/OCU7_1</td></tr>
<tr><td>8</td><td>0x012C</td><td>OCU8_0/OCU8_1</td></tr>
<tr><td>9</td><td>0x012C</td><td>OCU9_0/OCU9_1</td></tr>
<tr><td>10</td><td>0x0F90</td><td>OCU10_0/OCU10_1</td></tr>
<tr><td>11</td><td>0x0F90</td><td>OCU11_0/OCU11_1</td></tr>
</table>

(Correct)
Table 4-1 Table of Base_addr and External Pins
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th colspan="4">External pin (OCU output)</th></tr>
<tr><th>MB91F52xB
MB91F52xD</th><th>MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK
MB91F52xL</th></tr>
<tr><td>6</td><td>0x0120</td><td>none</td><td>OCU6_1</td><td>OCU6_0/
OCU6_1</td><td>OCU6_0/
OCU6_1</td></tr>
<tr><td>7</td><td>0x0120</td><td>OCU7_1</td><td>OCU7_1</td><td>OCU7_1</td><td>OCU7_0/
OCU7_1</td></tr>
<tr><td>8</td><td>0x012C</td><td>OCU8_0/
OCU8_1</td><td>OCU8_0/
OCU8_1</td><td>OCU8_0/
OCU8_1</td><td>OCU8_0/
OCU8_1</td></tr>
<tr><td>9</td><td>0x012C</td><td>OCU9_1</td><td>OCU9_1</td><td>OCU9_1</td><td>OCU9_0/
OCU9_1</td></tr>
<tr><td>10</td><td>0x0F90</td><td>none</td><td>OCU10_1</td><td>OCU10_1</td><td>OCU10_0/
OCU10_1</td></tr>
<tr><td>11</td><td>0x0F90</td><td>OCU11_1</td><td>OCU11_1</td><td>OCU11_0/
OCU11_1</td><td>OCU11_0/
OCU11_1</td></tr>
</table>
</td></tr>
</table>

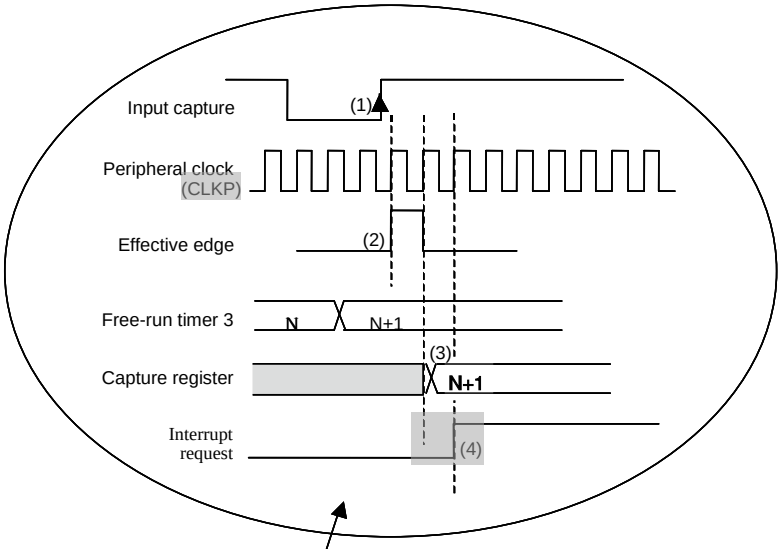
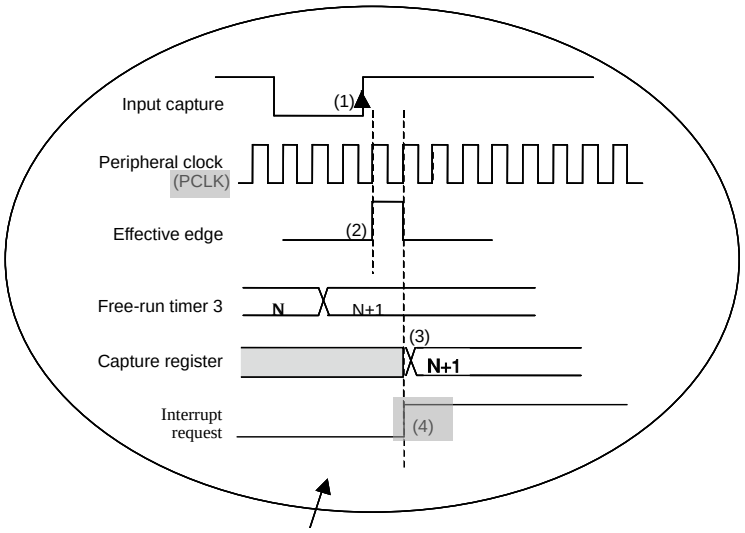
Page	Section	Change Results
852	22.5.1	The figure in "5.1. Output Compare Output (Independent Invert) CMOD = "0" should be modified as follows: (Error)  (Correct) 

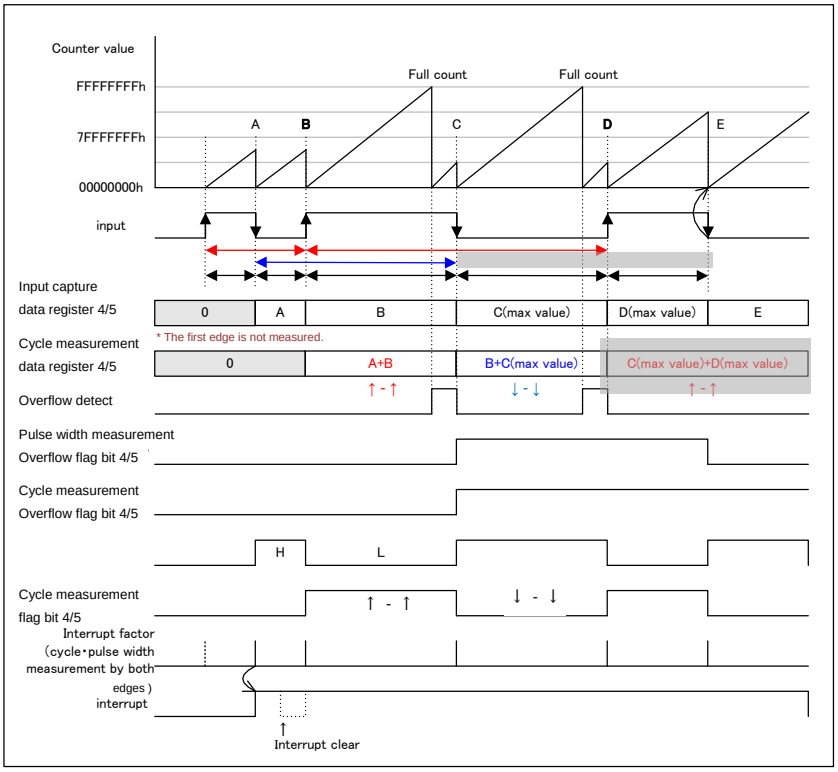
Page	Section	Change Results
871	22.7.11.1	(Example) To calculate a two-phase pulse with OCU6, 7, cycle A, and one-fourth phase difference in "7.11.1. Toggle Output Pulse" should be modified as follows: (Error) (Example) To calculate a two-phase pulse with OCU6, 7, cycle A, and one-fourth phase difference  (Correct) (Example) To calculate a two-phase pulse with OCU6, 7, cycle A, and one-fourth phase difference 

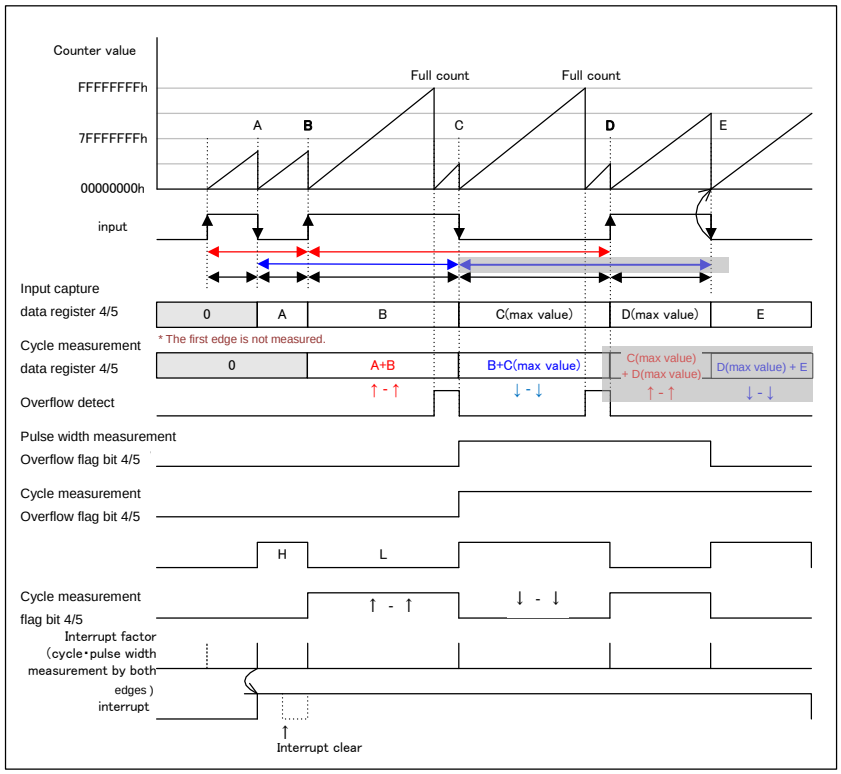
Page	Section	Change Results
872	22.7.11.2	(Example) To calculate the PWM with OCU6, 7, cycle A, and duty 1/4 in "7.11.2. PWM Output" should be modified as follows: (Error) (Example) To calculate the PWM with OCU6, 7, cycle A, and duty 1/4  (Correct) (Example) To calculate the PWM with OCU6, 7, cycle A, and duty 1/4 
878	23.1	The following description should be added in "1. Overview". (Correct) The numbers of available external input pins are shown below. MB91F52xB (64pin) : 5 MB91F52xD (80pin) : 5 MB91F52xF (100pin) : 6 MB91F52xJ (120pin) : 6 MB91F52xK (144pin) : 6 MB91F52xL (176pin) : 6

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>881</td><td>23.4</td><td>
Table 4-1 Table of Base Addresses (Base_addr) and External Pins in "4. Registers" should be modified as follows:

(Error)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th>External pin</th></tr>
<tr><th>ICU input</th></tr>
<tr><td>4</td><td>0x0FD0</td><td>ICU4_0/ICU4_1/ICU4_2</td></tr>
<tr><td>5</td><td>0x0FD0</td><td>ICU5_0/ICU5_1</td></tr>
<tr><td>6</td><td>0x0FDC</td><td>ICU6_0/ICU6_1</td></tr>
<tr><td>7</td><td>0x0FDC</td><td>ICU7_0/ICU7_1</td></tr>
<tr><td>8</td><td>0x0FE8</td><td>ICU8_0/ICU8_1</td></tr>
<tr><td>9</td><td>0x0FE8</td><td>ICU9_0/ICU9_1</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th colspan="5">External pin (ICU input)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD</th><th>MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK
MB91F52xL</th></tr>
<tr><td>4</td><td>0x0FD0</td><td>ICU4_2</td><td>ICU4_2</td><td>ICU4_2</td><td>ICU4_1/
ICU4_2</td><td>ICU4_0/
ICU4_1/
ICU4_2</td></tr>
<tr><td>5</td><td>0x0FD0</td><td>ICU5_0/
ICU5_1</td><td>ICU5_0/
ICU5_1</td><td>ICU5_0/
ICU5_1</td><td>ICU5_0/
ICU5_1</td><td>ICU5_0/
ICU5_1</td></tr>
<tr><td>6</td><td>0x0FDC</td><td>ICU6_0</td><td>ICU6_0/
ICU6_1</td><td>ICU6_0/
ICU6_1</td><td>ICU6_0/
ICU6_1</td><td>ICU6_0/
ICU6_1</td></tr>
<tr><td>7</td><td>0x0FDC</td><td>none</td><td>none</td><td>ICU7_0/
ICU7_1</td><td>ICU7_0/
ICU7_1</td><td>ICU7_0/
ICU7_1</td></tr>
<tr><td>8</td><td>0x0FE8</td><td>ICU8_0/
ICU8_1</td><td>ICU8_0/
ICU8_1</td><td>ICU8_0/
ICU8_1</td><td>ICU8_0/
ICU8_1</td><td>ICU8_0/
ICU8_1</td></tr>
<tr><td>9</td><td>0x0FE8</td><td>ICU9_1</td><td>ICU9_1</td><td>ICU9_1</td><td>ICU9_0/
ICU9_1</td><td>ICU9_0/
ICU9_1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
891	23.5.1	Figure 5-1 Example of 32-Bit Input Capture Operation in "5.1. Capture and Interrupt Timings" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
895	23.5.3	Figure 5-3 Example of the Cycle and Pulse Width Measurement Operation (The both edges are specified) in "5.3. Cycle and Pulse Width Measurement Operation" should be modified as follows: (Error) 

Page	Section	Change Results
		(Correct) 
908	23.9	The following sentences should be deleted from ● Notes when interrupt is processed in "9. Notes". (Error) When the level of external input pin (ICUn) switches while the interrupt routine is being processed after the interrupt request flag (ICPn) of the input capture control register (ICS) is set to "1", the effective edge instruction (IEIn) of the input capture control register (ICS) shows the detected latest edge. Moreover (Correct) Deleted.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>915</td><td>24.4</td><td>The following table should be added in "4. Registers".

(Correct)
■ Table of external pins
<table>
<tr><th rowspan="2">Channel</th><th colspan="2">External pins (FRCK)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD, MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL</th></tr>
<tr><td>0</td><td>FRCK0_0</td><td>FRCK0_0</td></tr>
<tr><td>1</td><td>FRCK1_1</td><td>FRCK1_0/FRCK1_1</td></tr>
<tr><td>2</td><td>FRCK2_0</td><td>FRCK2_0</td></tr>
</table>
■ List of registers</td></tr>
<tr><td>930</td><td>24.4.2.3</td><td>The description under the table in [bit22] STOP: Timer enable bit in "4.2.3. Timer State Control Register: TCCS0 to TCCS2" should be modified as follows:

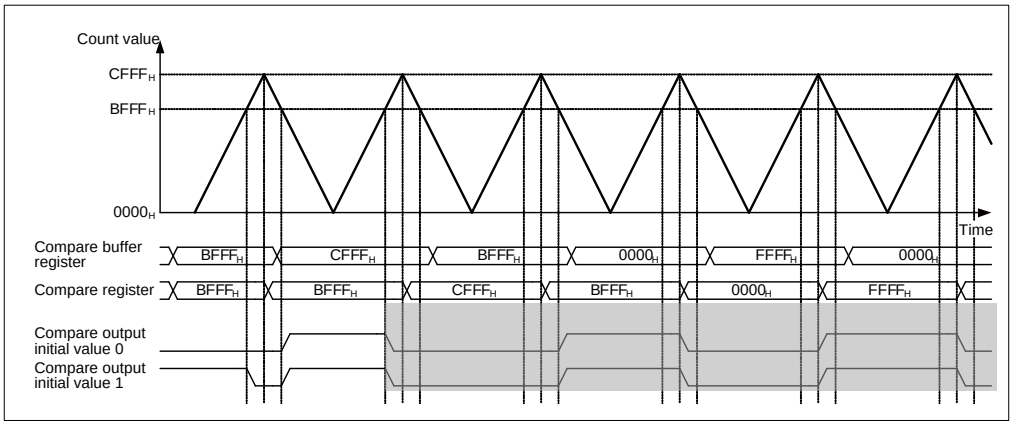
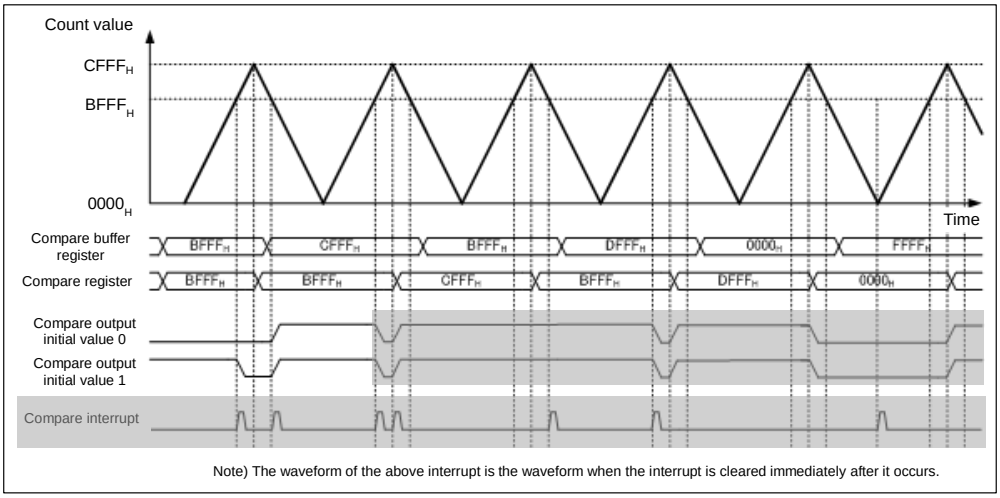
(Error)
· The free-run timer will not be initialized even when the SCLR: bit20 of the timer state control register (TCCS) is set to "1" while the free-run timer is inactive (this bit=0)

(Correct)
· The free-run timer will not be initialized even when the SCLR: bit20 of the timer state control register (TCCS) is set to "1" while the free-run timer is inactive (this bit=1).</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>957</td><td>24.5.3</td><td>The bit widths in Table 5-2 Table for Registers in "5.3. Operation of the Free-run Timer Selector" should be modified as follows:

(Error)
<table>
<tr><th>Resources</th><th>Register</th><th>Remarks</th></tr>
<tr><td>OCU0</td><td>FRS0:OS0[2:0]</td><td rowspan="2">16-bit output compare</td></tr>
<tr><td>OCU1</td><td>FRS0:OS1[2:0]</td></tr>
<tr><td>:
:</td><td>:
:</td><td>:
:</td></tr>
<tr><td>ADT22</td><td>FRS4:AS22[2:0]</td><td rowspan="2">A/D activation compare</td></tr>
<tr><td>ADT23</td><td>FRS4:AS23[2:0]</td></tr>
</table>

(Correct)
<table>
<tr><th>Resources</th><th>Register</th><th>Remarks</th></tr>
<tr><td>OCU0</td><td>FRS0:OS0[1:0]</td><td rowspan="2">16-bit output compare</td></tr>
<tr><td>OCU1</td><td>FRS0:OS1[1:0]</td></tr>
<tr><td>:
:</td><td>:
:</td><td>:
:</td></tr>
<tr><td>ADT22</td><td>FRS4:AS22[1:0]</td><td rowspan="2">A/D activation compare</td></tr>
<tr><td>ADT23</td><td>FRS4:AS23[1:0]</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
993	25.5.2.4	Figure 5-17 Case #5 Where the Free-run Timer Is in Up/Down Count Mode in "5.2.4. Operation of 16-bit Output Compare and Free-run timer" should be modified as follows: (Error)  (Correct)  Note) The waveform of the above interrupt is the waveform when the interrupt is cleared immediately after it occurs.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>995</td><td>25.5.3</td><td>"5.3. Notes on Using 16-bit Output Compare" should be modified as follows:

(Error)
● Read-modify-write
The interrupt request flag bits (IOP0), (IOP1), (IOP2), (IOP3), (IOP4) and (IOP5) are "1" when read using a read-modify-write instruction.

● Note on interrupts
When you set the interrupt flag bits (IOP1/IOP0) of the compare control register (OCS) to "1", then enable the interrupt request (set the interrupt request enable bits (IOE1/IOE0) of the OCS to "1"), the controller cannot return from the interrupt processing. Be sure to clear the interrupt flag bits (IOP1/IOP0).

(Correct)
● Read-modify-write
The interrupt request flag bits (IOP0) and (IOP1) are "1" when read using a read-modify-write instruction.</td></tr>
<tr><td>1001</td><td>26.4</td><td>The following table should be added in "4. Registers".

(Correct)
■ Table of external pins
<table>
<tr><th rowspan="2">Channel</th><th colspan="5">External pin (ICU input)</th></tr>
<tr><th>MB91F52xB</th><th>MB91F52xD</th><th>MB91F52xF</th><th>MB91F52xJ</th><th>MB91F52xK,
MB91F52xL</th></tr>
<tr><td>0</td><td>ICU0_3</td><td>ICU0_3</td><td>ICU0_2/
ICU0_3</td><td>ICU0_1/
ICU0_2/
ICU0_3</td><td>ICU0_0/
ICU0_1/
ICU0_2/
ICU0_3</td></tr>
<tr><td>1</td><td>ICU1_1/
ICU1_2</td><td>ICU1_1/
ICU1_2</td><td>ICU1_1/
ICU1_2/
ICU1_3</td><td>ICU1_1/
ICU1_2/
ICU1_3</td><td>ICU1_0/
ICU1_1/
ICU1_2/
ICU1_3</td></tr>
<tr><td>2</td><td>ICU2_2/
ICU2_3</td><td>ICU2_2/
ICU2_3</td><td>ICU2_1/
ICU2_2/
ICU2_3</td><td>ICU2_1/
ICU2_2/
ICU2_3</td><td>ICU2_0/
ICU2_1/
ICU2_2/
ICU2_3</td></tr>
<tr><td>3</td><td>ICU3_0/
ICU3_1/
ICU3_3</td><td>ICU3_0/
ICU3_1/
ICU3_2/
ICU3_3</td><td>ICU3_0/
ICU3_1/
ICU3_2/
ICU3_3</td><td>ICU3_0/
ICU3_1/
ICU3_2/
ICU3_3</td><td>ICU3_0/
ICU3_1/
ICU3_2/
ICU3_3</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1013</td><td>26.5.3</td><td>The following sentence should be deleted from "5.3. Notes on Using the 16-bit Input Capture".

(Error)
The input capture register value is undefined after a reset.</td></tr>
<tr><td>1021</td><td>27.4</td><td>Table 4-1 Correspondence between Pins and Channels in "4. Registers" should be modified as follows:

(Error)
<table>
<tr><th>Channel number</th><th colspan="3">External signal input pins</th></tr>
<tr><td>0</td><td>AIN0_0, AIN0_1</td><td>BIN0_0, BIN0_1</td><td>ZIN0_0, ZIN0_1, ZIN0_2</td></tr>
<tr><td>1</td><td>AIN1_0, AIN1_1</td><td>BIN1_0, BIN1_1</td><td>ZIN1_0, ZIN1_1</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="3">Channel</th><th colspan="9">External signal input pins</th></tr>
<tr><th colspan="3">MB91F52xB</th><th colspan="3">MB91F52xD</th><th colspan="3">MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL</th></tr>
<tr><th>AIN</th><th>BIN</th><th>ZIN</th><th>AIN</th><th>BIN</th><th>ZIN</th><th>AIN</th><th>BIN</th><th>ZIN</th></tr>
<tr><td>0</td><td>AIN0_0</td><td>BIN0_0/ BIN0_1</td><td>ZIN0_2</td><td>AIN0_0/ AIN0_1</td><td>BIN0_0/ BIN0_1</td><td>ZIN0_2</td><td>AIN0_0/ AIN0_1</td><td>BIN0_0/ BIN0_1</td><td>ZIN0_0/ ZIN0_1/ ZIN0_2</td></tr>
<tr><td>1</td><td>AIN1_0/ AIN1_1</td><td>BIN1_0</td><td>ZIN1_0/ ZIN1_1</td><td>AIN1_0/ AIN1_1</td><td>BIN1_0/ BIN1_1</td><td>ZIN1_0/ ZIN1_1</td><td>AIN1_0/ AIN1_1</td><td>BIN1_0/ BIN1_1</td><td>ZIN1_0/ ZIN1_1</td></tr>
</table>
</td></tr>
<tr><td>1056</td><td>28.4.2</td><td>The following description in "4.2. Sub-second Register" should be modified as follows:

(Error)
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.

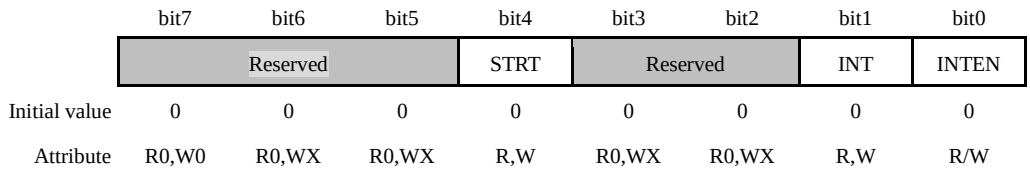
(Correct)
This value will be reloaded as soon as the sub-second counter (22-bit down counter) becomes "0". To modify the sub-second register, confirm that no reload operations are being performed during the writing instruction. Otherwise, the sub-second counter (22-bit down counter) will load a wrong value that combines both new and old data bytes. Generally, update the sub-second register 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.</td></tr>
</table>

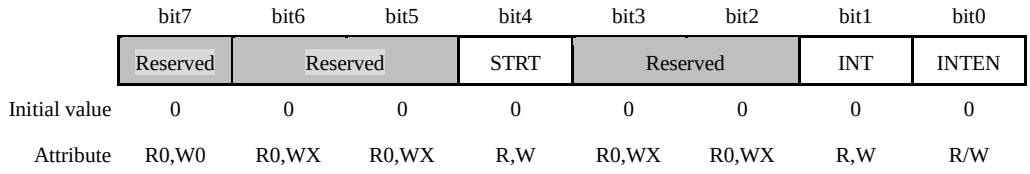
Page	Section	Change Results
1059	28.4.3	The following description in "4.3. Day/Hour/Minute/Second Register" should be modified. (Error) · As this series does not provide the RTC detection reset function, the Day/Hour/Minute/Second registers are cleared only in case of power-on reset. Therefore, when the microcomputer internal low-voltage detection flag is set, the Day/Hour/Minute/Second registers are recommended to be cleared. (Correct) · 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, clear the Day/Hour/Minute/Second registers.
1076	28.9	The following description in notes should be modified. (Error) · 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). (Correct) 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, 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, use interrupt (INT0) to read the number of days and time (Day/Hour/Minute/Second).

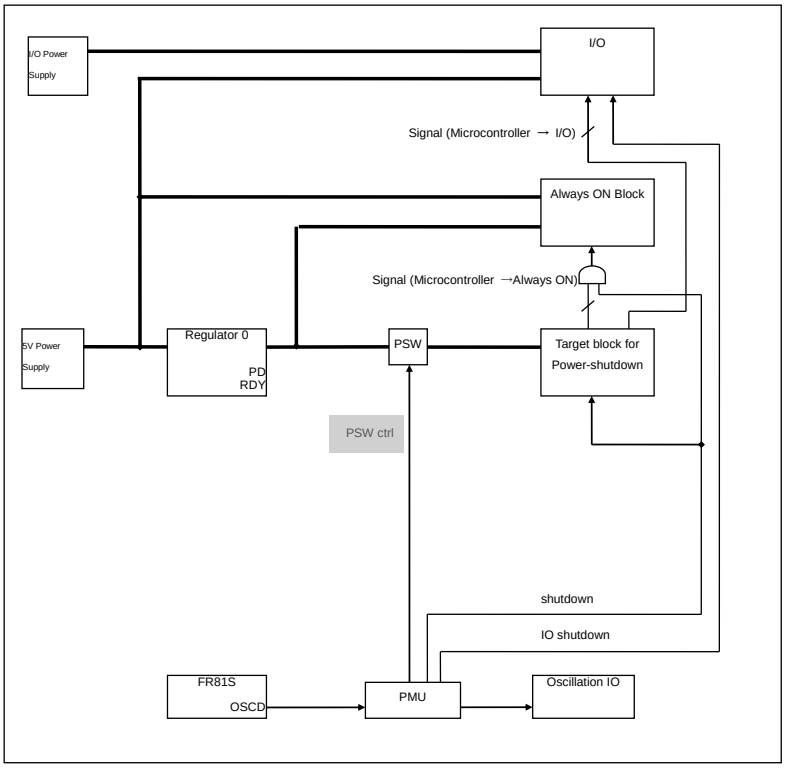
<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1076</td><td>28.9</td><td>The description in "9. Notes" should be modified as follows:

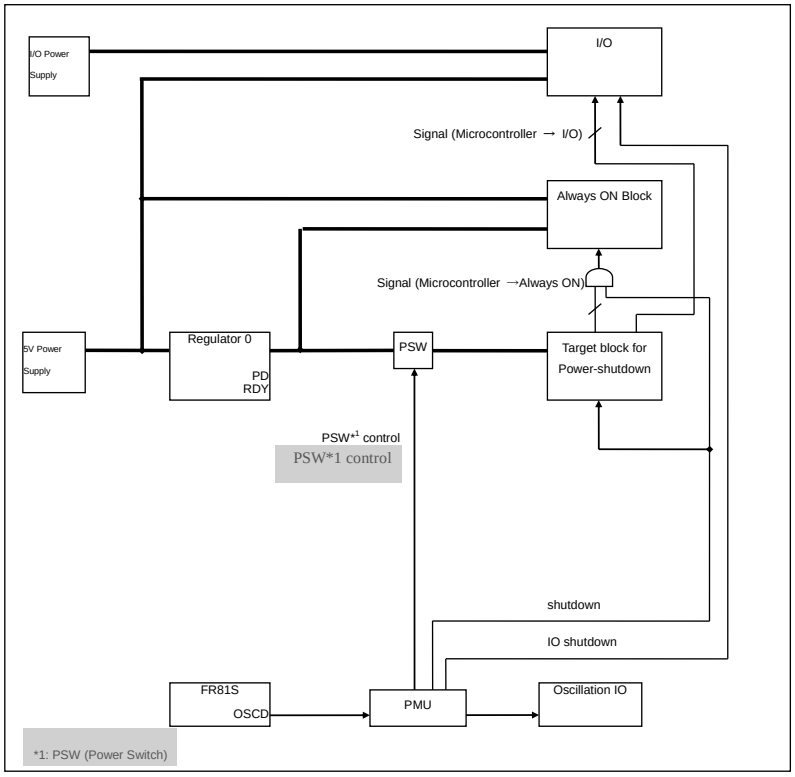
(Error)
· The internal reset is issued at the return from the standby watch mode (power shutdown). Therefore, only the reset factors (power-on reset, internal low-voltage reset, and simultaneous assert of RSTX and NMIX) are accepted. At this time, the register of the RTC is not initialized. If the reset 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

(Correct)
· 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 return, initialize the register of RTC before using.</td></tr>
<tr><td>1082</td><td>29.4.1</td><td>The bit configuration diagram of "4.1. Calibration Unit Control Register 0: CUCR0 (Calibration Unit Control Register 0) should be modified as follows:

(Error)


(Correct)

</td></tr>
</table>

Page	Section	Change Results
1093	29.5.3	The following sentences should be added at the end of the following description in "5.3. Note". (Error) The counter value will become invalid in such a case that transition to standby mode occurs. Write "0" to the STRT bit to stop, and then write "1" again to redo. (Correct) The counter value will become invalid in such a case that transition to standby mode occurs. Write "0" to the STRT bit to stop, and then write "1" again to redo. TOSC32/OSC100 > 2 × TOSC4 + 3 × TCLKP needs to be satisfied. TOSC4 : main clock cycle TOSC32 : sub clock cycle TOSC100 : oscillation cycle of CR oscillation circuit TCLKP : peripheral clock oscillation cycle
1098	30.3	Figure 3-1 in "3. Configuration" should be modified as follows: (Error) 

Page	Section	Change Results
		(Correct) 
1105	30.4.3	The description in "4.3. PoWeR on TiMing Control Register: PWRTMCTL (PoWeR on TiMing ConTroL register)" should be modified as follows: (Error) 4.3. PoWeR on TiMing Control Register : PWRTMCTL (PoWeR on TiMing ConTroL register) The bit configurations of the PoWeR on TiMing control register are shown below (Correct) 4.3. Power on Timing Control Register: PWRTMCTL (PoWeR on TiMing ConTroL register) The bit configurations of Power on Timing control register are shown below.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1105</td><td>30.4.3</td><td>The table in [bit2 to bit0] PTC in "4.3. PoWeR on TiMing Control Register: PWRTMCTL" should be modified as follows:

(Error)
<table>
<tr><th>PTC[2:0]</th><th>Rising time</th><th>Remarks (PMUCLK=32kHz)</th></tr>
<tr><td>000</td><td>3 × (1/PMUCLK)</td><td>90µS</td></tr>
<tr><td>001</td><td>9 × (1/PMU CLK)</td><td>270µS</td></tr>
<tr><td>010</td><td>15 × (1/PMU CLK)</td><td>450µS</td></tr>
<tr><td>011</td><td>27 × (1/PMU CLK)</td><td>810µS</td></tr>
<tr><td>100</td><td>Prohibit</td><td>-</td></tr>
<tr><td>101</td><td>6 × (1/PMU CLK)</td><td>180µS</td></tr>
<tr><td>110</td><td>12 × (1/PMU CLK)</td><td>360µS</td></tr>
<tr><td>111</td><td>21 × (1/PMU CLK)</td><td>630µS</td></tr>
</table>
(Correct)
<table>
<tr><th>PTC[2:0]</th><th>Rising time</th><th>Remarks (PMUCLK=32kHz)</th></tr>
<tr><td>000</td><td>5 × (1/PMUCLK)</td><td>150µs</td></tr>
<tr><td>001</td><td>11 × (1/PMU CLK)</td><td>330µs</td></tr>
<tr><td>010</td><td>17 × (1/PMU CLK)</td><td>510µs</td></tr>
<tr><td>011</td><td>29 × (1/PMU CLK)</td><td>870µs</td></tr>
<tr><td>100</td><td>Prohibit</td><td>-</td></tr>
<tr><td>101</td><td>8 × (1/PMU CLK)</td><td>240µs</td></tr>
<tr><td>110</td><td>14 × (1/PMU CLK)</td><td>420µs</td></tr>
<tr><td>111</td><td>23 × (1/PMU CLK)</td><td>690µs</td></tr>
</table></td></tr>
<tr><td>1108</td><td>30.4.6</td><td>The description in [bit5] MTIF in "4.6. PMU Interrupt Flag Register 2: PMUINTF2" should be modified as follows:

(Error)
This bit is enabled only at shutdown.
This bit is cleared by writing "0". Writing "1" does not affect operation

(Correct)
This bit is enabled only at shutdown.
This bit is cleared by writing "0". Writing "1" does not affect operation.
The internal reset is issued at the return from the standby mode (power-shutdown) and the main timer interrupt flag is not set.</td></tr>
</table>

Page	Section	Change Results
1108	30.4.6	The description in [bit4] STIF in "4.6. PMU Interrupt Flag Register 2: PMUINTF2" should be modified as follows: (Error) This bit is enabled only at shutdown. This bit is cleared by writing "0". Writing "1" does not affect operation. (Correct) This bit is enabled only at shutdown. This bit is cleared by writing "0". Writing "1" does not affect operation. The internal reset is issued at the return from the standby mode (power-shutdown) and the sub timer interrupt flag is not set.
1148	30.5.10	The description in "5.10. Restrictions on Power-Shutdown and Normal Standby Control" should be modified as follows: (Error) · When the CPU is operating with PLL · When the OCD is being enabled to operate · When missing the clock by CSV function *2 *3 The standby control does not operate in the states above, but the CPU is in the standby state (Correct) · When the CPU is operating with PLL · When the OCD is being enabled to operate The standby control does not operate in the state above, but the CPU is in the standby state. (See 5.8.) · When missing the clock by CSV function *2, *3. When standby (power-shutdown) or transition is directed in the state above, it usually becomes standby processing.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1149</td><td>30.5.10</td><td>Table 5-2 List of Registers stored at Return from Standby Mode with Power-shutdown in "5.10. Restrictions on Power-Shutdown and Normal Standby Control" should be modified as follows:

(Error)
<table>
<tr><td rowspan="4">Low-voltage detection setting register (External low-voltage detection)</td><td>LVD5R.LVD5R_F</td><td>Flag</td><td>0584H bit0</td><td></td></tr>
<tr><td>LVD5F.LVD5F_F</td><td>Flag</td><td>0585H bit0</td><td></td></tr>
<tr><td>LVD5F.LVD5F_PD</td><td>Register</td><td>0585H bit7</td><td></td></tr>
<tr><td>LVD5F.LVD5F_OE</td><td>Register</td><td>0585H bit3</td><td></td></tr>
<tr><td rowspan="3">Low-voltage detection setting register (Internal low-voltage detection)</td><td>LVD.LVD_F</td><td>Flag</td><td>0586H bit0</td><td></td></tr>
<tr><td>LVD.LVD_PD</td><td>Register</td><td>0586H bit7</td><td></td></tr>
<tr><td>LVD.LVD_OE</td><td>Register</td><td>0586H bit3</td><td></td></tr>
</table>
(Correct)
<table>
<tr><td rowspan="4">Low-voltage detection register (External low-voltage detection)</td><td>LVD5R.LVD5R_F</td><td>Flag</td><td>0584H bit0</td><td></td></tr>
<tr><td>LVD5F.LVD5F_F</td><td>Flag</td><td>0585H bit0</td><td></td></tr>
<tr><td>LVD5F.LVD5F_PD</td><td>Register</td><td>0585H bit7</td><td></td></tr>
<tr><td>LVD5F.LVD5F_OE</td><td>Register</td><td>0585H bit3</td><td></td></tr>
<tr><td rowspan="3">Low-voltage detection register (Internal low-voltage detection)</td><td>LVD.LVD_F</td><td>Flag</td><td>0586H bit0</td><td></td></tr>
<tr><td>LVD.LVD_PD</td><td>Register</td><td>0586H bit7</td><td></td></tr>
<tr><td>LVD.LVD_OE</td><td>Register</td><td>0586H bit3</td><td></td></tr>
</table></td></tr>
<tr><td>1149</td><td>30.5.10</td><td>Table 5-2 List of Registers stored at Return from Standby Mode with Power-shutdown in "5.10. Restrictions on Power-Shutdown and Normal Standby Control" should be modified as follows:

(Error)
<table>
<tr><td rowspan="6">RTC register</td><td>WTDR</td><td>Register</td><td>055EH-055FH</td><td></td></tr>
<tr><td>WTCR</td><td>Register</td><td>0561H-0563H</td><td>*1</td></tr>
<tr><td>WTBR</td><td>Register</td><td>0565H-0567H</td><td>*1</td></tr>
<tr><td>WTHR</td><td>Register</td><td>0568H</td><td></td></tr>
<tr><td>WTMR</td><td>Register</td><td>0569H</td><td></td></tr>
<tr><td>WTSR</td><td>Register</td><td>056AH</td><td></td></tr>
</table>
(Correct)
<table>
<tr><td rowspan="6">RTC register</td><td>WTDR</td><td>Register</td><td>055EH-055FH</td><td></td></tr>
<tr><td>WTCR</td><td>Register</td><td>0561H-0563H</td><td>*1</td></tr>
<tr><td>WTBR</td><td>Register</td><td>0565H-0567H</td><td></td></tr>
<tr><td>WTHR</td><td>Register</td><td>0568H</td><td></td></tr>
<tr><td>WTMR</td><td>Register</td><td>0569H</td><td></td></tr>
<tr><td>WTSR</td><td>Register</td><td>056AH</td><td></td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1156	31.4.1	The description in "4.1. Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect internal power fall register)" should be modified as follows: (Error) This register has the internal low-voltage detection flag (LVD_F). (Correct) This register has the internal low-voltage detection flag (LVD_F) and the control bit.
1157	31.4.1	The description under the table of [bit0] in "4.1. Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect internal power fall register) should be modified as follows: (Error) If a power-on reset or a fall in the internal power supply voltage is detected, the LVD_F bit is set to "1". (Correct) If a fall in the internal power supply voltage is detected, the LVD_F bit is set to "1".
1166	32.4.1	The description in "4.1. External Low-Voltage Detection Rise Detection Register: LVD5R (Low-Voltage Detect external 5v Rise register)" should be modified as follows: (Error) This register is used in order to clear the low-voltage detection reset flag, etc. (Correct) This register is the external power supply voltage rise detection flag.
1167	32.4.2	The description in "4.2. External Low-Voltage Detection Fall Detection Register: LVD5F (Low-Voltage Detect external 5v Fall register)" should be modified as follows: (Error) This register is used in order to clear the low-voltage detection reset flag, etc. (Correct) This register is used in order to clear the low-voltage detection reset flag and set the low-voltage detection circuit.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1170</td><td>32.6</td><td>The description in ● Hysteresis of detection/reset release voltage in "6. Notes" should be modified as follows:

(Error)
· Since the detection voltage and reset voltage exhibit hysteresis of 0.1V, the reset release voltage becomes the set detection voltage + 0.1V. For example, when LVD5F: 4.1V±8%

(Correct)
Since the detection voltage and reset voltage exhibit hysteresis of 0.1V, the reset release voltage becomes the set detection voltage + 0.1V.
For fall detection power supply voltage, the set detection voltage indicates the detection voltage.
For example, when 4.1V ± 8% is set, the reset release voltage becomes 4.2V ± 8%.
For rise detection power supply voltage, the set detection voltage indicates the reset release voltage.
For example, when 2.5V ± 8% is set, the detection voltage becomes 2.4V ± 8%.</td></tr>
<tr><td>1187</td><td>34.3.1</td><td>The attributes in bit6, bit5, and bit4 in "3.1. Clock Supervisor Control Register: CSVCR" should be modified.

(Error)
<table><tr><th></th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr><tr><td></td><td>SCKS</td><td>MM</td><td>SM</td><td>RCE</td><td>MSVE</td><td>SSVE</td><td>Reserved</td><td>Reserved</td></tr><tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>Attribute</td><td>R/W</td><td>R,W0</td><td>R,W0</td><td>R/W0</td><td>R/W</td><td>R/W</td><td>R0/W0</td><td>R0/W0</td></tr></table>
(Correct)
<table><tr><th></th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr><tr><td></td><td>SCKS</td><td>MM</td><td>SM</td><td>RCE</td><td>MSVE</td><td>SSVE</td><td>Reserved</td><td>Reserved</td></tr><tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>Attribute</td><td>R/W</td><td>R,W</td><td>R,W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R0/W0</td><td>R0/W0</td></tr></table></td></tr>
<tr><td>1187</td><td>34.3.1</td><td>The note in [bit6] MM (Main clock Missing) in "3.1. Clock Supervisor Control Register: CSVCR" should be modified as follows:

(Error)
Note:
Do not enable the PLL oscillation operation when this bit is "1".

(Correct)
Note:
Do not enable the PLL/SSCG oscillation operation when this bit is "1".</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1210</td><td>36.1</td><td>The following description should be added in "1. Overview".

(Correct)
External bus interface is available with the following models:
MB91F52xK (144 pin), MB91F52xL (176 pin)</td></tr>
<tr><td>1258</td><td>36.5.10</td><td>The content of ■ Program Example in Figure 5-12 CS1 Settings Sample Program in ● CS settings and update sample program in "5.10. CS Setting Flow" should be modified as follows:

(Error)
_set_ACR1
ldi #_ACR0, r0 //#_ACR1 is the ACR1 address value
ldi 0x40, r1 // Set ACR1 to 0x40
st r1, @r0
st r1, @r0

(Correct)
_set_ACR1
ldi #_ACR1, r0 //#_ACR1 is the ACR1 address value
ldi 0x40, r1 // Set ACR1 to 0x40
st r1, @r0</td></tr>
<tr><td>1258</td><td>36.5.10</td><td>A part of the content of ■ ACR1 Setting Example in Figure 5-12 CS1 Settings Sample Program in ● CS settings and update sample program in "5.10. CS Setting Flow" should be deleted as follows:

(Error)<table><tr><td>Data bus width</td><td>16bit</td></tr><tr><td>Address output type</td><td>Normal</td></tr><tr><td>Bus type</td><td>Address/data multiplexed bus</td></tr><tr><td>Write signal type</td><td>Write type 0</td></tr></table>
(Correct)<table><tr><td>Data bus width</td><td>16bit</td></tr><tr><td>Address output type</td><td>Normal</td></tr><tr><td>Bus type</td><td>Address/data multiplexed bus</td></tr></table></td></tr>
</table>

Page	Section	Change Results
1046	40.2	The following description should be added in "2. Features". The note should be modified. (Error) Note: I²C supports only ch.3 to ch.8, ch.10, and ch.11. (Correct) The numbers of available channels are shown below. MB91F52xB (64pin) : 8 MB91F52xD (80pin) : 9 MB91F52xF (100pin) : 12 MB91F52xJ (120pin) : 12 MB91F52xK (144pin) : 12 MB91F52xL (176pin) : 12 For available external signals, see Table 4-1, Table 4-2, Table 4-3, Table 4-4, and Table 4-5. Note: I²C supports only relocation 0 of ch.3 to ch.8, ch.10 and ch.11.
1050	40.2	The content of the reception interrupt in the interrupt request in the table in ● Assist Mode in ■ LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) in "2. Features" should be modified as follows: (Error) (3) Various error interrupts (LIN bus error, LIN ID parity error, LIN Sync Data error, LIN ID parity error, framing error, overrun error, and LIN checksum error) (Correct) (3) Various error interrupts (LIN bus error, LIN ID parity error, LIN Sync Data error, framing error, overrun error, and LIN checksum error)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1052</td><td>40.2</td><td>"Reception: Supported" in the function column for DMA transfer support in the table of ■ I²C in "2. Features" should be modified as follows:

(Error)
<table><tr><td>DMA transfer support</td><td>Transmission: Supported
Reception: Supported
Status: Not supported</td></tr></table>
(Correct)
<table><tr><td>DMA transfer support</td><td>Transmission: Supported
Reception: Not supported
Status: Not supported</td></tr></table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1054</td><td>40.4</td><td>■ Table of Base Addresses (Base_addr) and External Pins in "4. Registers" should be modified as follows:
The table should be divided into (MB91F52xB), (MB91F52xD), (MB91F52xF), (MB91F52xJ), and (MB91F52xK, MB91F52xL).

(Error)
Table 4-1 Table of Base Addresses (Base_addr) and External Pins
<table>
<tr><th rowspan="2">Channels</th><th rowspan="2">Base_addr</th><th colspan="4">External pin</th></tr>
<tr><th>SCK</th><th>SOT</th><th>SIN</th><th>SCS</th></tr>
<tr><td>0</td><td>0x1750</td><td>SCK0_0/
SCK0_1</td><td>SOT0_0/
SOT0_1</td><td>SIN0_0/
SIN0_1</td><td>-</td></tr>
<tr><td>1</td><td>0x1778</td><td>SCK1_0/
SCK1_1</td><td>SOT1_0</td><td>SIN1_0</td><td>SCS1_0/
SCS1_1</td></tr>
<tr><td>2</td><td>0x17A0</td><td>SCK2_0</td><td>SOT2_0/
SOT2_1</td><td>SIN2_0</td><td>SCS2_0</td></tr>
<tr><td>3</td><td>0x17C8</td><td>SCK3_0/
SCK3_1/
SCK3_2</td><td>SOT3_0/
SOT3_1/
SOT3_2</td><td>SIN3_0/
SIN3_1</td><td>SCS3_0/
SCS3_1</td></tr>
<tr><td>4</td><td>0x17F0</td><td>SCK4_0/
SCK4_1/
SCK4_2</td><td>SOT4_0/
SOT4_1/
SOT4_2</td><td>SIN4_0/
SIN4_1</td><td>SCS40_0/SCS40_1
SCS41_0/SCS41_1
SCS42_0/SCS42_1
SCS43_0/SCS43_1</td></tr>
<tr><td>5</td><td>0x1818</td><td>SCK5_0</td><td>SOT5_0</td><td>SIN5_0</td><td>SCS50_0
SCS51_0
SCS52_0
SCS53_0</td></tr>
<tr><td>6</td><td>0x1840</td><td>SCK6_0</td><td>SOT6_0</td><td>SIN6_0</td><td>SCS60_0
SCS61_0
SCS62_0
SCS63_0</td></tr>
<tr><td>7</td><td>0x1868</td><td>SCK7_0</td><td>SOT7_0</td><td>SIN7_0</td><td>SCS70_0
SCS71_0
SCS72_0
SCS73_0</td></tr>
<tr><td>8</td><td>0x1890</td><td>SCK8_0</td><td>SOT8_0</td><td>SIN8_0</td><td>SCS8_0</td></tr>
<tr><td>9</td><td>0x18B8</td><td>SCK9_0</td><td>SOT9_0</td><td>SIN9_0</td><td>SCS9_0</td></tr>
<tr><td>10</td><td>0x18E0</td><td>SCK10_0/
SCK10_1</td><td>SOT10_0/
SOT10_1</td><td>SIN10_0</td><td>SCS10_0/
SCS10_1</td></tr>
<tr><td>11</td><td>0x1908</td><td>SCK11_0</td><td>SOT11_0</td><td>SIN11_0</td><td>SCS11_0</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1054	40.4	(Continued) (Correct) Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB)

Channels	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_1	SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	none	none	none	none	none	none	none
2	0x17A0	none	none	none	none	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0	none	none	none
4	0x17F0	SCK4_2	SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	none	SCS42_0	SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	none
6	0x1840	SCK6_0	SOT6_0	SIN6_0	none	none	none	SCS63_0
7	0x1868	none	none	none	none	none	none	none
8	0x1890	none	none	none	none	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	none	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	none	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5, ch.6, and ch.11 can be used as I^2C (standard mode support).)

Page	Section	Change Results
1055	40.4	(Continued)

Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	none	none	none	none	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0/	none	none	none
4	0x17F0	SCK4_1/ SCK4_2	SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0	SCS42_0/ SCS42_1	SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	none
6	0x1840	SCK6_0	SOT6_0	SIN6_0	none	SCS61_0	none	SCS63_0
7	0x1868	none	none	none	none	none	none	none
8	0x1890	none	none	none	none	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	none	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5, ch.6, and ch.11 can be used as I^2C (standard mode support).)

Page	Section	Change Results
1056	40.4	(Continued)

Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_1	SOT1_0	SIN1_0	SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_2	SOT3_2	SIN3_1	SCS3_0/	none	none	none
4	0x17F0	SCK4_1/ SCK4_2	SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.5 to ch.8, and ch.11 can be used as I^2C (standard mode support).)

Page	Section	Change Results
1057	40.4	(Continued)

Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_0/ SCK1_1	SOT1_0	SIN1_0	SCS1_0/ SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_0/ SCK3_1/ SCK3_2	SOT3_0/ SOT3_1/ SOT3_2	SIN3_0/ SIN3_1	SCS3_0/ SCS3_1	none	none	none
4	0x17F0	SCK4_0/ SCK4_1/ SCK4_2	SOT4_0/ SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	none	none	none	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_1	SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.3 and ch.4 can be used as I^2C (fast mode/standard mode support). Ch.5 to ch.8, and ch.11 can be used as I^2C (standard mode support).)

Page	Section	Change Results
1058	40.4	(Continued) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)

Channel	Base_addr	External pin						
		SCK SCL(*)	SOT SDA(*)	SIN	SCS0	SCS1	SCS2	SCS3
0	0x1750	SCK0_0/ SCK0_1	SOT0_0/ SOT0_1	SIN0_0/ SIN0_1	none	none	none	none
1	0x1778	SCK1_0/ SCK1_1	SOT1_0	SIN1_0	SCS1_0/ SCS1_1	none	none	none
2	0x17A0	SCK2_0	SOT2_0/ SOT2_1	SIN2_0	SCS2_0	none	none	none
3	0x17C8	SCK3_0/ SCK3_1/ SCK3_2	SOT3_0/ SOT3_1/ SOT3_2	SIN3_0/ SIN3_1	SCS3_0/ SCS3_1	none	none	none
4	0x17F0	SCK4_0/ SCK4_1/ SCK4_2	SOT4_0/ SOT4_1/ SOT4_2	SIN4_0/ SIN4_1	SCS40_0/ SCS40_1	SCS41_0/ SCS41_1	SCS42_0/ SCS42_1	SCS43_0/ SCS43_1
5	0x1818	SCK5_0	SOT5_0	SIN5_0	SCS50_0	SCS51_0	SCS52_0	SCS53_0
6	0x1840	SCK6_0	SOT6_0	SIN6_0	SCS60_0	SCS61_0	SCS62_0	SCS63_0
7	0x1868	SCK7_0	SOT7_0	SIN7_0	SCS70_0	SCS71_0	SCS72_0	SCS73_0
8	0x1890	SCK8_0	SOT8_0	SIN8_0	SCS8_0	none	none	none
9	0x18B8	SCK9_0	SOT9_0	SIN9_0	SCS9_0	none	none	none
10	0x18E0	SCK10_0/ SCK10_1	SOT10_0/ SOT10_1	SIN10_0	SCS10_0/ SCS10_1	none	none	none
11	0x1908	SCK11_0	SOT11_0	SIN11_0	SCS11_0	none	none	none

* : Pin names at I^2C setting (Pins with relocation 0 can be used as I^2C. Ch.3 and ch.4 can be used as I^2C (fast mode/standard mode support). Ch.5 to ch.8, ch.10, and ch.11 can be used as I^2C (standard mode support).)

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1060</td><td>40.4</td><td>A part of the contents of the register functions in Table 4-2 Registers Map in ■ Registers Map in "4. Registers" should be deleted as follows:

(Error)
Table 4-2 Registers Map
<table>
<tr><td>Base+
18_H</td><td>[UART]
Reserved
[CSIO]
TBYTE3n
[LIN]
LAMESRn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE2n
[LIN]
LAMERTn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE1n
[LIN]
LAMIERn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE0n
[LIN]
LAMTIDn/L
AMRIDn
[I^2C]
Reserved</td><td>[CSIO] Transfer byte register
[LIN] Transfer byte register
[LIN] LIN assist mode error status register
[LIN] LIN assist mode trouble examination register
[LIN] LIN assist mode interrupt enable register
[LIN] LIN assist mode transmission/reception ID register</td></tr>
</table>
(Correct)
Table 4-6 Registers Map
<table>
<tr><td>Base+
18_H</td><td>[UART]
Reserved
[CSIO]
TBYTE3n
[LIN]
LAMESRn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE2n
[LIN]
LAMERTn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE1n
[LIN]
LAMIERn
[I^2C]
Reserved</td><td>[UART]
Reserved
[CSIO]
TBYTE0n
[LIN]
LAMTIDn/L
AMRIDn
[I^2C]
Reserved</td><td>[CSIO] Transfer byte register
[LIN] LIN assist mode error status register
[LIN] LIN assist mode trouble examination register
[LIN] LIN assist mode interrupt enable register
[LIN] LIN assist mode transmission/reception ID register</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1061</td><td>40.4.1.1</td><td>The attributes in "4.1.1. Serial Mode Register: SMR (Serial Mode Register)" should be modified as follows:

(Error)
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>bit</td></tr>
<tr><td colspan="3">MD[2:0]</td><td>Reserved</td><td>SBL/
SCINV/
RIE</td><td>BDS/TIE</td><td>SCKE/
(Reserved)</td><td>SOE/
(Reserved)</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W0</td><td>R/W</td><td>R/W</td><td>R/W
(R/W0)</td><td>R/W
(R/W0)</td><td>Attribute</td></tr>
</table>

(Correct)
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>bit</td></tr>
<tr><td colspan="3">MD[2:0]</td><td>Reserved</td><td>SBL/
SCINV/
RIE</td><td>BDS/TIE</td><td>SCKE/
(Reserved)</td><td>SOE/
(Reserved)</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W0</td><td>R/W</td><td>R/W
(R/W0)</td><td>R/W
(R/W0)</td><td>R/W
(R/W0)</td><td>Attribute</td></tr>
</table>
</td></tr>
<tr><td>1063</td><td>40.4.1.1</td><td>The description in [bit2] BDS/TIE in "4.1.1. Serial Mode Register: SMR (Serial Mode Register)" should be modified as follows:

(Error)
[LIN]
LIN does not use this bit. Writing a value to this bit does not affect the operation.

[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).

(Correct)
[LIN]
LIN must always write "0" to this bit.

[UART] [CSIO]
· This bit selects whether to transfer the transfer serial data from the least significant bit (LSB-first, BDS=0) or from the most significant bit (MSB-first, BDS=1).
· When chip select is used with the master mode (SCR:MS=0), this bit is used for communication with serial chip select pin 0.</td></tr>
</table>

Page	Section	Change Results
1063	40.4.1.1	The notes in [bit2] BDS/TIE: Transfer direction selection bit/transmission interrupt enable bit in "4.1.1. Serial Mode Register: SMR" should be modified as follows: (Error) · While in slave mode (SCR:MS=0) (Correct) · While in slave mode (SCR:MS=1)
1063	40.4.1.1	The description in [bit1] SCKE: Serial clock output enable bit in "4.1.1. Serial Mode Register: SMR" should be modified as follows: (Error) When this bit is set to "0": The SCK pin functions as a general-purpose I/O port or serial clock input pin. (Correct) When this bit is set to "0": The SCK pin functions as a serial clock input pin.
1064	40.4.1.1	The description in [bit0] SOE: Serial data output enable bit in "4.1.1. Serial Mode Register: SMR" should be modified as follows: (Error) When this bit is set to "0": The SOT pin functions as a general-purpose I/O port. When this bit is set to "1": The SOUT pin functions as a serial data output pin (SOUT). (Correct) When this bit is set to "0": Serial data output stops. When this bit is set to "1": The SOT pin functions as a serial data output pin (SOT).
1065	40.4.1.2	The following note should be added in the description in [bit3] FRIIE in "4.1.2. FIFO Control Register 1: FCR1 (FIFO Control Register 1)". (Correct) Note: When the reception FIFO is used, set this bit to "1".

Page	Section	Change Results
1067	40.4.1.3	The description in [bit1] FE2 in "4.1.3. FIFO Control Register 0: FCR0 (FIFO Control Register 0)" should be modified as follows: (Error) · [UART] [CSIO] When any data is present in FIFO2 and UART is enabled for transmission (SCR:TXE=1) after FIFO2 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit. (Correct) · [UART] [CSIO] When any data is present in FIFO2 and it is enabled for transmission (SCR:TXE=1) after FIFO2 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will begin immediately. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
1068	40.4.1.3	The description in [bit0] FE1 in "4.1.3. FIFO Control register 0: FCR0 (FIFO Control Register 0)" should be modified as follows: (Error) · [UART] [CSIO] When any data is present in FIFO1 and UART is enabled for transmission (SCR:TXE=1) after FIFO1 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit. (Correct) · [UART] [CSIO] When any data is present in FIFO1 and it is enabled for transmission (SCR:TXE=1) after FIFO1 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will begin immediately. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
1080	40.4.2.5	[bit15] STST: Serial test bit in "4.2.5. Serial Aid Control Status Register: SACSR" should be modified as follows: (Error) This bit selects to enable or disable serial test mode. When serial test mode is enabled, SOUT and SIN will be connected in the multi-function serial interface so that the data output from SOUT can be received from SIN without any modification. When serial test mode is enabled, SOUT pin will be fixed to "H" and data input to SIN pin will be ignored. (Correct) This bit selects to enable or disable serial test mode. When serial test mode is enabled, SOT and SIN will be connected in the multi-function serial interface so that the data output from SOT can be received from SIN without any modification. When serial test mode is enabled, the SOT pin will be fixed to "H" and data input to the SIN pin will be ignored.

Page	Section	Change Results
1080	40.4.2.5	The following description should be added under the table of [bit15] STST: Serial test bit in "4.2.5 Serial Aid Control Status Register: SACSR. (Correct) Note: This bit can be changed only when transmission and reception is disabled (SCR:TXE=0, SCR:RXE=0).
1081	40.4.2.5	The following sentence should be deleted from the notes in [bit8] TINT: Timer Interrupt Flag in "4.2.5. Serial Aid Control Status Register: SACSR". (Error) · Performing software reset (SCR:UPCL="1") will reset this bit to "0". · A read with a read-modify-write instruction will read "1". · When synchronous transmission enable bit (TSYNE) is "1", this bit will not be set to "1". (Correct) · Performing software reset (SCR:UPCL="1") will reset this bit to "0". · A read with a read-modify-write instruction will read "1".
1081	40.4.2.5	The following sentences should be deleted from the note in [bit0] TMRE: Serial timer enable Bit in "4.2.5. Serial Aid Control Status Register: SACSR". (Error) Note: When you make synchronous transmission with serial timer change this bit when any of following condition is met · Transmission is disabled (SCR:TXE="0") · Transmission bus is idling (SSR:TBI="1")

Page	Section	Change Results
1083	40.4.2.7	A part of the following sentences in [bit15 to bit0] TC15-0: Comparison Bits in "4.2.7. Serial Timer Comparison Register: STMCR" should be deleted. (Error) These bits will be compared with the Serial Timer Register (STMR) and when these bits and the values of the STMR matched at the update timing of the STMR, the STMR will be set to "0". At this timing while synchronous transmission is disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:INT) will be set to "1" but when synchronous transmission is enabled (SACSR:TSYNE="1"), a transmission will be started. The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0). ·SACSR:TINT is set to "1". ·Transmission is activated synchronizing with transmission of the serial timer Notes: · When $(0000)_H$ is set to this register, the Serial Timer Register still indicates "0" · With "0000_H" set to this register while synchronous transmission is disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) will be fixed to "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000_B" while the timer is running. · This register can be changed only when serial timer is disabled (SACSR:TMRE="0"). (Correct) These bits will be compared with the Serial Timer Register (STMR) and when these bits and the values of the STMR matched at the update timing of the STMR, the STMR will be set to "0". At this timing, the timer interrupt flag (SACSR:TINT) will be set to "1". The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0). · SACSR:TINT is set to "1". Notes: · When $(0000)_H$ is set to this register, the Serial Timer Register still indicates "0". · With this "0000_H" set to this register, the timer interrupt flag (SACSR:TINT) will be fixed to "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000_B" while the timer is running. · This register can be changed only when serial timer is disabled (SACSR:TMRE="0").
1087	40.4.3.2	The following sentence should be added in the note in bit4 in "4.3.2. Serial Status Register: SSR (Serial Status Register)". · To use the SPI mode with the slave mode while the transmit FIFO is enabled (FCR0.FE2,FE1=0), set this bit to "1".

Page	Section	Change Results
1089	40.4.3.3	The following sentences should be added (before ■ ESCRn (n=0 to 11)) in the description in "4.3.3. Extended Serial Control Register: ESCR". Setting to serial chip select pins without assignment to external pins is disabled. For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below. Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB) Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD) Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF) Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)
1090	40.4.3.3	The following description should be added in the note in ESCR bit5 (CSFE) in "4.3.3. Extended Serial Control Register: ESCR". (Correct) · When this bit is set to "1", set the following settings: · Enable reception FIFO · Set the hold delay to 2 or larger (setting CSHD7-0 bit in SCSTR1 register to 2 or larger) · Set the length of each serial chip select data to 9 bits or smaller, or 10 bits or larger to communicate with two or more slave devices · Disable setting the serial chip select pins separately to 9 bits and smaller, and 10 bits and larger (Examples of disabled settings) serial chip select 0 = 9 bits serial chip select 1 = 10 bits (Examples of enabled settings) serial chip select 0 = 16 bits serial chip select 1 = 10 bits
1091	40.4.3.3	The following note should be added in ESCR bit4 and bit3 (WT1 and WT0) in "4.3.3. Extended Serial Control Register: ESCR". (Correct) Note: If this register is used when all of the following conditions are satisfied, set WT1 and WT0 to "00". · Chip select is used. · The SPI mode (SCR:SPI=1) is used. · "01_H" is set in TBYTE register. · The SCAM bit in SCSCR register is set to "1".

Page	Section	Change Results
1092	40.4.3.4	A part of the sentence (byte) in "4.3.4. Receive Data Register/Transmit Data Register: RDR/TDR (Receive Data Register/Transmit Data Register)" should be deleted as follows: (Error) ■ RDR1n-0n(n=0 to 11): Address Base addr+04H(Access: Byte, Half-word, Word) (Correct) ■ RDR1n-0n (n=0 to 11): Address Base addr + 04H (Access: Half-word, Word)
1095	40.4.3.5	The following sentences should be added (before ■ SACSRn (n=0 to 11)) in the description in "4.3.5. Serial Aid Control Status Register: SACSR". (Correct) Setting to serial chip select pins without assignment to external pins is disabled. For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below. Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB) Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD) Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF) Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)
1095	40.4.3.5	SACSR [bit15] STST: Serial TeST bit in "4.3.5. Serial Aid Control Status Register: SACSR" should be modified as follows: (Error) This bit is used to enable or disable the serial test mode. When the serial test mode is enabled, SOUT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOUT can be received from SIN without being processed. When the serial test mode is enabled, the SOUT pin will be fixed to "H", and data input into the SIN pin will be ignored (Correct) This bit is used to enable or disable the serial test mode. When serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed. When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input into the SIN pin will be ignored.

Page	Section	Change Results
1100	40.4.3.7	[bit15 to bit0] TC15-0: Compare bits in "4.3.7. Serial Timer Compare Register: STMCR" should be modified as follows: (Error) These bits are used to set compare values of the serial timer. These bits will be compared with the serial timer register (SMTR), and when these bits match the value of the serial timer register immediately after the serial timer register (SMTR) is updated, they will set the serial timer register to "0". Then, if synchronous transmission is disabled (SACSR:TSYNE="0"), these bits will set the timer interrupt flag (SACSR:INT) to "1", and if synchronous transmission is enabled (SACSR:TSYNE="1"), these bits will activate transmission. (Correct) These bits are used to set compare values of the serial timer. These bits will be compared with the serial timer register (SMTR), and when these bits match the value of the serial timer register immediately after the serial timer register (SMTR) is updated, they will set the serial timer register to "0". Then, if synchronous transmission is disabled (SACSR:TSYNE="0"), these bits will set the timer interrupt flag (SACSR:TINT) to "1", and if synchronous transmission is enabled (SACSR:TSYNE="1"), these bits will activate transmission.
1101	40.4.3.8	The following sentences should be added (before ■ SCSCRn (n=0 to 11)) in "4.3.8. Serial Chip Select Control Status Register: SCSCR". (Correct) Setting to serial chip select pins without assignment to external pins is disabled. For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below. Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB) Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD) Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF) Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)
1103	40.4.3.8	The following sentences should be added in the notes in "4.3.8.Serial Chip Select Control Status Register: SCSCR". (Correct) · If this register is used when all of the following conditions are satisfied, set this bit to "0". · The master mode (SCR:MS=0) is used. · The chip select is used. · The SPI mode (SCR:SPI=1) is used. · "01_H" is set in TBYTE register. · A value other than "00" is set to WT1 and WT0 bits of ESCR register.

Page	Section	Change Results
1104	40.4.3.9	The following description should be added (before ■ SCSTR1n-0n (n=0 to 11)) in the description in "4.3.9. Serial Chip Select Timing Register: SCSTR3-0". (Correct) Setting to serial chip select pins without assignment to external pins is disabled. For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below. Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB) Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD) Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF) Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)
1104	40.4.3.9	The following note should be added in "4.3.9. Serial Chip Select Timing Register: SCSTR3-0". (Correct) Note: · In the master mode (SCR:MS=0) and the normal mode (SCR:SPI=0), set the setup delay time (CSSU7-0) or the hold delay time (CSHD7-0) so that either of the following conditions is satisfied. Baud rate/2[ns] < hold delay [ns] + 3 × bus clock [ns] Hold delay + setup delay < baud rate - 2 × bus clock [ns]
1108	40.4.3.10	The following sentences should be added (before ■ SCSFR1n-0n (n=0 to 11)) in the description in "4.3.10. Serial Chip Select Format Register: SCSFR2-0". (Correct) Setting to serial chip select pins without assignment to external pins is disabled. For assignment of serial chip select pins (SCS0/SCS1/SCS2/SCS3) and external pins, see the tables below. Table 4-1 Table of Base Addresses (Base_addr) and External Pins (MB91F52xB) Table 4-2 Table of Base Addresses (Base_addr) and External Pins (MB91F52xD) Table 4-3 Table of Base Addresses (Base_addr) and External Pins (MB91F52xF) Table 4-4 Table of Base Addresses (Base_addr) and External Pins (MB91F52xJ) Table 4-5 Table of Base Addresses (Base_addr) and External Pins (MB91F52xK, MB91F52xL)
1118	40.4.3.11	The following description should be added in the notes in "4.3.11. Transfer BYTE register: TBYTE3-0". (Correct) · If this register is used when all of the following conditions are satisfied, set TBYTE register to a value other than "01_H". · The master mode (SCR:MS=0) is used. · The chip select is used. · The SPI mode (SCR:SPI=1) is used. · "1" is set to SCAM bit of SCSCR register. · A value other than "00" is set to WT1 and WT0 bits of ESCR register.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1129</td><td>40.4.4.5</td><td>"4.4.5. Serial Aid Control Status Register: SACSR" should be modified as follows:

(Error)
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>bit</td></tr>
<tr><td>STST</td><td>Reserved</td><td>SFD</td><td>SFDE</td><td>AUTE</td><td colspan="2">Reserved</td><td>TINT</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R,W</td><td>RX,W0</td><td>R(RM1),W</td><td>R/W</td><td>R,W</td><td>RX,W0</td><td>RX,W0</td><td>R(RM1),W</td><td>Attribute</td></tr>
</table>

(Correct)
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>bit</td></tr>
<tr><td>STST</td><td>BST</td><td>SFD</td><td>SFDE</td><td>AUTE</td><td colspan="2">Reserved</td><td>TINT</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R,W</td><td>R,WX</td><td>R(RM1),W</td><td>R/W</td><td>R,W</td><td>RX,W0</td><td>RX,W0</td><td>R(RM1),W</td><td>Attribute</td></tr>
</table>
</td></tr>
<tr><td>1129</td><td>40.4.4.5</td><td>[bit15] STST: Serial test bit in "4.4.5. Serial Aid Control Status Register: SACSR should be modified as follows:

(Error)
This bit is used to enable or disable the serial test mode
When the serial test mode is enabled, SOUT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOUT can be received from SIN without being processed.
When the serial test mode is enabled, the SOUT pin will be fixed to "H", and data input into the SIN pin will be ignored

(Correct)
This bit is used to enable or disable the serial test mode.
When the serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed.
When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input to the SIN pin will be ignored.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1129
to
1140</td><td>40.4.4.5</td><td>[bit14] Reserved bit in "4.4.5. Serial Aid Control Status Register: SACSR" should be modified as follows:

(Error)
[bit14] Reserved bit
Always set this bit to "0".

(Correct)
[bit14] BST: Baud rate setting flag
This bit indicates that automatic baud rate adjustment due to Sync Field reception was executed.
This bit is updated if Sync Field detects the fifth fall of the LIN bus.

<table>
<tr><td rowspan="2">BST</td><td colspan="2">Baud rate setting flag</td></tr>
<tr><td>write</td><td>read</td></tr>
<tr><td>0</td><td rowspan="2">No influence</td><td>Automatic baud rate adjustment disabled</td></tr>
<tr><td>1</td><td>Automatic baud rate adjustment enabled</td></tr>
</table>

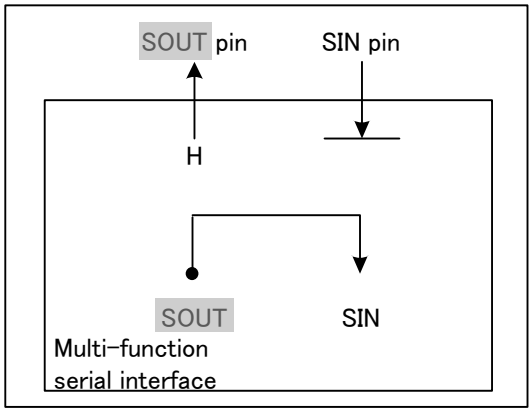
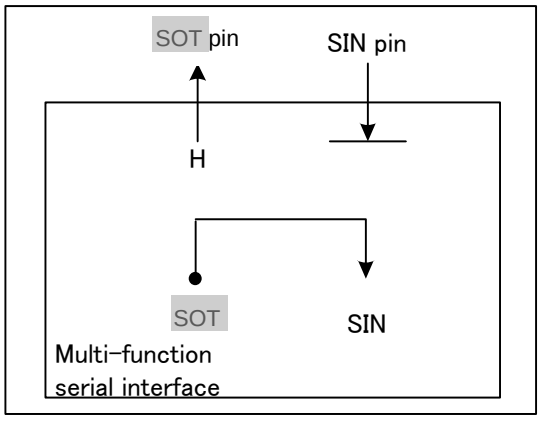
Notes:
· When automatic baud rate adjustment is disabled (AUTE=0), this bit will be fixed to "0".
· When a software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
· This bit is valid only when the sync field detection flag (SACSR:SFD) is "1".
· Writing to this bit has no effect.</td></tr>
<tr><td>1134</td><td>40.4.4.7</td><td>[bit15 to bit0] TC15-0: Compare bits in "4.4.7. Serial Timer Compare Register: STMCR" should be modified as follows:

(Error)
These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, the timer interrupt flag (SACSR:INT) is set to "1".

(Correct)
These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, the timer interrupt flag (SACSR:TINT) is set to "1".</td></tr>
</table>

Page	Section	Change Results
1157	40.4.5.2	The note in [bit2] RDRF in "4.5.2. Serial Status Register: SSR (Serial Status Register)" should be modified as follows: (Error) In the case where all of the conditions below are met, SCL is made "L" after 1-byte data is received and SCL releases the state of "L" when the RDRF bit becomes "0". · Reception FIFO is unused · DMA mode is enabled (IBCR:DMA="1") · RDRF bit is "1" while receiving second or latter byte data (IBSR:TRX="0") · IBCR:WSEL="0" (Correct) In the case where all of the conditions below are met, SCL is made "L" after 1-byte data is received and SCL releases the state of "L" when the RDRF bit becomes "0". · Reception FIFO is unused · DMA mode is enabled (IBCR:DMA="1") · RDRF bit is "1" while receiving second or latter byte data (IBSR:TRX="0") · IBCR:WSEL="1"
1168	40.4.5.7	The description in [bit15 to bit0] TC15 to TC0: Compare bits in "4.5.7. Serial Timer Compare Register: STMCR" should be modified as follows: (Error) These bits are used to set compared values of the serial timer. These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, these bits will set the timer interrupt flag (SACSR:INT) to "1" (Correct) These bits are used to set compared values of the serial timer. These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, these bits will set the timer interrupt flag (SACSR:TINT) to "1".
1185	40.5.2.10	The following description in Figure 5-10 Start by Using Serial Timer Enable Bit in "5.2.10. Operation of Serial Timer" should be modified. (Error) Figure 5-10 Start by Using Serial Timer Enable Bit (STMCR="10", SACSR:TSYNE="0") (Correct) Figure 5-10 Start by Using Serial Timer Enable Bit (STMCR="10")

Page	Section	Change Results
1186	40.5.2.10	The following description in ■ Timer Operation in "5.2.10. Operation of Serial Timer" should be modified. (Error) When the synchronous transmission enable bit (SAGSR:TSYNE) is set to "0", the serial timer operates as a timer. (Correct) The serial timer operates as a timer.
1186	40.5.2.10	The note in ■ Timer Operation in "5.2.10. Operation of Serial Timer" should be modified. (Error) When the timer comparison register (STMCR) is set to $(0000)_H$ with the synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B". (Correct) When the timer comparison register (STMCR) is set to $(0000)_H$, the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".

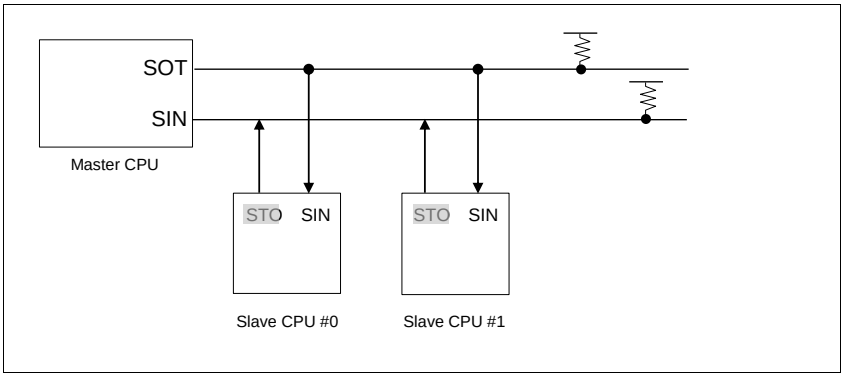
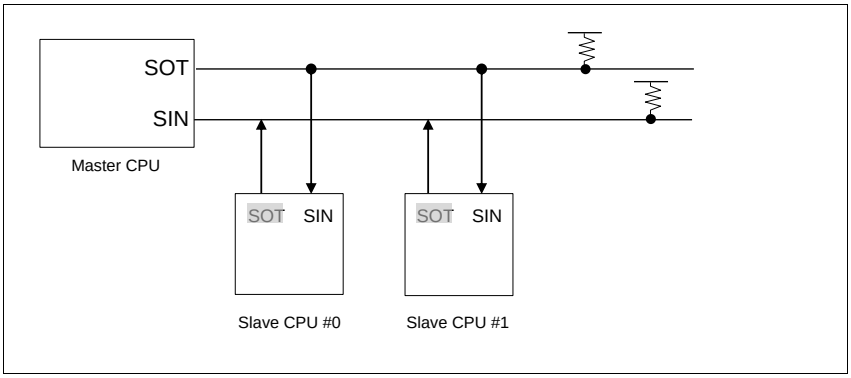
Page	Section	Change Results
1187	40.5.2.11	■ Serial Test Mode in "5.2.11. Test Mode" should be modified as follows: (Error) When the serial test mode is enabled (SACSR:STST="1"), SOUT and SIN are connected inside the multi-function serial interface, and then the data sent from SOUT can be received from SIN directly. When the serial test mode is enabled (SACSR:STST="1"), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored Figure 5-12 Serial Test Mode  (Correct) When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and the data sent from SOT can be received from SIN directly. When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored. Figure 5-12 Serial Test Mode 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1190</td><td>40.5.2.12</td><td>The formula for the allowed maximum transfer rate (FLmax) of ■ Allowed Baud Rate Error Range at Reception in "5.2.12. UART Baud Rate Selection/Setting should be modified as follows:

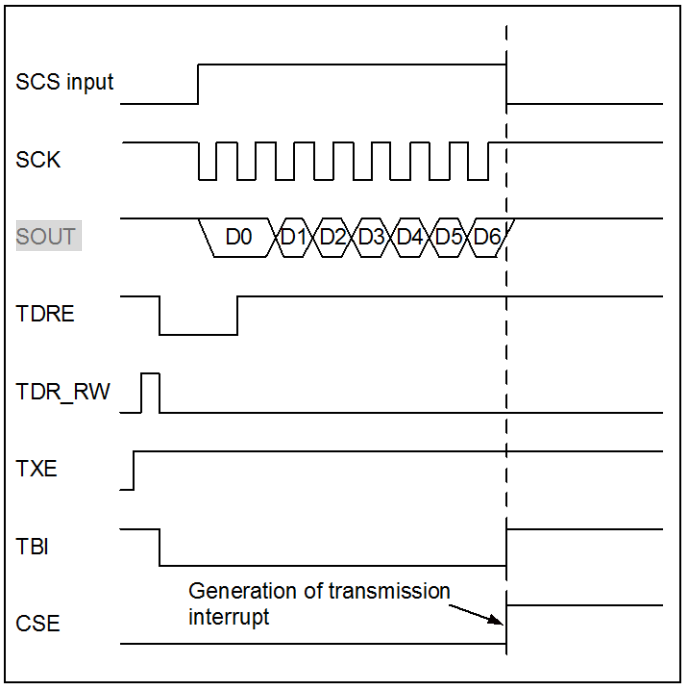
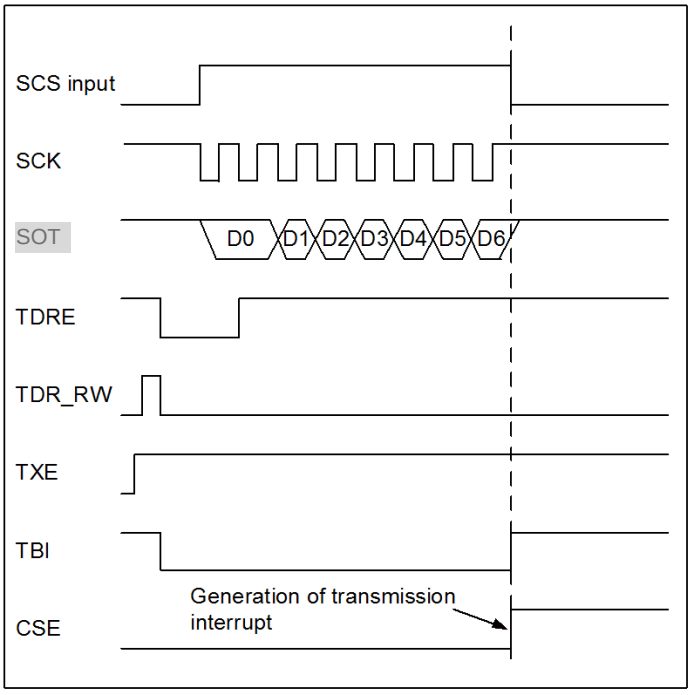
(Error)
10/11 × Flmax = (11bit × (V+1) – (V+1) / 2) / ϕ = (21/20 × 11 × (V+1) / 2) / ϕ (s)
V: Reload value ϕ: Bus clock

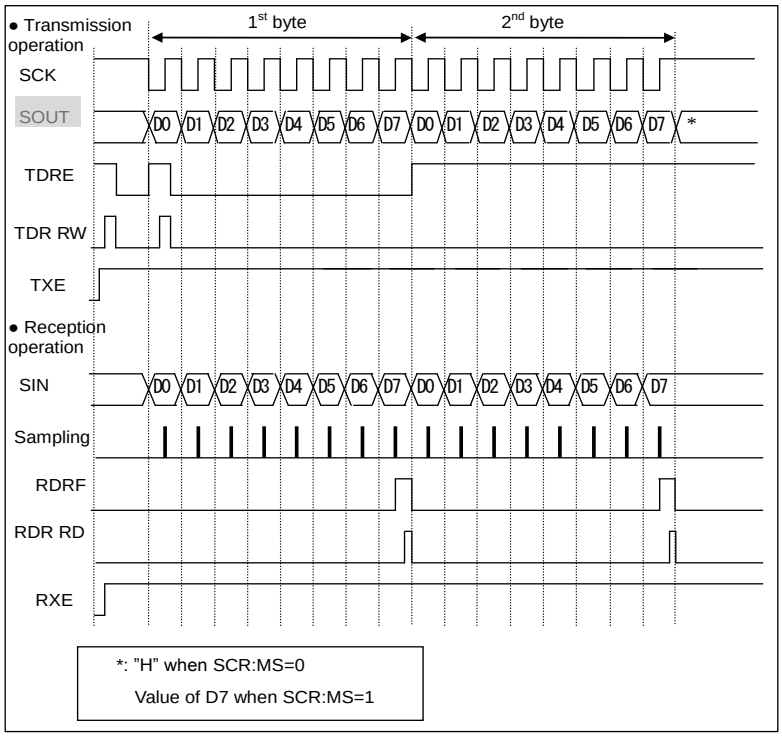
(Correct)
10/11 × Flmax = (11bit × (V + 1) – (V + 1) / 2) / ϕ
Flmax = (21/20 × 11 × (V + 1)) / ϕ (s)
V: Reload value ϕ: Bus clock</td></tr>
<tr><td>1191</td><td>40.5.2.12</td><td>The content of Table 5-3 Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate in ■ Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate in "5.2.12. UART Baud Rate Selection/Setting" should be modified as follows:

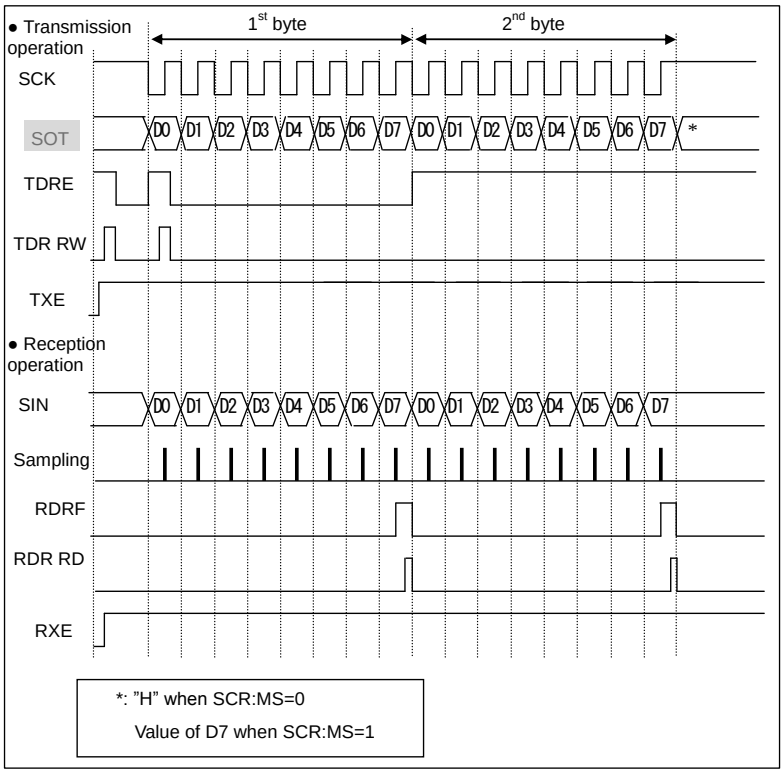
(Error)
<table>
<tr><td rowspan="2">Baud rate (bps)</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">24MHz</td></tr>
<tr><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td></tr>
<tr><td>300</td><td>26666</td><td>26666</td><td><0.01</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
(Correct)
<table>
<tr><td rowspan="2">Baud rate (bps)</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">24MHz</td></tr>
<tr><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td></tr>
<tr><td>300</td><td>26666</td><td><0.01</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

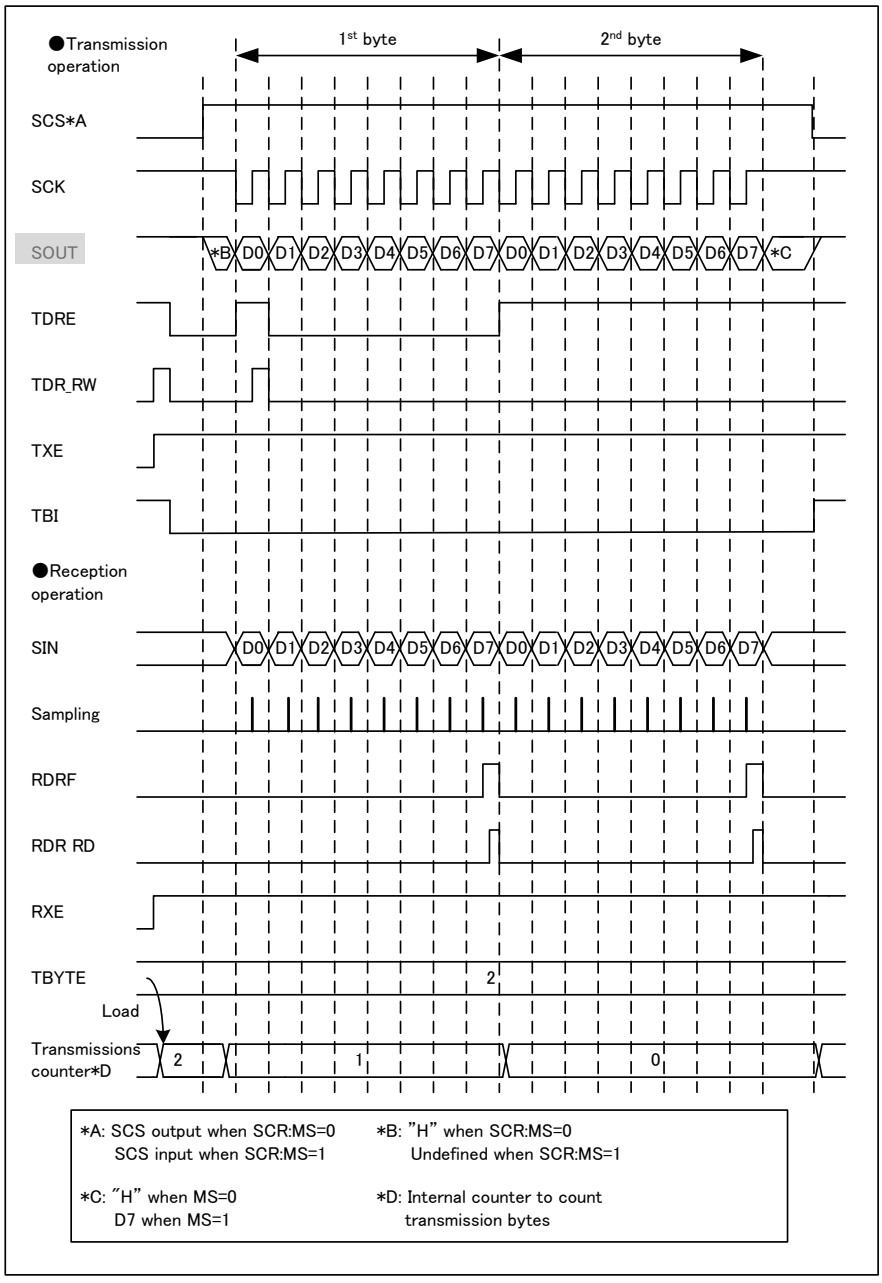
Page	Section	Change Results
1195	40.5.3.2	Figure 5-16 Example of Connection for Master-Slave Communications of UART in "5.3.2. Operation Mode 1 (One-to-N Connection)" should be modified as follows: (Error)  (Correct) 

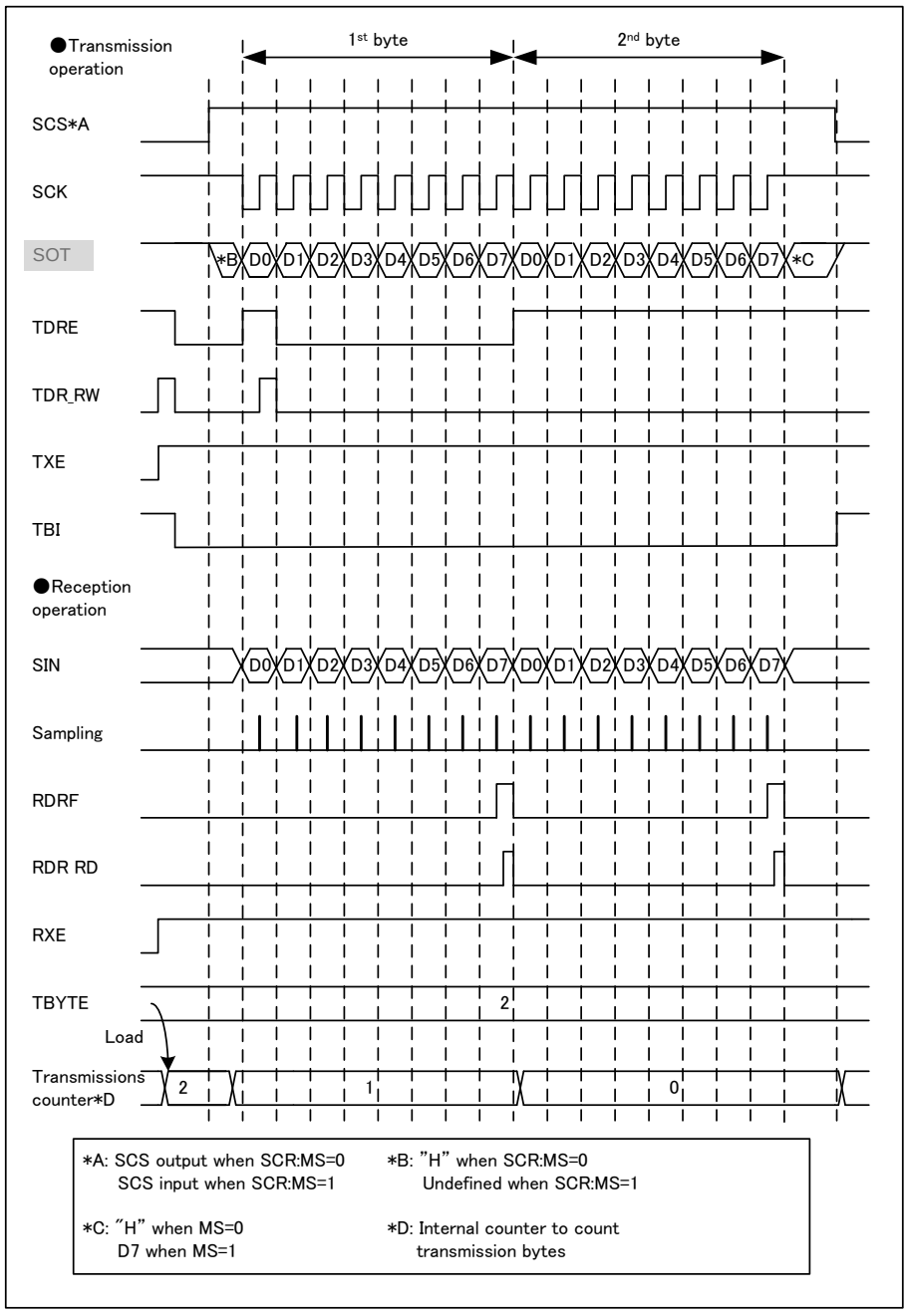
Page	Section	Change Results
1207	40.6.1.7	The notes in "6.1.7. Timing of Chip Select Error Generation and Flag Setting" should be modified as follows: (Error) · While the chip select error flag (SACSR:CSE) is set to "1", no transmission will be started even if transmission data is written to the transmit data register (TDR). · When the chip select error flag (SACSR:CSE) is set to "1" while synchronous transmission is used with the serial timer, no transmission will be started even if serial timer register (STMR) matched the serial timer comparison register. · When the chip select error flag (SACSR:CSE) is set to "1" while synchronous transmission is used with the serial timer, no transmission will be started even if serial timer register (STMR) matched the serial timer comparison register (Correct) · While the chip select error flag (SACSR:CSE) is set to "1", no transmission will be started even if transmission data is written to the transmit data register (TDR). · When the chip select error flag (SACSR:CSE) is set to "1" while synchronous transmission is used with the serial timer, no transmission will be started even if serial timer register (STMR) matched the serial timer comparison register.

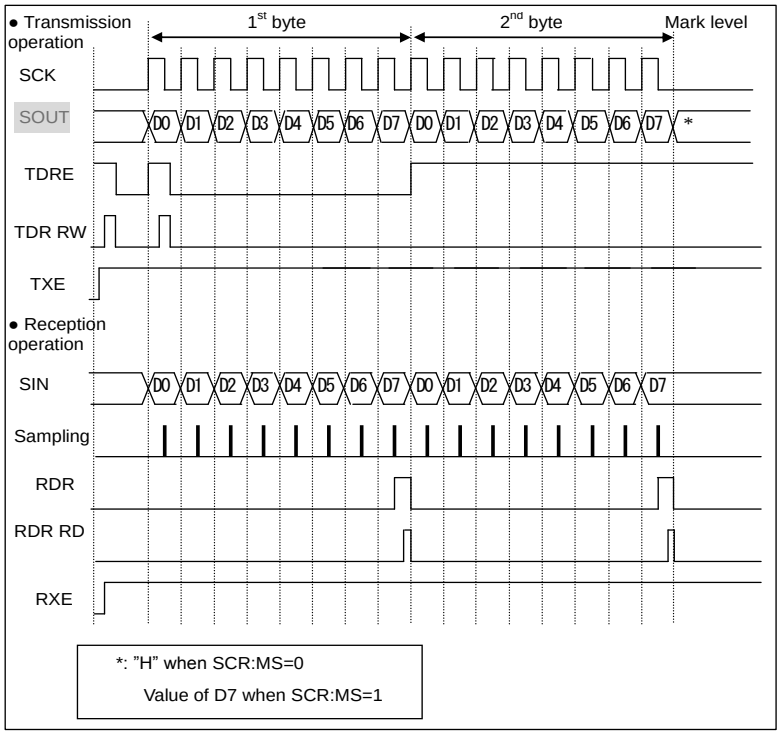
Page	Section	Change Results
1208	40.6.1.7	Figure 6-7 Chip Select Error Generation Timing in ● Slave Mode (SCR:MS="1") in "6.1.7. Timing of Chip Select Error Generation and Flag Setting" should be modified as follows: (Error)  (Correct) 

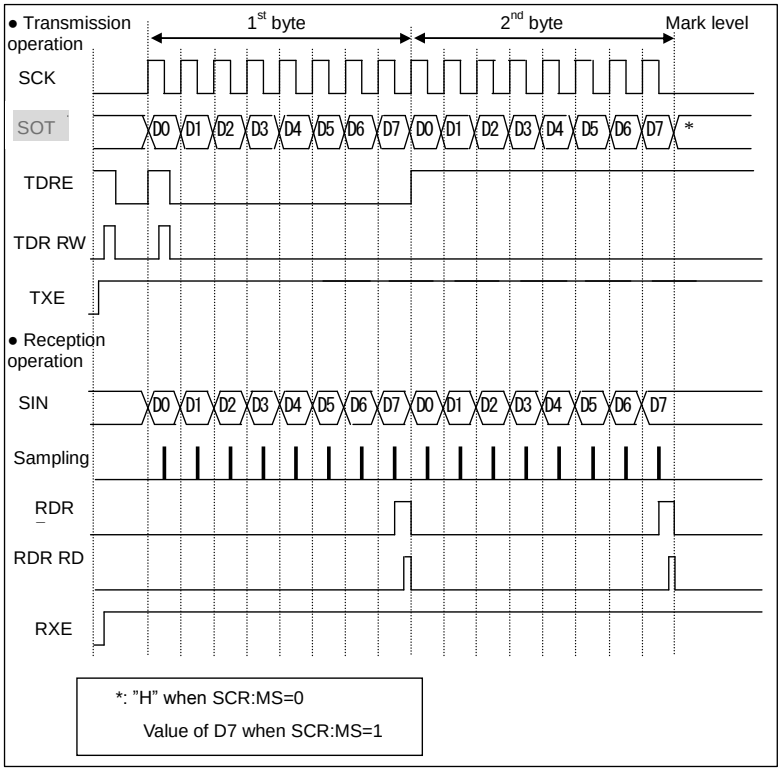
Page	Section	Change Results
1209	40.6.2.1	Figure 6-8 Normal Transfer (I) Timing Chart (Serial Chip Select Pin Unused) in "6.2.1. Normal Transfer (I)" should be modified as follows: (Error) 

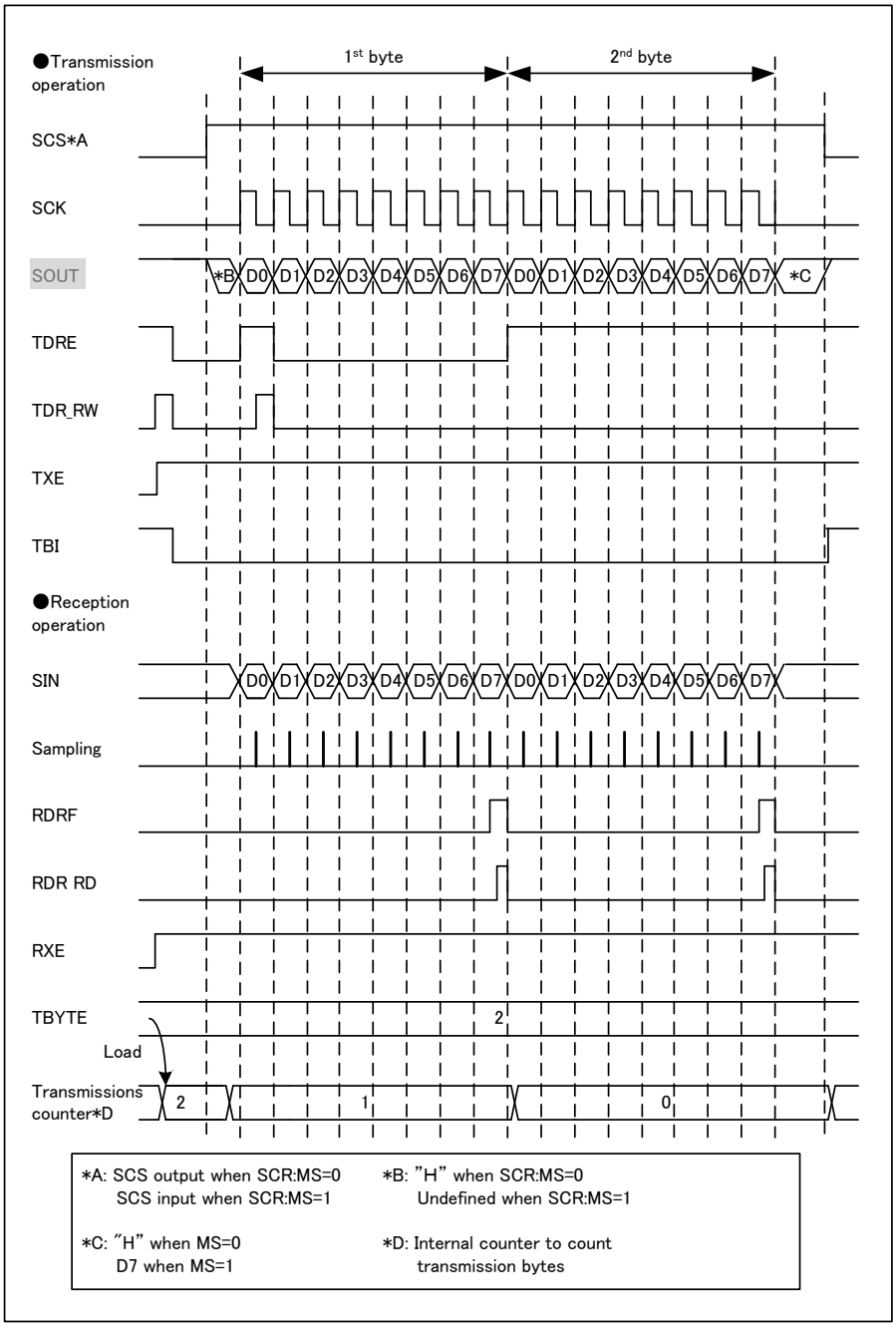
Page	Section	Change Results
		(Correct) 

Page	Section	Change Results
1213	40.6.2.1	Figure 6-9 Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used) in "6.2.1. Normal Transfer (I)" should be modified as follows: (Error) 

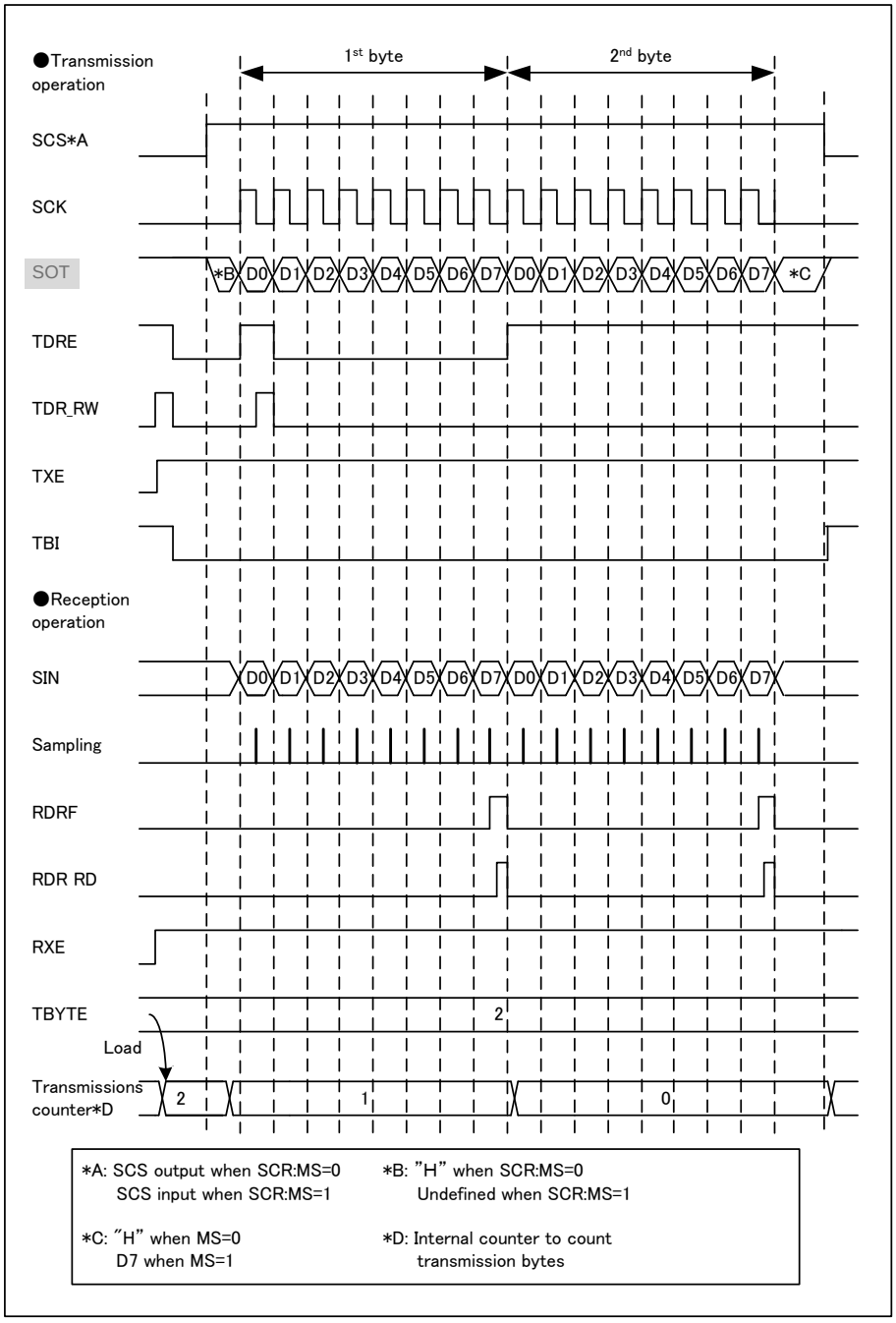
Page	Section	Change Results
1213	40.6.2.1	(Continued) (Correct) 

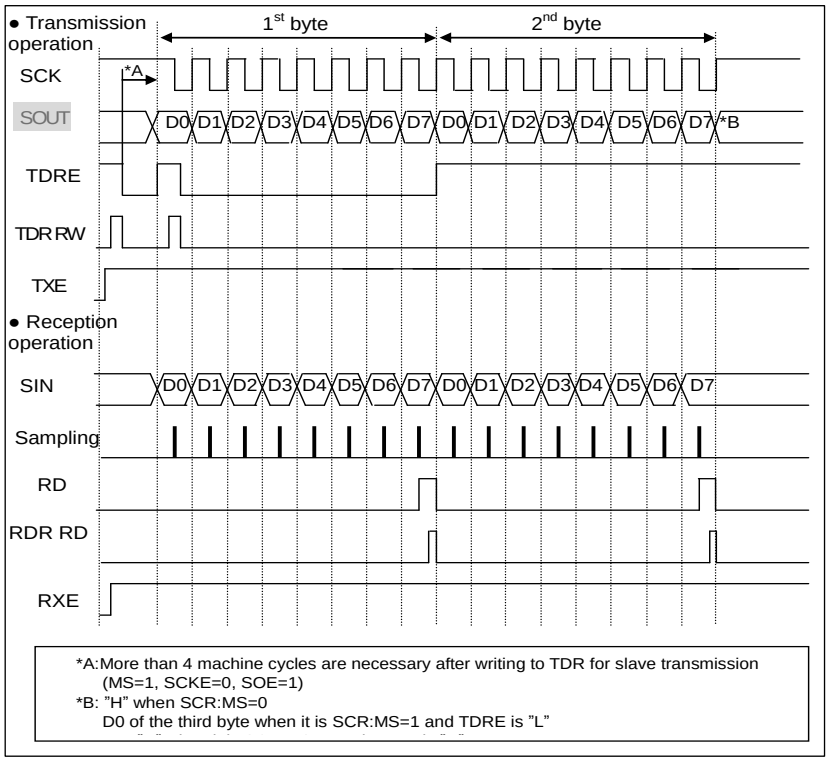
Page	Section	Change Results
1216	40.6.2.1	The description in (4) in ● Transmission/Reception operation in "6.2.1. Normal Transfer (I)" should be modified as follows: (Error) (4) Transmission/reception operation will be started when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOUT) becomes "H". (Correct) (4) Transmission/reception operation will be terminated when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOT) becomes "H".
1218	40.6.2.2	Figure 6-10 Normal Transfer (II) Timing Chart (Serial Chip Select Pin Unused) in "6.2.2. Normal Transfer (II)" should be modified as follows: (Error) 

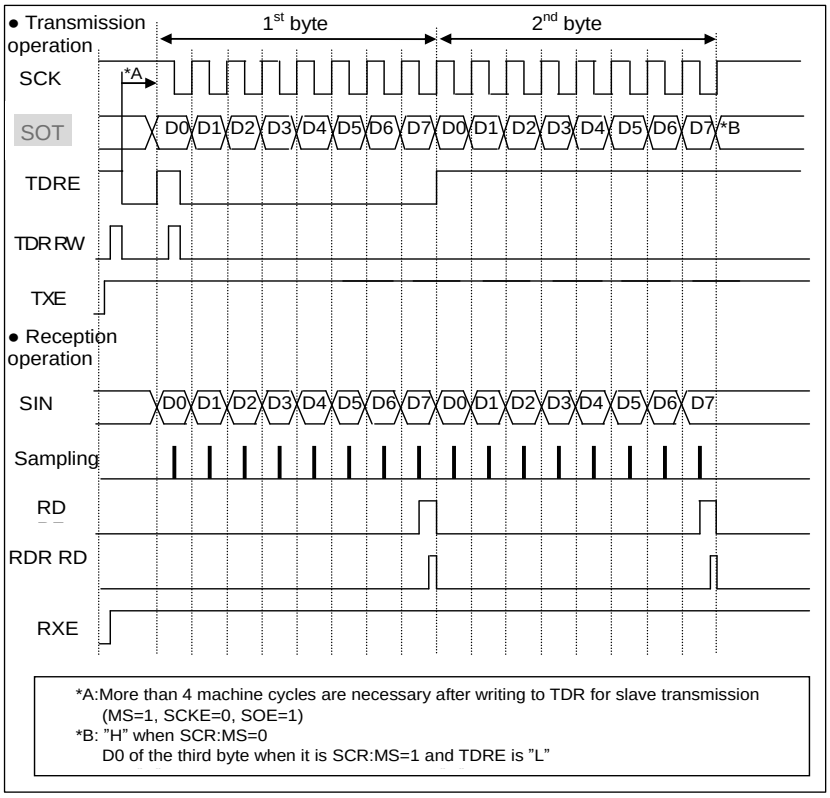
Page	Section	Change Results
		(Correct) 

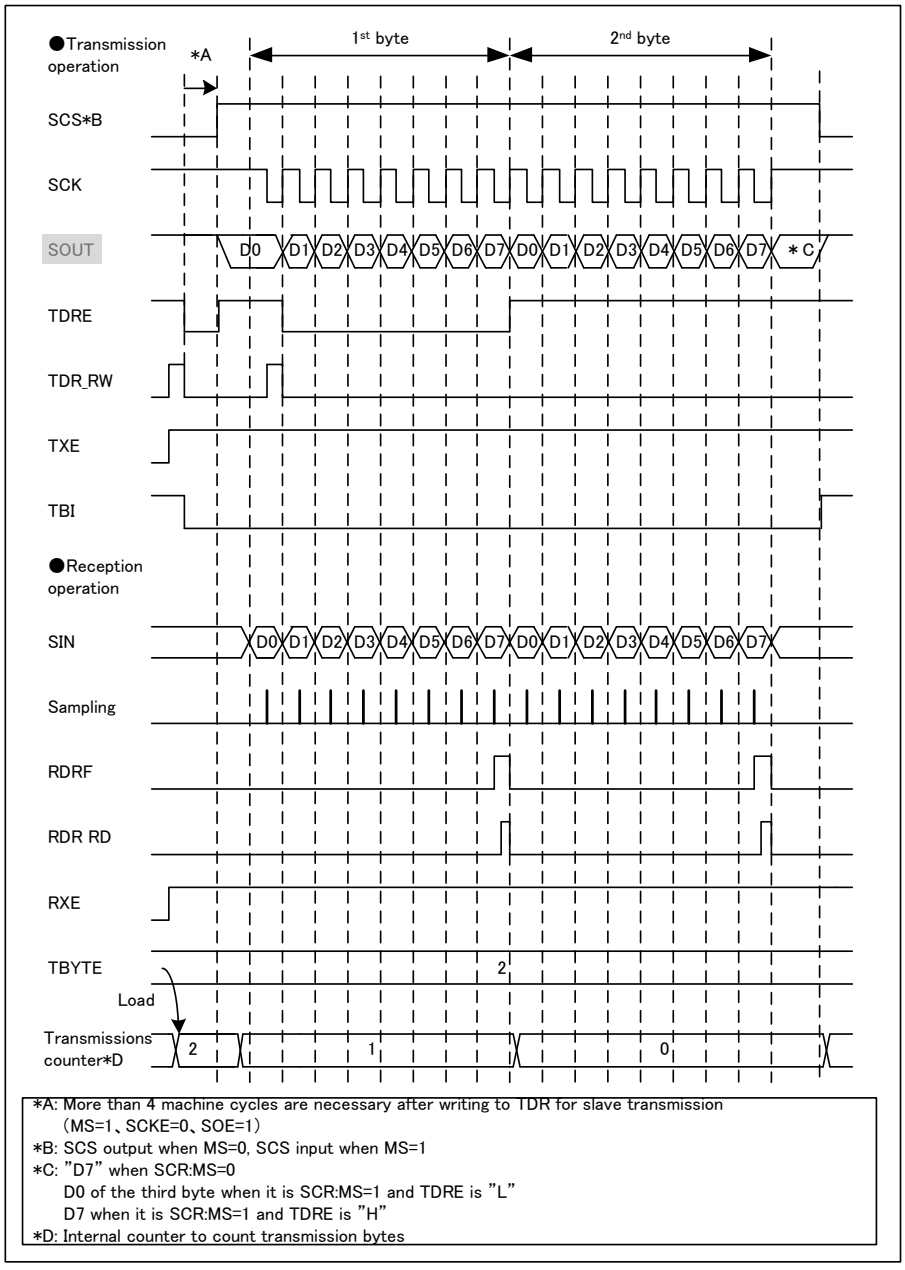
Page	Section	Change Results
1222	40.6.2.2	Figure 6-11 Normal Transfer (II) Timing Chart (Serial Chip Select Pin Used) in "6.2.2. Normal Transfer (II)" should be modified. (Error) 

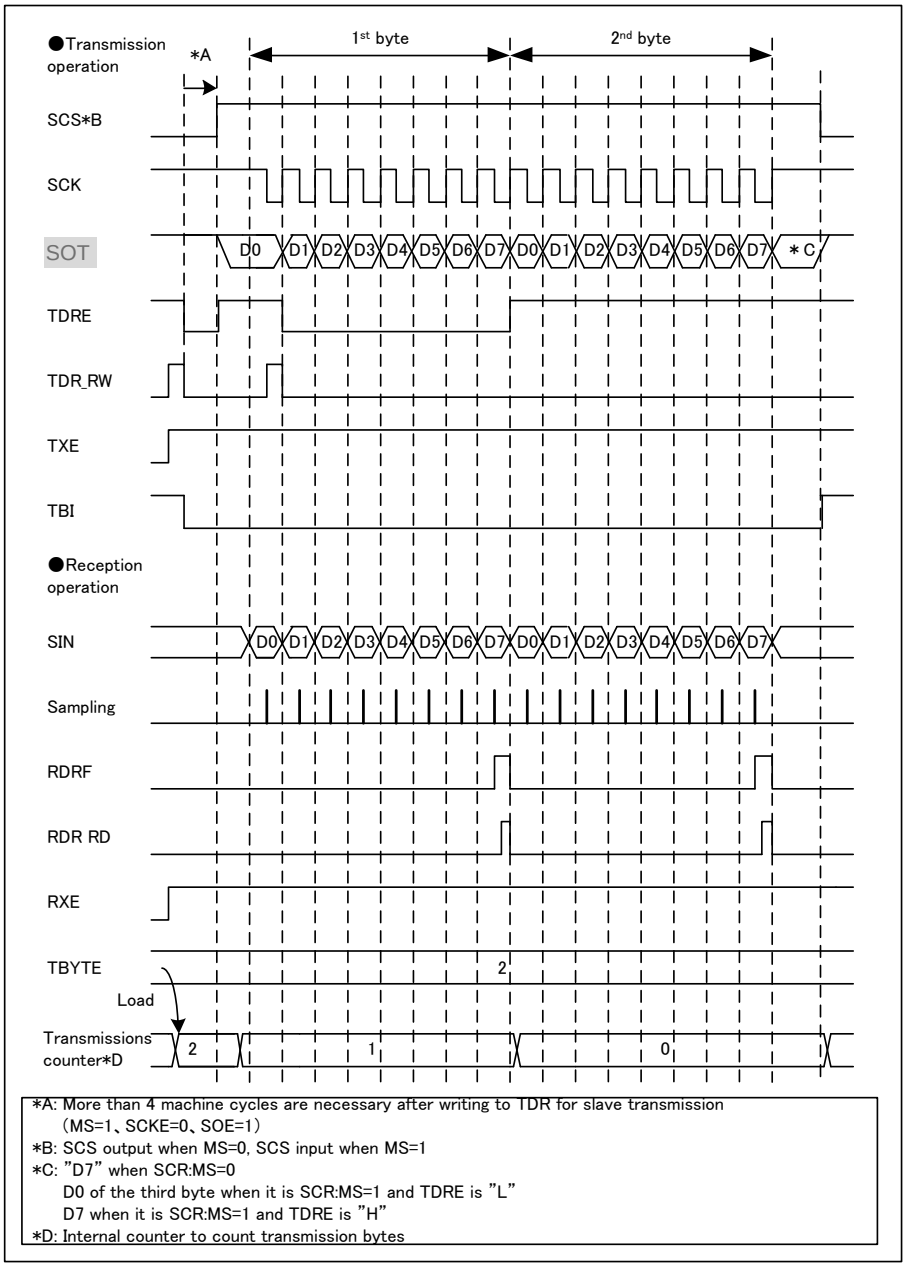
Page	Section	Change Results
1222	40.6.2.2	(Continued) (Correct)

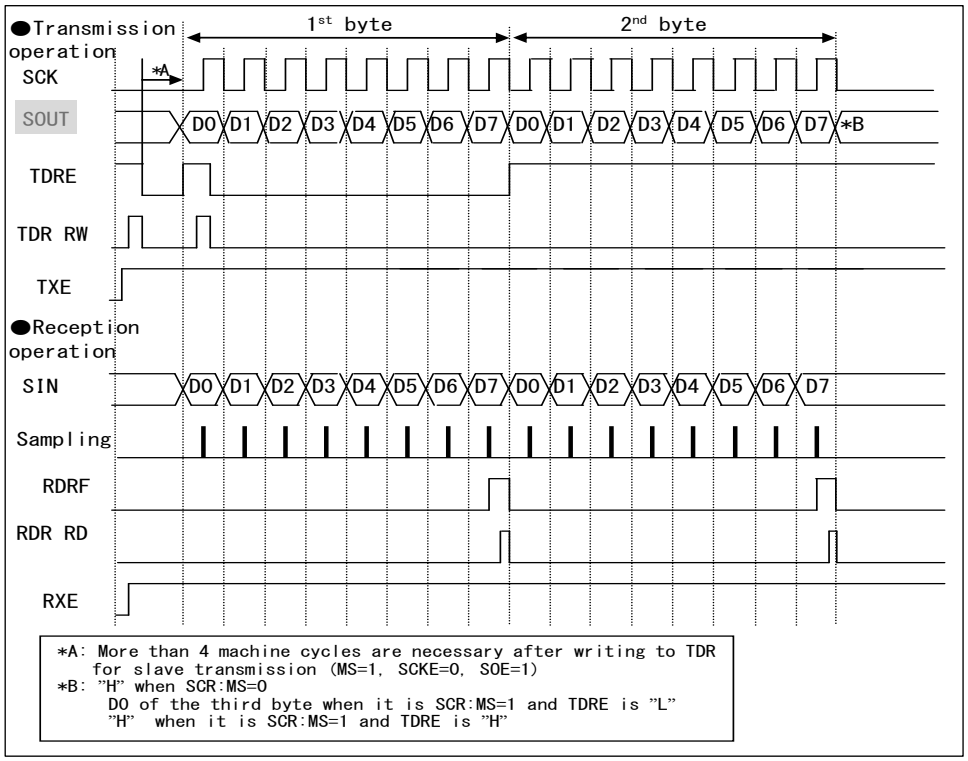


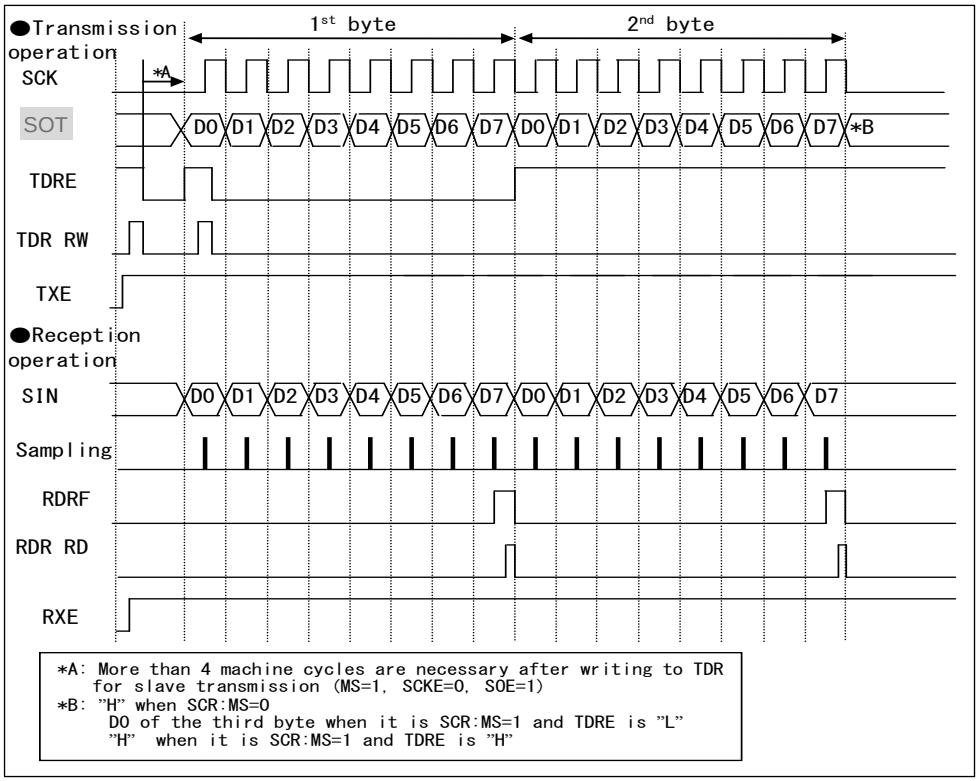
Page	Section	Change Results
1227	40.6.2.3	Figure 6-12 SPI Transfer (I) Timing Chart (Serial Chip Select Pin Unused) in "6.2.3. SPI Transfer (I)" should be modified as follows: (Error) 

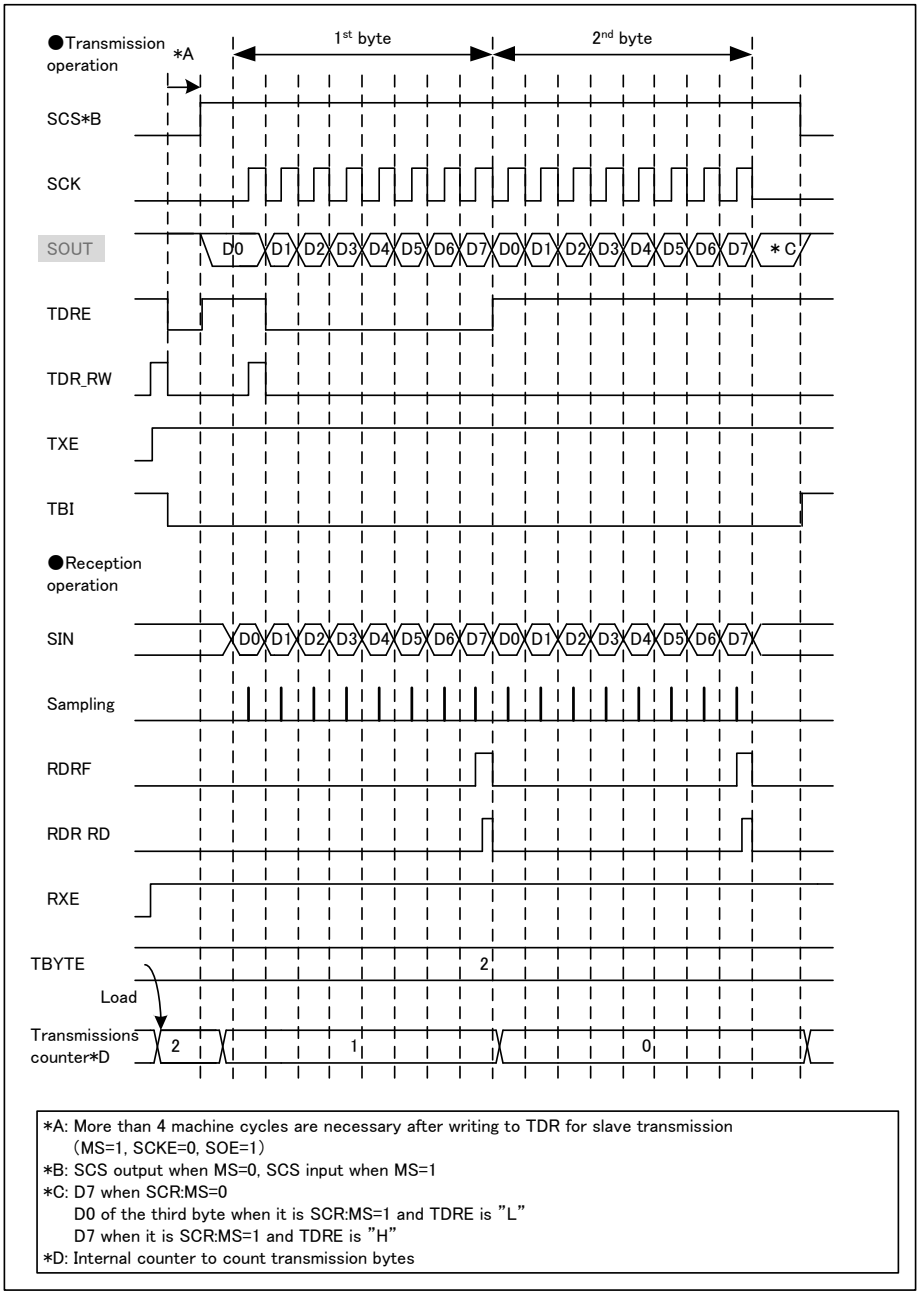
Page	Section	Change Results
		(Correct) 

Page	Section	Change Results
1231	40.6.2.3	Figure 6-13 SPI Transfer (I) Timing Chart (Serial Chip Select Pin Used) in "6.2.3. SPI Transfer (I)" should be modified as follows: (Error) 

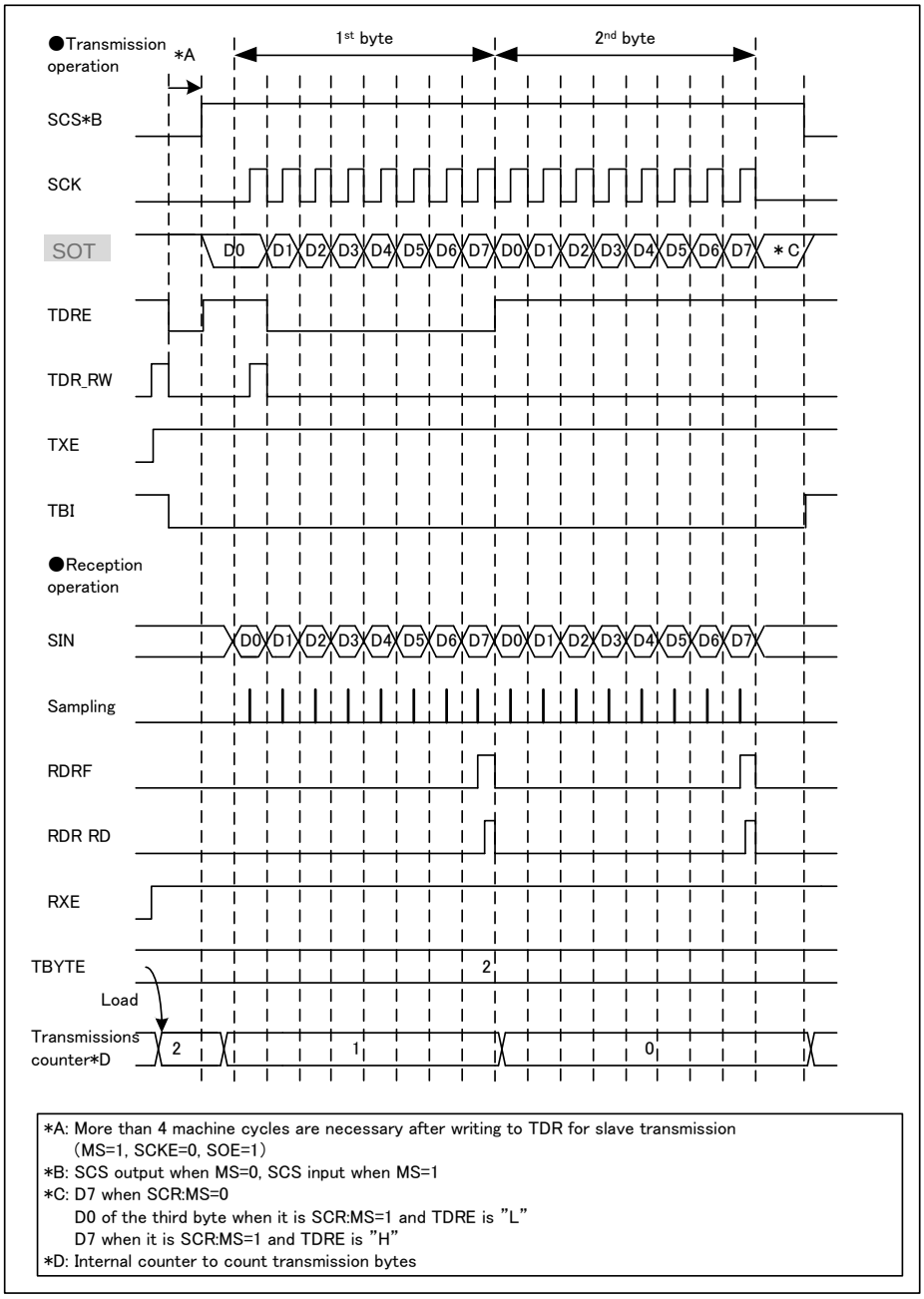
Page	Section	Change Results
1231	40.6.2.3	(Continued) (Correct) 

Page	Section	Change Results
1237	40.6.2.4	Figure 6-14 SPI Transfer (II) Timing Chart (Serial Chip Select Pin Unused) in "6.2.4. SPI Transfer (II)" should be modified as follows: (Error) 

Page	Section	Change Results
		(Correct) 

Page	Section	Change Results
1242	40.6.2.4	Figure 6-15 SPI Transfer (II) Timing Chart (When Serial Chip Select Pin Used) in "6.2.4. SPI Transfer (II)" should be modified as follows: (Error) 

Page	Section	Change Results
1242	40.6.2.4	(Continued) (Correct)



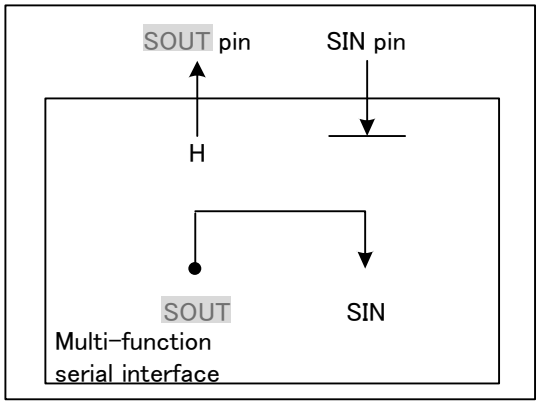
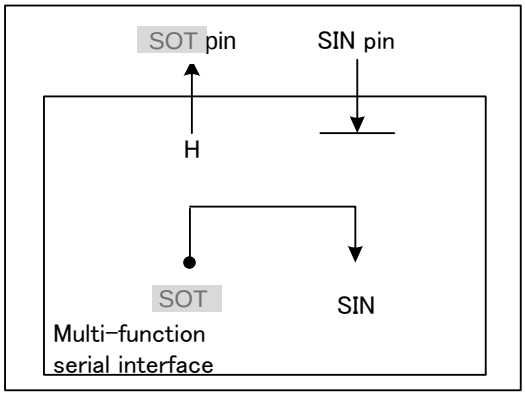
Page	Section	Change Results
1257	40.6.2.6	The following sentence should be deleted from the description in ● Operation of Serial Chip Select to Maintain Active in "6.2.6. Operation of Serial Chip Select". (Error) · If the serial chip select active maintaining bit is "1" and the serial timer synchronous transmission and the external trigger transmission are not used, the serial chip select pin is maintained to be active. (Correct) · If the serial chip select active maintaining bit is "1" upon serial timer synchronous transmission, the serial chip select pin is maintained to be active.
1257	40.6.2.6	The note in ● Operation of Serial Chip Select to Maintain Active in "6.2.6. Operation of Serial Chip Select" should be modified as follows: (Error) when the transfer byte error is enabled (SSR:TBEEN=1) (Correct) when the transfer byte error is enabled (SACSR:TBEEN=1)

Page	Section	Change Results
1257	40.6.2.6	Table 6-2 Format Setting of Serial Chip Select Pin in ● Format Setting of Serial Chip Select Pin in "6.2.6. Operation of Serial Chip Select" should be modified as follows: (Error)

Condition		Active level of chip select	Serial clock inversion	SPI setting	Data direction	Data length
Chip select format enabled (SCR:CSFE="1") and master mode (SCR:MS="0")	Serial chip select pin 0 output	SCSCR0: SCLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
	Serial chip select pin 1 output	SCSFR0: CS1SCLVL	SCSFR0: CS1SCLINV	SCSFR0: CS1SPI	SCSFR0 CS1BDS	SCSFR0: CS1L3-0
	Serial chip select pin 2 output	SCSFR1: CS2SCLVL	SCSFR1: CS2SCLINV	SCSFR1: CS2SPI	SCSFR1 CS2BDS	SCSFR1: CS2L3-0
	Serial chip select pin 3 output	SCSFR2: CS3SCLVL	SCSFR2: CS3SCLINV	SCSFR2: CS3SPI	SCSFR2 CS3BDS	SCSFR2: CS3L3-0
Chip select format disabled (SCR:CSFE="0")		SCSCR0: SCLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Slave mode (MS="1")						
Chip select unused (CSEN3 to CSEN0="0000"$_B$)						

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1258</td><td>40.6.2.6</td><td>(Continued)

(Correct)
<table>
<tr><th colspan="2">Condition</th><th>Active level of chip select</th><th>Serial clock inversion</th><th>SPI setting</th><th>Data direction</th><th>Data length</th></tr>
<tr><td rowspan="4">Chip select format enabled (SCR:CSFE="1") and master mode (SCR:MS="0")</td><td>Serial chip select pin 0 output</td><td>SCSCR0: CSLVL</td><td>SMR: SCINV</td><td>SCR:SPI</td><td>SMR:BDS</td><td>ESCR:L3-0</td></tr>
<tr><td>Serial chip select pin 1 output</td><td>SCSFR0: CS1CSLVL</td><td>SCSFR0: CS1SCLINV</td><td>SCSFR0: CS1SPI</td><td>SCSFR0 CS1BDS</td><td>SCSFR0: CS1L3-0</td></tr>
<tr><td>Serial chip select pin 2 output</td><td>SCSFR1: CS2CSLVL</td><td>SCSFR1: CS2SCLINV</td><td>SCSFR1: CS2SPI</td><td>SCSFR1 CS2BDS</td><td>SCSFR1: CS2L3-0</td></tr>
<tr><td>Serial chip select pin 3 output</td><td>SCSFR2: CS3CSLVL</td><td>SCSFR2: CS3SCLINV</td><td>SCSFR2: CS3SPI</td><td>SCSFR2 CS3BDS</td><td>SCSFR2: CS3L3-0</td></tr>
<tr><td colspan="2">Chip select format disabled (SCR:CSFE="0")</td><td rowspan="3">SCSCR0: SCLVL</td><td rowspan="3">SMR: SCINV</td><td rowspan="3">SCR:SPI</td><td rowspan="3">SMR:BDS</td><td rowspan="3">ESCR:L3-0</td></tr>
<tr><td colspan="2">Slave mode (MS="1")</td></tr>
<tr><td colspan="2">Chip select unused (CSEN3 to CSEN0="0000"$_B$)</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1259	40.6.2.7	● Serial Test Mode in "6.2.7. Test Mode" should be modified as follows: (Error) When the serial test mode is enabled (SACSR:STST="1"), SOUT and SIN are connected inside the multi-function serial interface, and then the data sent from SOUT can be received from SIN directly. When the serial test mode is enabled (SACSR:STST="1"), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored. Figure 6-28 Serial Test Mode  (Correct) When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and the data sent from SOT can be received from SIN directly. When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored. Figure 6-28 Serial Test Mode 

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1261</td><td>40.6.2.8</td><td>Table 6-3 Reload values and Baud Rate for Bus Clock Frequencies in ■ Reload values and Baud rate for Bus Clock Frequencies in "6.2.8. Baud Rate Generation" should be modified as follows:

(Error)
<table>
<tr><td rowspan="2">Baud Rate (bps)</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">24MHz</td></tr>
<tr><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td></tr>
<tr><td>300</td><td>26666</td><td>26666</td><td><0.01</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

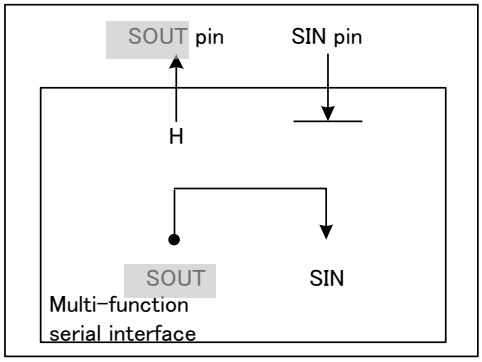
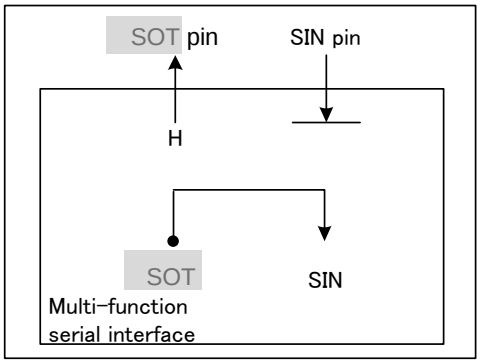
(Correct)
<table>
<tr><td rowspan="2">Baud Rate (bps)</td><td colspan="2">8MHz</td><td colspan="2">10MHz</td><td colspan="2">16MHz</td><td colspan="2">20MHz</td><td colspan="2">24MHz</td></tr>
<tr><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td><td>Value</td><td>ERR</td></tr>
<tr><td>300</td><td>26666</td><td><0.01</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
<tr><td>1284</td><td>40.7.3</td><td>The following description in Figure 7-16 Start by Using Serial Timer Enable Bit in "7.3. Operation of Serial Timer" should be modified.

(Error)
Figure 7-16 Start by Using Serial Timer Enable Bit (STMCR="10", SACSR:TSYNE="0")

(Correct)
Figure 7-16 Start by Using Serial Timer Enable Bit (STMCR="10")</td></tr>
<tr><td>1287</td><td>40.7.4.1</td><td>The following description in ■ Serial Test Mode in "7.4.1. Manual Mode" should be modified.

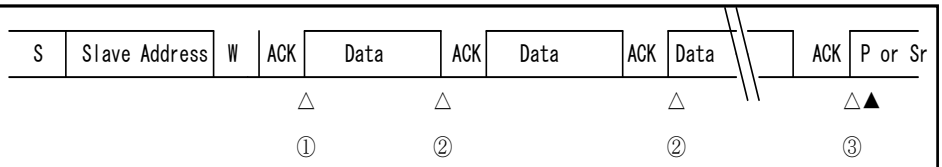
(Error)
When the serial test mode is enabled (SACSR:STST="1"), SOUT and SIN are connected inside the multi-function serial interface, and then the data sent from SOUT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOUT pin is fixed to "H", and the data input to the SIN pin is ignored.

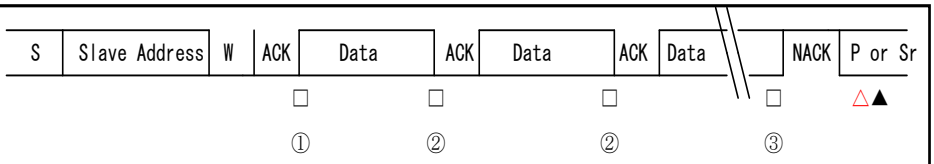
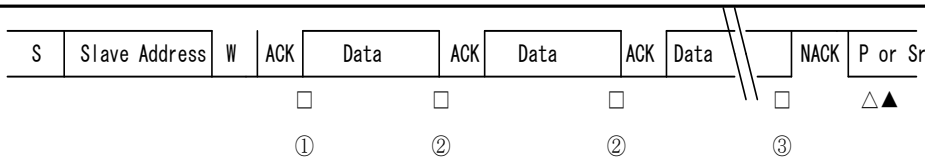
(Correct)
When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and the data sent from SOT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.</td></tr>
</table>

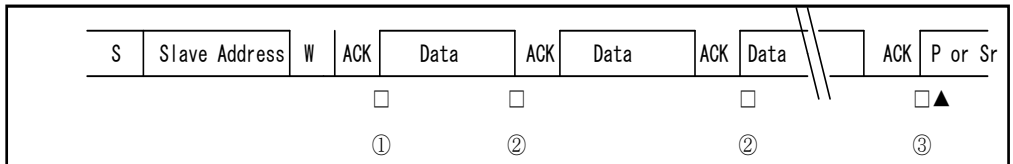
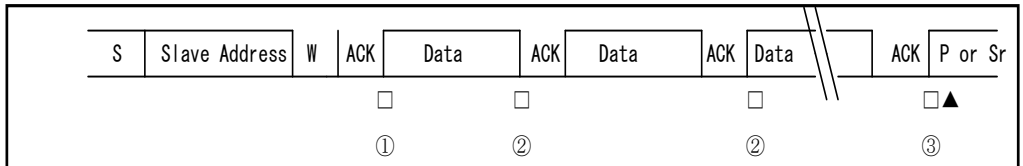
Page	Section	Change Results
1287	40.7.4.1	Figure 7-20 Serial Test Mode in "7.4.1. Manual Mode" should be modified as follows: (Error)  (Correct) 
1295	40.7.5.1	The following notes in "7.5.1 Manual mode" should be modified. (Error) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. It is recommended to leave a space between frames if successive frames are to be output. (Correct) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.

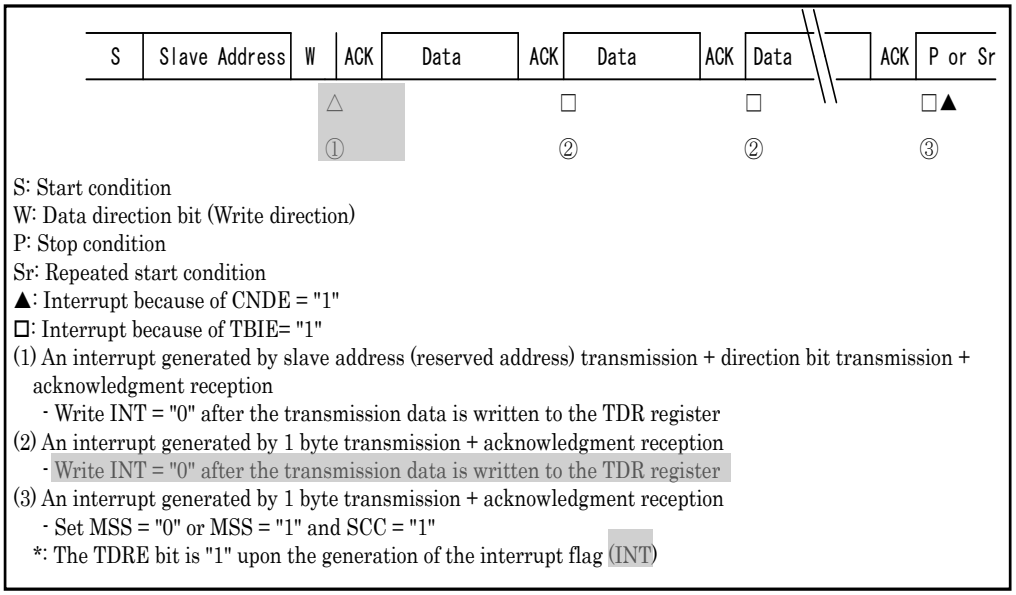
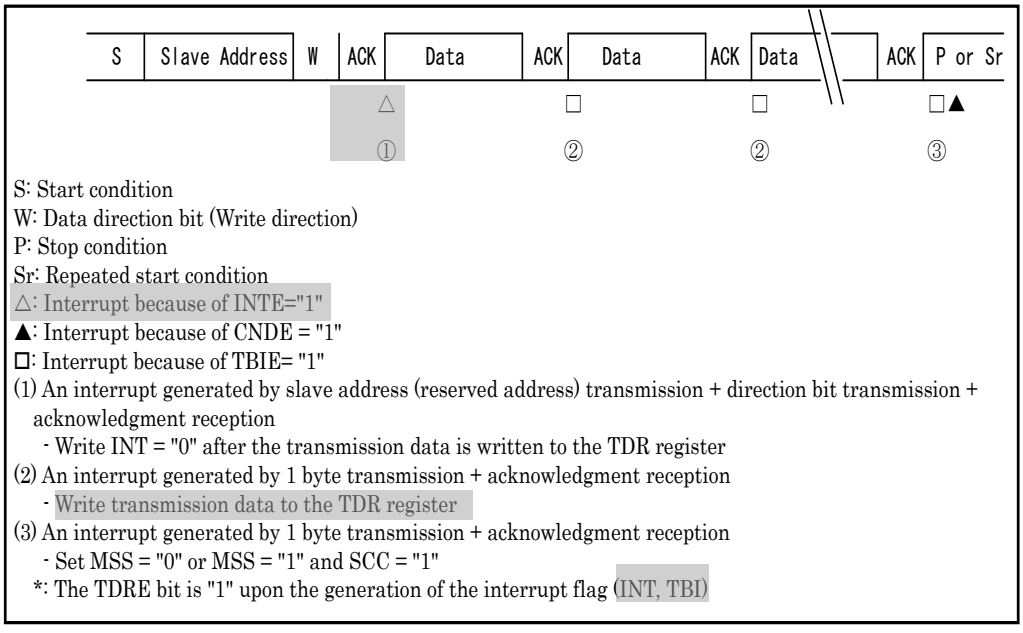
Page	Section	Change Results
1302	40.7.5.1	The following notes in "7.5.1 Manual mode" should be modified. (Error) ·If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive data normally. It is recommended to leave a space between frames if successive frames are to be output. (Correct) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.
1311	40.7.5.2	The notes in "7.5.2. Assist Mode" should be modified as follows: (Error) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. If successive frames are to be output, leaving a space between frames is recommended. (Correct) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.
1312	40.7.5.2	The notes in "7.5.2. Assist Mode" should be modified as follows: (Error) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. If successive frames are to be output, leaving a space between frames is recommended. (Correct) · If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. Leave a space between frames if successive frames are to be output.

Page	Section	Change Results
1357	40.8.3.4	The description in (6) in [1] Transmission to a destination that is not at the reserved address in ■ Data transmission to slave when DMA mode is enabled (SSR:DMA=1) in "8.3.4. Data Transmission by Master" should be modified as follows: (Error) (6)Put the I^2C bus in a wait by setting the transmission bus idle 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 (6) to (7) 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. (Correct) (6) Put the I^2C bus in a wait by setting the transmission bus idle 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 (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.

Page	Section	Change Results
1363	40.8.3.4	Figure 8-18 Master Transmission Interrupt (5)-when FIFO is Disabled in "8.3.4. Data Transmission by Master" should be modified as follows: (Error)  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 transmission buffer (2) An interrupt generated by 1 byte transmission - Write WSEL="0", INT = "0" after the transmission data is written to the transmission buffer (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) (Correct)  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 resister (2) An interrupt generated by 1 byte transmission - Write WSEL="0", INT = "0" after the transmission data is written to the TDR resister (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)

Page	Section	Change Results
1368	40.8.3.4	Figure 8-25 Master Transmission Interrupt (12)-when FIFO is Disabled in "8.3.4. Data Transmission by Master" should be modified as follows: (Error)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition □: Interrupt because of TBIE= "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) (Correct)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition △: Interrupt because of INTE="1" □: Interrupt because of TBIE= "1" ▲: Interrupt because of CNDE = "1" (1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception - Write transmission data to the TDR register (2) An interrupt generated by 1 byte transmission - Write transmission data 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)

Page	Section	Change Results
1370	40.8.3.4	Figure 8-27 Master Transmission Interrupt (14)-when FIFO is Disabled in "8.3.4. Data Transmission by Master" should be modified as follows: (Error)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition ▲: Interrupt because of CNDE = "1" □: Interrupt because of TBIE= "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 transmission buffer (2) An interrupt generated by 1 byte transmission - Write WSEL="0", INT = "0" after the transmission data is written to the transmission buffer (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) (Correct)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition ▲: Interrupt because of CNDE = "1" □: Interrupt because of TBIE= "1" (1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception - Write transmission data to the TDR register (2) An interrupt generated by 1 byte transmission - Write transmission data to the TDR register after WSEL="0" is written (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 (TBIE)

Page	Section	Change Results
1371	40.8.3.4	Figure 8-28 Master Transmission Interrupt (15)-when FIFO is Disabled in "8.3.4. Data Transmission by Master" should be modified as follows: (Error)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition ▲: Interrupt because of CNDE = "1" □: Interrupt because of TBIE= "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) (Correct)  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" □: Interrupt because of TBIE= "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 transmission data 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, TBI)

Page	Section	Change Results
1379	40.8.3.5	The title of a figure in "8.3.5. Data Reception by Master" should be modified as follows: (Error) Figure 8-36 Master Reception Interrupt (1)-when FIFO is Disabled (Correct) Figure 8-36 Master Reception Interrupt (5)-when FIFO is Disabled
1380	40.8.3.5	The title of a figure in "8.3.5. Data Reception by Master" should be modified as follows: (Error) Figure 8-37 Master Reception Interrupt (2)-when FIFO is Disabled (Correct) Figure 8-37 Master Reception Interrupt (6)-when FIFO is Disabled
1380	40.8.3.5	The title of a figure in "8.3.5. Data Reception by Master" should be modified as follows: (Error) Figure 8-38 Master Reception Interrupt (3)-when FIFO is Enabled (Correct) Figure 8-38 Master Reception Interrupt (7)-when FIFO is Enabled
1381	40.8.3.5	The title of a figure in "8.3.5. Data Reception by Master" should be modified as follows: (Error) Figure 8-39 Master Reception Interrupt (4)-when FIFO is Enabled (Correct) Figure 8-39 Master Reception Interrupt (8)-when FIFO is Enabled
1388	40.8.4.3	The title of Figure 8-46 Slave Reception Interrupt (6)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1") in ■ When DMA mode is enabled (SSR:DMA=1) in "8.4.3. Slave Mode Reception" should be modified as follows: (Error) Figure 8-46 Slave Reception Interrupt (6)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1") (Correct) Figure 8-46 Slave Reception Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1400</td><td>41</td><td>"6. Restrictions" should be added in the chapter list of "CHAPTER:CAN".

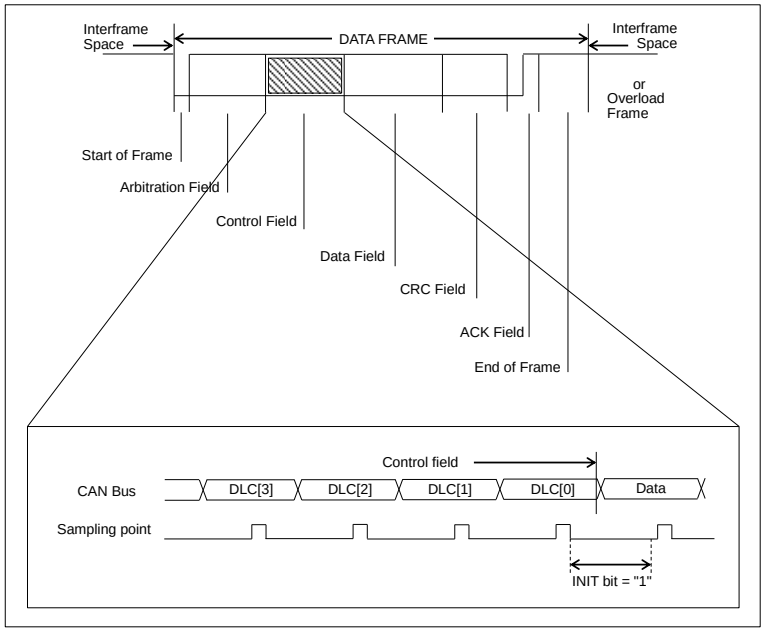
(Error)
1. Overview
2. Features
3. Configuration
4. Registers
5. Operation

(Correct)
1. Overview
2. Features
3. Configuration
4. Registers
5. Operation
6. Restrictions</td></tr>
<tr><td>1404</td><td>41.4.1.1</td><td>The table in "4.1.1. List of Base-addresses (Base-addr), External Pins and Buffer Size" should be modified as follows:

(Error)
<table>
<tr><th rowspan="2">Channel number</th><th rowspan="2">Base-addr</th><th colspan="2">External pin name</th><th rowspan="2">Buffer size</th></tr>
<tr><th>CANTX</th><th>CANRX</th></tr>
<tr><td>0</td><td>0x2000</td><td>TX0</td><td>RX0</td><td>128 msgb</td></tr>
<tr><td>1</td><td>0x2100</td><td>TX1</td><td>RX1</td><td>64 msgb</td></tr>
<tr><td>2</td><td>0x2200</td><td>TX2</td><td>RX2</td><td>64 msgb</td></tr>
</table>
(Correct)
<table>
<tr><th rowspan="3">Channel</th><th rowspan="3">Base-addr</th><th colspan="4">External pin name</th><th rowspan="3">Buffer size</th></tr>
<tr><th colspan="2">MB91F52xB,
MB91F52xD</th><th colspan="2">MB91F52xF,
MB91F52xJ,
MB91F52xK,
MB91F52xL</th></tr>
<tr><th>CANTX</th><th>CANRX</th><th>CANTX</th><th>CANRX</th></tr>
<tr><td>0</td><td>0x2000</td><td>TX0_1</td><td>RX0(128)</td><td>TX0(128)/
TX0_1</td><td>RX0(128)</td><td>128 msgb</td></tr>
<tr><td>1</td><td>0x2100</td><td>TX1(64)</td><td>RX1_1</td><td>TX1(64)</td><td>RX1(64)/
RX1_1</td><td>64 msgb</td></tr>
<tr><td>2</td><td>0x2200</td><td>TX2(64)</td><td>RX2(64)</td><td>TX2(64)</td><td>RX2(64)</td><td>64 msgb</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1414	41.4.2.1	The notes in "4.2.1. CAN Control Register: CTRLR" should be modified as follows: (Error) Notes: · 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, set "1" to the Init bit and initialize the CAN controller (Correct) Notes: · When "1" is set to the Init bit during transmission/reception, the transmission/reception is stopped immediately. · When the Init bit is set to "1" during transmission, set the Init bit to "1" after the transmission is completed. When the Init bit was set to "1" while the transmission was in progress, execute a transmission cancellation for the message buffer with "1" specified in the transmission request bit (TxRqst) (set the TxRqst bit to "0") and then set the Init bit to "0". After that, set the transmission request (TxRqst="1") after progress more than two bits time of CAN. · It is necessary to initialize the CAN controller by writing "1" to the Init bit before transiting to the low-power consumption mode (stop mode, clock mode) and before changing the supply clock.
1420	41.4.2.5	The word "byte" should be deleted from "4.2.5. CAN Interrupt Register: INTR" as follows: (Error) ■ CAN Interrupt Register (upper byte): Address Base + 08_H (Access: Byte, Half-word, Word) ■ CAN Interrupt Register (lower byte): Address Base + 09H (Access: Byte, Half-word, Word) (Correct) ■ CAN Interrupt Register (upper byte): Address Base + 08_H (Access: Half-word, Word) ■ CAN Interrupt Register (lower byte): Address Base + 09H (Access: Half-word, Word)
1428	41.4.3.2	The bit name should be added to bit3 in "4.3.2. IFx Command Mask Register (IFxCMSK)". (Error) [bit3] : Interrupt clear bit Operation of CAN controller will not be affected whether "0" or "1" is set. (Correct) [bit3] CIP: Interrupt clear bit Operation of CAN controller will not be affected whether "0" or "1" is set.

Page	Section	Change Results
1437	41.4.4.2	The notes in UMask in "4.4.2. Functions of Message Object" should be modified as follows: (Error) · 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 (Correct) · 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.
1439	41.4.4.2	The notes in EoB: End of Buffer bit in "4.4.2. Functions of Message Object" should be modified as follows: (Error) · The EoB bit is used to configure the FIFO buffer of 2 to 32 messages. (Correct) · The EoB bit is used to configure the FIFO buffer of 2 to 128 messages (ch.0) and 2 to 64 messages (ch.1, ch.2).
1456	41.5.2.2	The description in "5.2.2. Transmission Priority" should be modified as follows: (Error) 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 (Correct) Transmission priority of a message object is determined by its message number. Message object 1 has the highest priority; and message object 128 (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.

Page	Section	Change Results
1473	41.6	The following description should be added at the end of the chapter of "CAN". 6. Restrictions This section explains the restrictions on CAN. 6.1 INIT bit 6.1. INIT bit This section provides information on INIT bit. 6.1.1 Restrictions 6.1.2 Bypass method 6.1.1. Restrictions This section explains the restrictions. If the INIT bit of the CAN control register (CTRLR) is set to "1" (Figure 6-1) during the transmission of the last bit in the control field, the INIT bit is cleared and then the data field of the frame transmitted first is shifted to left by one bit. Note that subsequent messages are transmitted correctly. As for remote frames or data frames with the data length of zero, the INIT bit setting at this timing does not have any influence on them.  6.1.2. Bypass method This section explains the bypass method. Apply either of the following methods to bypass this restriction. 1. When "1" is set in the INIT bit of CAN control register (CTRLR), set the INIT bit of CAN control register (CTRLR) to "1" immediately after the transmission completed. 2. When "1" is set in the INIT bit of CAN control register (CTRLR) during transmission and then "0" is set in the INIT bit to transmit, first, set the INIT bit to "1" and execute the transmission cancel for the message buffer with "1" set in the transmission request bit (TxRqst) (set TxRqst bit to "0"). Then set the INIT bit to "0". Next, after a 2 bit interval of time of CAN has passed, set the transmission request bit (TxRqst) of the transmitted message buffer to "1".

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1476</td><td>42.1.</td><td>The following sentence should be added in "1. Overview".

(Error)
This module generates clocks (fsys) supplied from each clock source to the CAN macro.

(Correct)
This module generates clocks (fsys) supplied from each clock source to the CAN macro. Figure 3-1 indicates CAN and CAN interface, and CAN clock prescaler and clock source selector circuit.</td></tr>
<tr><td>1480</td><td>43.1</td><td>The following sentence should be added in "1. Overview".

(Correct)
In MB91F52xB (64 pin) and MB91F52xD (80 pin), only ch.0 is available.</td></tr>
<tr><td>1481</td><td>43.4</td><td>■ Table of Base Addresses (Base_addr) and External Pins in "4. Registers" should be modified as follows:

(Error)
<table>
<tr><th>Channel number</th><th>Base_addr</th><th>External pin</th></tr>
<tr><td>0</td><td>0x023C</td><td>DAO0</td></tr>
<tr><td>1</td><td>0x023E</td><td>DAO1</td></tr>
</table>

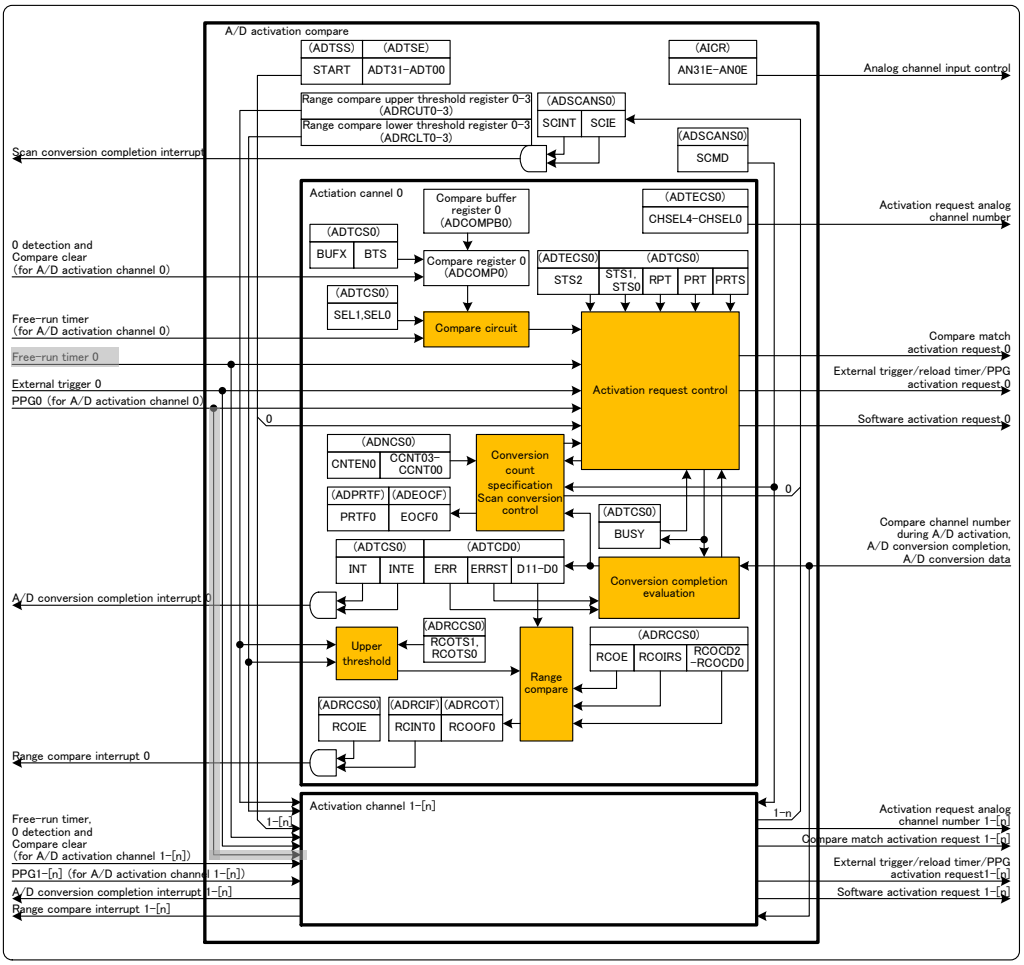
(Correct)
<table>
<tr><th rowspan="2">Channel</th><th rowspan="2">Base_addr</th><th colspan="2">External pin</th></tr>
<tr><th>MB91F52xB, MB91F52xD</th><th>MB91F52xF, MB91F52xJ, MB91F52xK, MB91F52xL</th></tr>
<tr><td>0</td><td>0x023C</td><td>DAO0</td><td>DAO0</td></tr>
<tr><td>1</td><td>0x023E</td><td>none</td><td>DAO1</td></tr>
</table>
</td></tr>
</table>

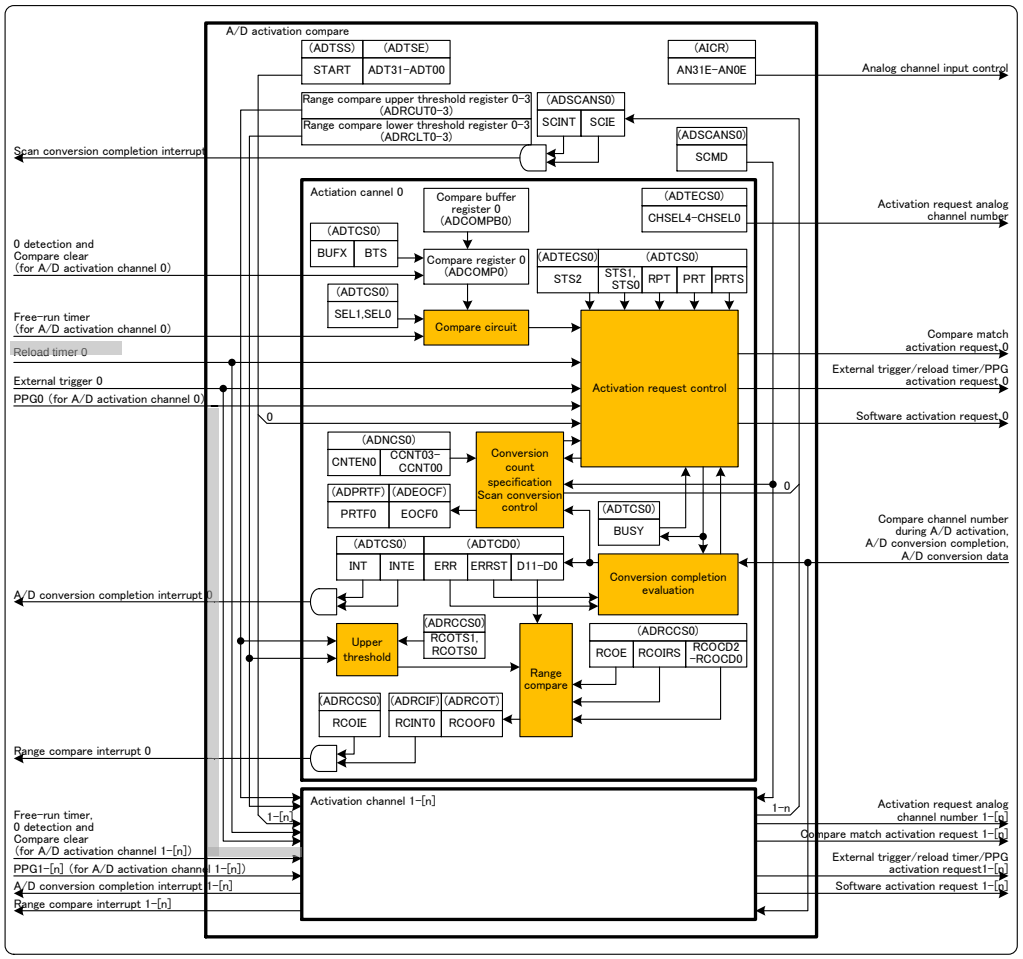
Page	Section	Change Results
1482	43.4.1	The note in "4.1. D/A Control Register: DACR (D/A Control Register)" should be modified as follows: (Error) When the D/A converter has been selected as a port function, if "0" is written in this bit, the port becomes a general-purpose port function. When the D/A converter has been selected as a port function and changed to the stop mode or the watch mode, the D/A converter stops the operation and the port becomes a general-purpose port function (Correct) When the D/A converter has been selected as a port function, if "0" is set in this bit, the D/A converter as a port function becomes disabled, regardless of the PFR/EPFR setting. When the D/A converter enters the stop mode or the watch mode, the pin function of the D/A converter becomes forcibly disabled. As written in "5.2. EPFR Setting Priority" in "I/O Ports", the D/A converter output function has the first priority. Therefore, note that if any functions other than the D/A converter, such as A/D, PPG, OCU, and SG, are enabled when the D/A converter function becomes disabled, such functions become enabled. If any resource functions other than the D/A converter are not selected in PFR/EPFR/ADCH, the port becomes a general-purpose port function. So, PDR/DDR can select the pin status.

Page	Section	Change Results
1484	43.5	The content of "5. Operation" should be modified as follows: (Error) 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 If "0" is written to the DAE bit of the D/A control register (DACR), 0.0V is output from the D/A converter. Moreover, 0.0V is output from the D/A converter even when the CPU is in the stop mode If "1" is written to the DAE bit of the D/A control register (DACR) and output from the D/A converter is enabled, the output voltage can be set from 0.0V to 255/256×AVCC (AVCC:AVCC pin voltage). Therefore, if AVCC pin voltage is adjusted, the output voltage range can be changed. For output of the conversion result of the D/A converter to the external pin, a pin must be set to D/A output by the DAE bit in the D/A control register (DACR), and PFR register and EPFR register. (Correct) 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. If the DAE bit of the D/A control register (DACR) is set to "0" when the D/A converter is selected as a port function, the D/A converter as a port function becomes disabled, regardless of PFR/EPFR setting. When the D/A converter enters the stop mode or the watch mode, the pin function of the D/A converter becomes forcibly disabled. As written in "5.2 EPFR Setting Priority" in "I/O Ports", the D/A converter output function has the first priority. Therefore, note that if any functions other than the D/A converter, such as A/D, PPG, OCU, and SG, are enabled when the D/A converter function becomes disabled, such functions become enabled. When any resource functions other than the D/A converter are not selected in PFR/EPFR/ADCH, the port becomes a general-purpose port function. So, PDR/DDR can select the pin status. For output of the conversion result of the D/A converter to the external pin, a pin must be set to D/A output by the DAE bit in the D/A control register (DACR), and PFR register and EPFR register.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1486</td><td>44.2.1</td><td>Ch.12 and ch.19 should be added as follows in Unit 0 of MB91F52xD in Table 2-1 Channel Allocation of Each Unit in ■ Activation channel in "2.1. Function of A/D Activation Compare".

(Error)
<table>
<tr><th></th><th>Unit0</th><th>Unit1</th></tr>
<tr><td>MB91F52xD</td><td>ch.0, ch.1, ch.7, ch.10, ch.11, ch.14 to ch.17, ch.22, ch.26, ch.27, ch.28, ch.31</td><td>ch.32 to ch.47</td></tr>
</table>
(Correct)
<table>
<tr><th></th><th>Unit 0</th><th>Unit 1</th></tr>
<tr><td>MB91F52xD</td><td>ch.0, ch.1, ch.7, ch.10, ch.11, ch.12, ch.14 to ch.17, ch.19, ch.22, ch.26, ch.27, ch.28, ch.31</td><td>ch.32 to ch.47</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
1490	44.3	Figure 3-1 Configuration of A/D Activation Compare (n=31 A/D converter unit 0) in "3. Configuration" should be modified as follows: (Error) 

Page	Section	Change Results
1490	44.3	(Continued) (Correct) 

Page	Section	Change Results
1502	44.4.1.1	A note should be added in ■ ADERH0 and ■ ADERL0 in "4.1.1. Analog Input Enable Register : ADER". (Error) Note: This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled. (Correct) Note: · This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled. · For pins with the A/D converter function, see Chapter-11 "5.6. Notes on Pins with the A/D pins converter Function".
1503	44.4.1.1	A note should be added in ■ ADERL1 in "4.1.1. Analog Input Enable Register : ADER". (Error) Note: This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled. (Correct) Note: · This register is a key code target register. Key code setting is required for writing. For the setting method, refer to sections "KEY CoDe Register : KEYCDR" and "Key code register function settings" in "CHAPTER: I/O PORTS". In addition, word access to this register is disabled. · For pins with the A/D converter function, see Chapter-11 "5.6. Notes on Pins with the A/D pins converter Function".

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1512</td><td>44.4.2.4</td><td>The following description should be added to the content of [bit4] BTS: Compare register buffer transfer control bit in "4.2.4. A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS47".

(Error)
· If this bit is set to "0", the compare buffer register value will be transferred to the compare register when 0 is detected as the free-run timer value.
· If this bit is set to "1", the compare buffer register value will be transferred to the compare register when it matches the compare clear register of the free-run timer.

(Correct)
· This bit sets the transfer condition when the buffer function of the compare register is enabled (ADTCS.BUFX="0").
· The BTS setting is enabled when the buffer function of the compare register is enabled (ADTCS.BUFX="0").
· If this bit is set to "0", the compare buffer register value will be transferred to the compare register when 0 is detected as the free-run timer value.
· If this bit is set to "1", the compare buffer register value will be transferred to the compare register when it matches the compare clear register of the free-run timer.</td></tr>
<tr><td>1520</td><td>44.4.2.9</td><td>The attribute in "4.2.9 Range Compare Control Status Register: ADRCCS0 to ADRCCS47" should be modified as follows:

(Error)
<table>
<tr><th></th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr>
<tr><td></td><td>RCOCD2</td><td>RCOCD1</td><td>RCOCD0</td><td>RCOIRS</td><td>RCOIE</td><td>RCOE</td><td>RCOTS1</td><td>RCOTS0</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>
(Correct)
<table>
<tr><th></th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr>
<tr><td></td><td>RCOCD2</td><td>RCOCD1</td><td>RCOCD0</td><td>RCOIRS</td><td>RCOIE</td><td>RCOE</td><td>RCOTS1</td><td>RCOTS0</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Attribute</td><td>R,W</td><td>R,W</td><td>R,W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table></td></tr>
<tr><td>1539</td><td>44.4.3.3</td><td>The title of the setting for [bit3, bit2] in "4.3.3. A/D Mode Setting Register" should be modified as follows:

(Error)
Recommended setting

(Correct)
Example of setting</td></tr>
</table>

Page	Section	Change Results
1539 to 1540	44.4.3.3	The title of the setting for [bit1, bit0] in "4.3.3. A/D Mode Setting Register" should be modified as follows: (Error) Recommended setting (Use conditions: AVcc=2.7V to 5.5V) Recommended setting (Use conditions: AVcc=4.5V to 5.5V) (Correct) Example of setting (Use conditions: AVcc=2.7V to 5.5V) Example of setting (Use conditions: AVcc=4.5V to 5.5V)
1563 to 1564	44.5.2.15	The description in "5.2.15. Forced termination of activation request" should be modified as follows: (Error) Whether an A/D activation request or a conversion operation is in progress can be reported using the A/D conversion request bit. The current A/D activation request or conversion operation can be terminated forcibly by resetting the appropriate bit to "0". (Correct) Whether an A/D activation request or a conversion operation is in progress can be reported using the A/D conversion request bit (ADTCS0-47.BUSY) of the A/D activation trigger control status register. The current A/D activation request or conversion operation can be terminated forcibly by resetting this bit to "0".

Page	Section	Change Results
1567	44.5.2.16	The following figure should replace Table 5-11 in ■ Example of operating range comparison in "5.2.16. Range comparison function".

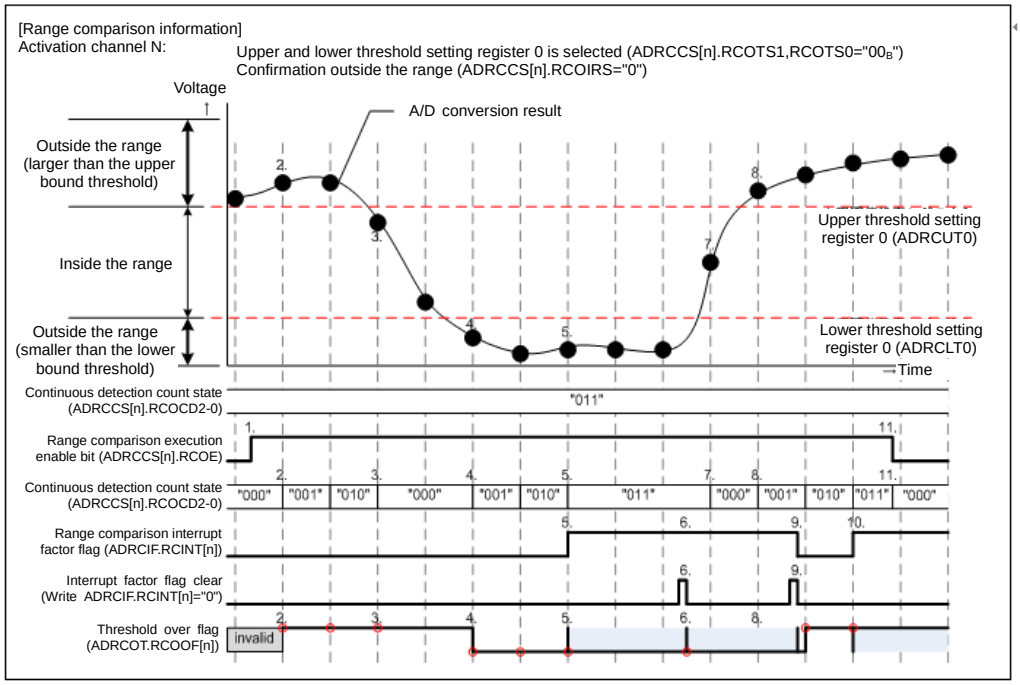
(Error)

Table 5-11 Range Comparison Upper and Lower Bound Threshold Select

Range comparison result	Range comparison threshold over flag bit (ADRCOT:RCOOF)	
	Confirmation outside the range (ADRCCS: RCOIRS="0")	Confirmation inside the range (ADRCCS: RCOIRS="1")
Outside the range (larger than upper bound threshold) A/D data bit > Upper bound threshold setting register	"1"	The last value is maintained
Inside the range A/D data bit ≤ Upper bound threshold setting register and A/D data bit ≥ Lower bound threshold setting register	The last value is maintained	The last value is maintained
Outside the range (Smaller than lower bound threshold) A/D data bit < Lower bound threshold setting register	"0"	The last value is maintained

(Correct)

Figure 5-22 Example of Operating Range Comparison



Page	Section	Change Results
1570	44.5.3.1	The figure number of Figure 5-22 Activation Arbitration in "5.3.1. A/D Activation Trigger Arbitration" should be modified as follows: (Error) Figure 5-22 Activation Arbitration (Correct) Figure 5-23 Activation Arbitration
1571	44.5.4.1	The figure number of Figure 5-23 12-bit A/D Converter Operation Timing in "5.4.1. Operation Timing" should be modified as follows: (Error) Figure 5-23 12-bit A/D Converter Operation Timing (Correct) Figure 5-24 12-bit A/D Converter Operation Timing
1573	44.5.4.7	The table numbers and the title of Table 5-12 and Table 5-13 in "5.4.7. A/D Conversion Time" should be modified as follows: (Error) Table 5-12 Sampling Time to Peripheral Clock Frequency Table 5-13 Sampling Time to Peripheral Clock Frequency (Correct) Table 5-11 Sampling Time to Peripheral Clock Frequency Table 5-12 Compare Time to Peripheral Clock Frequency

Page	Section	Change Results
1578	45.2	The description in "● Error correction code (ECC) security function" in "2. Features" should be modified as follows: (Error) ● 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 noread cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development (Correct) ● 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. · An error is detected when data is read in the chip erase/sector erase mode. Always write "FFFF" and then read the data when the data in the erase mode (FFFF) needs to be read correctly.
1591	45.4.1	The description in [bit9, bit8] FAW[1:0] : FLASH access/wait setting in "4.1. Flash Control Register: FCTLR (Flash ConTroL Register)" should be modified as follows: (Error) The wait cycle to the flash access at CPU mode is set. Because the reading time of the flash memory is 12.5ns, when it accesses the flash memory at over 80MHz, the access without waiting is impossible. It is indispensable to insert wait with these bits. Please set it to FAW=1(1wait) when you access it at over 80MHz. (Correct) The wait cycle to the flash access at CPU mode is set. Because the reading time of the flash memory is 12.5ns, access to the flash memory at over 80MHz is disabled. Please set it to FAW=1(1wait) when you access it at over 80MHz.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1594</td><td>45.4.3</td><td>A note should be added in [bit4] DFWDSBL (Data Fetch Wait cycle DiSaBle): Data fetch wait cycle disable in "4.3. Flash Interface Control Register: FLIFCTLR (Flash I/F Control Register)".

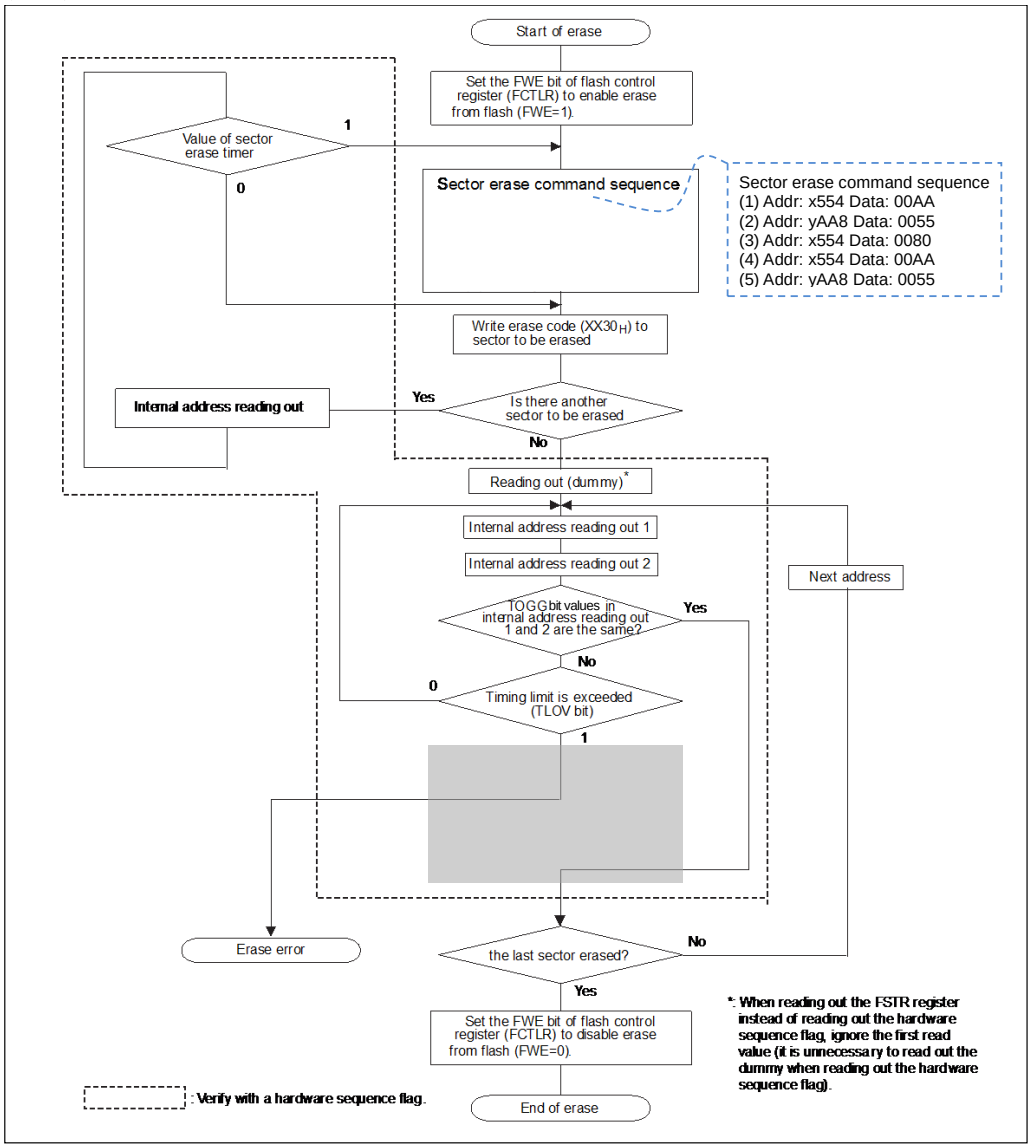
(Correct)
<table><tr><th>DFWDSBL</th><th>Description</th></tr><tr><td>0</td><td>Wait cycle enabled (Initial value)</td></tr><tr><td>1</td><td>Wait cycle disabled</td></tr></table>
Note:
When this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.</td></tr>
<tr><td>1601</td><td>45.5.3.1</td><td>The description in the note in ■ Sector Erase Suspend Command in "5.3.1. Command Sequence" should be modified as follows:

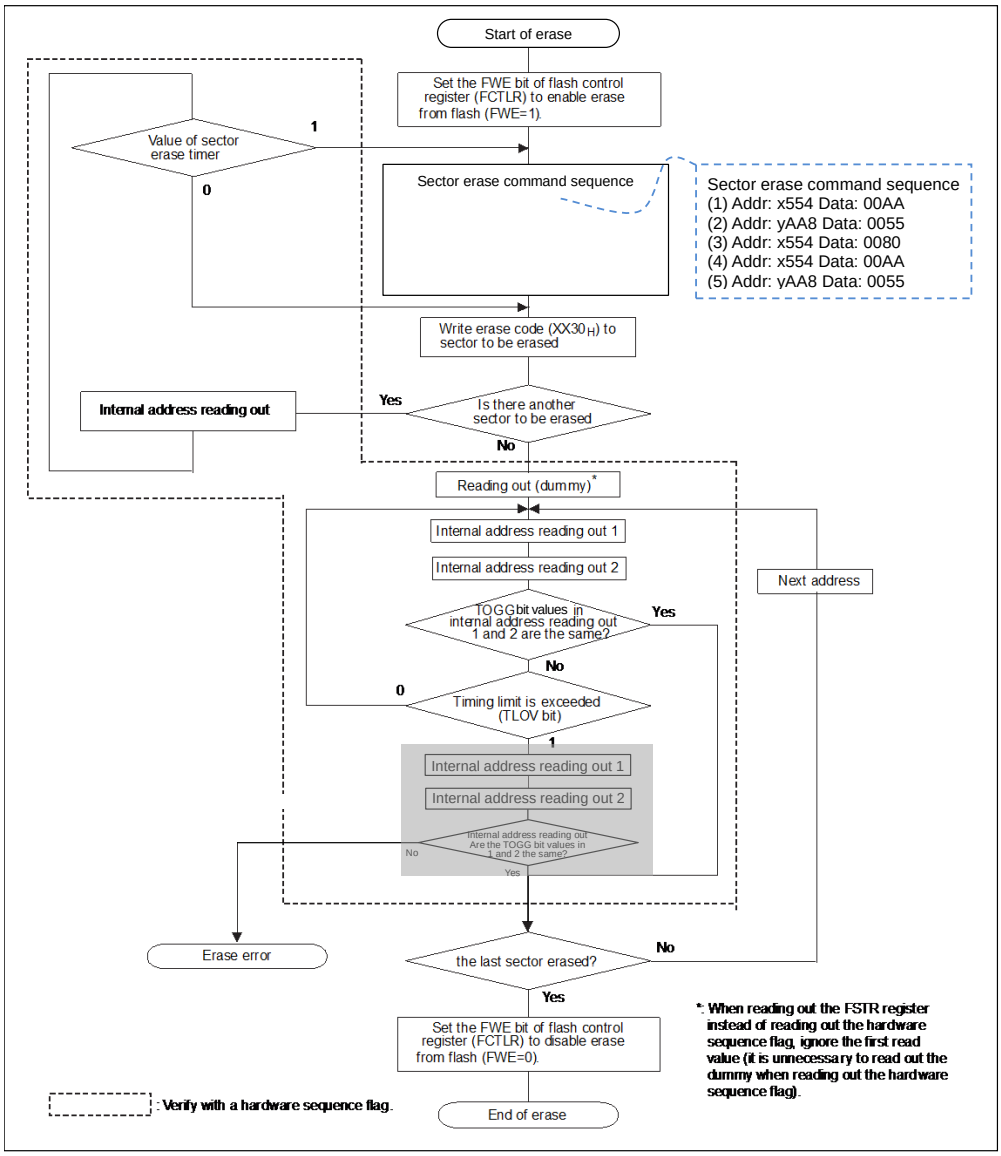
(Error)
16.7µs or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.

(Correct)
16.7µs + 2 cyc or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.</td></tr>
<tr><td>1605</td><td>45.5.4</td><td>The description in "5.4. Reset Command" should be modified as follows:

(Error)
When the power is turned on, there is no need to issue a data read command.

(Correct)
When the power is turned on, there is no need to issue a reset command.</td></tr>
</table>

Page	Section	Change Results
1609	45.5.7.	Figure 5-3 Example of Sector Erase Procedure in "5.7. Sector Erase Command" should be modified as follows: (Error) 

Page	Section	Change Results
1609	45.5.7	(Continued) (Correct) 

Page	Section	Change Results
1610	45.5.7	The notes in "5.7. Sector Erase Command" should be modified as follows: (Error) · 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. (Correct) · Only a command to add an erased sector and a command to suspend erasing are write commands acceptable in the command timeout state, while only a command to suspend erasing is an acceptable command in the sector erasing state.
1615	46.2	The description in "· Error correction code (ECC) function" in "2. Features" should be modified as follows: (Error) · Error correction code (ECC) function ·There is an error correction code (ECC) function that corrects errors of up to one bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read. Furthermore, ECC codes are automatically added when writing to the flash memory. Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development. (Correct) · Error correction code (ECC) function · There is an error correction code (ECC) function that corrects errors of up to one bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read. Furthermore, ECC codes are automatically added when writing to the flash memory. Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development. · An error is detected when data is read in the chip erase/sector erase state. Always write "FFFF" and then read data when data with erase state (FFFF) needs to be read without error detection.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1620</td><td>46.4.3</td><td>A note should be added in [bit4] DFWDSBL (Data Fetch Wait cycle DiSaBle): Data fetch wait cycle disabled in "4.3. Flash Interface Control Register: FLIFCTLR (Flash I/F ConTroL Register)".

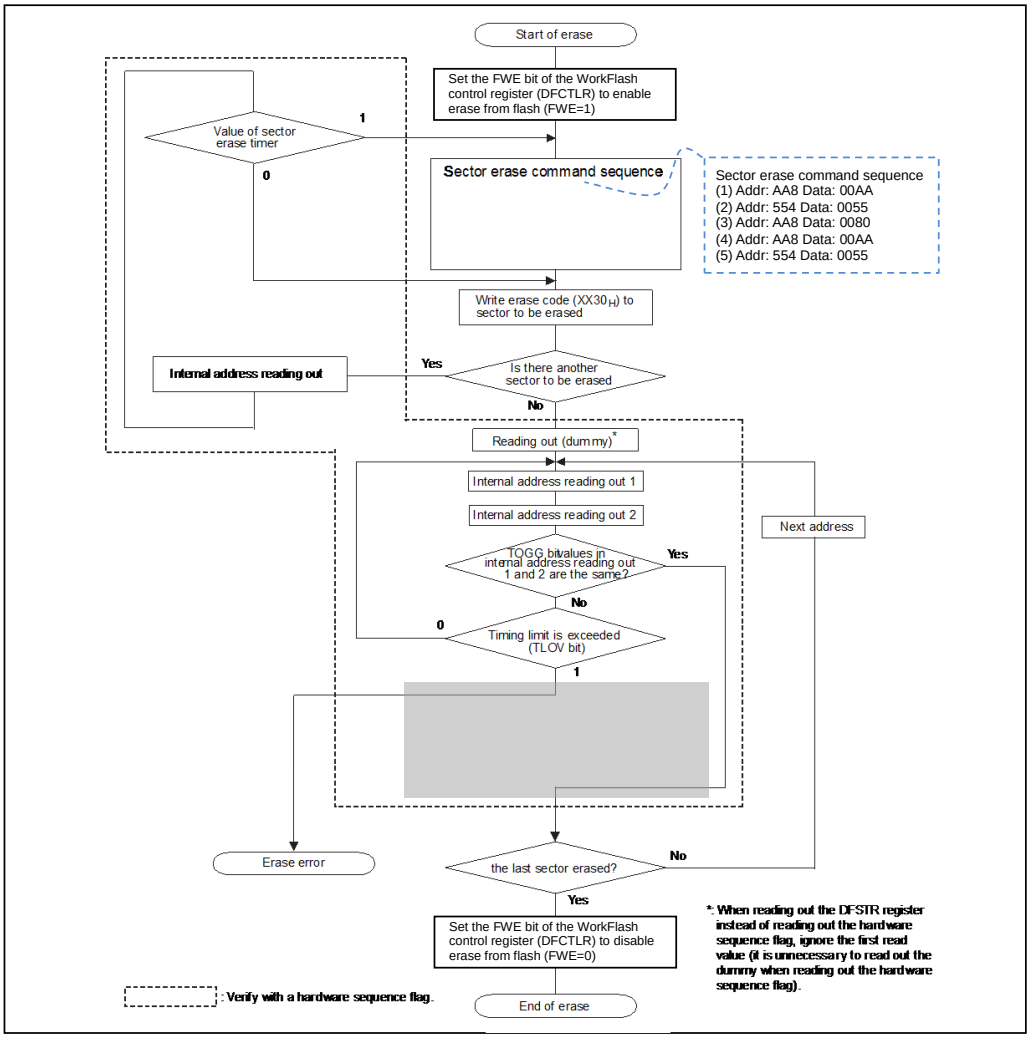
(Correct)
<table><tr><th>DFWDSBL</th><th>Description</th></tr><tr><td>0</td><td>Wait cycle enabled (Initial value)</td></tr><tr><td>1</td><td>Wait cycle disabled</td></tr></table>
Note:
When this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.</td></tr>
<tr><td>1626</td><td>46.5.3.1</td><td>The description in notes in ■ Sector Erase Suspend Command in "5.3.1. Command Sequence" should be modified as follows:

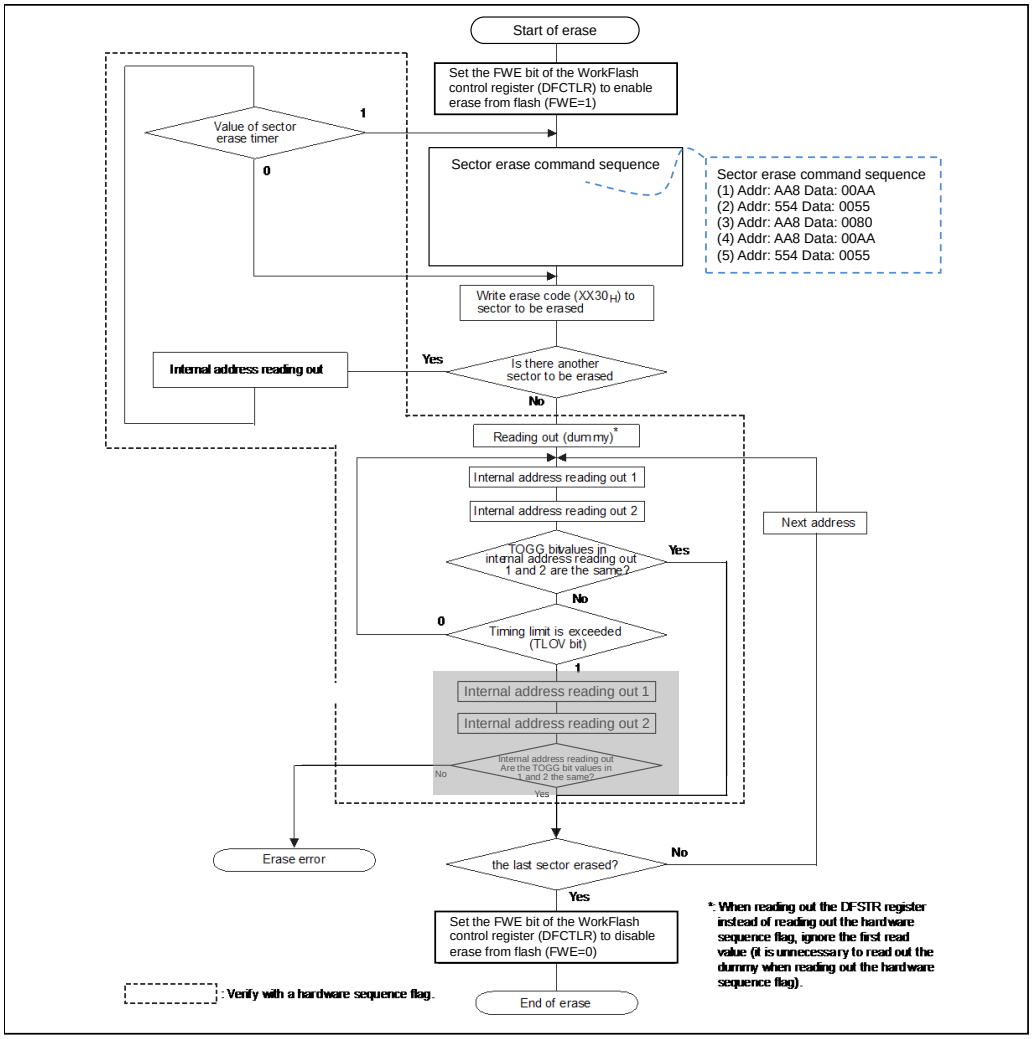
(Error)
16.7µs or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible

(Correct)
16.7µs + 2 cyc or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.</td></tr>
<tr><td>1630</td><td>46.5.4</td><td>The description in "5.4. Reset Command" should be modified as follows:

(Error)
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.

(Correct)
When the power is turned on, there is no need to issue a reset command. Furthermore, in reset state, data can be read using normal read access and programs can be accessed by the CPU, thus there is no need to issue a reset command when reading data.</td></tr>
</table>

Page	Section	Change Results
1634	46.5.7	Figure 5-3 Example of Sector Erase Procedure in "5.7. Sector Erase Command" should be modified as follows: (Error) 

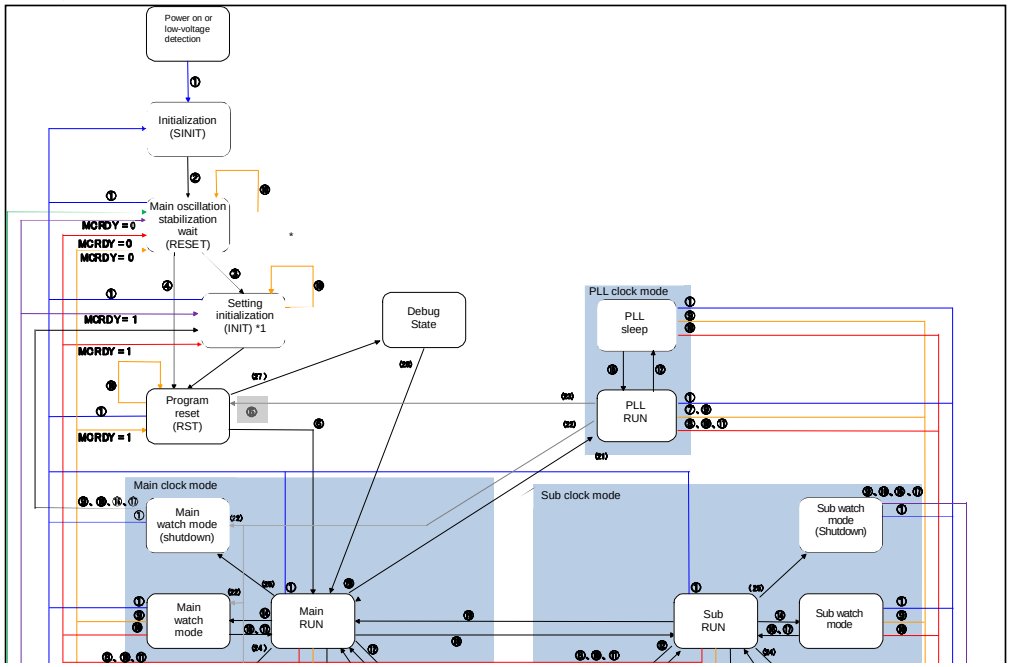
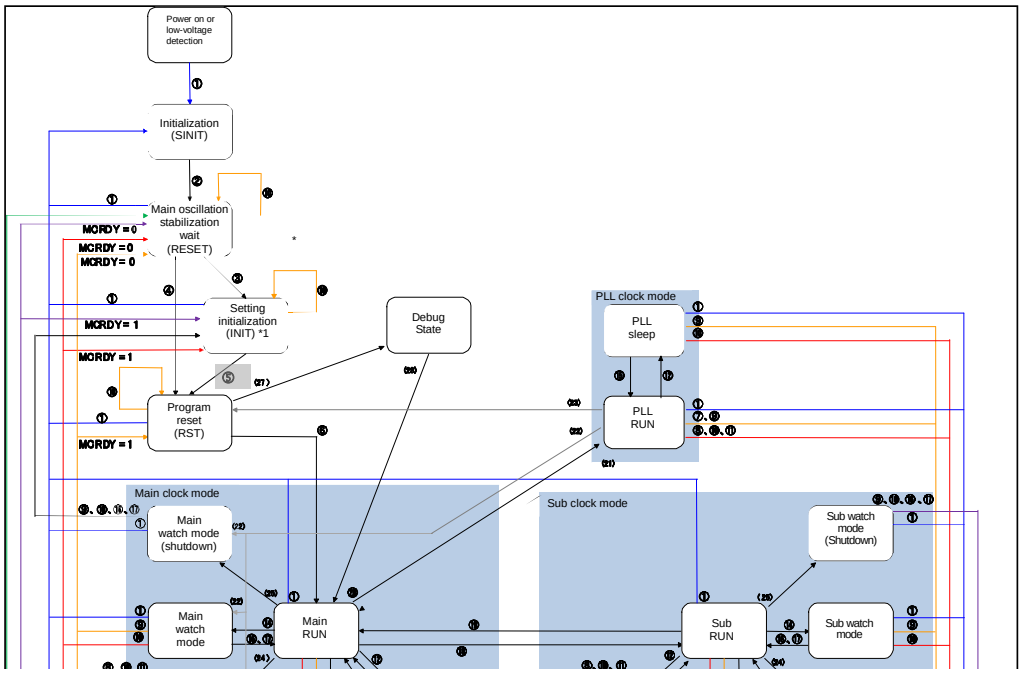
Page	Section	Change Results
1634	46.5.7	(Continued) (Correct) 

Page	Section	Change Results
1635	46.5.7	The notes in "5.7. Sector Erase Command" should be modified as follows: (Error) · 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. (Correct) · Only a command to add an erased sector and a command to suspend erasing are write commands acceptable in the command timeout state, while only a command to suspend erasing is an acceptable command in the sector erasing state.
1636	46.5.9.2	The description in "5.9.2. Flash Security Setting Method" should be modified as follows: (Error) When reset is input and released after writing 0x0001 to the flash security code area (see Figure 3-2 to Figure 3-6 Sector Configuration in "CHAPTER : FLASH MEMORY"), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased. (Correct) When reset is input and released after writing 0x0001 to the flash security code area (see Figure 3-2 Sector Configuration Diagram in "CHAPTER: FLASH MEMORY"), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.
1637	46.5.9.3	The note in "5.9.3. Unlocking Flash Security" should be modified as follows: (Error) 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. (Correct) 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. Execute chip erase to each flash macro in the above order from the viewpoint of the data protection stored in the flash macro.

Page	Section	Change Results
1638	46.5.10	All the words "flash memory" should be changed to "work flash memory" in "5.10. Notes on Using Flash Memory". (Error) 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 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 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 "5.2 Writing Flash Memory by CPU" for procedure. · Do not issue commands to multiple macros simultaneously (i.e., in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit. · If authentication by password of on-chip debugger (OCD) completes, you can read the content of flash memory from external by using OCD even if security is ON. When you want to 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.

Page	Section	Change Results
1638	46.5.10	(Continued) (Correct) 5.10. Notes on Using WorkFlash Memory Notes on using the workflash memory are shown below. · If this device is reset during a write, the data that was written cannot be guaranteed. · If CPU programming mode is set (FEW=1) using the FWE bit of the WorkFlash control register (DFCTLR), do not execute the program in workflash 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 workflash memory, do not generate interrupt requests. The program runs out of control without fetching the correct values. · Because this model has the ECC bit added, data always needs to be written as 32-bit by two 16-bit writes. See "5.2. Writing Flash Memory by CPU" for procedure. · Do not issue commands to multiple macros simultaneously (i.e., in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit. · If authentication by password of the on-chip debugger (OCD) has completed, you can read the content of work 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 standby state is prohibited during flash program/erase. · Because of the built-in ECC in this workflash memory, data superscription to an address where some values have already been written cannot be done.
1651	47.5.2.1	The description in ● INIT notification phase in "5.2.1. Chip Reset Sequence" should be modified as follows: (Error) INIT notification phase is the interval when the start phase is ended until 568 sampling clock cycles of the normal UART is counted. OCDU outputs L of 280 cycles to MDI twice while it is in this phase (The idle of eight cycles is inserted among), and notifies the tool the generation of INIT. (Correct) INIT notification phase is the interval when the start phase is ended until 568 sampling clock cycles of the normal UART is counted. OCDU outputs "L" of 280 cycles to DEBUG I/F twice while it is in this phase (An idle of eight cycles is inserted), and notifies the tool of the generation of INIT.

Page	Section	Change Results
1652	47.5.2.1	The description in ● Mode entry phase in "5.2.1. Chip Reset Sequence" should be modified as follows: (Error) If the mode command is received immediately after starting the mode entry phase, the mode command must be received after waiting for input of "H" to MDI for more than one cycle width using the UART reception sampling clock. If this condition is not met, the start bit of the mode command reception cannot be detected normally, the mode may not be entered correctly. (Correct) If the mode command is received immediately after starting the mode entry phase, the mode command must be received after waiting for the input of "H" to DEBUG I/F for more than one cycle width using the UART reception sampling clock. If this condition is not met, the start bit of the mode command reception cannot be detected normally, the mode may not be entered correctly.
1654	47.5.3.1	The following register name in the description in "5.3.1. Clock Setting" should be modified. (Error) CCPDSDIVR.PODS (Correct) CCPSDIVR.PODS

Page	Section	Change Results
1655	47.5.3.3	Figure 5-2 Device State in "5.3.3. Clock Reset State Transitions" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
1657	47.5.3.4	The following description should be deleted from the remarks for the reset factor "External power supply low-voltage detection reset" in the table in "5.3.4. Summary of Specification Restrictions". (Error) No voltage step-down circuit switch stabilization wait time (*2) *: Only recovery from main/sub stop mode or main/sub watch mode No main oscillation stabilization wait time *: Only recovery from main/sub stop mode or sub watch mode (Correct) No voltage step-down circuit switch stabilization wait time (*2) *: Only recovery from main/sub stop mode or main/sub watch mode

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1661</td><td>47.5.4</td><td>
Table 5-1 OCD-DSU ID Code of This Series in "5.4. OCD-DSU ID Code and Mount Type Information on This Series" should be modified as follows:

(Error)
<table>
<tr><th>ID name</th><th>bit width</th><th>Associated ID register name</th><th>Address in the OCD space</th><th>Value</th><th>Remarks</th></tr>
<tr><td>Manufacturer ID</td><td>16</td><td>E_IDMCR</td><td>0x000</td><td>0x0400</td><td>Fujitsu code</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="12">Device ID</td><td rowspan="12">16</td><td rowspan="12">E_IDDCR</td><td rowspan="12">0x002</td><td>0x0030</td><td>MB91F52xBW</td></tr>
<tr><td>0x0031</td><td>MB91F52xBS</td></tr>
<tr><td>0x0032</td><td>MB91F52xDW</td></tr>
<tr><td>0x0033</td><td>MB91F52xDS</td></tr>
<tr><td>0x0034</td><td>MB91F52xFW</td></tr>
<tr><td>0x0035</td><td>MB91F52xFS</td></tr>
<tr><td>0x0036</td><td>MB91F52xJW</td></tr>
<tr><td>0x0037</td><td>MB91F52xJS</td></tr>
<tr><td>0x0038</td><td>MB91F52xKW</td></tr>
<tr><td>0x0039</td><td>MB91F52xKS</td></tr>
<tr><td>0x003a</td><td>MB91F52xLW</td></tr>
<tr><td>0x003b</td><td>MB91F52xLS</td></tr>
<tr><td>Device version ID</td><td>4</td><td>E_IDVCR</td><td>0x003</td><td>0x1</td><td></td></tr>
</table>

(Correct)
<table>
<tr><th>ID name</th><th>bit width</th><th>Associated ID register name</th><th>Address in the OCD space</th><th>Value</th><th>Remarks</th></tr>
<tr><td>Manufacturer ID</td><td>16</td><td>E_IDMCR</td><td>0x000</td><td>0x0400</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="12">Device ID</td><td rowspan="12">16</td><td rowspan="12">E_IDDCR</td><td rowspan="12">0x002</td><td>0x0030</td><td>MB91F52xBW/Y/J/L</td></tr>
<tr><td>0x0031</td><td>MB91F52xBS/U/H/K</td></tr>
<tr><td>0x0032</td><td>MB91F52xDW/Y/J/L</td></tr>
<tr><td>0x0033</td><td>MB91F52xDS/U/H/K</td></tr>
<tr><td>0x0034</td><td>MB91F52xFW/Y/J/L</td></tr>
<tr><td>0x0035</td><td>MB91F52xFS/U/H/K</td></tr>
<tr><td>0x0036</td><td>MB91F52xJW/Y/J/L</td></tr>
<tr><td>0x0037</td><td>MB91F52xJS/U/H/K</td></tr>
<tr><td>0x0038</td><td>MB91F52xKW/Y/J/L</td></tr>
<tr><td>0x0039</td><td>MB91F52xKS/U/H/K</td></tr>
<tr><td>0x003a</td><td>MB91F52xLW/Y/J/L</td></tr>
<tr><td>0x003b</td><td>MB91F52xLS/U/H/K</td></tr>
<tr><td>Device version ID</td><td>4</td><td>E_IDVCR</td><td>0x003</td><td>0x2
0x3</td><td>Revision : B
Revision : C</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1664</td><td>48.3.</td><td>■ Pin Application of the Waveform Generator in "3. Configuration" should be deleted.

(Error)
■ Pin Application of the Waveform Generator in
Table 3-1 shows the pin application of the waveform generator.
Table 3-1 Pin Application of the Waveform Generator.
<table>
<tr><td>Waveform Generator.</td></tr>
<tr><td>Compare output 0/1/2/3/4/5</td></tr>
<tr><td>PPG0/2/4</td></tr>
<tr><td>GATE0/2/4</td></tr>
<tr><td>RTO0/1/2/3/4/5</td></tr>
<tr><td>Interrupt0/1/2</td></tr>
<tr><td>Interrupt reload 0/1/2</td></tr>
<tr><td>DTTI Interrupt0</td></tr>
<tr><td>DTTI Interrupt0</td></tr>
<tr><td>16-bit dead timer0/1/2</td></tr>
</table>
</td></tr>
<tr><td>1666</td><td>48.4</td><td>The following table should be added in "4. Registers". (before ■ List of Waveform Generator Register)

(Correct)
■ Table of External Pins
Table 4-1 Table of External Pins
<table>
<tr><td rowspan="2">Channel</td><td colspan="4">External pin</td></tr>
<tr><td>MB91F52xB</td><td>MB91F52xD</td><td>MB91F52xF</td><td>MB91F52xJ,
MB91F52xK,
MB91F52xL</td></tr>
<tr><td>DTTI</td><td>DTTI_0/
DTTI_1</td><td>DTTI_0/
DTTI_1</td><td>DTTI_0/
DTTI_1</td><td>DTTI_0/
DTTI_1/
DTTI_2</td></tr>
<tr><td>0</td><td>RTO0_1</td><td>RTO0_1</td><td>RTO0_1</td><td>RTO0_0/
RTO0_1</td></tr>
<tr><td>1</td><td>RTO1_1</td><td>RTO1_1</td><td>RTO1_1</td><td>RTO1_0/
RTO1_1</td></tr>
<tr><td>2</td><td>RTO2_1</td><td>RTO2_0/
RTO2_1</td><td>RTO2_0/
RTO2_1</td><td>RTO2_0/
RTO2_1</td></tr>
<tr><td>3</td><td>RTO3_0/
RTO3_1</td><td>RTO3_0/
RTO3_1</td><td>RTO3_0/
RTO3_1</td><td>RTO3_0/
RTO3_1</td></tr>
<tr><td>4</td><td>RTO4_0/
RTO4_1</td><td>RTO4_0/
RTO4_1</td><td>RTO4_0/
RTO4_1</td><td>RTO4_0/
RTO4_1</td></tr>
<tr><td>5</td><td>RTO5_1</td><td>RTO5_1</td><td>RTO5_0/
RTO5_1</td><td>RTO5_0/
RTO5_1</td></tr>
</table>
</td></tr>
</table>

Page	Section	Change Results
1669	48.4.1.2	The description in [bit6] GTEN1 of DTSCR0 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20.
1669	48.4.1.2	The description in [bit5] GTEN0 of DTSCR0 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL01, PSEL00 of SIGCR20.
1671	48.4.1.2	The description in [bit6] GTEN3 of DTSCR1 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20.
1671	48.4.1.2	The description in [bit5] GTEN2 of DTSCR1 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL11, PSEL10 of SIGCR20.

Page	Section	Change Results
1673	48.4.1.2	The description in [bit6] GTEN5 of DTSCR2 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20.
1673	48.4.1.2	The description in [bit5] GTEN4 of DTSCR2 in "4.1.2. 16-bit Dead Timer State Control Register" should be modified as follows: (Error) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20/21. (Correct) · If it is set to 1, GATE signal will be output. PPG of the output destination can be selected from PSEL21, PSEL20 of SIGCR20.
1678	48.4.1.4	The following sentence should be added in [bit2] MNS2: Dead time function selection bit (RTO4 and RTO5) in "4.1.4. 16-bit Dead Timer Minus Control Register (DTMNS)". (Error) · When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits. (Correct) · When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits. · The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.
1678	48.4.1.4	The following sentence should be added in [bit1] MNS1: Dead time function selection bit (RTO2 and RTO3) in "4.1.4. 16-bit Dead Timer Minus Control Register (DTMNS)". (Error) · When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits. (Correct) · When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits. · The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1679</td><td>48.4.1.4</td><td>The following description should be added in [bit0] MNS0: Dead time function selection bit (RTO0 and RTO1) in "4.1.4. 16-bit Dead Timer Minus Control Register (DTMNS).

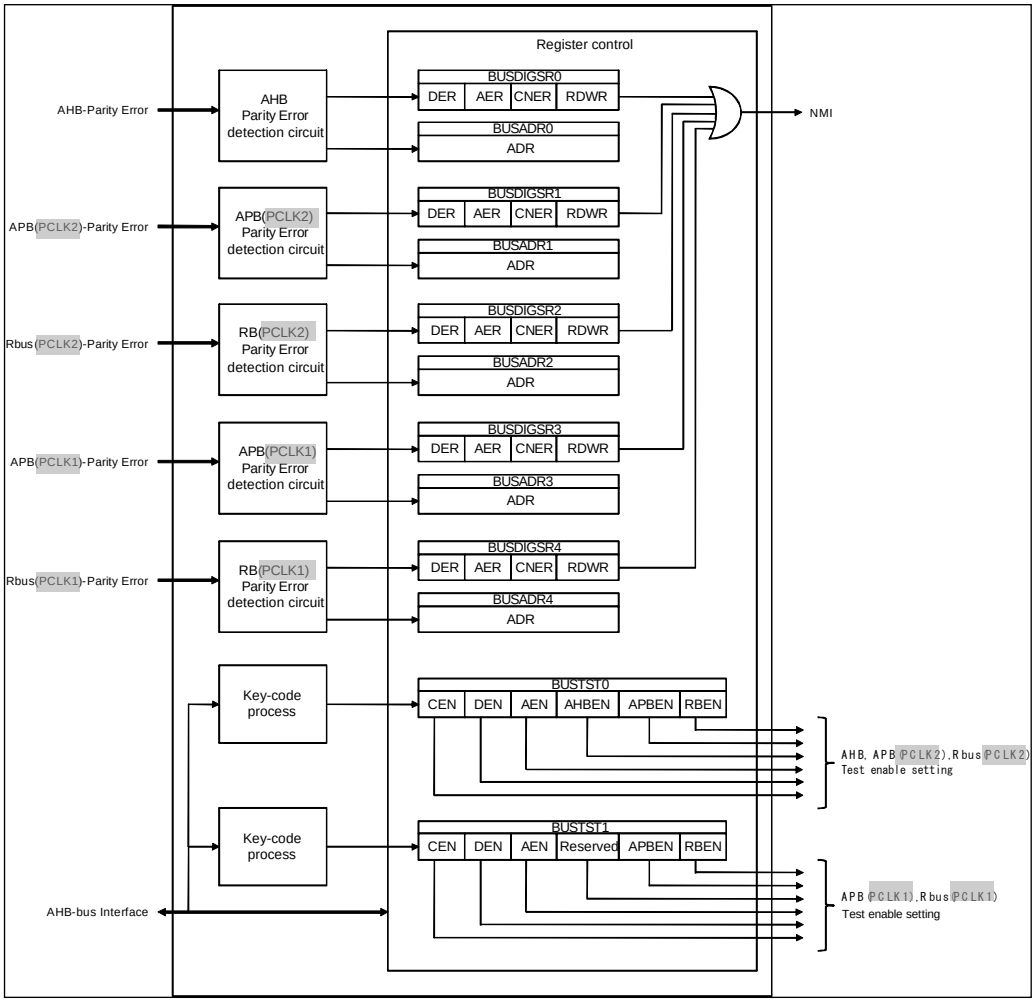
(Error)
· When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits.

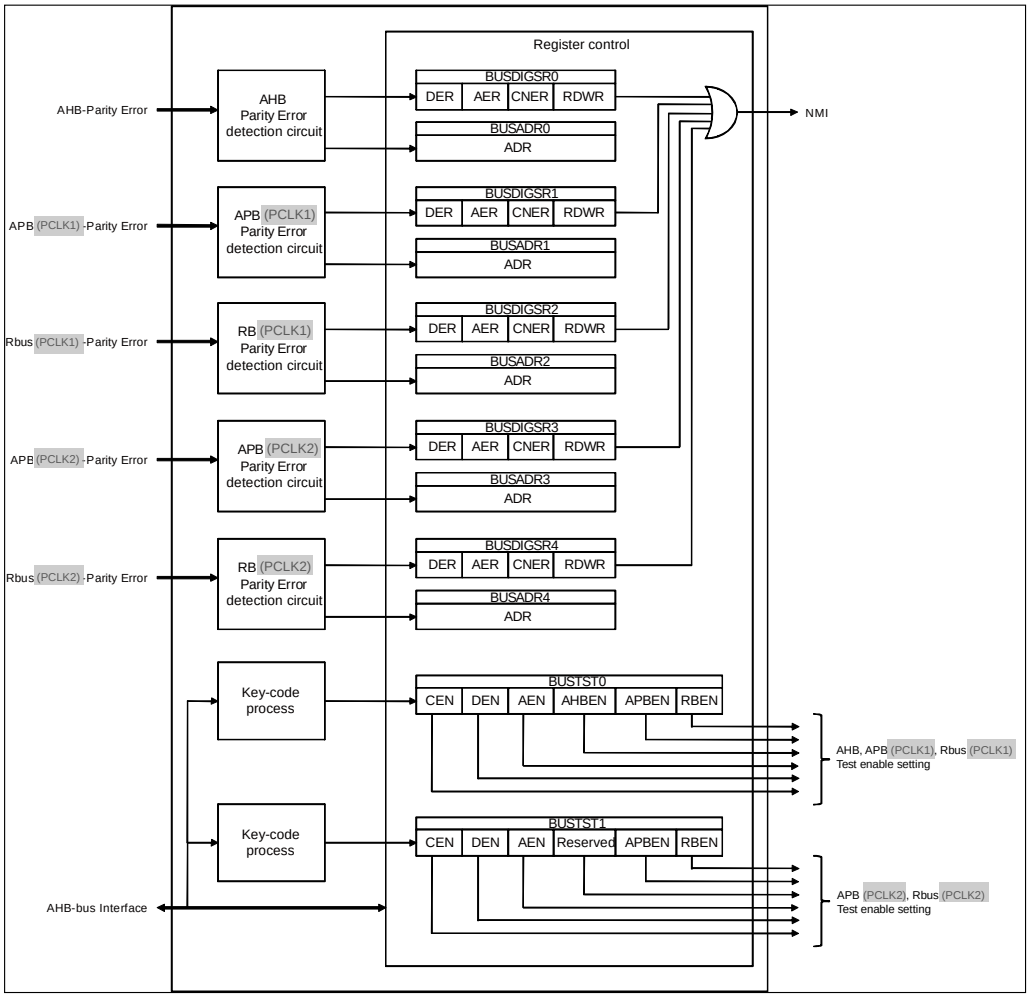
(Correct)
· When you set a value to this bit, follow the writing procedure using the KEY1, KEY0 bits.
· The key code setting becomes disabled when different data is written during writing of key enable bits "00", "01", "10", and "11", and the register needs to be set from the beginning.</td></tr>
<tr><td>1707</td><td>49.1</td><td>The table in "1. Overview" should be modified as follows:

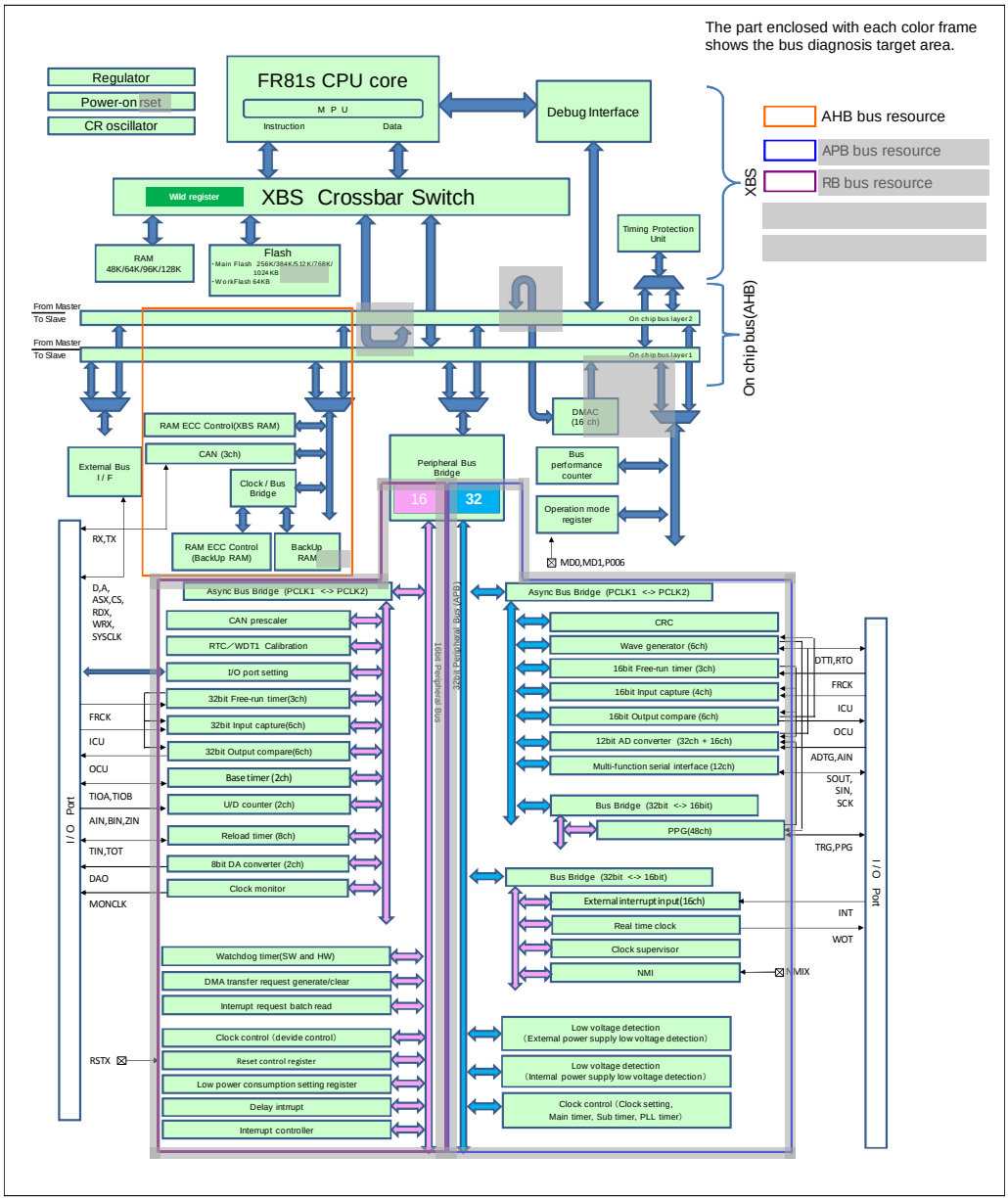
(Error)<table><tr><th></th><th></th><th>Function</th></tr><tr><td>2</td><td>Diagnosis bus</td><td>· AHB, APB(PCLK2), Rbus(PCLK2), APB(PCLK1), Rbus(PCLK1)</td></tr></table>
(Correct)<table><tr><th></th><th></th><th>Function</th></tr><tr><td>2</td><td>Diagnosis bus</td><td>· AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2), Rbus(PCLK2)</td></tr></table></td></tr>
<tr><td>1707</td><td>49.2</td><td>The description in "2. Features" should be modified as follows:

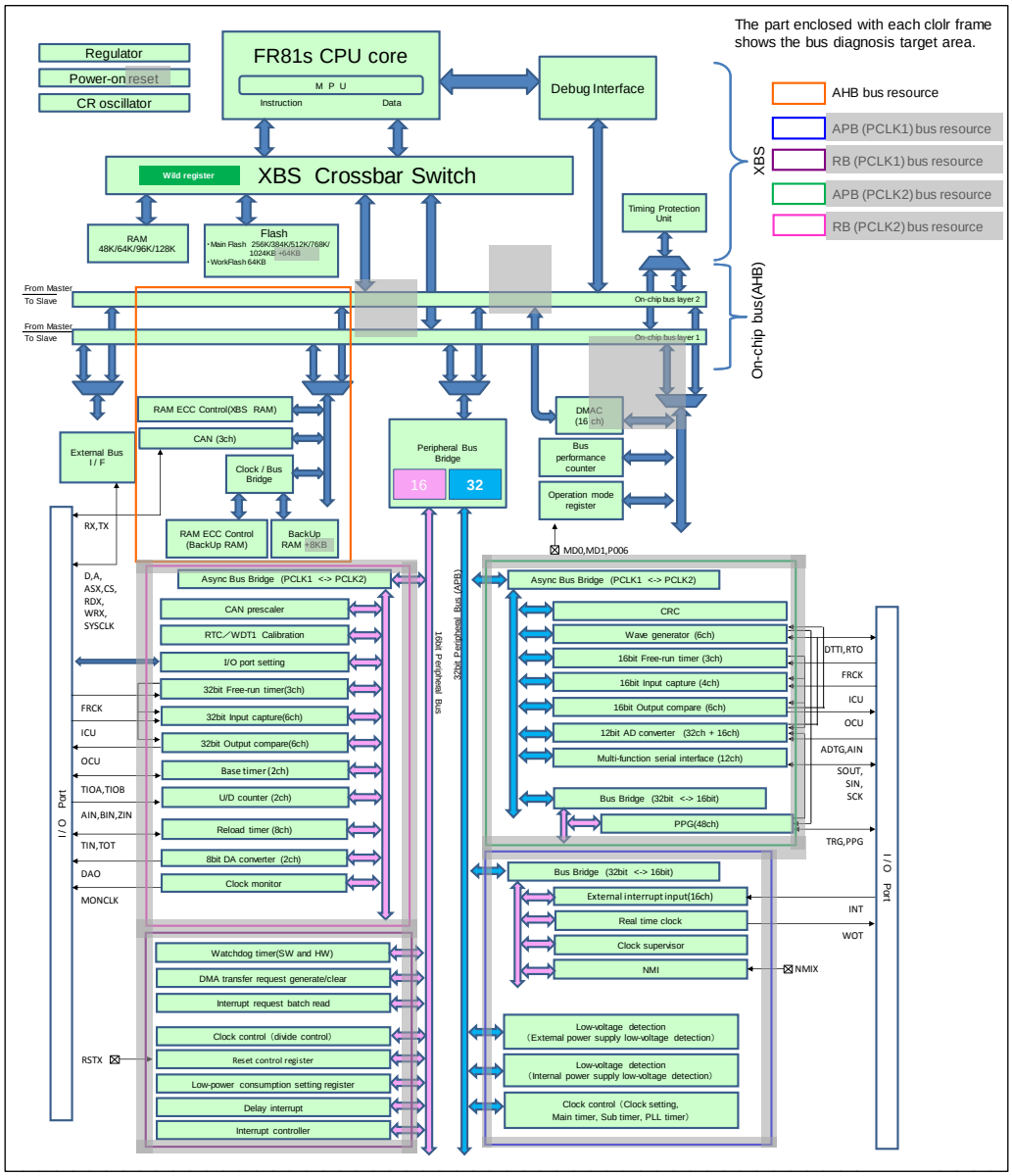
(Error)
· The internal bus of the CPU consists of AHB, APB(PCLK2), Rbus(PCLK2), APB(PCLK1) and Rbus(PCLK1).

(Correct)
· The internal bus of the CPU consists of AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2) and Rbus(PCLK2).</td></tr>
</table>

Page	Section	Change Results
1708	49.3	Figure 3-1 Configuration Diagram in "3. Configuration" should be modified as follows: (Error) 

Page	Section	Change Results
1708	49.3	(Continued) (Correct) 

Page	Section	Change Results
1709	49.3	Figure 3-2 Bus Diagnosis Target Area Diagram in "3. Configuration" should be modified as follows: (Error) 

Page	Section	Change Results
1709	49.3	(Continued) (Correct) 

Page	Section	Change Results
1711	49.4.1	"4.1. Bus DIaGnosis Status Register: BUSDIGSR" should be modified as follows: (Error) The bus diagnosis status register (BUSDIGSR) consists of a data parity error, address parity error, control parity error, data direction, and error flag clear. Bus diagnosis status register 0 indicates AHB error status. Bus diagnosis status register 1 indicates APB(PCLK2) error status. Bus diagnosis status register 2 indicates Rbus(PCLK2) error status. Bus diagnosis status register 3 indicates APB(PCLK1) error status. Bus diagnosis status register 4 indicates Rbus (PCLK1) error status. (Correct) The bus diagnosis status register (BUSDIGSR) consists of a data parity error, address parity error, control parity error, data direction, and error flag clear. Bus diagnosis status register 0 indicates AHB error status. Bus diagnosis status register 1 indicates APB(PCLK1) error status. Bus diagnosis status register 2 indicates Rbus(PCLK1) error status. Bus diagnosis status register 3 indicates APB(PCLK2) error status. Bus diagnosis status register 4 indicates Rbus (PCLK2) error status.
1712	49.4.1	The description in [bit1] CNER in "4.1. BUS DIaGnosis Status Register: BUSDIGSR" should be modified as follows: (Error) This bit is not updated while this bit is "1", any one of the AER bits is "1", or the DER bit is "1". (Correct) This bit is not updated while this bit is "1", any of the AER bits is "1", or the DER bit is "1".
1712	49.4.1	The following description should be deleted from the notes for BUSDIGSR in "4.1. BUS DIaGnosis Status Register: BUSDIGSR". (Error) · Bus diagnosis status register 0 (BUSDIGSR0) indicates AHB bus error status. Bus diagnosis status register 1 (BUSDIGSR1) indicates APB bus error status. Bus diagnosis status register 2 (BUSDIGSR2) indicates Rbus bus error status.
1714	49.4.2	[bit10] RBEN: Rbus parity error generation enable in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · This bit enables the generation of a Rbus (PCLK2) parity error. (Correct) · This bit enables the generation of a Rbus (PCLK1) parity error.

Page	Section	Change Results
1714	49.4.2	[bit9] APBEN: APB parity error generation enable in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · This bit enables the generation of an APB (PCLK2) parity error. (Correct) · This bit enables the generation of an APB (PCLK1) parity error.
1716	49.4.2	The note in [bit11] CEN in ■ BUSTSTR1: Address $311A_H$ (Access: Half-word, Word) in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · For RBEN=0, APBEN=0, and AHBEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit. (Correct) · For RBEN= 0 and APBEN= 0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.
1716	49.4.2	[bit10] RBEN: Rbus parity error generation enable in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · This bit enables the generation of a Rbus (PCLK1) parity error. (Correct) · This bit enables the generation of a Rbus (PCLK2) parity error.
1717	49.4.2	[bit9] APBEN: APB parity error generation enable in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · This bit enables the generation of an APB (PCLK1) parity error. (Correct) · This bit enables the generation of an APB (PCLK2) parity error.

Page	Section	Change Results
1717	49.4.2	The notes in [bit7 to bit4] DEN3 to DEN0: Data error in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits (Correct) · For RBEN=0 and APBEN=0, these bits are invalid. Under such conditions, it will be the same behavior as when "0000" is set to these bits.
1718	49.4.2	The note in [bit3 to bit0] AEN3 to AEN0: Address error in "4.2. BUS diagnosis TeST Register: BUSTSTR0/1" should be modified as follows: (Error) · For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits. (Correct) · For RBEN=0 and APBEN=0, these bits are invalid. Under such conditions, it will be the same behavior as when "0000" is set to these bits.
1718	49.4.3	The notation of PCLK1 and PCLK2 of bus diagnosis address register 1/2/3/4 in "4.3. BUS diagnosis ADdRess Register: BUSADR" should be modified as follows: (Error) Bus diagnosis address register 0 indicates an address in which an AHB diagnosis error was detected. Bus diagnosis address register 1 indicates an address in which an APB (PCLK2) diagnosis error was detected. Bus diagnosis address register 2 indicates an address in which an Rbus (PCLK2) diagnosis error was detected. Bus diagnosis address register 3 indicates an address in which an APB (PCLK1) diagnosis error was detected. Bus diagnosis address register 4 indicates an address in which an Rbus (PCLK1) diagnosis error was detected. (Correct) Bus diagnosis address register 0 indicates an address in which an AHB diagnosis error was detected. Bus diagnosis address register 1 indicates an address in which an APB (PCLK1) diagnosis error was detected. Bus diagnosis address register 2 indicates an address in which an Rbus (PCLK1) diagnosis error was detected. Bus diagnosis address register 3 indicates an address in which an APB (PCLK2) diagnosis error was detected. Bus diagnosis address register 4 indicates an address in which an Rbus (PCLK2) diagnosis error was detected.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1720</td><td>49.5.1</td><td>The table in ■ Data error detection in "5.1. Error Detection" should be modified as follows:

(Error)
<table>
<tr><th rowspan="3">Access size</th><th rowspan="3">Address</th><th colspan="4">BUSDIGSR0(AHB: On-chip bus)
BUSDIGSR1/3(APB:32bit peripheral bus)
BUSDIGSR2/4(R-bus:16bit peripheral bus)</th></tr>
<tr><th>DER[0]</th><th>DER[1]</th><th>DER[2]</th><th>DER[3]</th></tr>
<tr><th>Data
bit31-24</th><th>Data
bit21-16</th><th>Data
bit15-8</th><th>Data
bit7-0</th></tr>
<tr><td>Word access</td><td>Addr+0</td><td>○*</td><td>○*</td><td>○*</td><td>○*</td></tr>
<tr><td>Half-word access</td><td>Addr+0</td><td>○</td><td>○</td><td>-</td><td>-</td></tr>
<tr><td>Half-word access</td><td>Addr+2</td><td>-</td><td>-</td><td>○</td><td>○</td></tr>
<tr><td>Byte access</td><td>Addr+0</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+1</td><td>-</td><td>○</td><td>-</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+2</td><td>-</td><td>-</td><td>○</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+3</td><td>-</td><td>-</td><td>-</td><td>○</td></tr>
</table>

(Correct)
<table>
<tr><th rowspan="3">Access size</th><th rowspan="3">Address</th><th colspan="4">BUSDIGSR0 (AHB: On-chip bus)
BUSDIGSR1/3 (APB: 32bit peripheral bus)
BUSDIGSR2/4 (R-bus: 16bit peripheral bus)</th></tr>
<tr><th>DER[0]</th><th>DER[1]</th><th>DER[2]</th><th>DER[3]</th></tr>
<tr><th>Data
bit31-24</th><th>Data
bit21-16</th><th>Data
bit15-8</th><th>Data
bit7-0</th></tr>
<tr><td>Word access</td><td>Addr+0</td><td>○*</td><td>○*</td><td>○*</td><td>○*</td></tr>
<tr><td>Half-word access</td><td>Addr+0</td><td>○</td><td>○</td><td>-</td><td>-</td></tr>
<tr><td>Half-word access</td><td>Addr+2</td><td>-</td><td>-</td><td>○</td><td>○</td></tr>
<tr><td>Byte access</td><td>Addr+0</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+1</td><td>-</td><td>○</td><td>-</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+2</td><td>-</td><td>-</td><td>○</td><td>-</td></tr>
<tr><td>Byte access</td><td>Addr+3</td><td>-</td><td>-</td><td>-</td><td>○</td></tr>
</table>
</td></tr>
<tr><td>1734</td><td>50.4</td><td>The content of the register function of 0x3030 in Table 4-1 Register Map in "4. Registers" should be modified as follows:

(Error)
<table>
<tr><td>0x3030</td><td>TEAR0A</td><td>TEST error address register 0
XBS RAM</td></tr>
</table>

(Correct)
<table>
<tr><td>0x3030</td><td>TEAR0A</td><td>TEST error address register 0
BACKUP-RAM</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1857</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-1 Interrupt Vector MB91F52xB (64pin) in "B. List of Interrupt Vector".

(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
<tr><td>1861</td><td>Appendix B</td><td>The interrupt factor in Table B-2 Interrupt Vector MB91F52xD (80pin) in "B. List of Interrupt Vector" should be modified as follows:

(Error)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45*5</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>
(Correct)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table></td></tr>
<tr><td>1861</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-2 Interrupt Vector MB91F52xD (80pin) in "B. List of Interrupt Vector".

(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1865</td><td>Appendix B</td><td>The interrupt factor in Table B-3 Interrupt Vector MB91F52xF (100pin) in "B. List of Interrupt Vector" should be modified as follows:

(Error)
<table>
<tr><td>Base timer 0 IRQ0</td><td rowspan="2">60</td><td rowspan="2">3C</td><td rowspan="2">ICR44</td><td rowspan="2">$30C_H$</td><td rowspan="2">$000FFF0C_H$</td><td rowspan="2">44</td></tr>
<tr><td>Base timer 0 IRQ1</td></tr>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45*5</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>

(Correct)
<table>
<tr><td>–</td><td rowspan="2">60</td><td rowspan="2">3C</td><td rowspan="2">ICR44</td><td rowspan="2">$30C_H$</td><td rowspan="2">$000FFF0C_H$</td><td rowspan="2">44</td></tr>
<tr><td>–</td></tr>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table></td></tr>
<tr><td>1865</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-3 Interrupt Vector MB91F52xF (100pin) in "B. List of Interrupt Vector".

(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
<tr><td>1868</td><td>Appendix B</td><td>"42" is deleted as shown below from the interrupt factor in Table B-4 Interrupt Vector MB91F52xJ (120pin) in "B. List of Interrupt Vector".

(Error)
<table>
<tr><td>PPG2/3/12/13/22/23/32/33/42/43</td><td rowspan="2">41</td><td rowspan="2">29</td><td rowspan="2">ICR25</td><td rowspan="2">358_H</td><td rowspan="2">$000FFF58_H$</td><td rowspan="2">25*3</td></tr>
<tr><td>16bit free-run timer 2 (0 detection) / (compare clear)</td></tr>
</table>

(Correct)
<table>
<tr><td>PPG2/3/12/13/22/23/32/33/43</td><td rowspan="2">41</td><td rowspan="2">29</td><td rowspan="2">ICR25</td><td rowspan="2">358_H</td><td rowspan="2">$000FFF58_H$</td><td rowspan="2">25*3</td></tr>
<tr><td>16-bit free-run timer 2 (0 detection) / (compare clear)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1869</td><td>Appendix B</td><td>The interrupt factor in Table B-4 Interrupt Vector MB91F52xJ (120pin) in "B. List of Interrupt Vector" should be modified as follows:

(Error)
<table>
<tr><td>Base timer 1 IRQ0
Base timer 1 IRQ1
–
–</td><td>61</td><td>3D</td><td>ICR45</td><td>308_H</td><td>$000FFF08_H$</td><td>45*5</td></tr>
</table>
(Correct)
<table>
<tr><td>Base timer 1 IRQ0
Base timer 1 IRQ1
–
–</td><td>61</td><td>3D</td><td>ICR45</td><td>308_H</td><td>$000FFF08_H$</td><td>45</td></tr>
</table></td></tr>
<tr><td>1869</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-4 Interrupt Vector MB91F52xJ (120pin) in "B. List of Interrupt Vector".

(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
<tr><td>1872</td><td>Appendix B</td><td>"42" is deleted as shown below from the interrupt factor in Table B-5 Interrupt Vector MB91F52xK (144pin) in "B. List of Interrupt Vector".

(Error)
<table>
<tr><td>PPG2/3/12/13/22/23/32/33/42/43
16bit free-run timer 2 ("0" detection) / (compare clear)</td><td>41</td><td>29</td><td>ICR25</td><td>358_H</td><td>$000FFF58_H$</td><td>25*3</td></tr>
</table>
(Correct)
<table>
<tr><td>PPG2/3/12/13/22/23/32/33/43
16-bit free-run timer 2 ("0" detection) / (compare clear)</td><td>41</td><td>29</td><td>ICR25</td><td>358_H</td><td>$000FFF58_H$</td><td>25*3</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1873</td><td>Appendix B</td><td>The interrupt factor in Table B-5 Interrupt Vector MB91F52xK (144pin) in "B. List of Interrupt Vector" should be modified as follows:

(Error)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45*5</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>

(Correct)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>
</td></tr>
<tr><td>1873</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-5 Interrupt Vector MB91F52xK (144pin) in "B. List of Interrupt Vector".

(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
<tr><td>1877</td><td>Appendix B</td><td>The interrupt factor in Table B-6 Interrupt Vector MB91F52xL (176pin) in "B. List of Interrupt Vector" should be modified as follows:

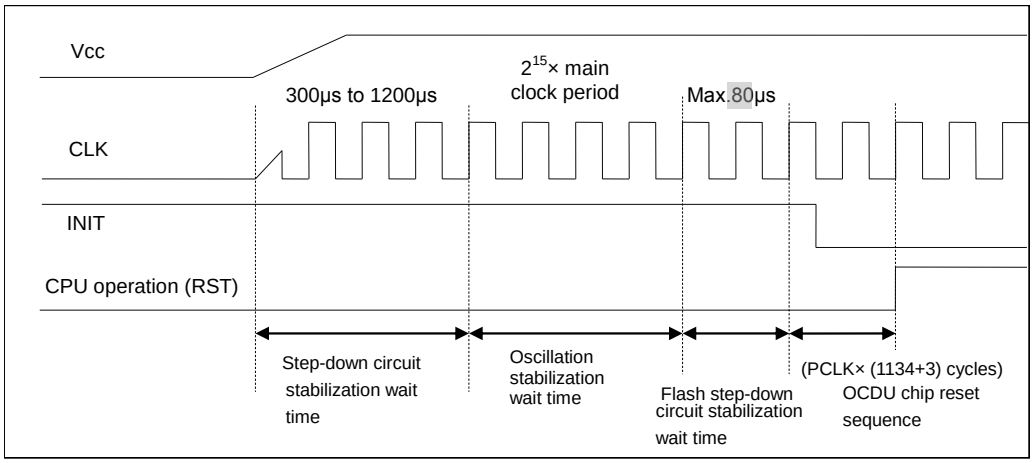
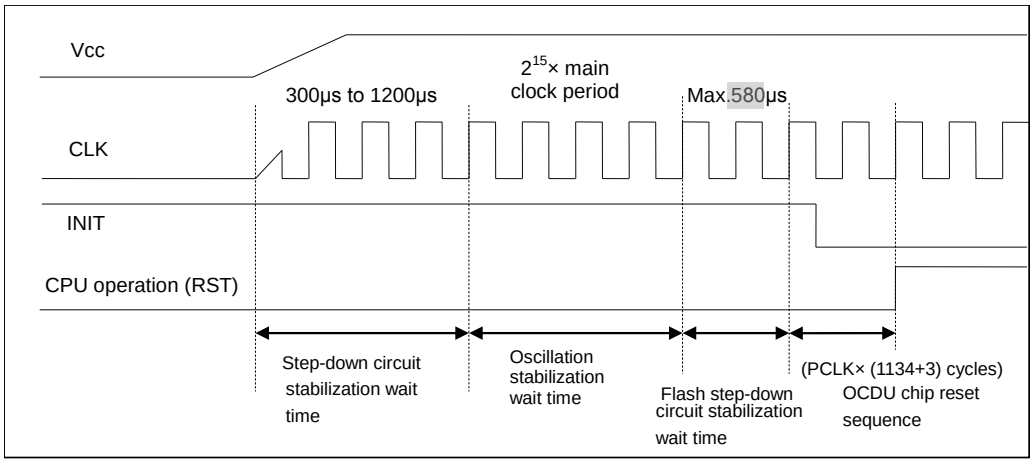
(Error)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45*5</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>

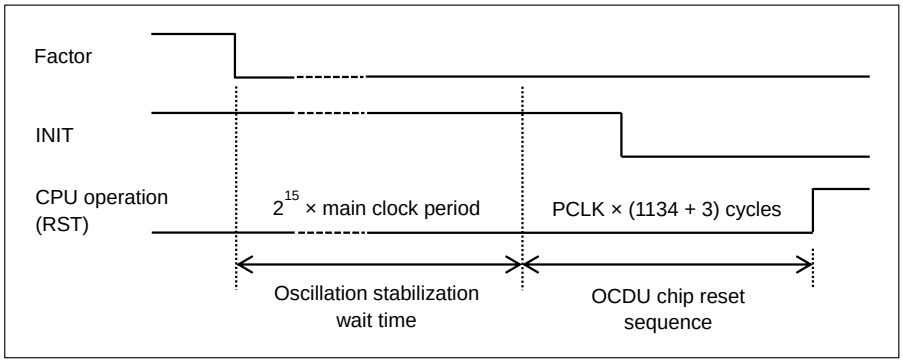
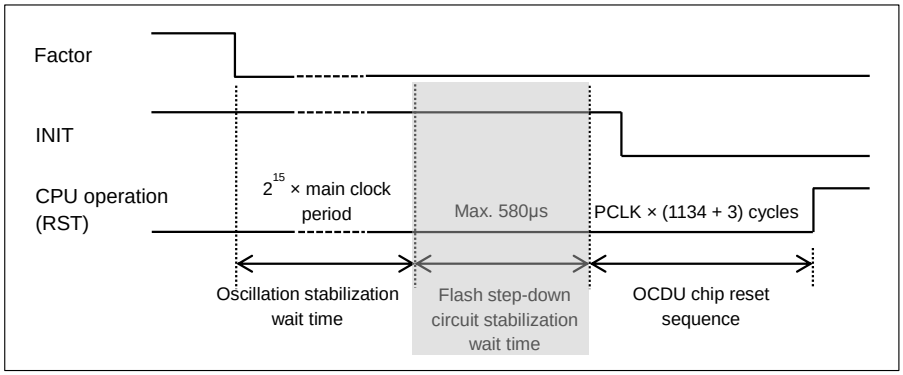
(Correct)
<table>
<tr><td>Base timer 1 IRQ0</td><td rowspan="4">61</td><td rowspan="4">3D</td><td rowspan="4">ICR45</td><td rowspan="4">308_H</td><td rowspan="4">$000FFF08_H$</td><td rowspan="4">45</td></tr>
<tr><td>Base timer 1 IRQ1</td></tr>
<tr><td>–</td></tr>
<tr><td>–</td></tr>
</table>
</td></tr>
<tr><td>1877</td><td>Appendix B</td><td>The following sentence should be deleted from Table B-6 Interrupt Vector MB91F52xL (176pin) in "B. List of Interrupt Vector".

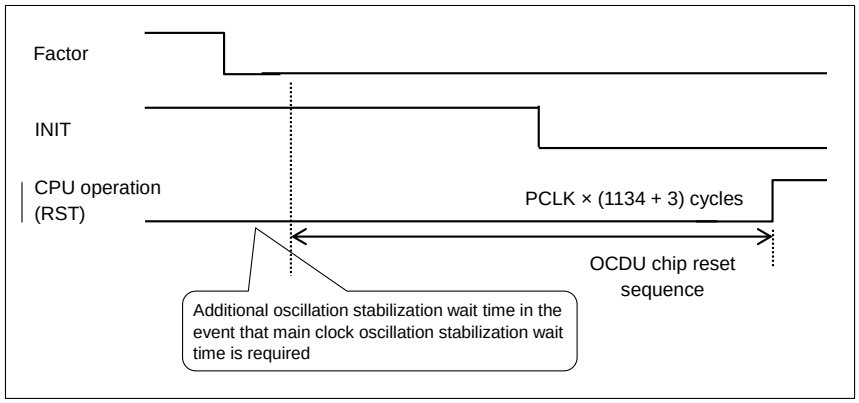
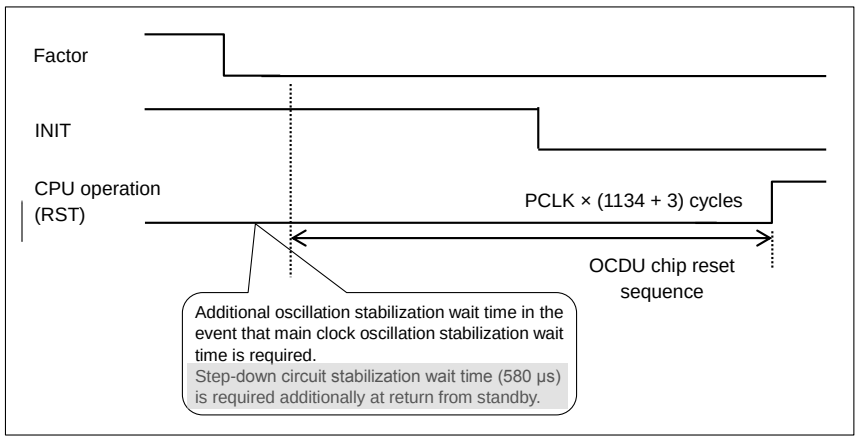
(Error)
*5: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.</td></tr>
</table>

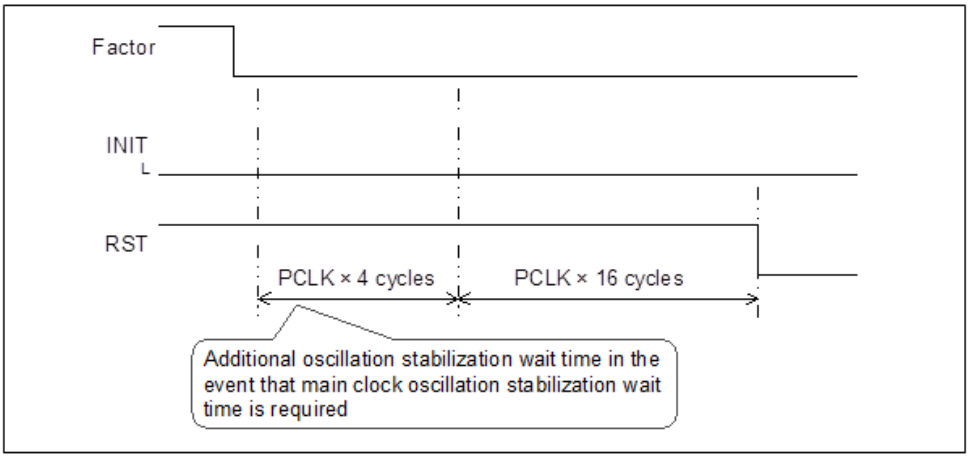
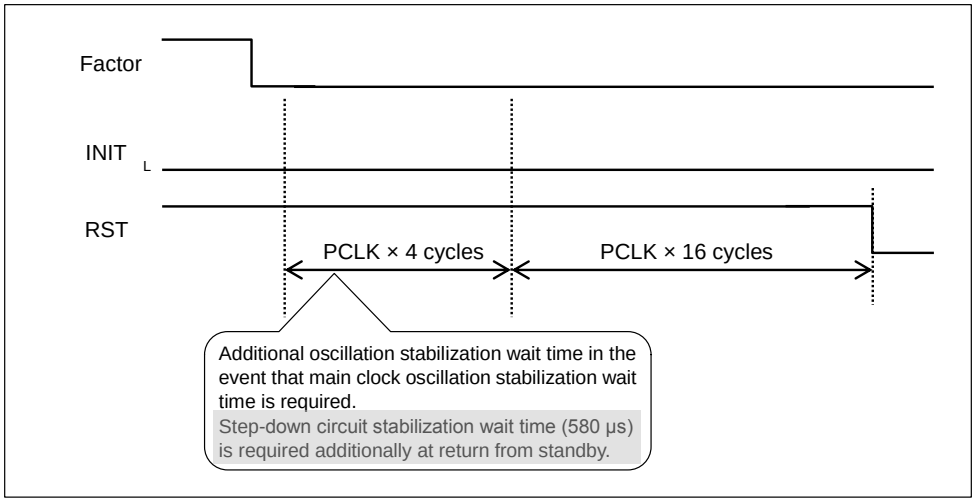
Page	Section	Change Results
1878 to 1881	Appendix C	The following parts in "APPENDIX C. Pins Statuses in State of CPU" should be modified. · Pin statuses before internal reset issuance of external factors after releases in the reset factor 2 (*2) should be changed from HiZ/input blocked to HiZ/input enabled. · Statuses (HiZ/input enabled) should be added in external factor after releases of the reset factor 3 (*3). (Error) Pin status after reset factor generation and after reset releasing reset factor2(*2) (*2 low-voltage detect(external power supply voltage)) / reset factor3(*3) external reset NMI disabled / internal reset factor4(*4) external factor generate / external factor After releases / external factor generate / external factor After releases Before internal reset issuance / while internal reset to After releases (Oscillation stabilization wait in progress) / Before internal reset issuance / Internal reset issuance in progress / while internal reset to After releases (Oscillation stabilization wait in progress) / After internal reset releasing / Before internal reset issuance / Internal reset issuance in progress (Oscillation stabilization wait in progress) / Before internal reset issuance / while internal reset to After releases (Oscillation stabilization wait in progress) / while internal reset to After releases(Oscillation stabilization wait in / After internal reset releasing - / - / - / - / - / - / - / - / - / - HiZ/input enabled / HiZ/input blocked / HiZ/input blocked / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input blocked HiZ/input enabled / HiZ/input blocked / HiZ/input blocked / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input blocked HiZ/input enabled / HiZ/input blocked / HiZ/input blocked / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input blocked HiZ/input enabled / HiZ/input blocked / HiZ/input blocked / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input enabled / HiZ/input blocked / HiZ/input blocked (Correct) Pin status after reset factor generation and after reset releasing Reset factor2(*2) (*2 Low-voltage detect (external power supply voltage)) / Reset factor3(*3) External reset NMI disabled / Internal reset factor4(*4) External factor generate / External factor After releases / External factor generate / External factor After releases Before internal reset issuance / Internal reset issuance in progress / Before internal reset issuance / While internal reset to after releases (oscillation stabilization wait in progress) / After internal reset releasing / Before internal reset issuance / Internal reset issuance in progress / Before internal reset issuance / While internal reset to after releases (oscillation stabilization wait in progress) / After internal reset releasing / While internal reset to After releases (oscillation stabilization wait in progress) / After internal reset releasing - / - / - / - / - / - / - / - / - / - / - / - HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/input blocked HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/input blocked HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/input blocked HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/ input enabled / HiZ/ input blocked / HiZ/ input enabled / HiZ/input blocked / HiZ/input blocked

Page	Section	Change Results
1881	Appendix C	The comment in "APPENDIX C. Pins Statuses in State of CPU" should be modified as follows: (Error) (*4) factor: software reset, soft/hardware watchdog reset, return from stand-by (PMUCTLR:IOCTMD=1) (including time out) (*14) factor: return from stand-by (power supply blocked) (PMUCTLR:IOCTMD=0) (Correct) (*4) factor: software reset, soft/hardware watchdog reset, return from stand-by where the I/O state is maintained until returning from the standby mode (including time out) (*14) factor: return from stand-by (power supply blocked) where the I/O state is maintained until the I/O maintain cancellation request has been cleared.
-	-	Company name and layout design change.
Revision 5.0		
6	1.2.2	The description in multi-function serial communication (built-in transmission/reception FIFO memory) in "2.2. Peripheral Functions" should be modified as follows: (Error) 5V tolerant input 4 channels (ch.6, ch.8, ch.9, ch.11) Automotive input (Correct) 5V tolerant input 4 channels (ch.6, ch.8, ch.9, ch.11) CMOS hysteresis input
37 to 43	1.8	The following sentences should be deleted from "1.8. Device Package". (Error) Please confirm the latest Package dimension by following URL. http://edevice.fujitsu.com/package/en-search/ (Correct) Deleted.
255	7.4.1	The following sentence should be added under Attribute in "4.1. Reset Source Register : RSTRR (ReSeT Result Register)": (Error) *: Due to a reset factor. (Correct) *: Due to a reset factor. *: These bits other than IRRST bit are undefined at power-on reset.

Page	Section	Change Results
282	7.5.4.1	The note in "5.4.1. Super Initialize Reset (SINIT)" should be modified as follows: (Error) These stabilization wait time (300µs to 1200µs and maximum 80µs) are needed at power-on reset. (Correct) These stabilization wait time (300µs to 1200µs and maximum 580µs) are needed at power-on reset.
282	7.5.4.1	"Figure 5-1 Oscillation Stabilization Wait Time for Power-on Reset" in "5.4.1. Super Initialize Reset (SINIT)" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
283	7.5.4.1	"Figure 5-2 Super Initialize Reset (SINIT) Sequence" in "5.4.1. Super Initialize Reset (SINIT)" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
284	7.5.4.2	"Figure 5-3 Initialize Reset (INIT) Sequence" in "5.4.2. Initialize Reset (INIT)" should be modified as follows: (Error)  (Correct) 

Page	Section	Change Results
285	7.5.4.3	"Figure 5-4 Reset (RST) Sequence" in "5.4.3. Reset (RST)" should be modified as follows: (Error)  (Correct) 
640	19.2.1	The description in ● Count clock in "2.1. 16/32-bit Reload Timer" should be modified as follows: (Error) You can select one of five internal (peripheral) clocks and three external clocks (ECK signals). (Correct) You can select one of eight internal (peripheral) clocks and three external clocks (ECK signals).

Page	Section	Change Results
641	19.2.2	The description in ● Count clock in "2.2. 16-bit PWM Timer" should be modified as follows: (Error) You can select one of five internal (peripheral) clocks and three external clocks (ECK signals). (Correct) You can select one of eight internal (peripheral) clocks and three external clocks (ECK signals).
642	19.2.3	The description in ● Count clock in "2.3. 16/32-bit PWC Timer" should be modified as follows: (Error) You can select one of the internal (peripheral) clocks obtained by dividing the frequency of the peripheral clock (PCLK) by five types. (Correct) You can select one of the internal (peripheral) clocks obtained by dividing the frequency of the peripheral clock (PCLK) by eight types.
643	19.2.4	The description in ● Count clock in "2.4. 16-bit PPG Timer" should be modified as follows: (Error) You can select one of five internal (peripheral) clocks and three external clocks (ECK signals). (Correct) You can select one of eight internal (peripheral) clocks and three external clocks (ECK signals).

Page	Section	Change Results
1601	45.5.3.1	A note should be added in ■ Sector Erase Suspend Command in "5.3.1. Command Sequence". (Error) Note: · 16.7µs + 2cyc 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. (Correct) Note: · 16.7µs + 2cyc 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. · At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed. · 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.
1610	45.5.8	A note should be added in "5.8. Sector Erase Suspend Command ". (Error) Note: It takes up to 16.7µs from the time the sector erase suspend command is issued until the time when the sector erase operation is stopped and it becomes possible to read out from sectors that are not the erase target. (Correct) Note: · It takes up to 16.7µs + 2cyc from the time the sector erase suspend command is issued until the time when the sector erase operation is stopped and it becomes possible to read out from sectors that are not the erase target. · At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.

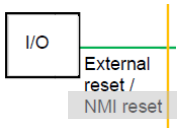
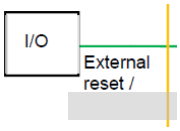
Page	Section	Change Results
1626	46.5.3.1	A note should be added in ■ Sector Erase Suspend Command in "5.3.1. Command Sequence". (Error) Note: · 16.7µs + 2cyc or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector 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. (Correct) Note: · 16.7µs + 2cyc 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. · At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed. · 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.
1635	46.5.8	A note should be added in "5.8. Sector Erase Suspend Command ". (Error) Note: 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. (Correct) Note: · 16.7µs + 2cyc 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. · At least 2ms of wait time is required after sector erase is restarted and before sector erase suspend command is executed.

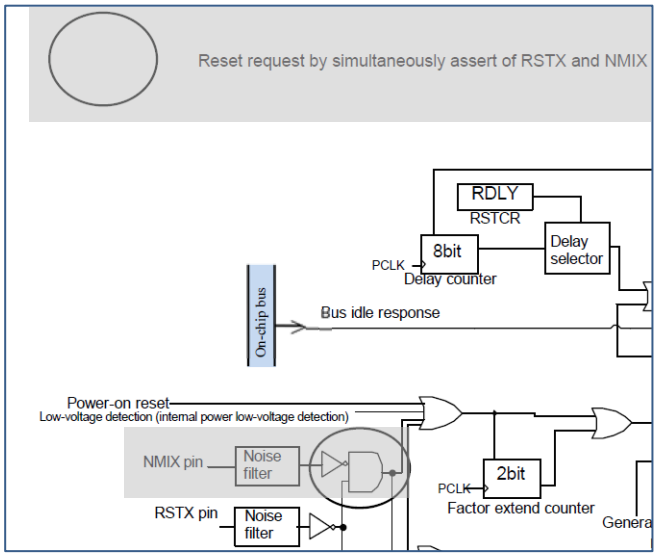
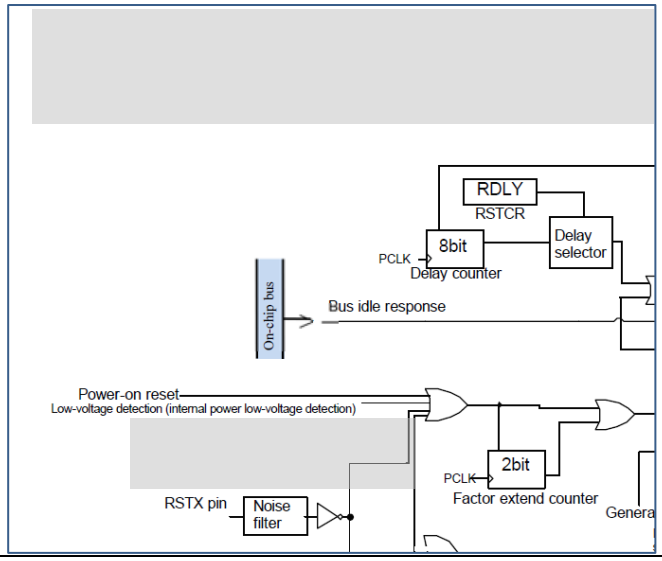
Page	Section	Change Results
1740	50.4.5	The note in "4.5. TEST End Address Register XBS RAM : TAEARX" should be modified as follows: (Error) (Absolute address) = (0001_0000_H) + (Offset address set with TAEARX + 2'b00) (Correct) (Absolute address) = (0001_0000_H) + (Offset address set with TAEARX + 2'b11)
1753	50.4.15	The note in "4.15. TEST End Address Register BACKUP-RAM : TAEARA" should be modified as follows: (Error) (Absolute address) = (0000_4000_H) + (Offset address set with TAEARA + 2'b00) (Correct) (Absolute address) = (0000_4000_H) + (Offset address set with TAEARA + 2'b11)
Cypress Document Number: 002-05528 Rev.*A		
-	-	Company name and layout design change
45	1.2.2	The following sentence should be modified as follows: (Error) Conversion time : 1µs (Correct) Conversion time : 1.4µs
45	1.2.2	The following sentence should be modified as follows: (Error) < I2C > ・2 channels (ch.3, ch.4) Standard mode / High-speed mode supported ・6 channels (ch.5 to ch.8, ch.10, ch.11) Standard mode supported ・Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory ・Standard mode (Max. 100kbps) / High-speed mode (Max. 400kbps) supported (Correct) < I2C > ・2 channels (ch.3, ch.4) Standard mode / Fast mode supported ・6 channels (ch.5 to ch.8, ch.10, ch.11) Standard mode supported ・Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory ・Standard mode (Max. 100kbps) / Fast mode (Max. 400kbps) supported
48	1.3	Changed the package type to JEDEC type in Table 3-1. (Error) LQFP-64 (Correct) LQD064

Page	Section	Change Results
48,50,52, 54,56,58	1.3	The following sentence should be added as follows: *2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD.
50	1.3	Changed the package type to JEDEC type in Table 3-2. (Error) LQFP-80 (Correct) LQH080
52	1.3	Changed the package type to JEDEC type in Table 3-3. (Error) LQFP-100 (Correct) LQI100
54	1.3	Changed the package type to JEDEC type in Table 3-4. (Error) LQFP-120 (Correct) LQM120
54, 56, 58	1.3	The following part should be modified as follows: (Error) *1: Only channel 3 and channel 4 support the I^2C (high-speed mode/standard mode). (Corret) *1: Only channel 3 and channel 4 support the I^2C (fast mode/standard mode).
56	1.3	Changed the package type to JEDEC type in Table 3-5. (Error) LQFP-144 (Correct) LQS144 / LQN144
58	1.3	Changed the package type to JEDEC type in Table 3-6. (Error) LQFP-176 (Correct) LQP176
59	1.3	Added Revision C, D, E in Table 3-7 Revision:B, C, D, E
61	1.4	The following sentence should be modified as follows: (Error) Conversion time : 12-bit A/D Converter 1µs (Correct) Conversion time : 12-bit A/D Converter 1.4µs
64, 65	1.5	The following part should be modified as follows in Figure 5-1, 5-2 Block Diagram: (Error) CAN 2ch (Correct) CAN 3ch
72	1.7	Changed the package type to JEDEC type in Figure 7-1. (Error) LQFP-64 (Correct) LQD064

Page	Section	Change Results
73	1.7	Changed the package type to JEDEC type in Figure 7-2. (Error) LQFP-80 (Correct) LQH080
73	1.7	The following sentence should be modified as follows in Figure 7-2. (Error) * In a single clock product, pin 71 and pin 72 are the general-purpose ports. (Correct) * In a single clock product, pin 70 and pin 71 are the general-purpose ports.
74	1.7	Changed the package type to JEDEC type in Figure 7-3. (Error) LQFP-100 (Correct) LQI100
75	1.7	Changed the package type to JEDEC type in Figure 7-4. (Error) LQFP-120 (Correct) LQM120
76	1.7	Changed the package type to JEDEC type in Figure 7-5. (Error) LQFP-144 (Correct) LQS144 / LQN144
77	1.7	Changed the package type to JEDEC type in Figure 7-6. (Error) LQFP-176 (Correct) LQP176
78 to 84	1.8	Changed the package type to JEDEC type
106	1.9.1	The following sentence should be modified as follows in TRG8_0 of 100pin: (Error) - (Correct) 23
125	2	The following sentence should be modified as follows: (Error) To prevent a malfunction of the voltage step-down circuit built in the device, set the voltage rise time to have 50µs or longer (between 0.2V and 2.7V) during power-on. (Correct) To prevent a malfunction of the voltage step-down circuit built in the device, the voltage rising must be monotonic increasing during power-on.
161	4.5.1	Changed the package type to JEDEC type in Table 5-2. (Error) LQFP-176 LQFP-144 LQFP-120 LQFP-100 LQFP-80 LQFP-64 (Correct) LQP176 LQS144 / LQN144 LQM120 LQI100 LQH080 LQD064

Page	Section	Change Results
194	5.4.9	The following sentence should be added in Note as follows: The period of the failure detection cycle is as follow. 2^{12} ×CR Oscillation time = approx. 40.96 ms
208	5.4.21	The following sentence should be added as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] The CSC bit and the CST bit are 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 and the CST bit are not initialized. Initialize these bits 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. [MB91F52xxxD] The CSC bit and the CST bit are 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 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 and the CST bit are not initialized. Initialize these bits 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.
218	5.5.2.3	The following sentences should be modified as folloing adjust the alignment: (Error) Display of the clock oscillation stabilization wait status and the oscillation stabilization status ・Main clock : CMONR:MCRDY ="0", CMONR:MCRDY ="1" ・PLL/SSCG clock (PLLSSCLK) : CMONR:PCRDY ="0", CMONR:PCRDY ="1" ・Sub clock (SBCLK) : CMONR:SCRDY ="0", CMONR:SCRDY ="1" (Correct) Display of the clock oscillation stabilization wait status Display of 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"

Page	Section	Change Results
233, 234	6.2.1	Figure 2-1 was modified as a diagram for MB91F52xxxB/MB91F52xxxC/MB91F52xxxE. Figure 2-2 was added as a diagram for MB91F52xxxD. Figure 2-1 and Figure 2-2 are same and different in the following sentences. Figure 2-1 Diagram of Device State Transitions [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] ① Power-on reset or internal low-voltage detection or external reset and simultaneous assertion of NMI ② Power-on reset release and internal low-voltage release and external reset and release simultaneous assertion of NMI ⑨ External reset input (NMI disabled) or external low-voltage detection ⑩ External reset input (NMI disabled + irregular) or external low-voltage detection (irregular) Figure 2-2 Diagram of Device State Transitions [MB91F52xxxD] ① Power-on reset or internal low-voltage detection or external reset ② Power-on reset release and internal low-voltage release and external reset release ⑨ External low-voltage detection ⑩ External low-voltage detection (irregular)
235	6.2.1	The following sentence should be modified as follows: (Error) See "Limitations of the standby control power shutdown/usually" in "CHAPTER : POWER CONSUMPTION CONTROL" for detail. (Correct) See "Restrictions on Power-Shutdown and Normal Standby Control" in "CHAPTER : POWER CONSUMPTION CONTROL" for detail.
241, 242	7.3	Figure 3-1 was modified as a diagram for MB91F52xxxB/MB91F52xxxC/MB91F52xxxE. Figure 3-2 was added as a diagram for MB91F52xxxD. (NMI reset deletes in Figure 3-2 as following) Figure 3-1 Configuration Diagram of Reset [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]  Figure 3-2 Configuration Diagram of Reset [MB91F52xxxD] 

Page	Section	Change Results
243, 244	7.3	Figure 3-2 was modified as a Figure 3-3 for MB91F52xxxB/MB91F52xxxC/MB91F52xxxE. Figure 3-4 was added as a diagram for MB91F52xxxD. (NMIX pin and Reset request by simultaneously of RSTX and NMIX deletes as follows.) Figure 3-3 Configuration Diagram of Reset (Reset Control) [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]  Figure 3-4 Configuration Diagram of Reset (Reset Control) [MB91F52xxxD] 
245	7.3	Figure 3-3 is modified as Figure 3-5. Figure 3-5 Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>246</td><td>7.4.1</td><td>The following sentence should be modified as follows:

[bit7] IRRST (IRregular ReSeT) : Irregular reset
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit indicates that any of power-on reset, internal low-voltage detection, reset timeout, or simultaneous assertion 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.

[MB91F52xxxD] This bit indicates that any of power-on reset, internal low-voltage detection, reset timeout, or assertion of RSTX 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.</td></tr>
<tr><td>246, 247</td><td>7.4.1</td><td>The following sentence should be modified as follows:
[bit6] ERST (External ReSeT)
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Reset pin input, illegal standby mode transition detection, external low-voltage detection, clock supervisor reset, simultaneous assertion 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 assertion of RSTX and NMIX external pins.
<table>
<tr><th>ERST</th><th>RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset, low-voltage detection (external low-voltage detection) or simultaneous assertion of RSTX and NMIX external pins</th></tr>
<tr><td>0</td><td>Undetected</td></tr>
<tr><td>1</td><td>Detected</td></tr>
</table>
[MB91F52xxxD] Reset pin input, illegal standby mode transition detection, external low-voltage detection, clock supervisor reset.
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.
<table>
<tr><th>ERST</th><th>RSTX pin reset detection, illegal standby mode transition detection, clock supervisor reset, low-voltage detection (external low-voltage detection)</th></tr>
<tr><td>0</td><td>Undetected</td></tr>
<tr><td>1</td><td>Detected</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>251</td><td>7.4.4</td><td>The following sentence should be modified as follows:
[bit7] PMUST (Power Management Unit STatus)
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit is initialized only by power-on reset, internal low-voltage detection reset, and simultaneous assertion of RSTX and NMIX. So, check other reset factors before checking the recovery from shutdown using this bit.

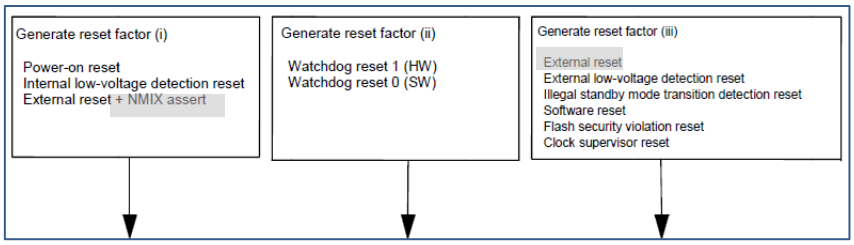
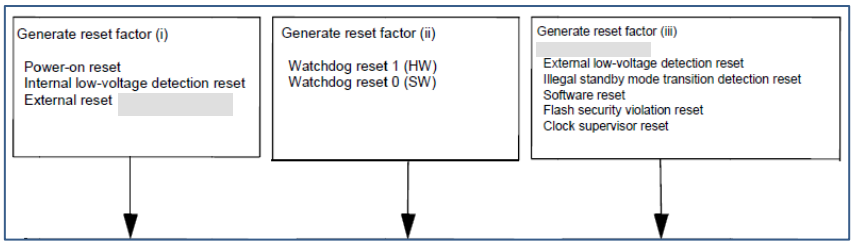
[MB91F52xxxD] This bit is initialized only by power-on reset, internal low-voltage detection reset, and assertion of RSTX. So, check other reset factors before checking the recovery from shutdown using this bit.</td></tr>
<tr><td>253</td><td>7.5.1.1</td><td>The following sentence should be modified as follows:
(Error)
It initializes all register settings and the entire hardware.

(Correct)
It initializes the CPU and all registers except the ones initialized only by the power-on reset or super initialize reset (SINIT) and registers with undefined initial value..</td></tr>
<tr><td>253</td><td>7.5.1.2</td><td>The following sentence should be modified as follows:
(Error)
It initializes the entire hardware and all registers except the ones initialized only by the initialize reset (INIT).

(Correct)
It initializes the CPU and all registers except the ones initialized only by the power-on reset, SINIT or INIT and registers with undefined initial value..</td></tr>
<tr><td>254</td><td>7.5.2.2</td><td>The following sentence should be modified as follows:
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Reset by this reset factor is detected as irregular reset only at the reset timeout or simultaneous assertion of the NMIX pin.
Other than the irregular reset detection, a reset (RST) will be issued.

[MB91F52xxxD] Reset by this reset factor is detected as irregular reset.</td></tr>
<tr><td>258</td><td>7.5.4.1</td><td>The following sentence should be modified as follows:
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] The super initialize reset (SINIT) will be issued first for power-on reset, internal low-voltage detection, or simultaneous assertion of RSTX and NMIX.

[MB91F52xxxD] The super initialize reset (SINIT) will be issued first for power-on reset, internal low-voltage detection, or assertion of RSTX.</td></tr>
<tr><td>258</td><td>7.5.4.1</td><td>The following sentence should be modified in Table 5-1 as follows:
Table 5-1 Oscillation Stabilization Wait Time (SINIT)
<table>
<tr><th>Type</th><th>Main clock oscillation stabilization wait time</th></tr>
<tr><td>Power-on reset</td><td>2^{15}× Main clock cycle</td></tr>
<tr><td>Internal low-voltage detection</td><td>2^{15}× Main clock cycle</td></tr>
<tr><td>[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Simultaneous assertion of RSTX and NMIX</td><td>2^{15}× Main clock cycle</td></tr>
<tr><td>[MB91F52xxxD] Assertion of RSTX</td><td>2^{15}× Main clock cycle</td></tr>
</table></td></tr>
</table>

Page	Section	Change Results
261	7.5.4.3	The following sentence should be added as follows: (Error) 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 "5.1.1. Initialize Reset (INIT)"). (Correct) If a reset factor that is not the SINIT or INIT level occurs, only a reset (RST) will be issued. This reset is used for initializing the CPU and all registers except some registers (see "5.1.1. Initialize Reset (INIT)").
262, 263	7.5.5	Figure 5-5 was modified as a diagram for MB91F52xxxB/MB91F52xxxC/MB91F52xxxE. Figure 5-6 was added as a diagram for MB91F52xxxD. (From Figure 5-5 to Figure 5-6, the following parts were deleted.) Figure 5-5 Reset Sequence [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]  Figure 5-6 Reset Sequence [MB91F52xxxD] 
265	7.5.5.6	The following sentence should be deleted: 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).
265	7.5.6	The following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] During return form standby watch mode (power-shutdown) and standby stop mode (power-shutdown), an internal reset is issued. Therefore any reset factor without power-on reset, internal low-voltage detection reset, reset by simultaneous assertion of RSTX and NMIX will not be accepted. [MB91F52xxxD] During return form standby watch mode (power-shutdown) and standby stop mode (power-shutdown), an internal reset is issued. Therefore any reset factor without power-on reset, internal low-voltage detection reset, reset by assertion of RSTX will not be accepted.

Page	Section	Change Results
344	11.4.3	The following sentence should be modified as follows: (Error) Initial value of the each bits can be referred "B. I/O Map" in "APPENDIX". (Correct) Initial value of the each bits can be referred "I/O Map" in "APPENDIX".
407	13.8	The following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] 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 causes (power-on reset, internal low-voltage detection, and simultaneous assertion of RSTX and NMIX) are recognized. At this time, the register of the external interrupt input is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the external interrupt input register before using it. [MB91F52xxxD] 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 causes (power-on reset, internal low-voltage detection, and assertion of RSTX) are recognized. At this time, the register of the external interrupt input is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the external interrupt input register before using it.
416	16.1	The following sentence should be modified as follows: (Error) FR80 (Correct) FR81
463	17.4.5	For [bit2] REMP, the following sentence should be modified as follows: (Error) The read value of this bit is always "0". This bit must always be written to "0". (Correct) The read value of this bit is always "1".
463	17.4.5	For [bit1] SREMP, the following sentence should be modified as follows: (Error) The read value of this bit is always "0". This bit must always be written to "0". (Correct) The read value of this bit is always "1".
722	23.5.3	The following sentence should be modified as follows: (Error) The edge of the external pin input is detected, and the cycle (rising or falling) and the pulse width (H or L) are measured with the counter. (Correct) The edge of the external pin input is detected, and the cycle (rising or falling) and the pulse width (H or L) are measured with a counter clocked by the peripheral clock PCLK2.

Page	Section	Change Results
868	28.9	The following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] The internal reset is issued at the return from the standby watch mode (power-shutdown). Therefore, only the reset causes (power-on reset, internal low-voltage reset, and simultaneous assertion of RSTX and NMIX) are recognized. At this time, the register of the RTC is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after wake-up, user needs to initialize the register of RTC before using it. [MB91F52xxxD] The internal reset is issued at the return from the standby watch mode (power-shutdown). Therefore, only the reset causes (power-on reset, internal low-voltage reset, and assertion of RSTX) are recognized. At this time, the register of the RTC is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after wake-up, user needs to initialize the register of RTC before using it.
884	30.4	The following sentence should be modified as follows: 3. [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Simultaneous assertion of RSTX and NMIX external pins [MB91F52xxxD] Assertion of RSTX external pin
885	30.4.2	For PMUCTLR, the following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset. [MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.
887	30.4.3	For PWRTMCTL, the following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset. [MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.
888	30.4.4	For PMUINTF0, the following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset. [MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.
889	30.4.5	For PMUINTF1, the following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset. [MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.

Page	Section	Change Results
890	30.4.6	For PMUINTF2, the following sentence should be modified as follows: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assertion of RSTX and NMIX, and hardware watchdog timer reset. [MB91F52xxxD] This register will be initialized by power-on reset, internal low-voltage reset, reset by assertion of RSTX, and hardware watchdog timer reset.
900	30.5.5.3	The following sentence should be modified as follows in Wake up from the watch mode with power-shutdown: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Only the reset factors (power-on reset, internal low-voltage reset, and simultaneous assertion of RSTX and NMIX) are accepted during wake-up. At this time, the register of the RTC and external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the RTC/external interrupt input register before using it. [MB91F52xxxD] Only the reset factors (power-on reset, internal low-voltage reset, and assertion of RSTX) are accepted during wake-up. At this time, the register of the RTC and external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the RTC/external interrupt input register before using it.
907	30.5.7.3	The following sentence should be modified as follows in Wake up from stop mode with power-shutdown: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] Only the reset factors (power-on reset, internal low-voltage reset and simultaneous assertion 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 flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the register before using it. [MB91F52xxxD] Only the reset factors (power-on reset, internal low-voltage reset, and assertion of RSTX) are accepted during wake-up. At this time, the register of the external interrupt input (IOCTMD=1) is not initialized. If the flag for RSTX reset or the flag for the external low-voltage detection reset is set after the start-up, the user needs to initialize the register before using it.
915	31.1	The following sentence should be added as follows: LVDV*: 0.9V ± 0.1V *: The detection voltage of the internal low voltage detection is 0.9V±0.1V. This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>917</td><td>31.4.1</td><td>The following table should be modified as follows:

(Error)
<table><tr><th>LVD_SEL[2:0]</th><th>Setting for detection level of fall of internal power supply voltage</th></tr><tr><td>100</td><td>0.9V ± 0.1V</td></tr><tr><td>Other than those above</td><td>Setting is prohibited</td></tr></table>(Note): These bits can be rewritten only when LVD_OE="1".

(Correct)
<table><tr><th>LVD_SEL[2:0]</th><th>Setting for detection level of fall of internal power supply voltage</th><th>Guaranteed MCU operation voltage range</th></tr><tr><td>100 *</td><td>0.9V ± 0.1V</td><td>No</td></tr><tr><td>Other than those above</td><td>Setting is prohibited</td><td>-</td></tr></table>(Note): These bits can be rewritten only when LVD_OE="1".

*: This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD.</td></tr>
<tr><td>919</td><td>31.6</td><td>The following sentence should be deleted as follows in Initial value of internal low-voltage detection flag (LVD:LVD_F):
(Error)
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).

(Correct)
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).</td></tr>
<tr><td>921</td><td>32.1</td><td>The following sentence should be added as follows:
(Note): Rising LVDV: 2.3V
Falling LVDV: 2.8 to 4.3V (11 steps) variable*
*: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage (2.7V). Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD.</td></tr>
<tr><td>923</td><td>32.4.1</td><td>The following sentence should be modified as follows in [bit0] LVD5R_F (Low-Voltage Detect external 5v Rise Flag):
(Error)
If a rise in external voltage is detected, the LVD5R_F bit is set to "1".

(Correct)
If a power-on reset is detected, the LVD5R_F bit is set to "1".</td></tr>
</table>

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>924, 925</td><td>32.4.2</td><td>The following table should be modified as follows:

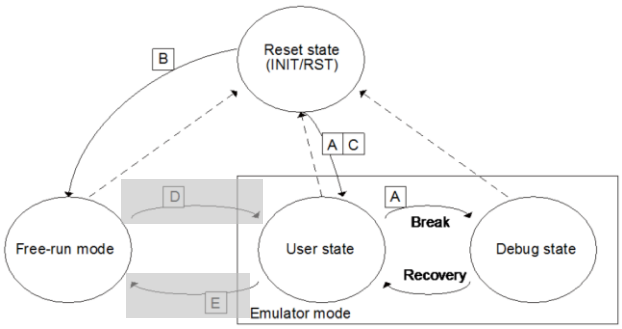
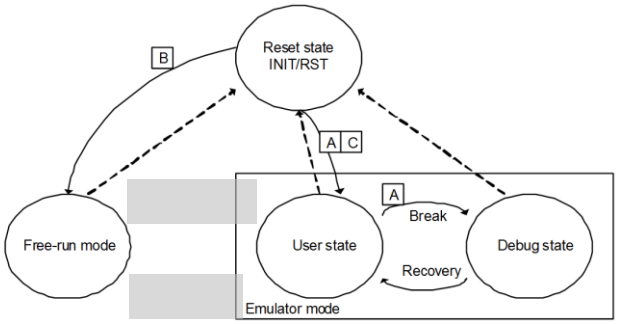
(Error)
<table>
<tr><th>LVD5F_SEL[3:0]</th><th>External power supply fall detection voltage setting</th></tr>
<tr><td>0000</td><td>2.80V ±8%</td></tr>
<tr><td>0001</td><td>3.00V ±8%</td></tr>
<tr><td>0010</td><td>3.20V ±8%</td></tr>
<tr><td>0011</td><td>3.60V ±8%</td></tr>
<tr><td>0100</td><td>3.70V ±8%</td></tr>
<tr><td>0101</td><td>3.80V ±8%</td></tr>
<tr><td>0110</td><td>3.90V ±8%</td></tr>
<tr><td>0111</td><td>4.00V ±8%</td></tr>
<tr><td>1000</td><td>4.10V ±8%</td></tr>
<tr><td>1001</td><td>4.20V ±8%</td></tr>
<tr><td>1010</td><td>4.30V ±8%</td></tr>
<tr><td>others</td><td>Setting prohibited</td></tr>
</table>
(Note): LVD5F_SEL[3:0] bits can be rewritten only when LVD5F_OE = "1".

(Crrect)
<table>
<tr><th>LVD5F_SEL[3:0]</th><th>External power supply fall detection voltage setting</th><th>Guaranteed MCU operation voltage range</th></tr>
<tr><td>0000 *</td><td>2.80V ±8%</td><td>No</td></tr>
<tr><td>0001</td><td>3.00V ±8%</td><td rowspan="12">Yes</td></tr>
<tr><td>0010</td><td>3.20V ±8%</td></tr>
<tr><td>0011</td><td>3.60V ±8%</td></tr>
<tr><td>0100</td><td>3.70V ±8%</td></tr>
<tr><td>0101</td><td>3.80V ±8%</td></tr>
<tr><td>0110</td><td>3.90V ±8%</td></tr>
<tr><td>0111</td><td>4.00V ±8%</td></tr>
<tr><td>1000</td><td>4.10V ±8%</td></tr>
<tr><td>1001</td><td>4.20V ±8%</td></tr>
<tr><td>1010</td><td>4.30V ±8%</td></tr>
<tr><td>others</td><td>Setting prohibited</td><td>-</td></tr>
</table>
(Note): LVD5F_SEL[3:0] bits can be rewritten only when LVD5F_OE = "1".
*: This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level (2.8V±8% = 2.576V to 3.024V) is below the minimum guaranteed MCU operation voltage (2.7V). Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD.</td></tr>
<tr><td>937</td><td>34.3.1</td><td>The following sentence should be modified as follows in [bit7] SCKS (Sub ClocK mode Select):
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] This bit will be cleared to "0" on power-on, external reset, or simultaneous assertion with NMIX. Other kind of reset does not affect this bit.

[MB91F52xxxD] This bit will be cleared to "0" on power-on, or external reset. Other kind of reset does not affect this bit.</td></tr>
</table>

Page	Section	Change Results
944	34.4.7	The following sentence should be modified as follows in Checking the Reset Factor Using the Clock Supervisor: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] 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 assertion of RSTX and NMIX external pins was generated. [MB91F52xxxD] 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, or clock supervisor reset was generated.
1093	40.4.3.4	The following sentence should be modified as follows in TDR1n-0n(n=0 to 11): (Error) TDR1n-0n(n=0 to 11) : Address Base addr + 04_H (Access: Byte, Half-word, Word) (Correct) TDR1n-0n(n=0 to 11) : Address Base addr + 04_H (Access: Half-word, Word)
1489	44.2.3	The following sentence should be modified as follows: (Error) The conversion time is at least 1.0 µs (including sampling time). (Correct) The conversion time is at least 1.4 µs (including sampling time).
1502	44.4.1.1	For ADERL0, the following sentence should be modified as follows: (Error) For pins with the A/D converter function, see Chapter-11 "5.6. Notes on Pins with the A/D pins converter Function". (Correct) For pins with the A/D converter function, see "5.6. Notes on Pins with the A/D pins converter Function" in "CHAPTER: I/O PORTS".
1503	44.4.1.1	For ADERL1, the following sentence should be modified as follows: (Error) For pins with the A/D converter function, see Chapter-11 "5.6. Notes on Pins with the A/D pins converter Function". (Correct) For pins with the A/D converter function, see "5.6. Notes on Pins with the A/D pins converter Function" in "CHAPTER: I/O PORTS".
1573	44.5.4.7	The following sentence should be added in the note of the sampling time: Set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 47) bits to become 700ns(4.5V to 5.5V)/1000ns(2.7V to 5.5V) or more at the sampling time. When the sampling time is less than 700ns/1000ns, a normal value of the analog conversion value might not be obtained.

Page	Section	Change Results
1573	44.5.4.7	The following sentence should be added in the note of the compare time: Set the CT1 and CT0 bits to become 700ns or more at the comparison time. When the compare time is less than 700ns, a normal value of the analog conversion value might not be obtained.
1575	44.6	The following sentence should be modified as follows: (Error) Please set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 47) bits to become 700ns(4.5V to 5.5V)/1000ns(2.7V to 5.5V) or more at the sampling time. When the sampling time is set to 700ns/1000ns or less, a normal value of the analog conversion value might not be obtained. Please set the CT1 and CT0 bits to become 700ns or more at the comparison time. When the compare time is set to 700ns or less, a normal value of the analog conversion value might not be obtained. (Correct) Set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 47) bits to become 700ns(4.5V to 5.5V)/1000ns(2.7V to 5.5V) or more at the sampling time. When the sampling time is less than 700ns/1000ns, a normal value of the analog conversion value might not be obtained. Set the CT1 and CT0 bits to become 700ns or more at the comparison time. When the compare time is less than 700ns, a normal value of the analog conversion value might not be obtained.
1649	47.5.1.2	The following sentence should be deleted as follows: (Error) 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.

Page	Section	Change Results
1650	47.5.1.2	In Figure 5 1 OCDU Operating Mode Transition Diagram, the following sentence should be modified as follows: (Error)  (Correct) 
1650	47.5.1.2	In Figure 5 1 OCDU Operating Mode Transition Diagram, the following sentence should be modified as follows: (Error) [Transition indicated by solid lines] 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
1653	47.5.2.2	The following sentence should be modified as follows: (Error) to the E_SLPR register for the number of times and specified length. (Correct) to the dedicated OCD register for the number of times and specified length.

<table>
<tr><th>Page</th><th>Section</th><th>Change Results</th></tr>
<tr><td>1657,
1659</td><td>47.5.3.4</td><td>The following sentence should be modified as follows in Summary of Specification Restrictions table 1) 2):

<table>
<tr><td>[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
RSTX pin input
(irregular)</td></tr>
<tr><td>[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
RSTX pin input</td></tr>
<tr><td>[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
RSTX pin input (+NMIX pin input)

[MB91F52xxxD]
RSTX pin input</td></tr>
</table>
</td></tr>
<tr><td>1661</td><td>47.5.4</td><td>The following sentence should be added in Table 5-1 as follows:

<table>
<tr><td>Device version ID</td><td>4</td><td>E_IDVCR</td><td>0x003</td><td>0x2
0x3
0x4
0x5</td><td>Revision : B
Revision : C
Revision : D
Revision : E</td></tr>
</table>
</td></tr>
<tr><td>1878 to 1881</td><td>Appendix.C</td><td>The following sentence should be modified as follows in Reset factor1(*1) of Table C-1 : Pin Statuses:
Reset factor1(*1)
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
(*1 Power-on reset, external reset NMI enable)
[MB91F52xxxD]
(*1 Power-on reset, external reset)</td></tr>
<tr><td>1878 to 1881</td><td>Appendix.C</td><td>The following sentence should be modified as follows in Reset factor3(*3) of Table C-1 : Pin Statuses:
Reset factor3(*3)
[MB91F52xxxB/MB91F52xxxC/MB91F52xxxE]
External reset NMI disabled</td></tr>
<tr><td>1881</td><td>Appendix.C</td><td>The following sentence should be modified as follows in the bottom of Table C-1 : Pin Statuses:
(*1) Factor: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] power-on reset, low-voltage detect (internal low-voltage detection), external reset (NMI enable) .
[MB91F52xxxD] power-on reset, low-voltage detect (internal low-voltage detection), external reset.</td></tr>
<tr><td>1881</td><td>Appendix.C</td><td>The following sentence should be modified as follows in the bottom of Table C-1 : Pin Statuses:
(*3) Factor: [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE] external reset (NMI disabled)</td></tr>
</table>

Page	Section	Change Results
Cypress Document Number: 002-05528 Rev.*B		
48,50,52, 54,56,58	1.3	The following sentence should be modified as follows: (Error) *2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD. (Correct) *2: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting and internal LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed operation voltage, as these detection levels are below the minimum guaranteed MCU operation voltage.
170	5.2	The following sentence should be added as follows: Note: If main timer or sub timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.
224	5.5.4.1	The following sentence should be added as follows: Note: If main timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.
224	5.5.4.2	The following sentence should be added as follows: Note: If sub timer is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.
226	5.5.4.7	The following sentence should be added as follows: Note: If main/sub timer or real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to ‘31’, before CPU state changes to the watch mode with power-shutdown.
407	13.8	The following sentence should be added as follows: ■ Note for using the external interrupt as source for recovering from the watch mode with power-shutdown Set the interrupt levels that are used as sources for recovering from the watch mode with power-shutdown to ‘31’, before CPU state changes to the watch mode with power-shutdown. And don’t use NMIX pin as source for recovering from the watch mode with power-shutdown.

Page	Section	Change Results
409	14.2	The following sentence should be modified as follows: (Error) Can be used in both stop mode (Power-shutdown is included) and watch mode (Power-shutdown is included). (Correct) Can be used in stop mode (Power-shutdown is included) and watch mode. Don't use NMI input as source for recovering from the watch mode (Power-shutdown).
850	28.2	The following sentence should be added as follows: Note: If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to '31', before CPU state changes to the watch mode with power-shutdown.
865	28.7.9	The following sentence should be added as follows: Note: If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to '31', before CPU state changes to the watch mode with power-shutdown.
868	28.9	The following sentence should be added as follows: - If the real-time clock is used as source for recovering from the watch mode with power-shutdown, set the interrupt level to '31', before CPU state changes to the watch mode with power-shutdown.
881	30.2	The following sentence should be added as follows: Note: In case of using the watch mode with power-shutdown, it is necessary to satisfy the below both conditions of (1) and (2). (1) Interrupt levels that are used as sources for recovering from the watch mode with power-shutdown are '31', before CPU state changes to the watch mode with power-shutdown. (2) Don't use NMIX pin as source for recovering from the watch mode with power-shutdown.
898	30.5.5.1	The following sentence should be added as follows: (2) Interrupt levels that are used as sources for recovering from the watch mode with power-shutdown are '31', before CPU state changes to the watch mode with power-shutdown. (3) Don't use NMIX pin as source for recovering from the watch mode with power-shutdown.
900	30.5.5.3	The following sentence should be deleted: - Generation of NMI request

Page	Section	Change Results
915	31.1	The following sentence should be modified as follows: (Error) *: The detection voltage of the internal low voltage detection is 0.9V±0.1V. This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD. (Correct) *: The detection voltage of the internal low voltage detection is 0.9V±0.1V. This LVD cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage.
917	31.4.1	The following sentence should be modified as follows: (Error) *: This LVD cannot used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage. Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD. (Correct) *: This LVD cannot used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage.
921	32.1	The following sentence should be modified as follows: (Error) *: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage (2.7V). Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD. (Correct) *: The initial detection voltage of the external low voltage detection is 2.8V±8% (2.576V to 3.024V). This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level is below the minimum guaranteed MCU operation voltage (2.7V).

Page	Section	Change Results
925	32.4.2	The following sentence should be modified as follows: (Error) *: This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level (2.8V±8% = 2.576V to 3.024V) is below the minimum guaranteed MCU operation voltage (2.7V). Below the minimum guaranteed MCU operation voltage, MCU operations are not guaranteed with the exception of LVD. (Crrect) *: This LVD setting cannot be used to reliably generate a reset before voltage dips below minimum guaranteed MCU operation voltage, as this detection level (2.8V±8% = 2.576V to 3.024V) is below the minimum guaranteed MCU operation voltage (2.7V).

Revision History

Document Revision History

Document Title:	MB91520 Series 32-bit Microcontroller FR Family FR81S Hardware Manual		
Document Number:	002-05528		
Revision	**ECN No.**	**Origin of Change**	**Description of Change**
**	-	KUME	New Specification
*A	5291625	KUME	Fixed some errors and added some sentences for version [MB91F52xxxB/MB91F52xxxC/MB91F52xxxE], [MB91F52xxxD]. For details, please see the chapter Major Changes in Appendix.
*B	5373289	KUME	For details, please see the chapter Major Changes in Appendix.