MB91580M/S Series

32-BIT MICROCONTROLLER

FR family FR81S
MB91F583AMG/AMH/AMJ/AMK/ASG/ASH/ASJ/ASK,
MB91F584AMG/AMH/AMJ/AMK/ASG/ASH/ASJ/ASK,
MB91F585AMG/AMH/AMJ/AMK/ASG/ASH/ASJ/ASK

HARDWARE MANUAL

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

http://www.spansion.com/jp/Support/microcontrollers/

Publication Number MB91F585MG_MN705-00015 **Revision** 2.0 **Issue Date** June 30, 2014

PREFACE

Thank you for your continued use of Spansion semiconductor products.
Read this manual and "Data Sheet" thoroughly before using products in the MB91580M/S series.

■ Purpose of this manual and intended readers

This series is a single-chip microcontroller with high-performance/high-speed CPU processing, which contains no wait accessible FLASH memory and uses a high-performance 32-bit RISC CPU. This series is best suited for automotive motor controls such as EPS, etc. This manual explains the functions and operations of the MB91580M/S series and describes how it is used. This manual is intended for engineers engaged in the actual development of products using the MB91580M/S series.

■ Trademark

FR is a product of Spansion Inc.. The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Sample programs and development environment

Spansion offers sample programs free of charge for using the peripheral functions of the FR81S family. Spansion also makes available descriptions of the development environment required for the MB91580M/S series. Feel free to use them to verify the operational specifications and usage of this Spansion semiconductor microcontroller.

- Microcontroller support information:
  http://www.spansion.com/Support/microcontrollers/Pages/default.aspx/

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

How to Use This Manual

■ Finding a function

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

- Search from the table of the contents
  The table of the contents lists the manual contents in the order of description.
- Search from the register
  The register list lists all the registers of this device. You can look up the name of a desired register on the list to find the address of its location or the page that explains it.

  The address where each register is located is not described in the text. To verify the address of a register, see "B. 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

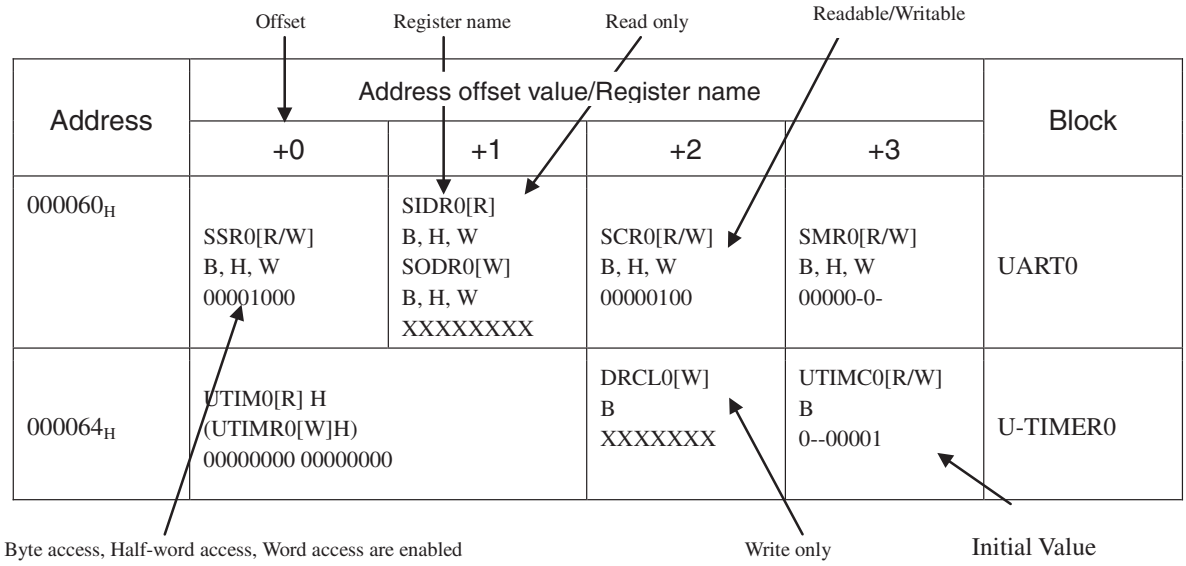
Terminology	Explanation
XBS	A 32-bit width, high-speed internal bus. The bus master is used for access from the CPU (for instruction fetch), the CPU (for data reading or writing), or the on-chip bus. The bus slave is used to access to the on-chip bus, RAM (via the XBS built-in wild register), and flash memory. The bus has a crossbar switch configuration, and a circuit from each bus master to each bus slave can operate simultaneously.

Terminology	Explanation
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 x 2.
External bus (External bus)*	8/16-bit width, low-speed external bus. It connects to memory devices, ASIC and others. This series is the bus master, and a device connected to the external bus is a bus slave.
Main clock (MCLK)	This is the reference clock for LSI operation, and it is supplied from the high-speed system oscillator. It is connected to the timer for main oscillation stabilization wait, the clock generator (PLL) and others.
Sub clock (SBCLK)	This is the reference clock for LSI operation, and it is supplied from the low-speed system oscillator. It is connected to the timer for sub oscillation stabilization wait and others. It can be used by the dual clock products only.
CR oscillation	The clock for watchdog timer 1 (hardware watchdog)
PLL clock (PLLCLK)	The main clock is multiplied by PLL.
CPU clock (CCLK)	The clock for peripherals operating under the XBS.
On-chip bus clock (HCLK)	The clock for peripherals operating under the on-chip bus.
Peripheral clock (PCLK)	The clock for peripherals operating under the 32-bit peripheral bus and 16-bit peripheral bus.
External bus clock (TCLK)*	The reference clock for an external bus interface connected to the X-bus and for the external clock output. It is generated from the base clock by the clock generator.
Main clock mode	The operation mode based on the main clock. The main clock mode has the main run, main sleep, main stop, oscillation stabilization wait run, oscillation stabilization wait reset, and program reset state.
Main run	The main clock mode is selected, and all circuits are operable.
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.

Terminology	Explanation
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 circuit that becomes an object is stopped, and power consumption is decreased.
Power supply ON block always	It is not a target division for the power shutdown.
PMU (Power Management Unit)	The power shutdown is controlled. PMU is always in the turning ON block.
PSW Power switch	This switch turns on/off the power supply controlled by the PMU.
SSCG	Spread Spectrum Clock Generator When the clock in electronic equipment generates a single frequency, the radiation because of the frequency and the higher harmonics wave grows. SSCG is a technology that does working that suppresses the peak of EMI to low especially depending that makes the clock frequency change slightly and oscillates it (= frequency modulation).

*: MB91F585LB/F586LB/F587LB/F585LD/F586LD/F587LD only

■ 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 "B. I/O Map" of "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)

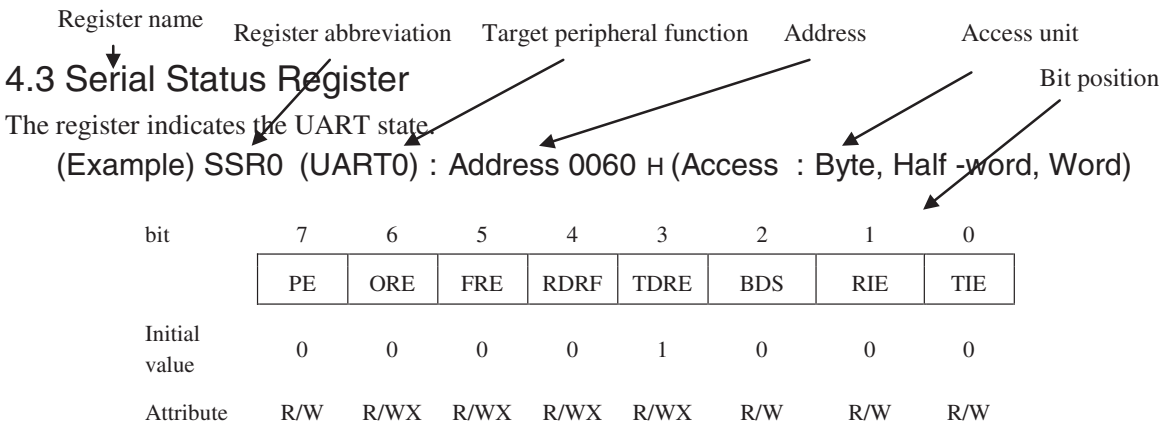
The following explains the address position during access.

- 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



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

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

Access size	Address	Bit position							
Word	060_H+1_H	7	6	5	4	3	2	1	0
Half-word	060_H+0_H	7	6	5	4	3	2	1	0
Word	060_H+0_H	23	22	21	20	19	18	17	16

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

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

Access size	Address	Bit position							
Word	060_H+3_H	7	6	5	4	3	2	1	0
Half-word	060_H+2_H	7	6	5	4	3	2	1	0
Word	060_H+0_H	7	6	5	4	3	2	1	0
Bit name		MD1	MD0	CS"	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 : Data 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) type 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) type instructions, "1" will be read out.) An example is an interrupt request flag.

- R,WX : Read only (Read enabled. This does not affect the writing 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.)
- R/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. This does not affect the writing operation.)
- R0,WX : Undefined bit (The read value is "0". This does not affect the writing operation.)

CONTENTS

CHAPTER 1 PREFACE 1
1.1 Handling Precautions 1
1.2 Handling Device 5
1.3 Application Notes 7
1.4 How to Read This Document 8

CHAPTER 2 OVERVIEW 13
2.1 Overview 13
2.2 Features 13
2.2.1 FR81S CPU Core 13
2.2.2 Peripheral Functions 14
2.3 Product Lineup 17
2.4 Functional Overview 20
2.5 Block Diagram 24
2.6 CPU 26
2.6.1 General-purpose Registers 26
2.6.2 Dedicated Registers 27
2.7 Pin Assignment 28
2.8 Device Package 30
2.8.1 MB91F583AM/F584AM/F585AM -- LQFP-100 (FPT-100P-M20) Package Dimensions 30
2.8.2 MB91F583AS/F584AS/F585AS -- LQFP-64 (FPT-64P-M24) Package Dimensions 31

CHAPTER 3 BASIC INFORMATION 33
3.1 List of Pin Functions 33
3.2 Pins of Each Function MB91F583AM/F584AM/F585AM 43
3.3 Pins of Each Function MB91F583AS/F584AS/F585AS 50
3.4 I/O circuit type 55

CHAPTER 4 CPU 59
4.1 Overview 59
4.2 Features 59
4.3 CPU Operating Description 61
4.3.1 CPU Operating States 61
4.4 Pipeline Operation 62
4.5 Floating Point Operation Processing 62
4.6 Data Structures 62
4.7 Addressing 63
4.8 Programming Model 64
4.9 Reset and EIT Processing 65
4.9.1 Reset 65
4.9.2 EIT Processing 65
4.9.3 Vector Table 66
4.10 Memory Protection Function (MPU) 68

4.10.1 Overview 68
4.10.2 Register List 69
4.10.3 Register Description 70
4.10.4 Operating Description of Memory Protection Function (MPU) 81
4.10.5 Note 83

CHAPTER 5 OPERATION MODE 85
5.1 Overview 85
5.2 Features 85
5.3 Configuration 85
5.4 Registers 86
5.5 Operation 86

CHAPTER 6 CLOCK 89
6.1 Overview 89
6.2 Features 90
6.3 Configuration 91
6.4 Registers 96
6.4.1 DIVR0: (DIVide clock configuration Register 0) Divide Configuration Register 0 97
6.4.2 DIVR2: (DIVide clock configuration Register 2) Divide Configuration Register 2 98
6.4.3 CSELR: (Clock source SELect Register) Clock Source Configuration Register 99
6.4.4 CMONR: (Clock source MONitor Register) Clock Source Monitor Register 101
6.4.5 MTMCR: (Main clock TiMer Control Register) Main Timer Control Register 103
6.4.6 PLLCR: (PLL Configuration Register) PLL Setting Register 105
6.4.7 CSTBR: (Clock STaBilization select Register) Clock Stabilization Select Register 107
6.4.8 PTMCR: (PLL clock osc TiMer Control Register) PLL Clock Oscillation Stabilization Wait Timer Setting Register 108
6.4.9 CCPSSELR (CCtl Pll/Sscg clock SELect Register) PLL/SSCG clock select register 109
6.4.10 PLL/SSCG output clock divide setting register: CCPSDIVR (CCtl Pll/Sscg clock DIVide Register) 110
6.4.11 PLL feed back divide setting register: CCPLLFBR (CCtl PLL FB clock divide Register) 112
6.4.12 SSCG feed back divide setting register 0: CCSSFBR0(CCtl SScg FB clock divide Register 0) 113
6.4.13 SSCG feed back divide setting register 1: CCSSFBR1(CCtl SScg FB clock divide Register 1) 114
6.4.14 SSCG config setting register 0: CCSSCCR0 (CCtl SSCg Config Register 0) 115
6.4.15 SSCG config setting register 1: CCSSCCR1 (CCtl SSCg Config Register 1) 117
6.4.16 Clock gear config setting register 0: CCCGRCR0 (CCtl Clock Gear Config Register 0) 118
6.4.17 Clock gear config setting register 1: CCCGRCR1 (CCtl Clock Gear Config Register 1) 120
6.4.18 Clock gear config setting register 2: CCCGRCR2 (CCtl Clock Gear Config Register 2) 121
6.4.19 PMU clock divide setting register 0: CCPMUCR0 (CCtl PMU Clock divide Register 0) 122
6.4.20 PMU clock divide setting register 1: CCPMUCR1 (CCtl PMU Clock divide Register 1) 123
6.4.21 Sync/Async Control Register: SACR (Sync/Async Control Register) 124

6.4.22 Peripheral Clock Divider Register: PICD (Peripheral Interface Clock Divider register) ... 125
6.5 Operation ... 126
6.5.1 Oscillation Control ... 126
6.5.2 Oscillation Stabilization Wait ... 129
6.5.3 Source Clock (SRCCLK) Select ... 130
6.5.4 Timer ... 134
6.5.5 Notes on Conflict of Clocks ... 135
6.5.6 Clock gear circuit ... 135
6.5.7 Operation under MDI communication ... 136
6.5.8 About PMU clock (PMUCLK) ... 136

CHAPTER 7 FLEXRAY DEDICATED CLOCK ... 139
7.1 Overview ... 139
7.2 Features ... 139
7.3 Configuration ... 140
7.4 Registers ... 140
7.4.1 PLL2DIVM:FlexRay PLL multiplication rate (M division) selection register ... 141
7.4.2 PLL2DIVN:FlexRay PLL multiplication rate (N division) selection register ... 142
7.4.3 PLL2DIVG:FlexRay PLL automatic gear multiplication rate (G Division) selection register ... 142
7.4.4 PLL2MULG:FlexRay PLL step multiplication rate (G division) selection register ... 144
7.4.5 PLL2CTRL: Automatic gear control register ... 145
7.4.6 PLL2DIVK: FlexRay PLL multiplication rate (K division) selection register ... 146
7.4.7 CLKR2:FlexRay PLL clock output control register ... 147
7.5 Recommended Setting ... 149
7.6 Clock Automatic Gear Up/Down ... 150
7.7 Operations ... 152
7.8 Notes ... 153

CHAPTER 8 RESET ... 155
8.1 Overview ... 155
8.2 Features ... 155
8.3 Configuration ... 156
8.4 Registers ... 158
8.4.1 RSTRR (ReSeT Result Register) Reset Source Register ... 158
8.4.2 RSTCR (ReSeT Control Register): Reset Control Register ... 160
8.4.3 CPUAR (CPU Abnormal operation Register): CPU abnormal operation register ... 161
8.4.4 PMUSTR (Power Management Unit STatus register): PMU Status Register ... 163
8.5 Operation ... 164
8.5.1 Reset Level ... 164
8.5.2 Reset Factor ... 165
8.5.3 Reset Acceptance ... 167
8.5.4 Reset Issue ... 168
8.5.5 Reset Sequence ... 172

CHAPTER 9 POWER CONSUMPTION CONTROL ... 175
9.1 Overview ... 175
9.2 Features ... 175
9.3 Configuration ... 176

9.4 Registers 178
9.4.1 STBCR: (STandby mode Control Register) Standby Control Register 178
9.4.2 PMU control register: PMUCTLR (Power Management Unit ConTroL register) 179
9.4.3 Power on Timing control register: PWRTMCTL (PoWeR on TiMing ConTroL register) 180
9.4.4 PMU interrupt flag register1: PMUINTF1(Power Management Unit INTerrupt Flag1 register) 181
9.4.5 PMU interrupt flag register 2: PMUINTF2 (Power Management Unit INTerrupt Flag2 register) 181
9.5 Operation 183
9.5.1 Clock Control 183
9.5.2 List of Clock Supply in Low-power Consumption Mode 183
9.5.3 Sleep Mode 183
9.5.4 Standby Mode: Watch Mode 185
9.5.5 Standby Mode: Watch Mode (Power Shutdown) 187
9.5.6 Standby Mode: Stop Mode 190
9.5.7 Standby Mode: Stop Mode (Power Shutdown) 191
9.5.8 Microcontroller Stop State 194
9.5.9 Transition to Illegal Standby Mode 194
9.5.10 Restrictions on Power Shutdown/Normal Standby Control 195
9.6 Usage Example 197

CHAPTER 10 CLOCK RESET STATE TRANSITIONS 199
10.1 Overview 199
10.2 Device States and Transitions 199
10.2.1 Diagram of State Transitions 199
10.2.2 Explanation of States 200
10.2.3 Priority of State Transition Requests 201
10.3 Device state and corresponding regulator mode 202

CHAPTER 11 CLOCK SUPERVISOR 203
11.1 Overview 203
11.2 Configuration 203
11.3 Registers 204
11.3.1 Clock Supervisor Control Register (CSVCR) 204
11.4 Operation Description 206
11.4.1 Initial State 206
11.4.2 Stopping CR Oscillator and the Clock Supervisor Function 207
11.4.3 Re-enabling the Clock Supervisor 207
11.4.4 Stop Mode 208
11.4.5 Stop mode (power shut-off) 209
11.4.6 Watch Mode 209
11.4.7 Watch Mode (Power Shutdown) 210
11.4.8 Checking the Reset Source Using the Clock Supervisor 211
11.4.9 Switchback from CR Clock 212

CHAPTER 12 REGULATOR CONTROL 213
12.1 Overview 213
12.2 Features 213
12.3 Configuration 213

12.4 Registers 214

CHAPTER 13 LOW-VOLTAGE DETECTION (INTERNAL LOW-VOLTAGE DETECTION) 217

13.1 Overview 217
13.2 Features 217
13.3 Configuration 217
13.4 Registers 218
13.4.1 Internal Low-Voltage Detection Register : LVD 218
13.5 Operation Description 220
13.6 Notes 220

CHAPTER 14 LOW-VOLTAGE DETECTION (EXTERNAL LOW-VOLTAGE DETECTION) 221

14.1 Overview 221
14.2 Features 221
14.3 Configuration 221
14.4 Registers 222
14.4.1 LVD5R (Low-Voltage Detect 5v Rise register) External low-voltage detection - rising edge detection register 222
14.4.2 LVD5F (Low-Voltage Detect 5v Fall register) External low-voltage detection falling edge detection register 222
14.5 Operation Description 224
14.6 Notes 224

CHAPTER 15 NMI INPUT 225

15.1 Overview 225
15.2 Features 225
15.3 Configuration 225
15.4 Registers 225
15.5 Operation Description 225
15.6 Usage Example 226

CHAPTER 16 WATCHDOG TIMER 227

16.1 Overview 227
16.2 Features 227
16.3 Configuration 228
16.4 Registers 229
16.5 Operation Description 232
16.5.1 Software Watchdog Function 232
16.5.2 Hardware Watchdog Function 234
16.6 Usage Example 235

CHAPTER 17 RAMECC FUNCTION 237

17.1 Overview 237
17.2 Features 237
17.3 Configuration 238
17.4 Registers 239
17.4.1 Single bit ECC Error Address Register XBS RAM:SEEARX 239

17.4.2 Double bit ECC Error Address Register XBS RAM : DEEARX 240
17.4.3 ECC Error Control Register XBS RAM (EECSRX) 241
17.4.4 ECC Fake Error Generation Address Register XBS RAM (EFEARX) 242
17.4.5 ECC Fake Error Generation Control Register XBS RAM (EFECRX) 243
17.4.6 Single bit ECC Error Address Register BACKUP-RAM: SEEARA 245
17.4.7 Single bit ECC Error Address Register BACKUP-RAM: DEEARA 246
17.4.8 ECC Error Control Register BACKUP-RAM: EECSRA 247
17.4.9 ECC Fake Error Generation Address Register BACKUP-RAM: EFEARA 248
17.4.10 ECC Fake Error Generation Control Register BACKUP-RAM: EFECRA 249
17.5 Operation 251
17.5.1 RAMECC Function 251
17.5.2 Interrupt-Related Register 252
17.5.3 Test Mode 252
17.5.4 Notes 253

CHAPTER 18 RAM DIAGNOSIS FUNCTION 255
18.1 Overview 255
18.2 Features 255
18.3 Configuration 256
18.4 Registers 257
18.4.1 TEST Error Address Register 0 XBS RAM : TEAR0X 258
18.4.2 TEST Error Address Register 1 XBS RAM : TEAR1X 259
18.4.3 TEST Error Address Register 2 XBS RAM : TEAR2X 260
18.4.4 TEST Start Address Register XBS RAM : TASARX 261
18.4.5 TEST End Address Register XBS RAM : TAEARX 262
18.4.6 TEST Diagnosis Function Register XBS RAM : TTCRX 263
18.4.7 TEST Initialization Function Register XBS RAM : TICRX 266
18.4.8 TEST Soft Reset Generation Control Register XBS RAM : TSRCRX 268
18.4.9 TEST Fake Error Generation Control Register XBS RAM : TFECRX 269
18.4.10 TEST Key Code Control Register XBS RAM : TKCCRX 270
18.4.11 TEST Error Address Register 0 BACKUP-RAM : TEAR0A 271
18.4.12 TEST Error Address Register 1 BACKUP-RAM : TEAR1A 272
18.4.13 TEST Error Address Register 2 BACKUP-RAM : TEAR2A 273
18.4.14 TEST Start Address Register BACKUP-RAM : TASARA 274
18.4.15 TEST End Address Register BACKUP-RAM : TAEARA 275
18.4.16 TEST Diagnosis Function Register BACKUP-RAM : TTCRA 276
18.4.17 TEST Initialization Function Register BACKUP-RAM : TICRA 279
18.4.18 TEST Soft Reset Generation Control Register BACKUP-RAM : TSRCRA 281
18.4.19 TEST Fake Error Generation Control Register BACKUP-RAM : TFECRA 282
18.4.20 TEST Key Code Control Register BACKUP-RAM : TKCCRA 283
18.5 Operation 284
18.5.1 RAM Diagnosis 284
18.5.2 RAM Initialization 284
18.5.3 Interrupt-Related Register 285
18.5.4 RAM diagnosis fake error generation procedure 285
18.5.5 Number of Required Cycles 286
18.5.6 Note 287

CHAPTER 19 BUS DIAGNOSIS FUNCTION 289
19.1 Overview 289

19.2 Features 289
19.3 Configuration 290
19.4 Registers 294
19.4.1 Bus Diagnosis Status Register (BUSDIGSR) 295
19.4.2 Bus Diagnosis Test Register (BUSTSTR0/1) 298
19.4.3 Bus Diagnosis Address Register (BUSADR) 302
19.5 Operation 303
19.5.1 Error detection 303
19.5.2 Test function 304
19.5.3 Notes 305
19.5.4 Example of operating bus diagnosis 306

CHAPTER 20 DELAYED INTERRUPT 313
20.1 Overview 313
20.2 Features 313
20.3 Configuration 313
20.4 Registers 313
20.5 Operation Description 314
20.6 Restrictions 314

CHAPTER 21 EXTERNAL INTERRUPT INPUT 315
21.1 Overview 315
21.2 Features 315
21.3 Configuration 315
21.4 Registers 315
21.5 Operation 318
21.6 Settings 319
21.7 Q&A 320
21.7.1 About Detection Levels and Their Setting Procedures 320
21.7.2 How to Make External Pins to Use for Input 320
21.7.3 About Interrupt Related Registers 320
21.7.4 About Interrupt Types 320
21.7.5 How to Enable/Disable/Clear Interrupts 320
21.7.6 How to Use CAN WAKEUP function 320
21.8 Notes 320

CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER) 321
22.1 Overview 321
22.2 Features 321
22.3 Configuration 321
22.4 Registers 322
22.4.1 ICR00 to ICR47 (Interrupt Control Register 00 to 47) Interrupt Controller Registers 00 to 47 322
22.5 Operation Description 323
22.5.1 Setup 323
22.5.2 Starting 323
22.5.3 Determining Priorities 323
22.5.4 Recovering From Stop Mode 323
22.5.5 Recovering From Standby Mode (Power Shutdown) 323

CHAPTER 23 INTERRUPT REQUEST BATCH READ 325
23.1 Overview 325
23.2 Features 325
23.3 Configuration 325
23.4 Registers 326
23.4.1 IRPR0H (Interrupt Request Peripheral Read register 0H) Interrupt request batch read register 0 high-order 327
23.4.2 IRPR0L (Interrupt Request Peripheral Read register 0L) Interrupt request batch read register 0 low-order 327
23.4.3 IRPR1H (Interrupt Request Peripheral Read register 1H) Interrupt request batch read register 1 high-order 328
23.4.4 IRPR1L (Interrupt Request Peripheral Read register 1L) Interrupt request batch read register 1 low-order 328
23.4.5 IRPR2H (Interrupt Request Peripheral Read register 2H) Interrupt request batch read register 2 high-order 328
23.4.6 IRPR2L (Interrupt Request Peripheral Read register 2L) Interrupt request batch read register 2 low-order 329
23.4.7 IRPR3H (Interrupt Request Peripheral Read register 3H) Interrupt request batch read register 3 high-order 329
23.4.8 IRPR3L (Interrupt Request Peripheral Read register 3L) Interrupt request batch read register 3 low-order 329
23.4.9 IRPR4H (Interrupt Request Peripheral Read register 4H) Interrupt request batch read register 4 high-order 330
23.4.10 IRPR4L (Interrupt Request Peripheral Read register 4L) Interrupt request batch read register 4 low-order 330
23.4.11 IRPR5H (Interrupt Request Peripheral Read register 5H) Interrupt request batch read register 5 high-order 331
23.4.12 IRPR5L (Interrupt Request Peripheral Read register 5L) Interrupt request batch read register 5 low-order 331
23.4.13 IRPR6H (Interrupt Request Peripheral Read register 6H) Interrupt request batch read register 6 high-order 332
23.4.14 IRPR6L (Interrupt Request Peripheral Read register 6L) Interrupt request batch read register 6 low-order 332
23.4.15 IRPR7H (Interrupt Request Peripheral Read register 7H) Interrupt request batch read register 7 high-order 333
23.4.16 IRPR7L (Interrupt Request Peripheral Read register 7L) Interrupt request batch read register 7 low-order 333
23.4.17 IRPR8H (Interrupt Request Peripheral Read register 8H) Interrupt request batch read register 8 high-order 333
23.4.18 IRPR8L (Interrupt Request Peripheral Read register 8L) Interrupt request batch read register 8 low-order 333
23.4.19 IRPR9H (Interrupt Request Peripheral Read register 9H) Interrupt request batch read register 9 high-order 334
23.4.20 IRPR9L (Interrupt Request Peripheral Read register 9L) Interrupt request batch read register 9 low-order 335
23.4.21 IRPR10H (Interrupt Request Peripheral Read register 10H) Interrupt request batch read register 10 high-order 335
23.4.22 IRPR10L (Interrupt Request Peripheral Read register 10L) Interrupt request batch read register 10 low-order 336
23.4.23 IRPR11H (Interrupt Request Peripheral Read register 11H) Interrupt request batch read register 11 high-order 336
23.4.24 IRPR11L (Interrupt Request Peripheral Read register 11L) Interrupt request batch read register 11 low-order 337

23.4.25 IRPR12H (Interrupt Request Peripheral Read register 12H) Interrupt request batch read register 12 high-order 338
23.4.26 IRPR12L (Interrupt Request Peripheral Read register 12L) Interrupt request batch read register 12 low-order 338
23.4.27 IRPR13H (Interrupt Request Peripheral Read register 13H) Interrupt request batch read register 13 high-order 339
23.4.28 IRPR13L (Interrupt Request Peripheral Read register 13L) Interrupt request batch read register 13 low-order 340
23.4.29 IRPR14H (Interrupt Request Peripheral Read register 14H) Interrupt request batch read register 14 high-order 340
23.4.30 IRPR14L (Interrupt Request Peripheral Read register 14L) Interrupt request batch read register 14 low-order 341
23.4.31 IRPR15H (Interrupt Request Peripheral Read register 15H) Interrupt request batch read register 15 high-order 341
23.4.32 IRPR15L (Interrupt Request Peripheral Read register 15L) Interrupt request batch read register 15 low-order 342
23.4.33 IRPR16H (Interrupt Request Peripheral Read register 16H) Interrupt request batch read register 16 high-order 343
23.4.34 IRPR16L (Interrupt Request Peripheral Read register 16L) Interrupt request batch read register 16 low-order 343
23.4.35 IRPR17H (Interrupt Request Peripheral Read register 17H) Interrupt request batch read register 17 high-order 344
23.4.36 IRPR17L (Interrupt Request Peripheral Read register 17L) Interrupt request batch read register 17 low-order 344
23.4.37 IRPR18H (Interrupt Request Peripheral Read register 18H) Interrupt request batch read register 18 high-order 345
23.4.38 IRPR18L (Interrupt Request Peripheral Read register 18L) Interrupt request batch read register 18 low-order 345
23.5 Operation 346

CHAPTER 24 I/O PORTS 347
24.1 Overview 347
24.2 Features 347
24.3 Configuration 348
24.4 Registers 348
24.4.1 PDR0 to PDR10 (Port Data Registers 0 to 10) Port Data Registers 0 to 10 350
24.4.2 DDR0 to 10 (Data Direction Registers 0 to 10) Data Direction Registers 0 to 10 351
24.4.3 PFR0 to PFR10 (Port Function Registers 0 to 10) Port Function Registers 0 to 10 ... 352
24.4.4 PDDR0 to PDDR10 (Port Data Direct Registers 0 to 10) Input Data Direct Registers 0 to 10 353
24.4.5 PPER0 to PPER10 (Port Pull-up Enable Registers 0 to 10) Port Pull-up Enable Registers 0 to 10 354
24.4.6 PILR0 to PILR10 (Port Input Level Registers 0 to 10) Port Input Level Selection Registers 0 to 10 355
24.4.7 PODR0 to PODR10 (Port Output Drive Registers 0 to 10) Port Output Drive Registers 0 to 10 356
24.4.8 EPFR00 to 14 (Extended Port Function Registers 00 to 14) Extended Port Function Registers 00 to 14 357
24.4.9 PORTEN (PORT ENable register) Port Input Enable Register 363
24.4.10 KEYCDR key code register 364
24.5 Operation 365
24.5.1 Pin I/O Assignment 365

24.5.2 Notes on Port Changes 372
24.5.3 EPFR Setting Priority 372
24.5.4 Notes on Setting Input I/O Relocation 372
24.5.5 Noise Filter 373
24.5.6 Disabling of input by GPORTEN 373
24.5.7 Notes on Pins with the A/D Converter Function 373
24.5.8 Notes on Pins with the D/A Converter Function 373
24.5.9 Setting when Using the Base Timer TIOA1 Pin 374
24.5.10 Key Code Register Function Settings 375

CHAPTER 25 CLOCK MONITOR 377
25.1 Overview 377
25.2 Features 377
25.3 Configuration 378
25.4 Registers 379
25.4.1 Clock Monitor Configuration Registers 379
25.5 Operation Description 382
25.6 Setup 383
25.7 Q&A 383
25.7.1 How Do I Configure the Output Pin (MONCLK)? 383
25.7.2 How Do I Select the Output Frequency? 383
25.7.3 How Do I Enable or Disable Clock Monitor Output? 383
25.7.4 How Do I Set the Clock Output Mark Level? 383
25.8 Notes 384

CHAPTER 26 WDT1 CALIBRATION 385
26.1 Overview 385
26.2 Features 385
26.3 Configuration 385
26.4 Registers 386
26.4.1 CUCR1: (Calibration Unit Control Register 1) Calibration Unit Control Register 1 386
26.4.2 CUTD1: (Calibration Unit Timer Data register 1) CR Clock Timer Data Register 387
26.4.3 CUTR1: (Calibration Unit Timer Result register 1) Main Oscillation Timer Data Register 1 387
26.4.4 CRTR: (CR oscillator calibration Trimming Register) CR Oscillation Trimming Setting Register 388
26.5 Operation 389
26.5.1 WDT1 Calibration (CR Clock Calibration) 389
26.5.2 Notes 389

CHAPTER 27 DMA CONTROLLER (DMAC) 391
27.1 Overview 391
27.2 Features 391
27.3 Configuration 392
27.4 Registers 393
27.4.1 DMACR: (DMA Control Register) DMA Control Register 394
27.4.2 DCCR0 to DCCR7: (DMA Channel Control Register 0 to 7) DMA Channel Control Registers 0 to 7 396
27.4.3 DCSR0 to DCSR7: (DMA Channel Status Register 0 to 7) DMA Channel Status Registers 0 to 7 401

27.4.4 DTCR0 to DTCR7:(DMA Transfer Count Register 0 to 7) DMA Transfer Count Registers 0 to 7 402
27.4.5 DSAR0 to DSAR7: (DMA Source Address Registers 0 to 7) DMA Transfer Source Registers 0 to 7 403
27.4.6 DDAR0 to DDAR7: (DMA Destination Address Registers 0 to 7) DMA Transfer Destination Registers 0 to 7 404
27.4.7 DNMIR: (DMA-halt by NMI Register) DMA Transfer Suppression NMI Flag Register 405
27.4.8 DILVR: (DMA-halt by Interrupt Level Register) DMA Transfer Suppress Level Register 406
27.5 Operation 407
27.5.1 Configuration 407
27.5.2 Table for On-chip Bus IP and DMAC Channels 422
27.6 DMA Example 423

CHAPTER 28 GENERATION AND CLEARING OF DMA TRANSFER REQUESTS 425

28.1 Overview 425
28.2 Features 425
28.3 Configuration 426
28.4 Registers 426
28.4.1 ICSEL0 (Interrupt Clear SELect register 0) DMA request clear register 0 428
28.4.2 ICSEL1 (Interrupt Clear SELect register 1) DMA request clear register 1 428
28.4.3 ICSEL2 (Interrupt Clear SELect register 2) DMA request clear register 2 429
28.4.4 ICSEL3 (Interrupt Clear SELect register 3) DMA request clear register 3 429
28.4.5 ICSEL4 (Interrupt Clear SELect register 4) DMA request clear register 4 430
28.4.6 ICSEL5 (Interrupt Clear SELect register 5) DMA request clear register 5 430
28.4.7 ICSEL6 (Interrupt Clear SELect register 6) DMA request clear register 6 431
28.4.8 ICSEL7 (Interrupt Clear SELect register 7) DMA request clear register 7 431
28.4.9 ICSEL8 (Interrupt Clear SELect register 8) DMA request clear register 8 432
28.4.10 ICSEL9 (Interrupt Clear SELect register 9) DMA request clear register 9 432
28.4.11 ICSEL10 (Interrupt Clear SELect register 10) DMA request clear register 10 433
28.4.12 ICSEL11 (Interrupt Clear SELect register 11) DMA request clear register 11 433
28.4.13 ICSEL12 (Interrupt Clear SELect register 12) DMA request clear register 12 434
28.4.14 ICSEL13 (Interrupt Clear SELect register 13) DMA request clear register 13 434
28.4.15 ICSEL14 (Interrupt Clear SELect register 14) DMA request clear register 14 434
28.4.16 ICSEL15 (Interrupt Clear SELect register 15) DMA request clear register 15 435
28.4.17 ICSEL16 (Interrupt Clear SELect register 16) DMA request clear register 16 435
28.4.18 ICSEL17 (Interrupt Clear SELect register 17) DMA request clear register 17 436
28.4.19 ICSEL18 (Interrupt Clear SELect register 18) DMA request clear register 18 436
28.4.20 ICSEL19 (Interrupt Clear SELect register 19) DMA request clear register 19 437
28.4.21 ICSEL20 (Interrupt Clear SELect register 20) DMA request clear register 20 437
28.4.22 ICSEL21 (Interrupt Clear SELect register 21) DMA request clear register 21 438
28.4.23 ICSEL22 (Interrupt Clear SELect register 22) DMA request clear register 22 438
28.4.24 ICSEL23 (Interrupt Clear SELect register 23) DMA request clear register 23 439
28.4.25 ICSEL24 (Interrupt Clear SELect register 24) DMA request clear register 24 439
28.4.26 ICSEL25 (Interrupt Clear SELect register 25) DMA request clear register 25 440
28.4.27 ICSEL26 (Interrupt Clear SELect register 26) DMA request clear register 26 440
28.4.28 ICSEL27 (Interrupt Clear SELect register 27) DMA request clear register 27 441

28.4.29 IORR0-7 (IO triggered DMA Request Register for ch. 0-7) IO transfer request setting registers 0 to 7 442
28.5 Operation 443

CHAPTER 29 BUS PERFORMANCE COUNTERS 445
29.1 Overview 445
29.2 Features 445
29.3 Configuration 445
29.4 Registers 446
29.4.1 BPCCRA (Bus Performance Counter Control Register A) BPC-A Control Register 446
29.4.2 BPCCRB (Bus Performance Counter Control Register B) BPC-B control register 447
29.4.3 BPCCRC (Bus Performance Counter Control Register C) BPC-C control register 447
29.4.4 BPCTRA (Bus Performance CounTer Register A) BPC-A Count Register 447
29.4.5 BPCTRB (Bus Performance CounTer Register B) BPC-B Count Register 448
29.4.6 BPCTRC (Bus Performance CounTer Register C) BPC-C Count Register 448
29.5 Operation Description 449
29.5.1 Setup 449
29.5.2 Starting and Stopping 449
29.5.3 Operation 450
29.5.4 Measurement and Result Processing 450

CHAPTER 30 FLASH MEMORY 453
30.1 Overview 453
30.2 Features 453
30.3 Configuration 454
30.3.1 Block Diagram 454
30.3.2 Sector Configuration Diagram 455
30.3.3 Sector Number and Flash Macro Number Correspondence Chart 457
30.4 Registers 460
30.4.1 FCTLR (Flash ConTroL Register) Flash Control Register 460
30.4.2 FSTR (Flash STatus Register) Flash Status Register 463
30.4.3 Flash interface control register: FLIFCTLR(Flash I/F Control Register) 464
30.4.4 Flash interface feature extension register 1 : FLIFFER1 (Flash I/F Feature Extension Register 1) 465
30.4.5 Flash interface feature extension register 2 : FLIFFER2 (Flash I/F Feature Extension Register 2) 465
30.5 Operation Description 466
30.5.1 Access Mode Setting 466
30.5.2 Programming Flash Memory by CPU 467
30.5.3 Automatic Algorithm 468
30.5.4 Reset command 474
30.5.5 Write Command 474
30.5.6 Chip Erase Command 476
30.5.7 Sector Erase Command 477
30.5.8 Sector erase suspend command 479
30.5.9 Security Function 480
30.5.10 Warnings When Using Flash Memory 481

CHAPTER 31 WorkFlash MEMORY 483
31.1 Overview 483
31.2 Features 483
31.3 Configuration 484
31.3.1 Block Diagram 484
31.3.2 Sector configuration diagram 484
31.4 Registers 485
31.4.1 DFCTLR (Data Flash ConTroL Register) WorkFlash Control Register 485
31.4.2 DFSTR (WorkFlash STatus Register) WorkFlash Status Register 486
31.4.3 FLIFCTLR(Flash I/F Control Register) Flash Interface Control Register 487
31.5 Operation 488
31.5.1 Access Mode Setting 488
31.5.2 Programming Flash Memory by CPU 489
31.5.3 Automatic Algorithm 490
31.5.4 Reset command 496
31.5.5 Write Command 496
31.5.6 Chip Erase Command 499
31.5.7 Sector Erase Command 500
31.5.8 Sector Erase Suspend Command 502
31.5.9 Security Function 503
31.5.10 Warnings When Using Flash Memory 504

CHAPTER 32 FIXEDVECTOR FUNCTION 505
32.1 Overview 505
32.2 Features 505
32.3 Configuration 505
32.4 Registers 505
32.5 Operation Description 505
32.5.1 Operation After Reset Released 505
32.5.2 Usage 506
32.5.3 Notes 506

CHAPTER 33 WILD REGISTER 507
33.1 Overview 507
33.2 Features 507
33.3 Configuration 507
33.4 Registers 508
33.4.1 WREN (Wild Register ENable register) Wile Register Data Enable Register 509
33.4.2 WRAR00 to 15 (Wild Register Address Register 00 to 15)Wild Register Address Registers 00 to 15 509
33.4.3 WRDR00 to 15 (Wild Register Data Register 00 to 15) Wild Register Data Registers 00 to 15 510
33.5 Operation Description 510
33.6 Usage Example 511

CHAPTER 34 CRC 513
34.1 Overview 513
34.2 Features 513
34.3 Configuration 513

34.4 Registers 514
34.4.1 CRCCR (CRC Control Register) 514
34.4.2 CRCINIT (CRC Initial value register) 515
34.4.3 CRCIN (CRC INput data register) 515
34.4.4 CRCR (CRC Register) 516
34.5 Operation 517
34.5.1 CRC Definition 517
34.5.2 Reset Operation 517
34.5.3 Initialization 517
34.5.4 Byte and Bit Orders 517
34.5.5 CRC Calculation Sequence 518
34.6 Example 519
34.6.1 Example 1 CRC16, Fixed Byte Input 519
34.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths 520
34.6.3 Example 3 CRC32, Byte Order, Big Endian 521
34.6.4 Example 4 CRC32, Byte Order, Little Endian 522

CHAPTER 35 MULTI-FUNCTION SERIAL INTERFACE 523
35.1 Overview 523
35.2 Features 523
35.3 Configuration 528
35.4 Registers 529
35.4.1 Common Registers 534
35.4.2 Registers for UART 544
35.4.3 Registers for CSIO 557
35.4.4 Registers for the LIN Interface (v2.1) 581
35.4.5 Registers for I^2C 600
35.5 Operation of UART 624
35.5.1 Interrupt of UART 624
35.5.2 Operation of UART 630
35.5.3 Setup Procedure and Program Flow 643
35.6 Operation of CSIO 646
35.6.1 Interrupts of CSIO 646
35.6.2 Operation of CSIO 654
35.6.3 Setup Procedure and Program Flow 702
35.7 Operation of LIN Interface (v2.1) 703
35.7.1 Interrupt of LIN Interface (v2.1) 703
35.7.2 Operation of LIN Interface (v2.1) 713
35.7.3 Setup Procedure and Program Flow 730
35.8 Operation of I^2C 734
35.8.1 Interrupts of I^2C 734
35.8.2 Operation for I^2C Interface Communication 737
35.8.3 I^2C Master Mode 743
35.8.4 I^2C Slave Mode 773
35.8.5 Example of I^2C Flowchart 781

CHAPTER 36 CAN 789
36.1 Overview 789
36.2 Features 789
36.3 Configuration 789

36.4 Registers 790
36.4.1 Overall Control Register 798
36.4.2 Message Interface Register 809
36.4.3 Message Object 820
36.4.4 Message Handler Register 825
36.4.5 CAN Prescaler Register 831
36.5 Operation 832
36.5.1 Message Object 832
36.5.2 Message Transmission Operation 833
36.5.3 Message Reception Operation 835
36.5.4 FIFO buffer function 837
36.5.5 Interrupt Function 839
36.5.6 Bit Timing and CAN System Clock (fsys) Generation 840
36.5.7 Test Mode 842
36.5.8 Software Initialization 845
36.5.9 CAN Wakeup Function 845

CHAPTER 37 FLEXRAY 847
37.1 Overview 847
37.2 Features 847
37.3 Configuration of FlexRay 850
37.3.1 Customer register 855
37.3.2 Special Registers 861
37.3.3 Interrupt-Related Register 862
37.3.4 Communication controller (CC) control registers 876
37.3.5 Communication Controller (CC) Status Register 894
37.3.6 Message Buffer Control Registers 908
37.3.7 Message Buffer Status Register 912
37.3.8 Identification Register 921
37.3.9 Input buffer 922
37.3.10 Output buffer 928
37.4 Operation of FlexRay 938
37.4.1 Communication cycle 938
37.4.2 Communication mode 941
37.4.3 Clock synchronization 942
37.4.4 Error handling 944
37.4.5 Communication Controller State 946
37.4.6 Network Management 962
37.4.7 Filtering and Masking 963
37.4.8 Transmission Procedure 966
37.4.9 Reception Procedure 969
37.4.10 FIFO Function 971
37.4.11 Message handling 973
37.4.12 Message RAM 982
37.4.13 Interrupts 990

CHAPTER 38 RELOAD TIMER 993
38.1 Overview 993
38.2 Features 993
38.3 Configuration 995

38.4 Registers 996
38.4.1 TMCSR (TiMer Control and Status Registers) Control Status Registers 996
38.4.2 TMR (16bit TiMer Register) 16-bit Timer Register 1000
38.4.3 TMRLRA, TMRLRB(16bit TiMer ReLoad Register A/B) 16-bit Timer Reload Register A, 16-bit Timer Reload Register B 1000
38.5 Operation 1003
38.5.1 Configuration 1003
38.5.2 Operation Procedures 1005
38.5.3 Operation of Counters 1009
38.5.4 Cascade Input 1020
38.5.5 Priority of Concurrent Operations 1020
38.6 Application Note 1021
38.6.1 Single One-shot Timer 1022
38.6.2 Reload Timer 1024
38.6.3 PPG 1026
38.6.4 PWM 1029
38.6.5 PWC 1031

CHAPTER 39 UP/DOWN COUNTER 1033
39.1 Overview 1033
39.2 Features 1033
39.3 Configuration 1035
39.4 Registers 1037
39.4.1 Reload compare register (RCR0, RCR1) 1038
39.4.2 Up/down count register (UDCR0, UDCR1) 1039
39.4.3 Counter control register (CCR0, CCR1) 1040
39.4.4 Counter Status Register (CSR0, CSR1) 1045
39.5 Interrupt 1047
39.6 Operation and Setting Procedure Examples 1049
39.6.1 Operation in Timer Mode 1053
39.6.2 Operation in Up/down Count Mode 1055
39.6.3 Operation in the Phase Difference Count Mode (Multiply-by-Two) 1058
39.6.4 Operation in the Phase Difference Count Mode (Multiply-by-Four) 1060

CHAPTER 40 BASE TIMERS 1063
40.1 Overview 1063
40.2 Features 1063
40.3 Configuration 1067
40.4 Registers 1068
40.4.1 Common Registers 1070
40.4.2 Register for 16/32-bit Reload Timer 1078
40.4.3 Register for 16-bit PWM Timer 1080
40.4.4 Register for 16-bit PPG Timer 1083
40.4.5 Register for 16/32-bit PWC Timer 1086
40.5 Operation 1088
40.5.1 Selection of Timer Function 1088
40.5.2 I/O Assignment 1088
40.5.3 32-bit Mode Operation 1091
40.5.4 16/32-bit Reload Timer Operation 1092
40.5.5 16-bit PWM Timer Operation 1102

40.5.6 16-bit PPG Timer Operation 1110
40.5.7 16/32-bit PWC Timer Operation 1120

CHAPTER 41 PPG 1131
41.1 Overview 1131
41.2 Features 1131
41.3 Configuration 1133
41.4 Registers 1134
41.4.1 PPG Control Status Register (PCN0-5) 1135
41.4.2 PPG Cycle Setting Register (PCSR 0 to PCSR5) 1139
41.4.3 PPG Duty Setting Register (PDUT0 to PDUT5) 1140
41.4.4 PPG Timer Register (PTMR0 to PTMR5) 1141
41.4.5 General-purpose Trigger Selection Register (GTRS0 to GTRS2) 1142
41.4.6 General-purpose trigger setting register (GTREN0) 1144
41.4.7 GATE Function Control Register GATEC0, 2, and 4 1145
41.5 Operation Explanation 1146
41.5.1 PWM Operation (Normal Waveform Selected) 1146
41.5.2 PWM Operation (Center-aligned Waveform Selected) 1148
41.5.3 One-shot Operation (Normal Waveform Selected) 1150
41.5.4 One-shot Operation (Center-aligned Waveform Selected) 1151
41.5.5 Restart Operation 1153
41.5.6 GATE Operation 1154
41.6 Notes 1155

CHAPTER 42 FREE-RUN TIMER 1157
42.1 Overview 1157
42.2 Features 1157
42.3 Configuration 1158
42.4 Registers 1161
42.4.1 Registers for the Free-run Timer Simultaneous Activation 1162
42.4.2 Registers for the 16-bit Free-run Timer 1165
42.4.3 Register for the Free-run Timer Selector 1176
42.5 Operation 1197
42.5.1 Interrupt for the 16-bit Free-run Timer 1198
42.5.2 Operation of the 16-bit Free-run Timer 1199
42.5.3 Operation of the Free-run Timer Selector 1207
42.5.4 Notes on Operating Specifications 1210

CHAPTER 43 INPUT CAPTURE 1211
43.1 Overview 1211
43.2 Features 1211
43.3 Configuration 1212
43.4 Registers 1213
43.4.1 16-bit Input Capture Registers 1214
43.5 Operation 1218
43.5.1 Interrupts for 16-bit Input Capture 1218
43.5.2 Operation of 16-bit Input Capture 1219
43.5.3 Notes on Using the 16-bit Input Capture 1220

CHAPTER 44 OUTPUT COMPARE 1221
44.1 Overview 1221
44.2 Features 1221
44.3 Configuration 1222
44.4 Registers 1223
44.4.1 16-bit Output Compare Registers 1225
44.5 Operation Explanation 1233
44.5.1 Interrupts for 16-bit Output Compare 1233
44.5.2 Operation of 16-bit Output Compare 1234
44.5.3 Notes on Using 16-bit Output Compare 1247

CHAPTER 45 12-BIT A/D CONVERTER 1249
45.1 Overview 1249
45.2 Features 1249
45.3 Configuration 1252
45.4 Registers 1255
45.4.1 Analog Input Enable Registers 1258
45.4.2 A/D Activation Compare Register 1260
45.4.3 12-bit A/D Converter Control Registers 1270
45.5 Operation 1274
45.5.1 Interrupt for A/D Activation Compare 1274
45.5.2 A/D Activation Compare Operation 1275
45.5.3 A/D Activation Arbitration Operation 1287
45.5.4 12-bit A/D Converter Operation 1289
45.6 Caution 1291

CHAPTER 46 WAVEFORM GENERATOR 1293
46.1 Overview 1293
46.2 Features 1293
46.3 Configuration 1294
46.4 Registers 1297
46.4.1 Registers for the Waveform Generator 1298
46.5 Operation Explanation 1322
46.5.1 Interrupts for Waveform Generator 1322
46.5.2 Operation of the Waveform Generator 1324
46.6 Notes: 1341

CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION 1343
47.1 Overview 1343
47.2 Features 1343
47.3 Configuration 1344
47.4 Registers 1345
47.4.1 Motor Control Function Operating Clock Selection Register: MTRCSR 1346
47.4.2 Waveform Generator Output Level Conversion Setting Register: RTOSEL0, RTOSEL1 1347
47.5 Operation Explanation 1349
47.5.1 Operating Clock Selection Control 1349
47.5.2 Waveform Generator Output Level Conversion Control 1350
47.5.3 A/D Mode Setting Register of the A/D Converter (ADMD) 1352

47.6 Notes 1353

CHAPTER 48 D/A CONVERTER 1355
48.1 Overview 1355
48.2 Features 1355
48.3 Configuration 1355
48.4 Registers 1356
48.4.1 DACR (D/A Control Register) 1356
48.4.2 DADR (D/A Data Register) 1356
48.4.3 DAER (D/A Enable Register) Analog Output control Register 1357
48.5 Operating Explanation 1358

CHAPTER 49 ON-CHIP DEBUGGER (OCD) 1359
49.1 Overview 1359
49.2 Features 1359
49.3 Configuration 1360
49.3.1 Clock for DEBUG I/F 1361
49.4 Registers 1362
49.4.1 DBG Registers 1362
49.4.2 User IO Registers 1362
49.5 Operation 1363
49.5.1 OCDU Operating Mode 1363
49.5.2 Overview of DEBUG I/F 1365
49.5.3 Specification Restrictions at Connection to OCD Tool of This Product 1369
49.5.4 Summary of Restrictions 1371
49.5.5 OCD-DSU ID Code of This Product Type 1375

APPENDIX 1377
APPENDIX A. Memory Map 1377
APPENDIX B. I/O Map 1378
APPENDIX C. Interrupt Vector 1445
APPENDIX D. Pin Statuses in CPU Status 1451
Major Changes 1459

CHAPTER 1 PREFACE

1.1 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 Spansion semiconductor devices.

■ Precautions for Product Design

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

● Absolute Maximum Ratings

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

● Recommended Operating Conditions

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

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

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

● Processing and Protection of Pins

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

(1) Preventing Over-Voltage and Over-Current Conditions

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

(2) Protection of Output Pins

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

Therefore, avoid this type of connection.

(3) Handling of Unused Input Pins

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

Code: DS00-00004-3E
Code: PREV-1v0-91580MS-1-E

● 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

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

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

■ Precautions for Package Mounting

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

● Lead Insertion Type

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

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

Condition: 125 °C/24 h

Static Electricity

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

(1) Maintain relative humidity in the working environment between 40% and 70%.
Use of an apparatus for ion generation may be needed to remove electricity.

(2) Electrically ground all conveyors, solder vessels, soldering irons and peripheral equipment.

(3) Eliminate static body electricity by the use of rings or bracelets connected to ground through high resistance (on the level of 1 MΩ).
Wearing of conductive clothing and shoes, use of conductive floor mats and other measures to minimize shock loads is recommended.

(4) Ground all fixtures and instruments, or protect with anti-static measures.

(5) Avoid the use of styrofoam or other highly static-prone materials for storage of completed board assemblies.

■ Precautions for Use Environment

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

For reliable performance, do the following:

(1) Humidity

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

(2) Discharge of Static Electricity

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

(3) Corrosive Gases, Dust, or Oil

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

(4) Radiation, Including Cosmic Radiation

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

(5) Smoke, Flame

CAUTION: Plastic molded devices are flammable, and therefore should not be 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 Spansion products in other special environmental conditions should consult with sales representatives.

Please check the latest handling precautions at the following URL.

http://www.spansion.com/fjdocuments/fj/datasheet/e-ds/DS00-00004.pdf

1.2 Handling Device

■ Notes on Handling Device

The latch-up prevention and pin processing are explained below.

● 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 supplies (AVCC0, AVCC1, AVRH0, AVRH1) and analog input must not exceed the digital power supply (VCC5) when the power supply to the analog system is turned on or off.

In the correct power-on sequence, turn on the digital power supply (VCC5) and analog power supplies (AVCC0, AVCC1, AVRH0, AVRH1) simultaneously. Alternatively, turn on the digital power supply (VCC5) first, and then turn on the analog power supplies (AVCC0, AVCC1, AVRH0, AVRH1).

● Treatment of unused pins

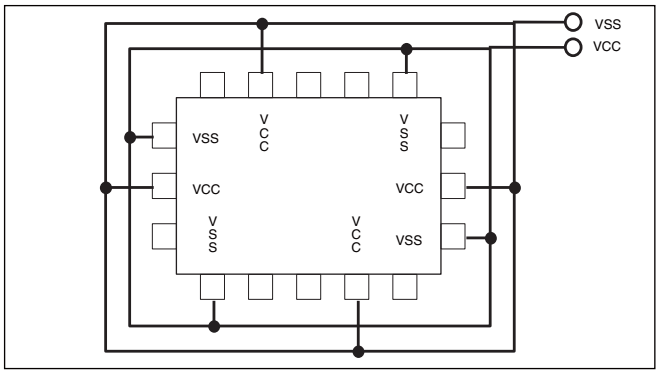
If unused input pins are left open, they may cause a permanent damage to the device due to device malfunction or latch-up. Connect a 2kΩ or higher resistor to each of unused input pins for pull-up or pull-down processing.

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

● Power supply pins

The device is designed to ensure that if the device contains multiple VCC or VSS pins, the pins that should be at the same potential are interconnected to prevent latch-up or other malfunctions. Further, connect these pins to an external power supply or ground to reduce unwanted radiation, prevent strobe signals from malfunctioning due to a raised ground level, and fulfill the total output current standard, etc. As shown in Figure 1.2-1, all VSS power supply pins must be treated in the similar way. If multiple VCC or VSS systems are connected, the device cannot operate correctly even within the guaranteed operating range.

Figure 1.2-1 Power Supply Input Pins



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

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

- Crystal oscillation circuit

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

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

- Mode pin (MD[1:0])

Connect the MD[1:0] mode pin to the VCC or VSS pin directly. To prevent an erroneous selection of test mode caused by the noise, reduce the pattern length between each mode pin and VCC or VSS pin on the printed circuit board. Also, use the low-impedance pin connection.

- During power-on

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

- Notes during PLL clock operation

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

- Treatment of A/D converter power supply pins

Connect the pins to have AVCC0 = AVCC1 = AVRH0 = AVRH1 = VCC, AVSS0/AVRL0 = AVSS1/AVRL1 = VSS even if the A/D converter are not used.

- Note on using external clock

The external clock is unsupported.

External direct clock input cannot use.

- Power-on sequence of A/D converter analog inputs

Be sure to turn on the digital power supply (VCC5) first, and then turn on the A/D converter power supplies (AVCC0, AVCC1, AVRH0, AVRH1, AVRL0, AVRL1) and analog inputs (AN0 to 14, AN16 to AN23). Also, turn off the A/D converter power supplies (AVCC0, AVCC1, AVRH0, AVRH1, AVRL0, AVRL1) and analog inputs (AN0 to 14, AN16 to AN23) first, and then turn off the digital power supply (VCC5). When the AVRH0 and AVRH1 pin voltages are turned on or off, they must not exceed AVCC0 and AVCC1. Even if a common analog input pin is used as an input port, its input voltage must not exceed AVCC0 or AVCC1. (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.

- Operating voltages

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

1.3 Application Notes

■ Function Switching of a Multiplexed Port

To switch between the port function and the multiplexed pin function, use the PFR (port function register). For details, see "CHAPTER 24 I/O PORTS".

■ Low-power Consumption Mode

- To set Sleep mode / Watch mode / Stop mode, or Watch mode (power shut-off) / Stop mode (power shut-off), refer to the section "Launching Sleep mode / Watch mode / Stop mode" or "Launching Watch mode (power shut-off) / Stop mode (power shut-off)" of "CHAPTER 9 POWER CONSUMPTION CONTROL", and follow the procedures.

- Do not perform the following when using a monitor debugger.
  - Do not set a break point for the low-power consumption transition program.
  - Do not execute an operation step for the low-power consumption transition program.

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

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

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

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

Note: These points can be ignored because the bit instructions already take the points into consideration for registers that support read-modify-write (RMW) operations. These points must be considered when using the bit instructions for registers that do not support RMW operations.

1.4 How to Read This Document

■ Primary Terms

The following explains the primary terms used in this series.

Terminology	Meaning
XBS	A 32-bit width, high-speed internal bus. The bus master is used for access from the CPU (for instruction fetch), the CPU (for data reading or writing), or the on-chip bus. The bus slave is used to access to the on-chip bus, RAM (via the XBS built-in wild register), and flash memory. The bus has a crossbar switch configuration, and a circuit from each bus master to each bus slave can operate simultaneously.
On-chip bus	A 32-bit width, high-speed internal bus. It has a 2-layer structure for XBS and DMA, and they can operate simultaneously. The bus master of the XBS layer is accessed from the XBS. The bus master of the DMA layer is accessed from the DMA. The bus slave of both layers has CAN, 16/32-bit peripheral bus bridge and others. The bus slave of only DMA layer has an access to the XBS.
32-bit peripheral bus	A 32-bit width, low-speed internal bus. It connects to various types of peripherals.
16-bit peripheral bus (R-bus)	A 16-bit width, low-speed internal bus. It connects to various types of peripherals. The 32-bit width access to this bus is divided into 16 bits × 2.
Main clock (MCLK)	This is the reference clock for LSI operation, and it is supplied from the high-speed system oscillator. It is connected to the timer for main oscillation stabilization wait, the clock generator (PLL) and others.
CR oscillation	The clock for watchdog timer 1 (hardware watchdog)
PLL clock (PLLCLK)	The main clock is multiplied by PLL.
CPU clock (CCLK)	The clock for peripherals operating under the XBS.
On-chip bus clock (HCLK)	The clock for peripherals operating under the on-chip bus
Peripheral clock (PCLK)	The clock for peripherals operating under the 32-bit peripheral bus and 16-bit peripheral bus
Motor control function operating clock (MTRCLK)	The operating clock for the motor control function*. *: For details on the motor control function, see the next page.
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 device
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.

Terminology	Meaning
Power shut-off	This stops power supply to a specific circuit, to reduce power consumption.
Constant power ON block	This is the block where power shut-off feature will not be applied.
PMU Power Management Unit	Power shut-off can be controlled. PMU is located in the constant power ON block.
PSW	Power switch This switch turns on/off the power supply controlled by the PMU.
SSCG	Spread Spectrum Clock Generator If single frequency is generated by the built-in clock of the electronic device, radiation at the frequency and its higher harmonics increases. SSCG is a technology that reduces EMI peak by slightly modifying the clock frequency (= by applying frequency modulation).
Motor control function	Motor control function is a collective term for free-run timer, input capture, output compare, 12-bit A/D converter, and waveform generator.

■ Access Unit and Address Position

Offset Register name Read only Read and write enabled

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000060_H	SSR0[R/W] B, H, W 00001000	SIDR0[R] B, H, W SODR0[W] B, H, W XXXXXXXX	SCR0[R/W] B, H, W 00000100	SMR0[R/W] B, H, W 00000-0-	UART0
000064_H	UTIM0[R] H (UTIMR0[W]H) 00000000 00000000		DRCL0[W] B XXXXXXX	UTIMC0[R/W] B 0--00001	U-TIMER0

Byte access, Half-word access, or Word access is enabled. Write only Initial value

Although three types of access (Byte, Half-word, and Word access) are enabled, some registers have access restrictions. For details, see the "Appendix B. I/O MAP", 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 or Half-word access only (The Word access is not allowed.)
H, W : Half-word access or 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

Register name Register abbreviation Target peripheral function Address Access unit Bit position

4.3 Serial Status Register

The register indicates the UART state.

SSR0(UART0):Address 0060_H (Access:Byte, Half-word, or Word)

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

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							
Byte	060_H+0_H	07	06	05	04	03	02	01	00
Half-word	060_H+0_H	15	14	13	12	11	10	09	08
Word	060_H+0_H	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							
Byte	060_H+1_H	07	06	05	04	03	02	01	00
Half-word	060_H+0_H	07	06	05	04	03	02	01	00
Word	060_H+0_H	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							
Byte	060_H+2_H	07	06	05	04	03	02	01	00
Half-word	060_H+2_H	15	14	13	12	11	10	09	08
Word	060_H+0_H	15	14	13	12	11	10	09	08
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							
Byte	060_H+3_H	07	06	05	04	03	02	01	00
Half-word	060_H+2_H	07	06	05	04	03	02	01	00
Word	060_H+0_H	07	06	05	04	03	02	01	00
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 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 : "0" must be written.
W1 : "1" must be written.
(RM0) : "0" is read by read-modify-write operation.
(RM1) : "1" is read by read-modify-write operation.
RX : The read value is undefined. (A reserved bit or an undefined bit)
WX : Data 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 instruction.) An example is a port data register.
- R(RM1), W :Read and write enabled (The read value is different from the written value. 1 is read by the read-modify-write instruction.) An example is an interrupt request flag.
- R,WX :Read only (Read enabled, and the data writing does not affect on the operation.)
- R1,W : Write only (Write enabled, and the read value is 1.)
- R0,W : Write only (Write enabled, and the read value is 0.)
- RX,W : Write only (Write enabled, but the read value is undefined.)
- R/W0 : Reserved bit (The written value is 0, and the read value is written value.)
- R0,W0 : Reserved bit (The written value is 0, and the read value is 0.)
- R1,W0 : Reserved bit (The written value is 0, and the read value is 1.)
- RX,W0 : Reserved bit (The written value is 0, but the read value is undefined.)
- R/W1 : Reserved bit (The written value is 1, and the read value is written value.)
- R1,W1 : Reserved bit (The written value is 1, and the read value is 1.)
- R0,W1 : Reserved bit (The written value is 1, and the read value is 0.)
- RX,W1 : Reserved bit (The written value is 1, but the read value is undefined.)
- RX,WX : Undefined bit (The read value is undefined, but the data writing does not affect on the operation.)
- R0,WX : Undefined bit (The read value is 0, but the data writing does not affect on the operation.)

CHAPTER 2 OVERVIEW

2.1 Overview

This series has Spansin 32-bit microcontrollers for automobile motor control. They use the FR81S CPU that is compatible with the FR family.

2.2 Features

2.2.1 FR81S CPU Core

- 32-bit RISC, load/store architecture, pipeline 5-stage structure
- Maximum operating frequency: 128MHz (Source oscillation= 4.0MHz, 32 multiplied (PLL clock multiplication system))
- General-purpose register: 32 bits, 16 sets
- 16-bit fixed length instructions (basic instructions), 1 instruction per cycle
- Instructions appropriate to embedded applications
  - Memory-to-memory transfer instructions
  - Bit manipulation instructions
  - Barrel shift instructions
- 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
  Overhead decrement 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 data.
  - Control access privilege in both privilege mode and user mode
- Built-in FPU (floating-point operation)
  - IEEE754 compliant
  - Floating-point register: 32 bits × 16 sets

Code: OVERVIEW-1v0-91580MS-1-E

2.2.2 Peripheral Functions

- Clock generation (SSCG function is available)
  - Main oscillation (4 to 20 MHz)
  - PLL multiplication rate:1 to 32 times
- CR oscillation
  - Oscillation frequency: 100kHz, with frequency accuracy ± 50% (pre-trimming)
  - Trimming is enabled
  - To be used as a count clock of hardware watchdog
  - Oscillation stop feature during stand-by is not available
    MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
    MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK
  - Oscillation stop feature during stand-by is available
    MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
    MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH
- Built-in program flash memory capacity
  MB91F583: 256+64 Kbytes
  MB91F584: 384+64 Kbytes
  MB91F585: 512+64 Kbytes
- Built-in data flash (WorkFlash) 64 Kbytes
- Built-in RAM capacity
  - Main RAM
    MB91F583: 32 Kbytes
    MB91F584: 48 Kbytes
    MB91F585: 48 Kbytes
  - Backup RAM 8 Kbytes
- General-purpose ports: MB91F583AM/F584AM/F585AM 76 ports
  Including eight I^2C open drain corresponding ports 8
  MB91F583AS/F584AS/F585AS 44ports
  Including eight I^2C open drain corresponding ports 2
- DMA controller
  - Up to 8 channels can be started simultaneously.
  - 2 transfer factors (Internal peripheral request and software)
- External interrupt input
  MB91F583AM/F584AM/F585AM: 8 channels
  MB91F583AS/F584AS/F585AS: 7 channels
  Level ("H" / "L") or edge detection (rising or falling) enabled
- Multi-function serial communication (built-in transmission/reception FIFO memory)
  MB91F583AM/F584AM/F585AM: 4 channels
  MB91F583AS/F584AS/F585AS: 2 channels
  < UART (Asynchronous serial interface) >
  - Full-duplex double buffering system, 16-byte transmission FIFO memory, 16-byte reception FIFO memory
  - Parity or no parity is selectable.
  - Built-in dedicated baud rate generator
  - An external clock can be used as the transfer clock
  - Parity, frame, and overrun error detect functions provided
  - DMA transfer supported

<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 to 16, 20, 24, 32-bit data length can be set.
- Built-in dedicated baud rate generator (Master operation)
- An external clock can be entered. (Slave operation)
- Overrun error detect function is provided.
- Built-in chip selection function
- DMA transfer supported

<LIN interface (v2.1)>

- Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
- LIN protocol revision2.1 supported.
- Master and slave systems supported
- Framing error and overrun error detection
- LIN sync break generation and detection; LIN sync delimiter generation
- Built-in dedicated baud rate generator
- An external clock can be adjusted by the reload counter.
- DMA transfer supported

< I^2C >

- MB91F583AM/F584AM/F585AM: Supported for 3 channels: ch.0,ch.1,and ch.3
  MB91F583AS/F584AS/F585AS: Supported for 1 channel: ch.0
- Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO
- Standard mode (Max. 100Kbps) / high-speed mode (Max. 400Kbps) supported
- DMA transfer supported (for transmission only)

- CAN controller (CAN)
  MB91F583AM/F584AM/F585AM: 2 channels
  MB91F583AS/F584AS/F585AS: 1 channel
  - Transfer speed: Up to 1Mbps
  - 64-transmission/reception message buffering
- FlexRay controller
  MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ/
  F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ: 1 unit (ch.A/ch.B)
  - FlexRay Specifications Version 2.1 supported
  - Up to 128 message buffers
  - 8K bytes of message RAM
  - Variable length of message buffers
  - Each message buffer can be allocated as a part of reception buffer, transmission buffer or reception FIFO
  - Host access to the message buffer via input and output buffers
  - Filtering for slot counter, cycle counter and channels
  - Maskable interrupts are supported
- PPG: 16 bits × 6 channels
- Reload timer: 16 bits × 4 channels
- A/D converter (successive approximation type)
  - 12-bit resolution
    MB91F583AM/F584AM/F585AM: 3 units (23 channels)

MB91F583AS/F584AS/F585AS: 3 units (17 channels)
  - Conversion time: 1 μs
- Free-run timer: 16 bits × 6 channels (1 channel can be selected for input capture, and 1 channel for output compare.)
- Input capture: 16 bits × 4 channels (linked to the free-run timer)
- Output compare: 16 bits × 7 channels (linked to the free-run timer)
- Waveform generator: 2 units (7 channels)
- 10-bit D/A converter: 1 channel
- Calibration: The hardware watchdog for CR oscillation drive
  The CR oscillation frequency can be trimmed.
- Clock Supervisor
  - Anomaly supervisory feature (by damaged quartz, etc.) of external main oscillation (4MHz)
  - When anomaly is detected, clock is switched to CR.
- Up/ down counter: 2 channels
  - 8/16bit Up/ down counter
- Base timer: 2 channels
  - 16-bit timer
  - Any of four PWM/PPG/PWC/reload timer functions can be selected and used.
  - As for the functions of PWC and reload timer, 2 channels of cascade mode can be used as 32-bit timer.
- CRC generation
- Watchdog timer
  - Hardware watchdog
  - Software watchdog
- NMI
- Interrupt controller
- Interrupt request batch read
  Multiple interrupts from peripherals can be read by a series of registers.
- I/O relocation (MB91F583AM/F584AM/F585AM)
  Change of terminal position of periphery functions
- Low-power consumption mode
  - Sleep/Stop/Watch
  - Stop (Power shut-off)/Watch (Power shut-off)
- Power-on reset/internal low-voltage detection reset
- Low-voltage detection reset
- Package
  MB91F583AM/F584AM/F585AM: LQFP-100
  MB91F583AS/F584AS/F585AS: LQFP-64
- CMOS 90 nm technology
- Power supplies
  - Single 5V power
  - Stepdown transformer circuit brings the 5.0V down to generate 1.2V internally
  - I/O 5.0V

2.3 Product Lineup

Table 2.3-1 MB91580M Series Product Lineup Comparison

- **Memory size**

Items	MB91F583AMG MB91F583AMH MB91F583AMJ MB91F583AMK	MB91F584AMG MB91F584AMH MB91F584AMJ MB91F584AMK	MB91F585AMG MB91F585AMH MB91F585AMJ MB91F585AMK
Flash memory capacity (program)	256+64 Kbytes	384+64 Kbytes	512+64 Kbytes
Flash memory capacity (work)	64 Kbytes		
RAM capacity (main)	32 Kbytes	48 Kbytes	48 Kbytes
RAM capacity (backup)	8 Kbytes		

- **Function**

Items		MB91F583AMG MB91F584AMG MB91F585AMG	MB91F583AMH MB91F584AMH MB91F585AMH	MB91F583AMJ MB91F584AMJ MB91F585AMJ	MB91F583AMK MB91F584AMK MB91F585AMK
System clock		On-chip PLL clock multiplication system (Up to 32 times of multiplication) Minimum instruction execution time: 7.81ns (128MHz, source oscillation 4MHz × 32 times of multiplication)			
CR oscillation		Provided			
	Oscillation stop feature during stand-by	Provided	Provided	Not provided	Not provided
External bus interface		Not provided			
DMA transfer		8 channels			
16-bit base timer		2 channels			
Free-run timer		6 channels			
Input capture		4 channels			
Output compare		7 channels			
Waveform generator		2 units (7 channels)			
16-bit reload timer		4 channels			
PPG		6 channels			
External interrupt		8 channels			
A/D converter		3 units (23 channels)			
R/D converter		Not provided			
D/A converter		Provided			
Up/ down counter		2 channels			
Multi-function serial interface		4 channels			
CAN		64msb × 2 channels (ch.0/ch.1)			
FlexRay		128msb × 1 unit (ch.A / ch.B)	Not provided	128msb × 1 unit (ch.A / ch.B)	Not provided
Software watchdog		Provided			
Hardware watchdog		Provided			
CRC generation		2 channels			
Internal power supply low-voltage detection reset		Provided			
Low-voltage detection reset		Provided			
Device package		LQFP-100			
Debug interface		Built-in OCD (On Chip Debug Unit)			

Table 2.3-2 MB91580S Series Product Lineup Comparison

- **Memory size**

Items	MB91F583ASG MB91F583ASH MB91F583ASJ MB91F583ASK	MB91F584ASG MB91F584ASH MB91F584ASJ MB91F584ASK	MB91F585ASG MB91F585ASH MB91F585ASJ MB91F585ASK
Flash memory capacity (program)	256+64 Kbytes	384+64 Kbytes	512+64 Kbytes
Flash memory capacity (work)	64 Kbytes		
RAM capacity (main)	32 Kbytes	48 Kbytes	48 Kbytes
RAM capacity (backup)	8 Kbytes		

- **Function**

Items		MB91F583ASG MB91F584ASG MB91F585ASG	MB91F583ASH MB91F584ASH MB91F585ASH	MB91F583ASJ MB91F584ASJ MB91F585ASJ	MB91F583ASK MB91F584ASK MB91F585ASK
System clock		On-chip PLL clock multiplication system (Up to 32 times of multiplication) Minimum instruction execution time: 7.81ns (128MHz, source oscillation 4MHz × 32 times of multiplication)			
CR oscillation		Provided			
	Oscillation stop feature during stand-by	Provided	Provided	Not provided	Not provided
External bus interface		Not provided			
DMA transfer		8 channels			
16-bit base timer		2 channels			
Free-run timer		6 channels			
Input capture		4 channels			
Output compare		7 channels			
Waveform generator		2 units (7 channels)			
16-bit reload timer		4 channels			
PPG		6 channels			
External interrupt		7 channels			
A/D converter		3 units (17 channels)			
R/D converter		Not provided			
D/A converter		Provided			
Up/ down counter		2 channels			
Multi-function serial interface		2 channels			
CAN		64msb × 1 channel (ch.0)			
FlexRay		128msb × 1unit (ch.A / ch.B)	Not provided	128msb × 1unit (ch.A / ch.B)	Not provided
Software watchdog		Provided			
Hardware watchdog		Provided			
CRC generation		2 channels			
Internal power supply low-voltage detection reset		Provided			
Low-voltage detection reset		Provided			
Device package		LQFP-64			
Debug interface		Built-in OCD (On Chip Debug Unit)			

Table 2.3-3 MB91580L Series Product Lineup Comparison

- **Memory size**

Items	MB91F585LA MB91F585LB MB91F585LC MB91F585LD	MB91F586LA MB91F586LB MB91F586LC MB91F586LD	MB91F587LA MB91F587LB MB91F587LC MB91F587LD
Flash memory capacity (program)	512+64 Kbytes	768+64 Kbytes	1024+64 Kbytes
Flash memory capacity (work)	64 Kbytes		
RAM capacity (main)	48 Kbytes	64 Kbytes	96 Kbytes
RAM capacity (backup)	8 Kbytes		

- **Function**

Items		MB91F585LA MB91F586LA MB91F587LA	MB91F585LB MB91F586LB MB91F587LB	MB91F585LC MB91F586LC MB91F587LC	MB91F585LD MB91F586LD MB91F587LD
System clock		On-chip PLL clock multiplication system (Up to 32 times of multiplication) Minimum instruction execution time: 7.81ns (128MHz, source oscillation 4MHz × 32 times of multiplication)			
CR oscillation		Provided			
	Oscillation stop feature during stand-by	Provided	Provided	Not provided	Not provided
External bus interface		Not provided	Address: 22 bits Data: 16 bits	Not provided	Address: 22 bits Data: 16 bits
DMA transfer		8 channels			
16-bit base timer		2 channels			
Free-run timer		6 channels			
Input capture		8 channels			
Output compare		12 channels			
Waveform generator		2 units (12 channels)			
16-bit reload timer		4 channels			
PPG		24 channels			
External interrupt		8 channels			
A/D converter		3 units (24 channels)			
R/D converter		Provided	Not provided	Provided	Not provided
D/A converter		Not provided	Provided	Not provided	Provided
Up/ down counter		2 channels			
Multi-function serial interface		5 channels			
CAN		64 msb × 3 channels (ch.0/ch.1/ch.2)			
FlexRay		128 msb × 1 unit (ch.A / ch.B)			
Software watchdog		Provided			
Hardware watchdog		Provided			
CRC generation		1 channel			
Internal power supply low-voltage detection reset		Provided			
Low-voltage detection reset		Provided			
Device package		LQFP-144			
Debug interface		Built-in OCD (On Chip Debug Unit)			

(Note) For details on the MB91580L series, see the "MB91580L Series HARDWARE MANUAL".

2.4 Functional Overview

Table 2.4-1 MB91580M/S Series Functional Overview (1 / 4)

Function	Features
CPU	32-bit RISC microcontroller FR81S CPU core Built-in memory protection function (MPU) 8ch Built-in floating-point operation (FPU)
System clock	Main oscillation: 4MHz (up to 20MHz can be input) PLL multiplication rate: Up to 32 times of multiplication
CR oscillation	Oscillation frequency: 100kHz, with frequency precision ± 10% Trimming is enabled To be used as a count clock of hardware watchdog Oscillation stop feature during stand-by is not available MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK Oscillation stop feature during stand-by is available MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH
I/O ports	Each bit can be programmed for I/O or peripheral signals. Input thresholds, driving capacity, and pull-up can be set.
Internal bus interface	On-chip bus: 32 bits; maximum operating frequency: 128MHz
Peripheral bus interface (Excluding motor control function)	Maximum operating frequency: 40MHz 32-bit peripheral bus, or 16-bit peripheral bus (R-bus) (Note) Both of them operate in the same frequency.
Peripheral bus interface (Motor control function)	32-bit peripheral bus, peripheral clock or on-chip bus clock is selectable. Maximum operating frequency When peripheral clock is selected: 40MHz When on-chip bus clock is selected: 128MHz
Flash memory interface	Wild register function provided. However, it can be used only during no wait operation. If operating frequency exceeds 80MHz, add 1wait. Small sectors (64KB) supported
DMA controller	Up to 8 channels can be started simultaneously. A transfer cause (internal peripheral request or software) is selectable. Burst or block transfer mode is selectable.
Generation and clear of DMA transfer request	If multiple interrupts occur for a single interrupt vector, generation of DMA request from any interrupt can be selected. If multiple interrupts occur for a single interrupt vector, an interrupt to be cleared at the end of DMA transfer can be selected.
Base timer (2ch)	16-bit timer Any of four PWM/PPG/PWC/reload timer functions can be selected and used. A 32-bit timer can be used in 2 channels of cascade mode for the reload timer/PWC function.
Free-run timer (6ch)	16-bit up counter

Table 2.4-1 MB91580M/S Series Functional Overview (2 / 4)

Function	Features
Input capture (4ch)	16-bit capture register to detect a rising edge, a falling edge, or both of them When an edge of pin input is detected, the counter value of 16-bit free-run timer is latched and an interrupt request is generated. LIN sync break/sync field linkage: MB91F583AM/F584AM/F585AM: Input capture 0ch → Multi-function serial 0ch Input capture 1ch → Multi-function serial 1ch Input capture 2ch → Multi-function serial 2ch Input capture 3ch → Multi-function serial 3ch MB91F583AS/F584AS/F585AS: Input capture 0ch → Multi-function serial 0ch Input capture 1ch → Multi-function serial 1ch
Output compare (7ch)	An interrupt signal is output during collating with the 16-bit free-run timer.
Waveform generator (2 units (12ch))	- Generates various outputs: Real-time output 16-bit PPG waveform output Non-overlap 3-phase waveform output (for inverter control) DC chopper waveform output - Built-in dead time timer function - Built-in GATE function - Built-in DTTI function - Built-in output inversion control function - Maximum output pulse width: 32MHz
Reload timer (4ch.)	16-bit reload timer operation (Toggle output or one-shop output can be selected.) Event count function can be selected.
PPG (6 ch.)	Used for one-shot (square wave) output and PWM (Pulse Width Modulation) output. The cycle and duty can be changed by the software. Operating clock frequency: Selectable from PCLK × 1, $1/2^2$, $1/2^4$, or $1/2^6$
Delayed interrupt	An interrupt for task switching is generated. The CPU interrupt request can be generated or canceled by the software.
External interrupt	MB91F583AM/F584AM/F585AM: 8 channels, independent MB91F583AS/F584AS/F585AS: 7 channels, independent Interrupt cause: "L"→"H" Edge/"H"→"L" Edge/"L" level/"H" level can be selected. Support of edge input detection when returned to standby state.
A/D converter	MB91F583AM/F584AM/F585AM: 12-bit resolution × 23 ch. (3 built-in units) MB91F583AS/F584AS/F585AS: 12-bit resolution × 17 ch. (3 built-in units) An analog value can be sampled from the 24-channel or 17-channel input port. Conversion time: 1 µs External trigger activation (ADTG) Can be started by the internal timer (16-bit reload timer)

Table 2.4-1 MB91580M/S Series Functional Overview (3 / 4)

Function	Features
Multi-function Serial	MB91F583AM/F584AM/F585AM : 4 channels MB91F583AS/F584AS/F585AS : 2 channels Any of UART/CSIO/LIN interface (v2.1)/I^2C-UART functions can be selected and used. Transmission FIFO memory 64 bytes, and reception FIFO memory 64 bytes provided Receive interrupt cause (3 types) – Receive error detection (parity, overrun, and frame error) – Data which amount is set for FIFO memory can be received. – Data below the FIFO memory capacity is received, and an idle period longer than 8 clocks of baud rate clock is detected. Transmission interrupt cause (2 types) – No transmission is in progress. – Empty transmission FIFO memory (including the time of transmission) SPI (Serial Peripheral Interface) supported LIN protocol Revision 2.1 supported
Up/ down counter (2ch)	8/16-bit up/down counter
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	MB91F583AM/F584AM/F585AM : 2 channels MB91F583AS/F584AS/F585AS : 1 channel CAN Specifications Version 2.0, Part A and Part B satisfied 64 message buffers Each message object has a unique identifier mask. Up to 1Mbps is supported. The clock has the CAN prescaler. CAN wake up function
FlexRay controller	MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ/F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ : 1unit ch.A/ch.B FlexRay Specifications Version 2.1 supported Up to 128 message buffers 8K bytes of message RAM Variable length of message buffers Each message buffer can be allocated as a part of reception buffer, transmission buffer or reception FIFO Host access to the message buffer via input and output buffers Filtering for slot counter, cycle counter and channels Maskable interrupts are supported
D/A converter	10-bit resolution × 1 ch.
Calibration	CR oscillation frequency can be corrected by comparison with the main clock.
Software watchdog	Counts the CPU program operations. Stops counting when the CPU is stopped. Any of 16 cycles, PCLK × (2^9 to 2^{24}) can be selected.
Hardware watchdog	CR-based CPU operation detect counter Used against program overrun. Cycle: 18.6 to 22.8 ms Calibration can be performed with the "WDT1 calibration" circuit. The range in the cycle is due to variations in production. Note that the cycle cannot be arbitrarily set.
CRC generation	When data is sequentially written in the input registers, the CRC code is displayed in the result register.

Table 2.4-1 MB91580M/S Series Functional Overview (4 / 4)

Function	Features
Low-voltage detection (External low-voltage detection)	A reset is generated when a low-voltage is detected.
Internal low-voltage detection	The voltages of 1.2V are monitored. A reset is generated when a low-voltage is detected. The internal power supply voltage is monitored to set a flag when a low-voltage is detected. If the flag is set, the contents of RAM are not guaranteed.
Low-power consumption mode	Sleep mode Stop mode Watch mode Stop mode (Power shut-off) Watch mode (Power shut-off)
I/O relocation	MB91F583AM/F584AM/F585AM only Relocation target peripheral functions and number of branches - Multi-function serial ch.1 to ch.3: no branch, ch.0: 2 branches.
NMI request	Non-maskable interrupt signal that is entered from NMIX pin or interrupt signal for the functional safety.
Debug interface	Built-in OCD (On Chip Debug Unit)

2.5 Block Diagram

Figure 2.5-1 Block Diagram (MB91F583AM/F584AM/F585AM)

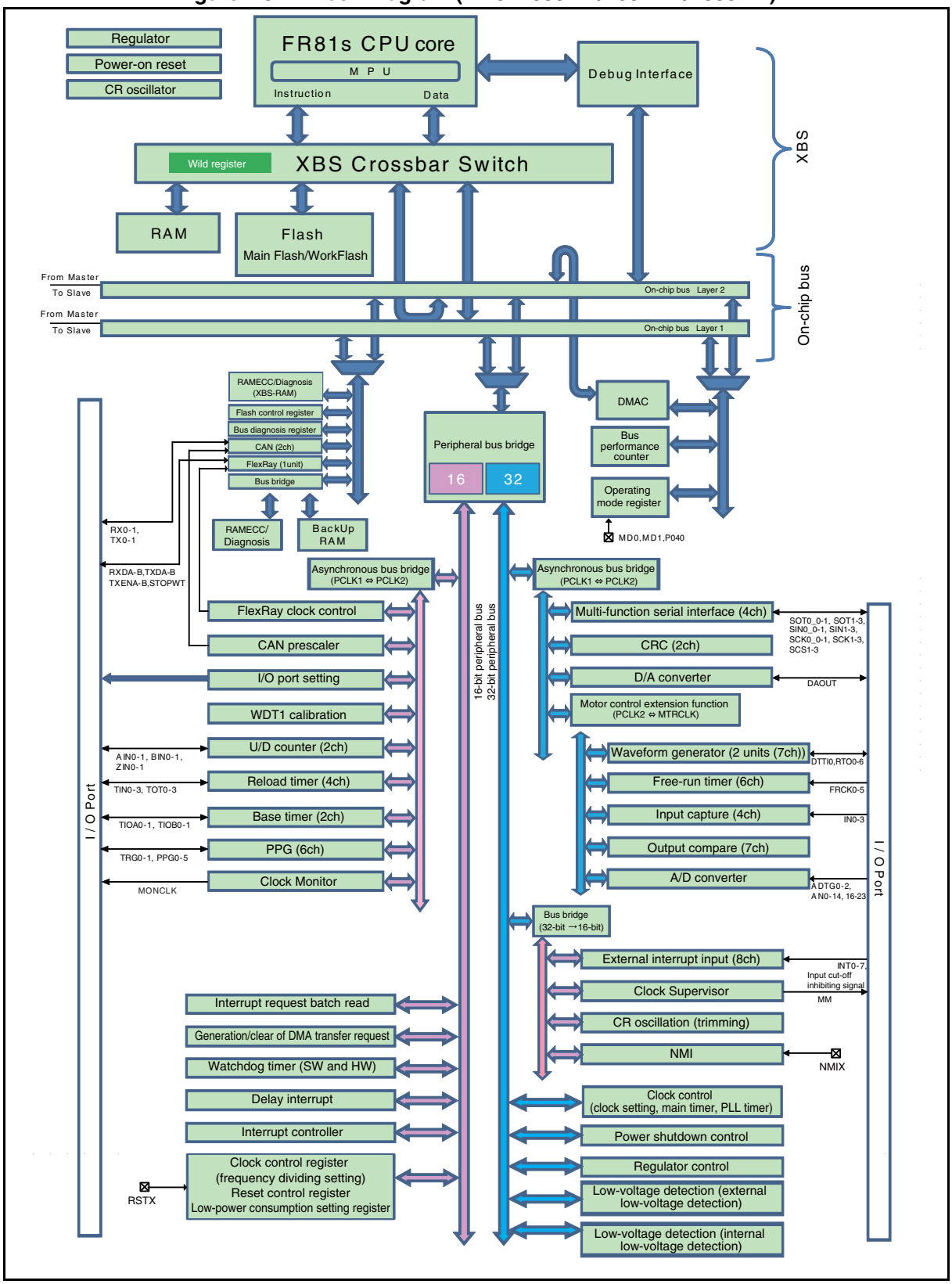
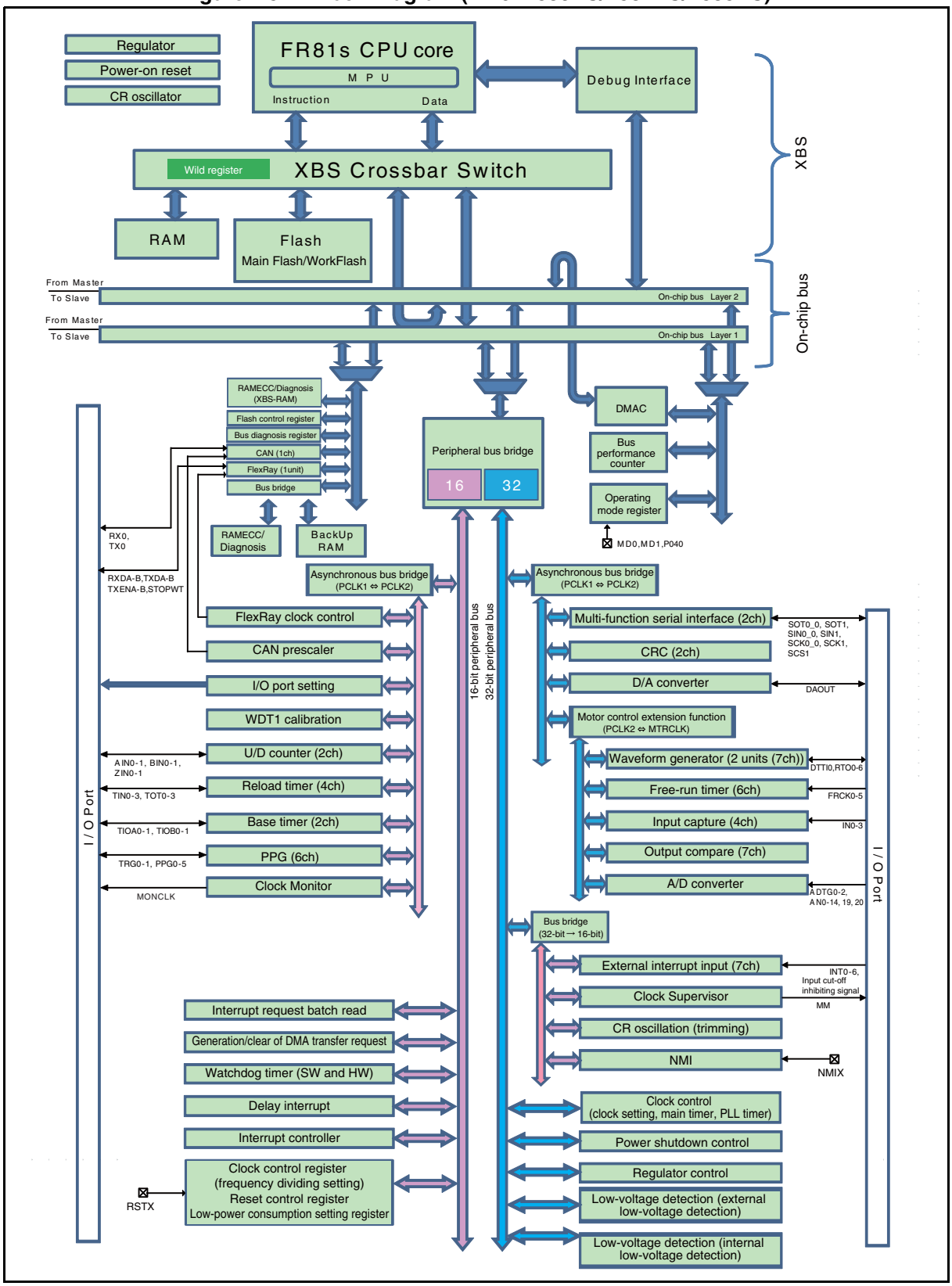


Figure 2.5-2 Block Diagram (MB91F583AS/F584AS/F585AS)



2.6 CPU

2.6.1 General-purpose Registers

Figure 2.6-1 General-purpose Registers

	32-bit	Initial value
R0		Undefined
R1		Undefined
R2		Undefined
R3		Undefined
R4		Undefined
R5		Undefined
R6		Undefined
R7		Undefined
R8		Undefined
R9		Undefined
R10		Undefined
R11		Undefined
R12		Undefined
R13	Accumulator (AC)	Undefined
R14	Frame Pointer (FP)	Undefined
R15	SSP or USP	00000000_H

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

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

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

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

2.6.2 Dedicated Registers

Nine 32-bit dedicated registers are used for various purposes. One 64-bit dedicated register is used for multiplication and division.

Figure 2.6-2 List of Dedicated Registers

		Initial value
PC		Reset entry address
PS		SSR=3_H,ILM=01111_B,SCR=$XX0_B$,CCR=$0000XXXX_B$
TBR		$000FFC00_H$
RP		Undefined
SSP		00000000_H
USP		Undefined
BP		Undefined
FCR		Undefined
ESR		00000000_H
MD		Undefined

The dedicated registers are used for special purposes.

The following dedicated registers have been prepared for the FR family.

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

2.7 Pin Assignment

Figure 2.7-1 LQFP-100Pin Assignment (MB91F583AM/F584AM/F585AM)

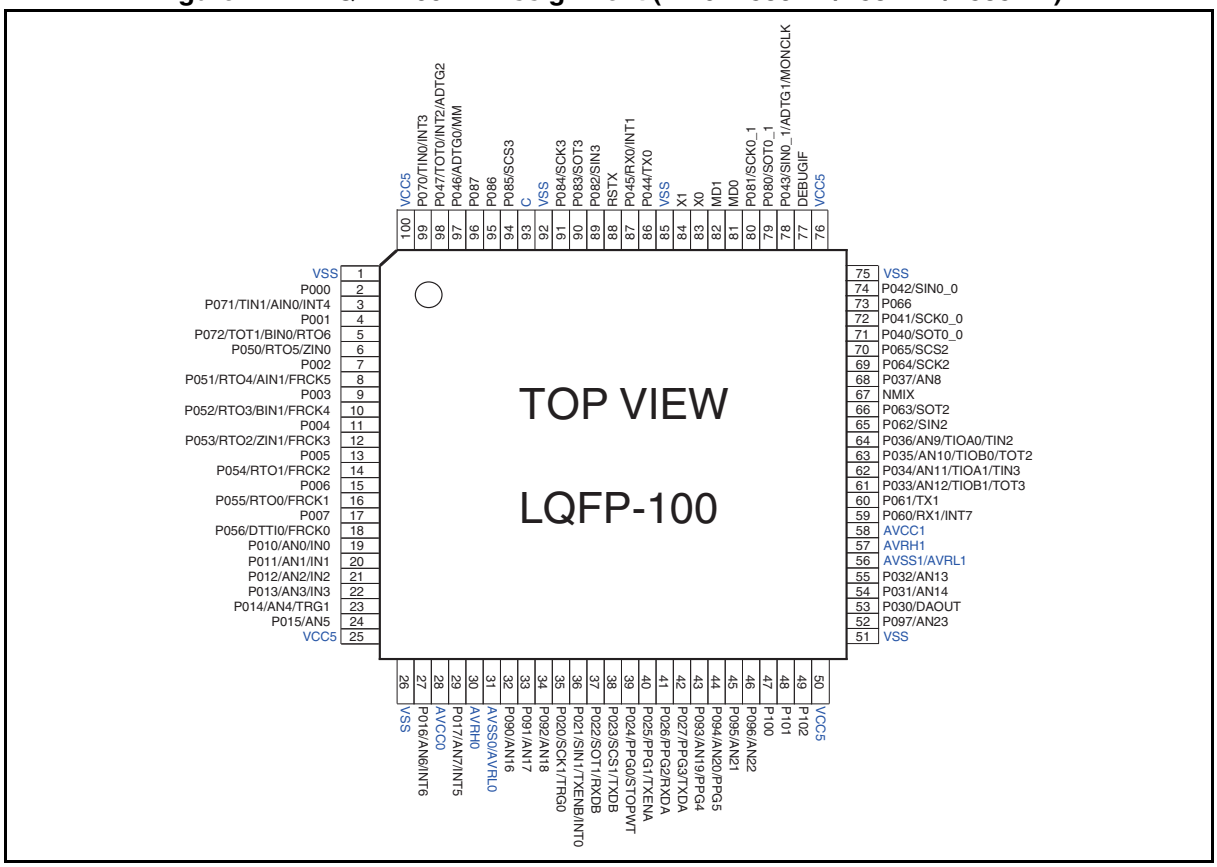
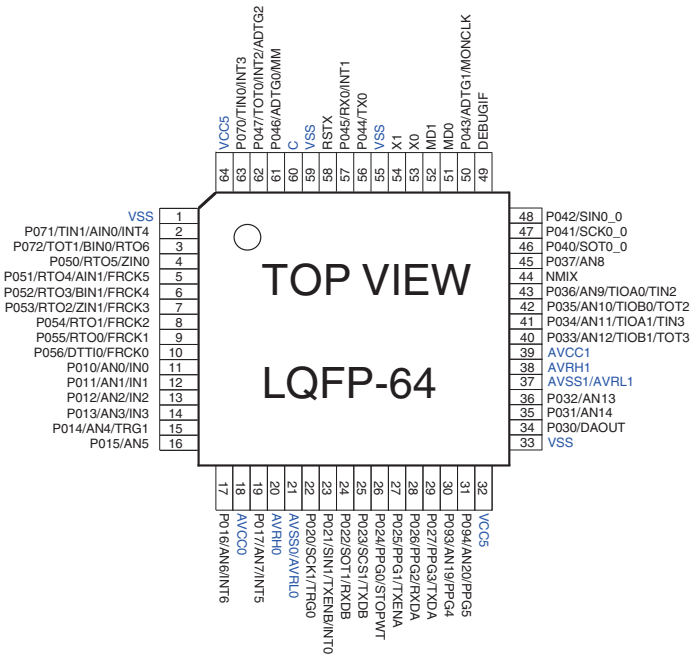


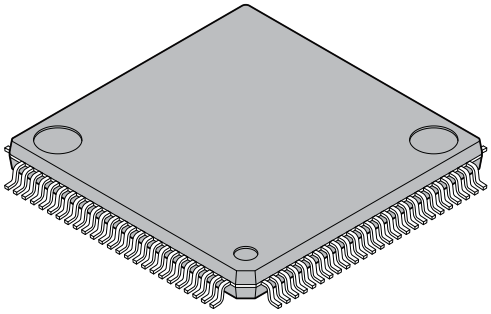
Figure 2.7-2 LQFP-64Pin Assignment (MB91F583AS/F584AS/F585AS)

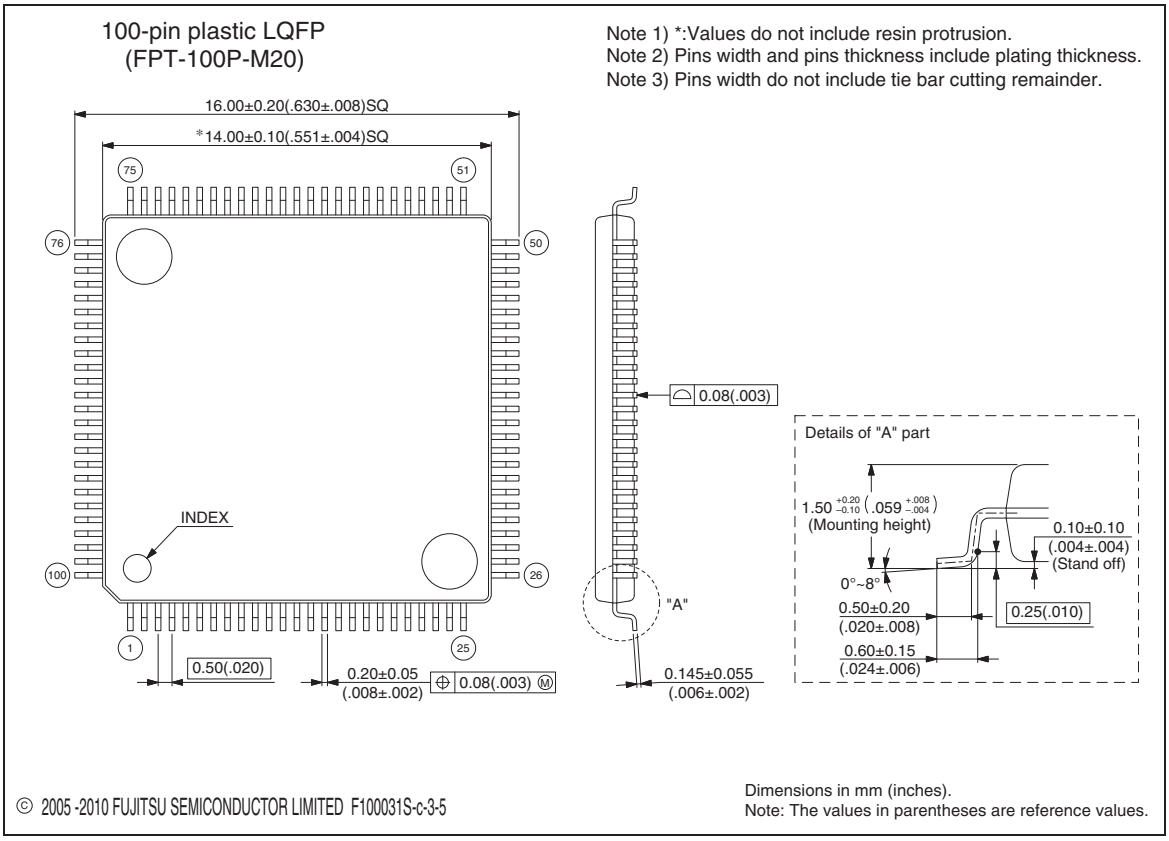


2.8 Device Package

2.8.1 MB91F583AM/F584AM/F585AM -- LQFP-100 (FPT-100P-M20) Package Dimensions

Figure 2.8-1 FPT-100P-M20 Package Dimensions

100-pin plastic LQFP		
	Lead pitch	0.50 mm
	Package width × package length	14.0 × 14.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.65 g
(FPT-100P-M20)	Code (Reference)	P-LFQFP100-14×14-0.50

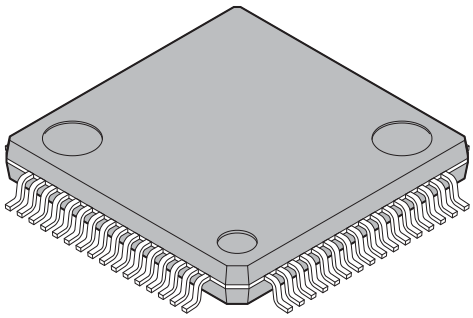


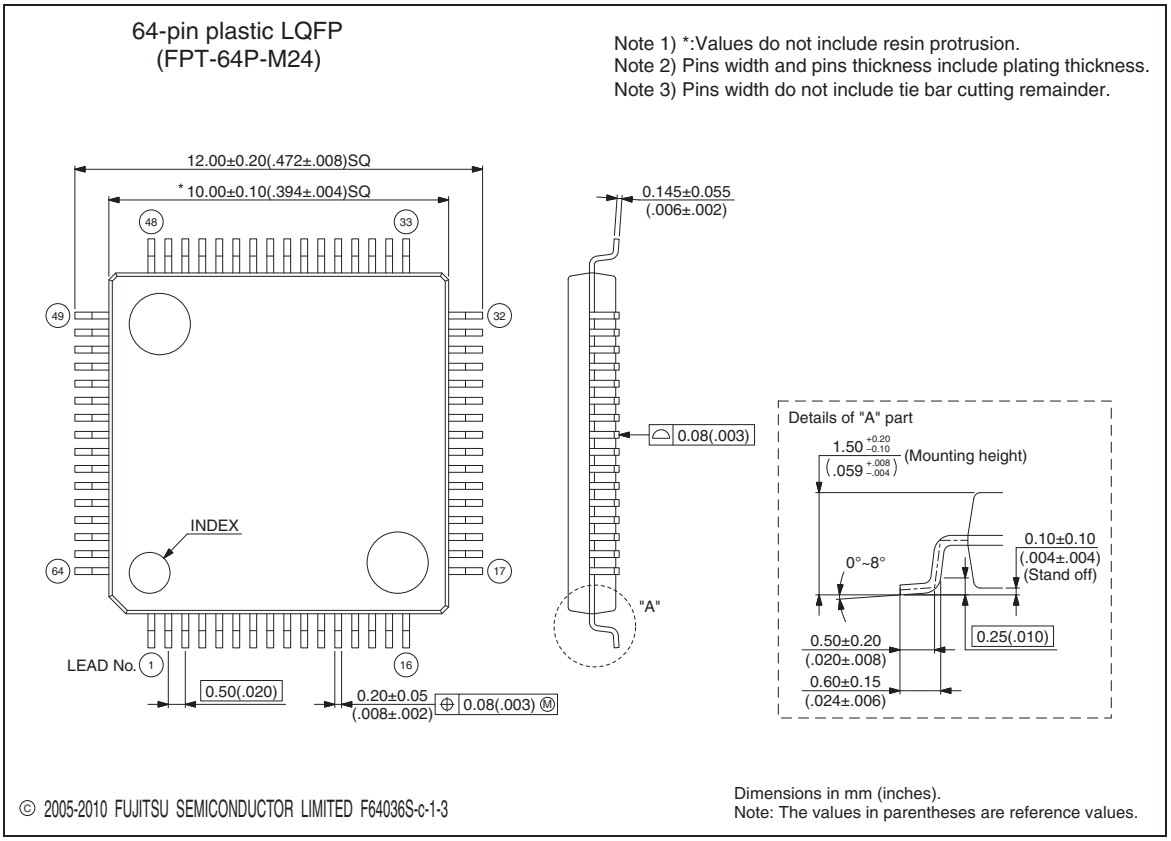
Please check the latest package dimension at the following URL.

http://edevice.fujitsu.com/package/en-search/

2.8.2 MB91F583AS/F584AS/F585AS -- LQFP-64 (FPT-64P-M24) Package Dimensions

Figure 2.8-2 FPT-64P-M24 Package Dimensions

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



Please check the latest package dimension at the following URL.

http://edevice.fujitsu.com/package/en-search/

CHAPTER 3 BASIC INFORMATION

3.1 List of Pin Functions

* For I/O circuit type, see Section "3.4 I/O circuit type"

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (1 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
83	X0	A	Main clock oscillation input pin
84	X1		Main clock oscillation output pin
67	NMIX	B	Interrupt input without mask pin
88	RSTX	B	External reset input pin
81	MD0	C	Mode pin 0 (with high-voltage control)
82	MD1	C	Mode pin 1 (with high-voltage control)
2	P000	D	General-purpose I/O ports
4	P001	D	General-purpose I/O ports
7	P002	D	General-purpose I/O ports
9	P003	D	General-purpose I/O ports
11	P004	D	General-purpose I/O ports
13	P005	D	General-purpose I/O ports
15	P006	D	General-purpose I/O ports
17	P007	D	General-purpose I/O ports
19	P010	F	General-purpose I/O ports
	IN0		16-bit input capture ch.0 external pulse input pin
	AN0		ADC analog 0 input pin
20	P011	F	General-purpose I/O ports
	IN1		16-bit input capture ch.1 external pulse input pin
	AN1		ADC analog 1 input pin
21	P012	F	General-purpose I/O ports
	IN2		16-bit input capture ch.2 external pulse input pin
	AN2		ADC analog 2 input pin
22	P013	F	General-purpose I/O ports
	IN3		16-bit input capture ch.3 external pulse input pin
	AN3		ADC analog 3 input pin
23	P014	F	General-purpose I/O ports
	TRG1		PPG ch.4, ch.5 external trigger
	AN4		ADC analog 4 input pin
24	P015	F	General-purpose I/O ports
	AN5		ADC analog 5 input pin
27	P016	G	General-purpose I/O ports
	AN6		ADC analog 6 input pin
	INT6		INT6 external interrupt input pin

Code: BASICINFO-1v0-91580MS-1-E

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (2 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
29	P017	G	General-purpose I/O ports
	AN7		ADC analog 7 input pin
	INT5		INT5 external interrupt input pin
35	P020	D	General-purpose I/O ports
	SCK1		Multi-function serial ch.1 clock I/O pin
	TRG0		PPG ch.0 to ch.3 external trigger
36	P021	L	General-purpose I/O ports
	SIN1		Multi-function serial ch.1 serial data input pin
	TXENB		FlexRay ch.B operation enable output pin
	INT0		INT0 external interrupt input pin
37	P022	K	General-purpose I/O ports
	SOT1		Multi-function serial ch.1 serial data output pin
	RXDB		FlexRay ch.B data input pin
38	P023	K	General-purpose I/O ports
	SCS1		Multi-function serial ch.1 serial chip select I/O pin
	TXDB		FlexRay ch.B data output pin
39	P024	D	General-purpose I/O ports
	PPG0		PPG ch.0 output pin
	STOPWT		FlexRay Stopwatch input pin
40	P025	K	General-purpose I/O ports
	PPG1		PPG ch.1 output pin
	TXENA		FlexRay ch.A operation enable output pin
41	P026	K	General-purpose I/O ports
	PPG2		PPG ch.2 output pin
	RXDA		FlexRay ch.A data input pin
42	P027	K	General-purpose I/O ports
	PPG3		PPG ch.3 output pin
	TXDA		FlexRay ch.A data output pin
53	P030	M	General-purpose I/O ports
	DAOUT		DAC analog output pin
54	P031	F	General-purpose I/O ports
	AN14		ADC analog 14 input pin
55	P032	F	General-purpose I/O ports
	AN13		ADC analog 13 input pin
61	P033	F	General-purpose I/O ports
	TIOB1		Base timer ch.1 TIOB input pin
	TOT3		Reload timer ch.3 output pin
	AN12		ADC analog 12 input pin

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (3 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
62	P034	F	General-purpose I/O ports
	TIOA1		Base timer ch.1 TIOA I/O pin
	TIN3		Reload timer ch.3 event input pin
	AN11		ADC analog 11 input pin
63	P035	F	General-purpose I/O ports
	TIOB0		Base timer ch.0 TIOB input pin
	TOT2		Reload timer ch.2 output pin
	AN10		ADC analog 10 input pin
64	P036	F	General-purpose I/O ports
	TIOA0		Base timer ch.0 TIOA output pin
	TIN2		Reload timer ch.2 event input pin
	AN9		ADC analog 9 input pin
68	P037	F	General-purpose I/O ports
	AN8		ADC analog 8 input pin
71	P040	H	General-purpose I/O ports
	SOT0_0		Multi-function serial ch.0 serial data output pin (0)/I^2C ch.0 serial data I/O pin (SDA)
72	P041	H	General-purpose I/O ports
	SCK0_0		Multi-function serial ch.0 clock I/O pin (0)/I^2C ch.0 clock I/O pin (SCL)
74	P042	D	General-purpose I/O ports
	SIN0_0		Multi-function serial ch.0 serial data input pin (0)
78	P043	D	General-purpose I/O ports
	SIN0_1		Multi-function serial ch.0 serial data input pin (1)
	ADTG1		A/D converter ch.8 to ch.14 external trigger input pin
	MONCLK		Clock monitor output pin
86	P044	D	General-purpose I/O ports
	TX0		CAN transmission data 0 output pin
87	P045	E	General-purpose I/O ports
	RX0		CAN reception data 0 input pin
	INT1		INT1 external interrupt input pin
97	P046	D	General-purpose I/O ports
	ADTG0		A/D converter ch.0 to ch.7 external trigger input pin
	MM		Clock supervisor main clock stop detection output pin
98	P047	E	General-purpose I/O ports
	TOT0		Reload timer ch.0 output pin
	INT2		INT2 external interrupt input pin
	ADTG2		A/D converter ch.16-ch.23 external trigger input pin
6	P050	D	General-purpose I/O ports
	RTO5		Waveform generator ch.5 output pin
	ZIN0		Up/down counter ch.0 ZIN input pin

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (4 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
8	P051	D	General-purpose I/O ports
	RTO4		Waveform generator ch.4 output pin
	AIN1		Up/down counter ch.1 AIN input pin
	FRCK5		Free-run timer ch.5 external clock input pin
10	P052	D	General-purpose I/O ports
	RTO3		Waveform generator ch.3 output pin
	BIN1		Up/down counter ch.1 BIN input pin
	FRCK4		Free-run timer ch.4 external clock input pin
12	P053	D	General-purpose I/O ports
	RTO2		Waveform generator ch.2 output pin
	ZIN1		Up/down counter ch.1 ZIN input pin
	FRCK3		Free-run timer ch.3 external clock input pin
14	P054	D	General-purpose I/O ports
	RTO1		Waveform generator ch.1 output pin
	FRCK2		Free-run timer ch.2 external clock input pin
16	P055	D	General-purpose I/O ports
	RTO0		Waveform generator ch.0 output pin
	FRCK1		Free-run timer ch.1 external clock input pin
18	P056	D	General-purpose I/O ports
	DTTI0		Waveform generator output stop signal input pin 0
	FRCK0		Free-run timer ch.0 external clock input pin
59	P060	E	General-purpose I/O ports
	RX1		CAN reception data 1 input pin
	INT7		INT7 external interrupt input pin
60	P061	D	General-purpose I/O ports
	TX1		CAN transmission data 1 output pin
65	P062	D	General-purpose I/O ports
	SIN2		Multi-function serial ch.2 serial data input pin
66	P063	H	General-purpose I/O ports
	SOT2		Multi-function serial ch.2 serial data output pin/I^2C ch.2 serial data I/O pin (SDA)
69	P064	H	General-purpose I/O ports
	SCK2		Multi-function serial ch.2 clock I/O pin/I^2C ch.2 clock I/O pin (SCL)
70	P065	D	General-purpose I/O ports
	SCS2		Multi-function serial ch.2 serial chip select I/O pin
73	P066	D	General-purpose I/O ports
99	P070	E	General-purpose I/O ports
	TIN0		Reload timer ch.0 event input pin
	INT3		INT3 external interrupt input pin

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (5 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
3	P071	E	General-purpose I/O ports
	TIN1		Reload timer ch.1 event input pin
	AIN0		Up/down counter ch.0 AIN input pin
	INT4		INT4 external interrupt input pin
5	P072	D	General-purpose I/O ports
	TOT1		Reload timer ch.1 output pin
	BIN0		Up/down counter ch.0 BIN input pin
	RTO6		Waveform generator ch.6 output pin
79	P080	H	General-purpose I/O ports
	SOT0_1		Multi-function serial ch.0 serial data output pin (1)/I^2C ch.0 serial data I/O pin (1) (SDA)
80	P081	H	General-purpose I/O ports
	SCK0_1		Multi-function serial ch.0 clock I/O pin (1)/I^2C ch.0 clock I/O pin (1) (SCL)
89	P082	D	General-purpose I/O ports
	SIN3		Multi-function serial ch.3 serial data input pin
90	P083	H	General-purpose I/O ports
	SOT3		Multi-function serial ch.3 serial data output pin/I^2C ch.3 serial data I/O pin (SDA)
91	P084	H	General-purpose I/O ports
	SCK3		Multi-function serial ch.3 clock I/O pin/I^2C ch.3 clock I/O pin (SCL)
94	P085	D	General-purpose I/O ports
	SCS3		Multi-function serial ch.3 serial chip select I/O pin
95	P086	D	General-purpose I/O ports
96	P087	D	General-purpose I/O ports
32	P090	F	General-purpose I/O ports
	AN16		ADC analog 16 input pin
33	P091	F	General-purpose I/O ports
	AN17		ADC analog 17 input pin
34	P092	F	General-purpose I/O ports
	AN18		ADC analog 18 input pin
43	P093	F	General-purpose I/O ports
	PPG4		PPG ch.4 output pin
	AN19		ADC analog 19 input pin
44	P094	F	General-purpose I/O ports
	PPG5		PPG ch.5 output pin
	AN20		ADC analog 20 input pin
45	P095	F	General-purpose I/O ports
	AN21		ADC analog 21 input pin
46	P096	F	General-purpose I/O ports
	AN22		ADC analog 22 input pin

Table 3.1-1 Pin Functions MB91F583AM/F584AM/F585AM (6 / 6)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
52	P097	F	General-purpose I/O ports
	AN23		ADC analog 23 input pin
47	P100	D	General-purpose I/O ports
48	P101	D	General-purpose I/O ports
49	P102	D	General-purpose I/O ports
77	DEBUGIF	I	DEBUG I/F pin
28	AVCC0	-	A/D converter analog power supply
58	AVCC1	-	A/D converter analog power supply
30	AVRH0	-	A/D converter upper limit reference voltage
57	AVRH1	-	A/D converter upper limit reference voltage
31	AVSS0	-	A/D converter GND
	AVRL0		A/D converter lower limit reference voltage
56	AVSS1	-	A/D converter GND
	AVRL1		A/D converter lower limit reference voltage
93	C	-	External capacity connection output pin
25, 50, 76, 100	VCC5	-	+5.0v power supply
1, 26, 51, 75, 85, 92	VSS	-	GND

Table 3.1-2 Pin Functions MB91F583AS/F584AS/F585AS (1 / 4)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
53	X0	A	Main clock oscillation input pin
54	X1		Main clock oscillation output pin
44	NMIX	B	Interrupt input without mask pin
58	RSTX	B	External reset input pin
51	MD0	C	Mode pin 0 (with high-voltage control)
52	MD1	C	Mode pin 1 (with high-voltage control)
11	P010	F	General-purpose I/O ports
	IN0		16-bit input capture ch.0 external pulse input pin
	AN0		ADC analog 0 input pin
12	P011	F	General-purpose I/O ports
	IN1		16-bit input capture ch.1 external pulse input pin
	AN1		ADC analog 1 input pin
13	P012	F	General-purpose I/O ports
	IN2		16-bit input capture ch.2 external pulse input pin
	AN2		ADC analog 2 input pin
14	P013	F	General-purpose I/O ports
	IN3		16-bit input capture ch.3 external pulse input pin
	AN3		ADC analog 3 input pin
15	P014	F	General-purpose I/O ports
	TRG1		PPG ch.4, ch.5 external trigger
	AN4		ADC analog 4 input pin
16	P015	F	General-purpose I/O ports
	AN5		ADC analog 5 input pin
17	P016	G	General-purpose I/O ports
	AN6		ADC analog 6 input pin
	INT6		INT6 external interrupt input pin
19	P017	G	General-purpose I/O ports
	AN7		ADC analog 7 input pin
	INT5		INT5 external interrupt input pin
22	P020	D	General-purpose I/O ports
	SCK1		Multi-function serial ch.1 clock I/O pin
	TRG0		PPG ch.0 to ch.3 external trigger
23	P021	L	General-purpose I/O ports
	SIN1		Multi-function serial ch.1 serial data input pin
	TXENB		FlexRay ch.B operation enable output pin
	INT0		INT0 external interrupt input pin
24	P022	K	General-purpose I/O ports
	SOT1		Multi-function serial ch.1 serial data output pin
	RXDB		FlexRay ch.B data input pin

Table 3.1-2 Pin Functions MB91F583AS/F584AS/F585AS (2 / 4)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
25	P023	K	General-purpose I/O ports
	SCS1		Multi-function serial ch.1 serial chip select I/O pin
	TXDB		FlexRay ch.B data output pin
26	P024	D	General-purpose I/O ports
	PPG0		PPG ch.0 output pin
	STOPWT		FlexRay Stopwatch input pin
27	P025	K	General-purpose I/O ports
	PPG1		PPG ch.1 output pin
	TXENA		FlexRay ch.A operation enable output pin
28	P026	K	General-purpose I/O ports
	PPG2		PPG ch.2 output pin
	RXDA		FlexRay ch.A data input pin
29	P027	K	General-purpose I/O ports
	PPG3		PPG ch.3 output pin
	TXDA		FlexRay ch.A data output pin
34	P030	M	General-purpose I/O ports
	DAOUT		DAC analog output pin
35	P031	F	General-purpose I/O ports
	AN14		ADC analog 14 input pin
36	P032	F	General-purpose I/O ports
	AN13		ADC analog 13 input pin
40	P033	F	General-purpose I/O ports
	TIOB1		Base timer ch.1 TIOB input pin
	TOT3		Reload timer ch.3 output pin
	AN12		ADC analog 12 input pin
41	P034	F	General-purpose I/O ports
	TIOA1		Base timer ch.1 TIOA I/O pin
	TIN3		Reload timer ch.3 event input pin
	AN11		ADC analog 11 input pin
42	P035	F	General-purpose I/O ports
	TIOB0		Base timer ch.0 TIOB input pin
	TOT2		Reload timer ch.2 output pin
	AN10		ADC analog 10 input pin
43	P036	F	General-purpose I/O ports
	TIOA0		Base timer ch.0 TIOA output pin
	TIN2		Reload timer ch.2 event input pin
	AN9		ADC analog 9 input pin
45	P037	F	General-purpose I/O ports
	AN8		ADC analog 8 input pin

Table 3.1-2 Pin Functions MB91F583AS/F584AS/F585AS (3 / 4)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
46	P040	H	General-purpose I/O ports
	SOT0_0		Multi-function serial ch.0 serial data output pin(0)/ I^2C ch.0 serial data I/O pin (0) (SDA)
47	P041	H	General-purpose I/O ports
	SCK0_0		Multi-function serial ch.0 clock I/O pin (0)/ I^2C ch.0 clock I/O pin (0) (SCL)
48	P042	D	General-purpose I/O ports
	SIN0_0		Multi-function serial ch.0 serial data input pin (0)
50	P043	D	General-purpose I/O ports
	ADTG1		A/D converter ch.8 to ch.14 external trigger input pin
	MONCLK		Clock monitor output pin
56	P044	D	General-purpose I/O ports
	TX0		CAN transmission data 0 output pin
57	P045	E	General-purpose I/O ports
	RX0		CAN reception data 0 input pin
	INT1		INT1 external interrupt input pin
61	P046	D	General-purpose I/O ports
	ADTG0		A/D converter ch.0 to ch.7 external trigger input pin
	MM		Clock supervisor main clock stop detection output pin
62	P047	E	General-purpose I/O ports
	TOT0		Reload timer ch.0 output pin
	INT2		INT2 external interrupt input pin
	ADTG2		A/D converter ch.19 to ch.20 external trigger input pin
4	P050	D	General-purpose I/O ports
	RTO5		Waveform generator ch.5 output pin
	ZIN0		Up/down counter ch.0 ZIN input pin
5	P051	D	General-purpose I/O ports
	RTO4		Waveform generator ch.4 output pin
	AIN1		Up/down counter ch.1 AIN input pin
	FRCK5		Free-run timer ch.5 external clock input pin
6	P052	D	General-purpose I/O ports
	RTO3		Waveform generator ch.3 output pin
	BIN1		Up/down counter ch.1 BIN input pin
	FRCK4		Free-run timer ch.4 external clock input pin
7	P053	D	General-purpose I/O ports
	RTO2		Waveform generator ch.2 output pin
	ZIN1		Up/down counter ch.1 ZIN input pin
	FRCK3		Free-run timer ch.3 external clock input pin
8	P054	D	General-purpose I/O ports
	RTO1		Waveform generator ch.1 output pin
	FRCK2		Free-run timer ch.2 external clock input pin

Table 3.1-2 Pin Functions MB91F583AS/F584AS/F585AS (4 / 4)

Pin No.	Pin name	I/O circuit type*	Function (For switching, see "CHAPTER 24 I/O PORTS".)
9	P055	D	General-purpose I/O ports
	RTO0		Waveform generator ch.0 output pin
	FRCK1		Free-run timer ch.1 external clock input pin
10	P056	D	General-purpose I/O ports
	DTTI0		Waveform generator output stop signal input pin 0
	FRCK0		Free-run timer ch.0 external clock input pin
63	P070	E	General-purpose I/O ports
	TIN0		Reload timer ch.0 event input pin
	INT3		INT3 external interrupt input pin
2	P071	E	General-purpose I/O ports
	TIN1		Reload timer ch.1 event input pin
	AIN0		Up/down counter ch.0 AIN input pin
	INT4		INT4 external interrupt input pin
3	P072	D	General-purpose I/O ports
	TOT1		Reload timer ch.1 output pin
	BIN0		Up/down counter ch.0 BIN input pin
	RTO6		Waveform generator ch.6 output pin
30	P093	F	General-purpose I/O ports
	PPG4		PPG ch.4 output pin
	AN19		ADC analog 19 input pin
31	P094	F	General-purpose I/O ports
	PPG5		PPG ch.5 output pin
	AN20		ADC analog 20 input pin
49	DEBUGIF	I	DEBUG I/F pin
18	AVCC0	-	A/D converter analog power supply
39	AVCC1	-	A/D converter analog power supply
20	AVRH0	-	A/D converter upper limit reference voltage
38	AVRH1	-	A/D converter upper limit reference voltage
21	AVSS0	-	A/D converter GND
	AVRL0		A/D converter lower limit reference voltage
37	AVSS1	-	A/D converter GND
	AVRL1		A/D converter lower limit reference voltage
60	C	-	External capacity connection output pin
32, 64	VCC5	-	+5.0v power supply
1, 33, 55, 59	VSS	-	GND

* For I/O circuit type, see Section "3.4 I/O circuit type"

3.2 Pins of Each Function MB91F583AM/F584AM/F585AM

■ A/D Converter Pins (3 units (23 channels))

- A/D converter external trigger 0 input (Pin name) ADTG0 (Pin No.) 97
- A/D converter external trigger 1 input (Pin name) ADTG1 (Pin No.) 78
- A/D converter external trigger 2 input (Pin name) ADTG2 (Pin No.) 98
- A/D converter analog 0 input (Pin name) AN0 (Pin No.) 19
- A/D converter analog 1 input (Pin name) AN1 (Pin No.) 20
- A/D converter analog 2 input (Pin name) AN2 (Pin No.) 21
- A/D converter analog 3 input (Pin name) AN3 (Pin No.) 22
- A/D converter analog 4 input (Pin name) AN4 (Pin No.) 23
- A/D converter analog 5 input (Pin name) AN5 (Pin No.) 24
- A/D converter analog 6 input (Pin name) AN6 (Pin No.) 27
- A/D converter analog 7 input (Pin name) AN7 (Pin No.) 29
- A/D converter analog 8 input (Pin name) AN8 (Pin No.) 68
- A/D converter analog 9 input (Pin name) AN9 (Pin No.) 64
- A/D converter analog 10 input (Pin name) AN10 (Pin No.) 63
- A/D converter analog 11 input (Pin name) AN11 (Pin No.) 62
- A/D converter analog 12 input (Pin name) AN12 (Pin No.) 61
- A/D converter analog 13 input (Pin name) AN13 (Pin No.) 55
- A/D converter analog 14 input (Pin name) AN14 (Pin No.) 54
- A/D converter analog 16 input (Pin name) AN16 (Pin No.) 32
- A/D converter analog 17 input (Pin name) AN17 (Pin No.) 33
- A/D converter analog 18 input (Pin name) AN18 (Pin No.) 34
- A/D converter analog 19 input (Pin name) AN19 (Pin No.) 43
- A/D converter analog 20 input (Pin name) AN20 (Pin No.) 44
- A/D converter analog 21 input (Pin name) AN21 (Pin No.) 45
- A/D converter analog 22 input (Pin name) AN22 (Pin No.) 46
- A/D converter analog 23 input (Pin name) AN23 (Pin No.) 52
- A/D converter GND (Pin name) AVSS 0 (Pin No.) 31
- A/D converter GND (Pin name) AVSS 1 (Pin No.) 56
- A/D converter analog power supply (Pin name) AVCC0 (Pin No.) 28
- A/D converter analog power supply (Pin name) AVCC1 (Pin No.) 58
- A/D converter lower limit reference voltage (Pin name) AVRL0 (Pin No.) 31
- A/D converter lower limit reference voltage (Pin name) AVRL1 (Pin No.) 56
- A/D converter upper limit reference voltage (Pin name) AVRH0 (Pin No.) 30
- A/D converter upper limit reference voltage (Pin name) AVRH1 (Pin No.) 57

■ CAN Pins (ch.0, ch.1)

- CAN reception data 0 input (Pin name) RX0 (Pin No.) 87
- CAN reception data 1 input (Pin name) RX1 (Pin No.) 59
- CAN transmission data 0 output (Pin name) TX0 (Pin No.) 86
- CAN transmission data 1 output (Pin name) TX1 (Pin No.) 60

■ External Interrupt Input Pins (ch.0 to ch.7)

- INT0 external interrupt input (Pin name) INT0 (Pin No.) 36
- INT1 external interrupt input (Pin name) INT1 (Pin No.) 87
- INT2 external interrupt input (Pin name) INT2 (Pin No.) 98
- INT3 external interrupt input (Pin name) INT3 (Pin No.) 99
- INT4 external interrupt input (Pin name) INT4 (Pin No.) 3
- INT5 external interrupt input (Pin name) INT5 (Pin No.) 29
- INT6 external interrupt input (Pin name) INT6 (Pin No.) 27
- INT7 external interrupt input (Pin name) INT7 (Pin No.) 59

■ Multi-function serial interface Pins (ch.0 to ch.3)

- Multi-function serial interface ch.0 clock input/output (0) /I^2C ch.0 clock input/output (0)
  (Pin name) SCK0_0 (Pin No.) 72
- Multi-function serial interface ch.0 serial data output (0) /I^2C ch.0 serial data input/output (0)
  (Pin name) SOT0_0 (Pin No.) 71
- Multi-function serial interface ch.0 serial data input (0) (Pin name) SIN0_0 (Pin No.) 74
- Multi-function serial interface ch.0 clock input/output (1) /I^2C ch.0 clock input/output (1)
  (Pin name) SCK0_1 (Pin No.) 80
- Multi-function serial interface ch.0 serial data output (1) /I^2C ch.0 serial data input/output (1)
  (Pin name) SOT0_1 (Pin No.) 79
- Multi-function serial interface ch.0 serial data input (1) (Pin name) SIN0_1 (Pin No.) 78
- Multi-function serial interface ch.1 serial chip select input/output
  (Pin name)SCS1 (Pin No.) 38
- Multi-function serial interface ch.1 clock input/output (Pin name) SCK1 (Pin No.) 35
- Multi-function serial interface ch.1 serial data output (Pin name) SOT1 (Pin No.) 37
- Multi-function serial interface ch.1 serial data input (Pin name) SIN1 (Pin No.) 36
- Multi-function serial interface ch.2 serial chip select input/output
  (Pin name) SCS2 (Pin No.) 70
- Multi-function serial interface ch.2 clock input/output /I^2C ch.2 clock input/output
  (Pin name) SCK2 (Pin No.) 69
- Multi-function serial interface ch.2 serial data output /I^2C ch.2 serial data input/output
  (Pin name) SOT2 (Pin No.) 66
- Multi-function serial interface ch.2 serial data input (Pin name) SIN2 (Pin No.) 65
- Multi-function serial interface ch.3 serial chip select input/output
  (Pin name) SCS3 (Pin No.) 94
- Multi-function serial interface ch.3 clock input/output /I^2C ch.3 clock input/output
  (Pin name) SCK3 (Pin No.) 91
- Multi-function serial interface ch.3 serial data output /I^2C ch.3 serial data input/output
  (Pin name) SOT3 (Pin No.) 90
- Multi-function serial interface ch.3 serial data input (Pin name) SIN3 (Pin No.) 89

■ PPG Pins (ch.0 to ch.5)

• PPG ch.0 output	(Pin name) PPG0	(Pin No.) 39
• PPG ch.1 output	(Pin name) PPG1	(Pin No.) 40
• PPG ch.2 output	(Pin name) PPG2	(Pin No.) 41
• PPG ch.3 output	(Pin name) PPG3	(Pin No.) 42
• PPG ch.4 output	(Pin name) PPG4	(Pin No.) 43
• PPG ch.5 output	(Pin name) PPG5	(Pin No.) 44
• PPG trigger 0 input (ch.0 to ch.3)	(Pin name) TRG0	(Pin No.) 35
• PPG trigger 1 input (ch.4, ch.5)	(Pin name) TRG1	(Pin No.) 23

■ Input Capture Pins (ch.0 to ch.3)

• Input capture ch.0 input	(Pin name) IN0	(Pin No.) 19
• Input capture ch.1 input	(Pin name) IN1	(Pin No.) 20
• Input capture ch.2 input	(Pin name) IN2	(Pin No.) 21
• Input capture ch.3 input	(Pin name) IN3	(Pin No.) 22

■ Free-Run Timer Pins (ch.0 to ch.5)

• Free-run timer ch.0 clock input	(Pin name) FRCK0	(Pin No.) 18
• Free-run timer ch.1 clock input	(Pin name) FRCK1	(Pin No.) 16
• Free-run timer ch.2 clock input	(Pin name) FRCK2	(Pin No.) 14
• Free-run timer ch.3 clock input	(Pin name) FRCK3	(Pin No.) 12
• Free-run timer ch.4 clock input	(Pin name) FRCK4	(Pin No.) 10
• Free-run timer ch.5 clock input	(Pin name) FRCK5	(Pin No.) 8

■ Base Timer Pins (ch.0, ch.1)

• Base timer TIOA0 output	(Pin name) TIOA0	(Pin No.) 64
• Base timer TIOB0 input	(Pin name) TIOB0	(Pin No.) 63
• Base timer TIOA1 output/input	(Pin name) TIOA1	(Pin No.) 62
• Base timer TIOB1 input	(Pin name) TIOB1	(Pin No.) 61

■ Reload Timer Pins (ch.0 to ch.3)

• Reload timer ch.0 event input	(Pin name) TIN0	(Pin No.) 99
• Reload timer ch.0 output	(Pin name) TOT0	(Pin No.) 98
• Reload timer ch.1 event input	(Pin name) TIN1	(Pin No.) 3
• Reload timer ch.1 output	(Pin name) TOT1	(Pin No.) 5
• Reload timer ch.2 event input	(Pin name) TIN2	(Pin No.) 64
• Reload timer ch.2 output	(Pin name) TOT2	(Pin No.) 63
• Reload timer ch.3 event input	(Pin name) TIN3	(Pin No.) 62
• Reload timer ch.3 output	(Pin name) TOT3	(Pin No.) 61

■ Waveform Generator Pins (2 units (7ch))

- Waveform generator output stop signal input pin (Pin name) DTTI0 (Pin No.) 18
- Waveform generator ch.0 output (Pin name) RTO0 (Pin No.) 16
- Waveform generator ch.1 output (Pin name) RTO1 (Pin No.) 14
- Waveform generator ch.2 output (Pin name) RTO2 (Pin No.) 12
- Waveform generator ch.3 output (Pin name) RTO3 (Pin No.) 10
- Waveform generator ch.4 output (Pin name) RTO4 (Pin No.) 8
- Waveform generator ch.5 output (Pin name) RTO5 (Pin No.) 6
- Waveform generator ch.6 output (Pin name) RTO6 (Pin No.) 5

■ Flex Ray Pins (1unit ch.A/ch.B)

- FlexRay ch.A data input (Pin name) RXDA (Pin No.) 41
- FlexRay ch.A data output (Pin name) TXDA (Pin No.) 42
- FlexRay ch.A operation enable output (Pin name) TXENA (Pin No.) 40
- FlexRay ch.B data input (Pin name) RXDB (Pin No.) 37
- FlexRay ch.B data output (Pin name) TXDB (Pin No.) 38
- FlexRay ch.B operation enable output (Pin name) TXENB (Pin No.) 36
- FlexRay stopwatch input (Pin name) STOPWT (Pin No.) 39

■ Clock Monitor Pins

- Clock monitor output (Pin name) MONCLK (Pin No.) 78

■ Up/down Counter Pins (ch.0, ch.1)

- U/D counter ch.0 AIN input (Pin name) AIN0 (Pin No.) 3
- U/D counter ch.0 BIN input (Pin name) BIN0 (Pin No.) 5
- U/D counter ch.0 ZIN input (Pin name) ZIN0 (Pin No.) 6
- U/D counter ch.1 AIN input (Pin name) AIN1 (Pin No.) 8
- U/D counter ch.1 BIN input (Pin name) BIN1 (Pin No.) 10
- U/D counter ch.1 ZIN input (Pin name) ZIN1 (Pin No.) 12

■ Clock Supervisor Pins

- Clock stop detection output (Pin name) MM (Pin No.) 97

■ D/A Converter Pins

- D/A converter output (Pin name) DAOUT (Pin No.) 53

■ Port Function (General-Purpose I/O) Pins

- General-purpose I/O ports (Pin name) P000 (Pin No.) 2
- General-purpose I/O ports (Pin name) P001 (Pin No.) 4
- General-purpose I/O ports (Pin name) P002 (Pin No.) 7
- General-purpose I/O ports (Pin name) P003 (Pin No.) 9
- General-purpose I/O ports (Pin name) P004 (Pin No.) 11
- General-purpose I/O ports (Pin name) P005 (Pin No.) 13
- General-purpose I/O ports (Pin name) P006 (Pin No.) 15
- General-purpose I/O ports (Pin name) P007 (Pin No.) 17
- General-purpose I/O ports (Pin name) P010 (Pin No.) 19
- General-purpose I/O ports (Pin name) P011 (Pin No.) 20
- General-purpose I/O ports (Pin name) P012 (Pin No.) 21
- General-purpose I/O ports (Pin name) P013 (Pin No.) 22
- General-purpose I/O ports (Pin name) P014 (Pin No.) 23
- General-purpose I/O ports (Pin name) P015 (Pin No.) 24
- General-purpose I/O ports (Pin name) P016 (Pin No.) 27
- General-purpose I/O ports (Pin name) P017 (Pin No.) 29
- General-purpose I/O ports (Pin name) P020 (Pin No.) 35
- General-purpose I/O ports (Pin name) P021 (Pin No.) 36
- General-purpose I/O ports (Pin name) P022 (Pin No.) 37
- General-purpose I/O ports (Pin name) P023 (Pin No.) 38
- General-purpose I/O ports (Pin name) P024 (Pin No.) 39
- General-purpose I/O ports (Pin name) P025 (Pin No.) 40
- General-purpose I/O ports (Pin name) P026 (Pin No.) 41
- General-purpose I/O ports (Pin name) P027 (Pin No.) 42
- General-purpose I/O ports (Pin name) P030 (Pin No.) 53
- General-purpose I/O ports (Pin name) P031 (Pin No.) 54
- General-purpose I/O ports (Pin name) P032 (Pin No.) 55
- General-purpose I/O ports (Pin name) P033 (Pin No.) 61
- General-purpose I/O ports (Pin name) P034 (Pin No.) 62
- General-purpose I/O ports (Pin name) P035 (Pin No.) 63
- General-purpose I/O ports (Pin name) P036 (Pin No.) 64
- General-purpose I/O ports (Pin name) P037 (Pin No.) 68
- General-purpose I/O ports (Pin name) P040 (Pin No.) 71
- General-purpose I/O ports (Pin name) P041 (Pin No.) 72
- General-purpose I/O ports (Pin name) P042 (Pin No.) 74
- General-purpose I/O ports (Pin name) P043 (Pin No.) 78
- General-purpose I/O ports (Pin name) P044 (Pin No.) 86
- General-purpose I/O ports (Pin name) P045 (Pin No.) 87
- General-purpose I/O ports (Pin name) P046 (Pin No.) 97
- General-purpose I/O ports (Pin name) P047 (Pin No.) 98

- General-purpose I/O ports (Pin name) P050 (Pin No.) 6
- General-purpose I/O ports (Pin name) P051 (Pin No.) 8
- General-purpose I/O ports (Pin name) P052 (Pin No.) 10
- General-purpose I/O ports (Pin name) P053 (Pin No.) 12
- General-purpose I/O ports (Pin name) P054 (Pin No.) 14
- General-purpose I/O ports (Pin name) P055 (Pin No.) 16
- General-purpose I/O ports (Pin name) P056 (Pin No.) 18
- General-purpose I/O ports (Pin name) P060 (Pin No.) 59
- General-purpose I/O ports (Pin name) P061 (Pin No.) 60
- General-purpose I/O ports (Pin name) P062 (Pin No.) 65
- General-purpose I/O ports (Pin name) P063 (Pin No.) 66
- General-purpose I/O ports (Pin name) P064 (Pin No.) 69
- General-purpose I/O ports (Pin name) P065 (Pin No.) 70
- General-purpose I/O ports (Pin name) P066 (Pin No.) 73
- General-purpose I/O ports (Pin name) P070 (Pin No.) 99
- General-purpose I/O ports (Pin name) P071 (Pin No.) 3
- General-purpose I/O ports (Pin name) P072 (Pin No.) 5
- General-purpose I/O ports (Pin name) P080 (Pin No.) 79
- General-purpose I/O ports (Pin name) P081 (Pin No.) 80
- General-purpose I/O ports (Pin name) P082 (Pin No.) 89
- General-purpose I/O ports (Pin name) P083 (Pin No.) 90
- General-purpose I/O ports (Pin name) P084 (Pin No.) 91
- General-purpose I/O ports (Pin name) P085 (Pin No.) 94
- General-purpose I/O ports (Pin name) P086 (Pin No.) 95
- General-purpose I/O ports (Pin name) P087 (Pin No.) 96
- General-purpose I/O ports (Pin name) P090 (Pin No.) 32
- General-purpose I/O ports (Pin name) P091 (Pin No.) 33
- General-purpose I/O ports (Pin name) P092 (Pin No.) 34
- General-purpose I/O ports (Pin name) P093 (Pin No.) 43
- General-purpose I/O ports (Pin name) P094 (Pin No.) 44
- General-purpose I/O ports (Pin name) P095 (Pin No.) 45
- General-purpose I/O ports (Pin name) P096 (Pin No.) 46
- General-purpose I/O ports (Pin name) P097 (Pin No.) 52
- General-purpose I/O ports (Pin name) P100 (Pin No.) 47
- General-purpose I/O ports (Pin name) P101 (Pin No.) 48
- General-purpose I/O ports (Pin name) P102 (Pin No.) 49

■ Other Pins

- +5.0V power supply (Pin name) VCC5 (Pin No.) 25, 50, 76, 100
- GND (Pin name) VSS (Pin No.) 1, 26, 51, 75, 85, 92
- External capacity connection output (Pin name) C (Pin No.) 93
- Main clock oscillation output (Pin name) X1 (Pin No.) 84

- Main clock oscillation input (Pin name) X0 (Pin No.) 83
- Mode pin 0 (with high-voltage control) (Pin name) MD0 (Pin No.) 81
- Mode pin 1 (with high-voltage control) (Pin name) MD1 (Pin No.) 82
- NMI interrupt input (Pin name) NMIX (Pin No.) 67
- DEBUG I/F (Pin name) DEBUGIF (Pin No.) 77
- External reset input (Pin name) RSTX (Pin No.) 88

3.3 Pins of Each Function MB91F583AS/F584AS/F585AS

■ A/D Converter Pins (3 units (17 channels))

- A/D converter external trigger 0 input (Pin name) ADTG0 (Pin No.) 61
- A/D converter external trigger 1 input (Pin name) ADTG1 (Pin No.) 50
- A/D converter external trigger 2 input (Pin name) ADTG2 (Pin No.) 62
- A/D converter analog 0 input (Pin name) AN0 (Pin No.) 11
- A/D converter analog 1 input (Pin name) AN1 (Pin No.) 12
- A/D converter analog 2 input (Pin name) AN2 (Pin No.) 13
- A/D converter analog 3 input (Pin name) AN3 (Pin No.) 14
- A/D converter analog 4 input (Pin name) AN4 (Pin No.) 15
- A/D converter analog 5 input (Pin name) AN5 (Pin No.) 16
- A/D converter analog 6 input (Pin name) AN6 (Pin No.) 17
- A/D converter analog 7 input (Pin name) AN7 (Pin No.) 19
- A/D converter analog 8 input (Pin name) AN8 (Pin No.) 45
- A/D converter analog 9 input (Pin name) AN9 (Pin No.) 43
- A/D converter analog 10 input (Pin name) AN10 (Pin No.) 42
- A/D converter analog 11 input (Pin name) AN11 (Pin No.) 41
- A/D converter analog 12 input (Pin name) AN12 (Pin No.) 40
- A/D converter analog 13 input (Pin name) AN13 (Pin No.) 36
- A/D converter analog 14 input (Pin name) AN14 (Pin No.) 35
- A/D converter analog 19 input (Pin name) AN19 (Pin No.) 30
- A/D converter analog 20 input (Pin name) AN20 (Pin No.) 31
- A/D converter GND (Pin name) AVSS 0 (Pin No.) 21
- A/D converter GND (Pin name) AVSS 1 (Pin No.) 37
- A/D converter analog power supply (Pin name) AVCC0 (Pin No.) 18
- A/D converter analog power supply (Pin name) AVCC1 (Pin No.) 39
- A/D converter lower limit reference voltage (Pin name) AVRL0 (Pin No.) 21
- A/D converter lower limit reference voltage (Pin name) AVRL1 (Pin No.) 37
- A/D converter upper limit reference voltage (Pin name) AVRH0 (Pin No.) 20
- A/D converter upper limit reference voltage (Pin name) AVRH1 (Pin No.) 38

■ CAN Pins (ch.0)

- CAN reception data 0 input (Pin name) RX0 (Pin No.) 57
- CAN transmission data 0 output (Pin name) TX0 (Pin No.) 56

■ External Interrupt Input Pins (ch.0 to ch.6)

- INT0 external interrupt input (Pin name) INT0 (Pin No.) 23
- INT1 external interrupt input (Pin name) INT1 (Pin No.) 57
- INT2 external interrupt input (Pin name) INT2 (Pin No.) 62
- INT3 external interrupt input (Pin name) INT3 (Pin No.) 63
- INT4 external interrupt input (Pin name) INT4 (Pin No.) 2
- INT5 external interrupt input (Pin name) INT5 (Pin No.) 19
- INT6 external interrupt input (Pin name) INT6 (Pin No.) 17

■ Multi-function serial interface Pins (ch.0, ch.1)

- Multi-function serial interface ch.0 clock input/output (0) /I^{2}C ch.0 clock input/output (0) (Pin name) SCK0_0 (Pin No.) 47
- Multi-function serial interface ch.0 serial data output (0) /I^{2}C ch.0 serial data input/output (0) (Pin name) SOT0_0 (Pin No.) 46
- Multi-function serial interface ch.0 serial data input (0) (Pin name) SIN0_0 (Pin No.) 48
- Multi-function serial interface ch.1 serial chip select input/output (Pin name) SCS1 (Pin No.)25
- Multi-function serial interface ch.1 clock input/output (Pin name) SCK1 (Pin No.)22
- Multi-function serial interface ch.1 serial data output (Pin name) SOT1 (Pin No.)24
- Multi-function serial interface ch.1 serial data input (Pin name) SIN1 (Pin No.)23

■ PPG Pins (ch.0 to ch.5)

- PPG ch.0 output (Pin name) PPG0 (Pin No.) 26
- PPG ch.1 output (Pin name) PPG1 (Pin No.) 27
- PPG ch.2 output (Pin name) PPG2 (Pin No.) 28
- PPG ch.3 output (Pin name) PPG3 (Pin No.) 29
- PPG ch.4 output (Pin name) PPG4 (Pin No.) 30
- PPG ch.5 output (Pin name) PPG5 (Pin No.) 31
- PPG trigger 0 input (ch.0 to ch.3) (Pin name) TRG0 (Pin No.) 22
- PPG trigger 1 input (ch.4, ch.5) (Pin name) TRG1 (Pin No.) 15

■ Input Capture Pins (ch.0 to ch.3)

- Input capture ch.0 input (Pin name) IN0 (Pin No.) 11
- Input capture ch.1 input (Pin name) IN1 (Pin No.) 12
- Input capture ch.2 input (Pin name) IN2 (Pin No.) 13
- Input capture ch.3 input (Pin name) IN3 (Pin No.) 14

■ Free-Run Timer Pins (ch.0 to ch.5)

- Free-run timer ch.0 clock input (Pin name) FRCK0 (Pin No.) 10
- Free-run timer ch.1 clock input (Pin name) FRCK1 (Pin No.) 9
- Free-run timer ch.2 clock input (Pin name) FRCK2 (Pin No.) 8
- Free-run timer ch.3 clock input (Pin name) FRCK3 (Pin No.) 7
- Free-run timer ch.4 clock input (Pin name) FRCK4 (Pin No.) 6
- Free-run timer ch.5 clock input (Pin name) FRCK5 (Pin No.) 5

■ Base Timer Pins (ch.0, ch.1)

- Base timer TIOA0 output (Pin name) TIOA0 (Pin No.) 43
- Base timer TIOB0 input (Pin name) TIOB0 (Pin No.) 42
- Base timer TIOA1 output/input (Pin name) TIOA1 (Pin No.) 41
- Base timer TIOB1 input (Pin name) TIOB1 (Pin No.) 40

■ Reload Timer Pins (ch.0 to ch.3)

- Reload timer ch.0 event input (Pin name) TIN0 (Pin No.) 63
- Reload timer ch.0 output (Pin name) TOT0 (Pin No.) 62

- Reload timer ch.1 event input (Pin name) TIN1 (Pin No.) 2
- Reload timer ch.1 output (Pin name) TOT1 (Pin No.) 3
- Reload timer ch.2 event input (Pin name) TIN2 (Pin No.) 43
- Reload timer ch.2 output (Pin name) TOT2 (Pin No.) 42
- Reload timer ch.3 event input (Pin name) TIN3 (Pin No.) 41
- Reload timer ch.3 output (Pin name) TOT3 (Pin No.) 40

■ Waveform Generator Pins (2 units (7ch))

- Waveform generator output stop signal input pin (Pin name)DTTI0 (Pin No.)10
- Waveform generator ch.0 output (Pin name) RTO0 (Pin No.) 9
- Waveform generator ch.1 output (Pin name) RTO1 (Pin No.) 8
- Waveform generator ch.2 output (Pin name) RTO2 (Pin No.) 7
- Waveform generator ch.3 output (Pin name) RTO3 (Pin No.) 6
- Waveform generator ch.4 output (Pin name) RTO4 (Pin No.) 5
- Waveform generator ch.5 output (Pin name) RTO5 (Pin No.) 4
- Waveform generator ch.6 output (Pin name) RTO6 (Pin No.) 3

■ Flex Ray Pins (1unit ch.A/ch.B)

- FlexRay ch.A data input (Pin name) RXDA (Pin No.) 28
- FlexRay ch.A data output (Pin name) TXDA (Pin No.) 29
- FlexRay ch.A operation enable output (Pin name) TXENA (Pin No.) 27
- FlexRay ch.B data input (Pin name) RXDB (Pin No.) 24
- FlexRay ch.B data output (Pin name) TXDB (Pin No.) 25
- FlexRay ch.B operation enable output (Pin name) TXENB (Pin No.) 23
- FlexRay stopwatch input (Pin name) STOPWT (Pin No.) 26

■ Clock Monitor Pins

- Clock monitor output (Pin name) MONCLK (Pin No.) 50

■ Up/down Counter Pins (ch.0, ch.1)

- U/D counter ch.0 AIN input (Pin name) AIN0 (Pin No.) 2
- U/D counter ch.0 BIN input (Pin name) BIN0 (Pin No.) 3
- U/D counter ch.0 ZIN input (Pin name) ZIN0 (Pin No.) 4
- U/D counter ch.1 AIN input (Pin name) AIN1 (Pin No.) 5
- U/D counter ch.1 BIN input (Pin name) BIN1 (Pin No.) 6
- U/D counter ch.1 ZIN input (Pin name) ZIN1 (Pin No.) 7

■ Clock Supervisor Pins

- Clock stop detection output (Pin name) MM (Pin No.) 61

■ D/A Converter Pins

- D/A converter output (Pin name) DAOUT (Pin No.) 34

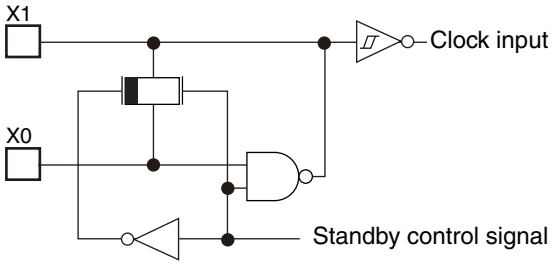
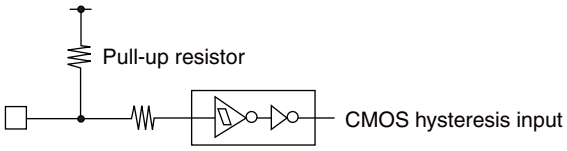
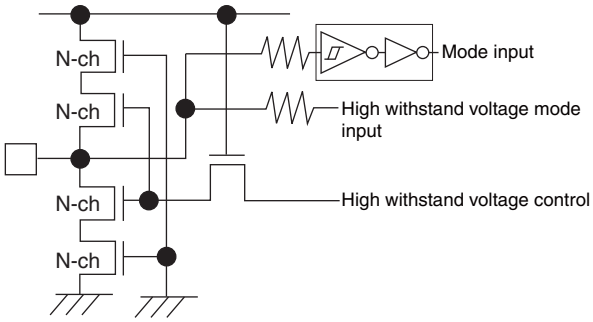
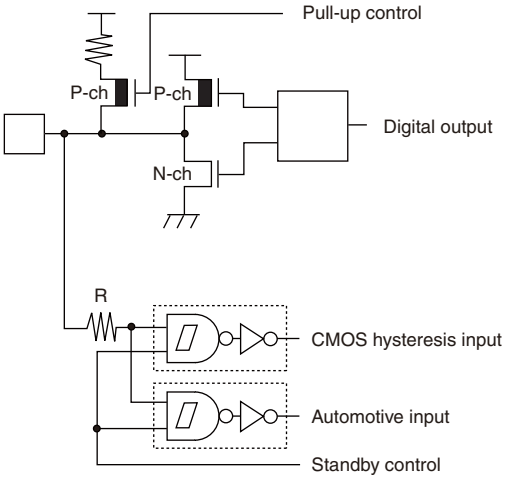
■ Port Function (General-Purpose I/O) Pins

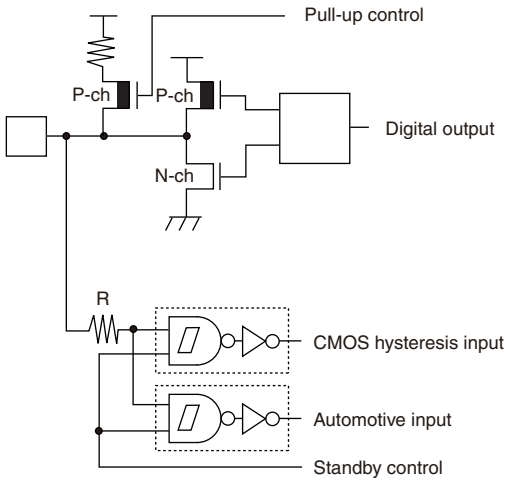
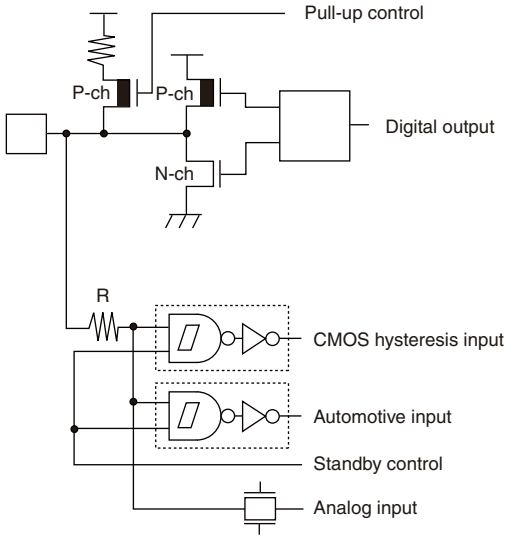
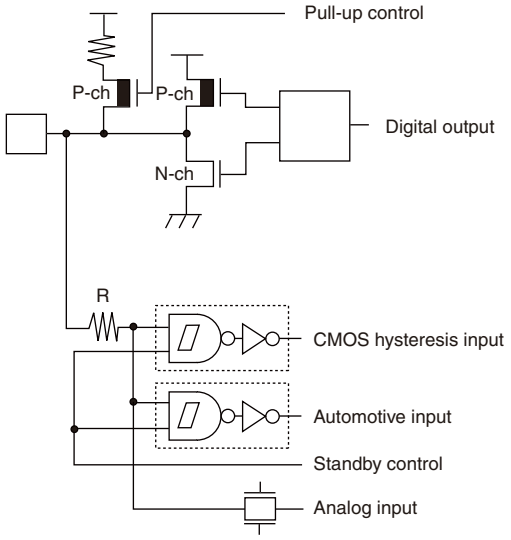
- General-purpose I/O ports (Pin name) P010 (Pin No.) 11
- General-purpose I/O ports (Pin name) P011 (Pin No.) 12
- General-purpose I/O ports (Pin name) P012 (Pin No.) 13
- General-purpose I/O ports (Pin name) P013 (Pin No.) 14
- General-purpose I/O ports (Pin name) P014 (Pin No.) 15
- General-purpose I/O ports (Pin name) P015 (Pin No.) 16
- General-purpose I/O ports (Pin name) P016 (Pin No.) 17
- General-purpose I/O ports (Pin name) P017 (Pin No.) 19
- General-purpose I/O ports (Pin name) P020 (Pin No.) 22
- General-purpose I/O ports (Pin name) P021 (Pin No.) 23
- General-purpose I/O ports (Pin name) P022 (Pin No.) 24
- General-purpose I/O ports (Pin name) P023 (Pin No.) 25
- General-purpose I/O ports (Pin name) P024 (Pin No.) 26
- General-purpose I/O ports (Pin name) P025 (Pin No.) 27
- General-purpose I/O ports (Pin name) P026 (Pin No.) 28
- General-purpose I/O ports (Pin name) P027 (Pin No.) 29
- General-purpose I/O ports (Pin name) P030 (Pin No.) 34
- General-purpose I/O ports (Pin name) P031 (Pin No.) 35
- General-purpose I/O ports (Pin name) P032 (Pin No.) 36
- General-purpose I/O ports (Pin name) P033 (Pin No.) 40
- General-purpose I/O ports (Pin name) P034 (Pin No.) 41
- General-purpose I/O ports (Pin name) P035 (Pin No.) 42
- General-purpose I/O ports (Pin name) P036 (Pin No.) 43
- General-purpose I/O ports (Pin name) P037 (Pin No.) 45
- General-purpose I/O ports (Pin name) P040 (Pin No.) 46
- General-purpose I/O ports (Pin name) P041 (Pin No.) 47
- General-purpose I/O ports (Pin name) P042 (Pin No.) 48
- General-purpose I/O ports (Pin name) P043 (Pin No.) 50
- General-purpose I/O ports (Pin name) P044 (Pin No.) 56
- General-purpose I/O ports (Pin name) P045 (Pin No.) 57
- General-purpose I/O ports (Pin name) P046 (Pin No.) 61
- General-purpose I/O ports (Pin name) P047 (Pin No.) 62
- General-purpose I/O ports (Pin name) P050 (Pin No.) 4
- General-purpose I/O ports (Pin name) P051 (Pin No.) 5
- General-purpose I/O ports (Pin name) P052 (Pin No.) 6
- General-purpose I/O ports (Pin name) P053 (Pin No.) 7
- General-purpose I/O ports (Pin name) P054 (Pin No.) 8
- General-purpose I/O ports (Pin name) P055 (Pin No.) 9
- General-purpose I/O ports (Pin name) P056 (Pin No.) 10
- General-purpose I/O ports (Pin name) P070 (Pin No.) 63
- General-purpose I/O ports (Pin name) P071 (Pin No.) 2
- General-purpose I/O ports (Pin name) P072 (Pin No.) 3
- General-purpose I/O ports (Pin name) P093 (Pin No.) 30
- General-purpose I/O ports (Pin name) P094 (Pin No.) 31

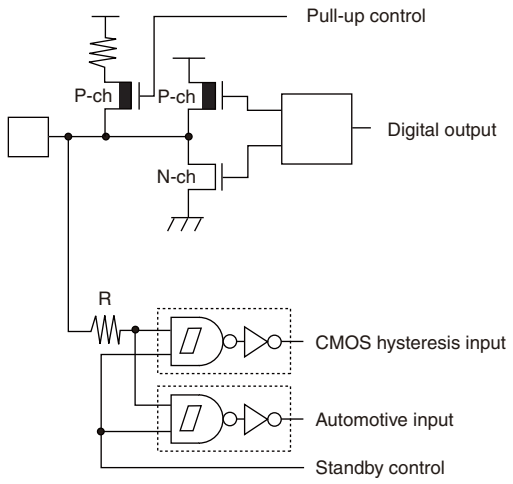
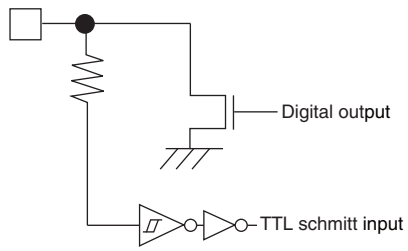
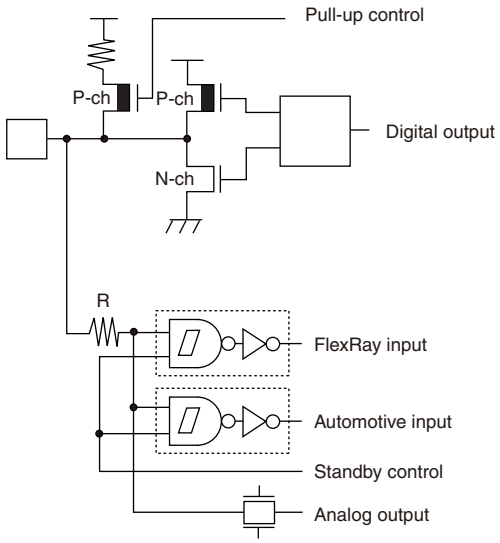
■ Other Pins

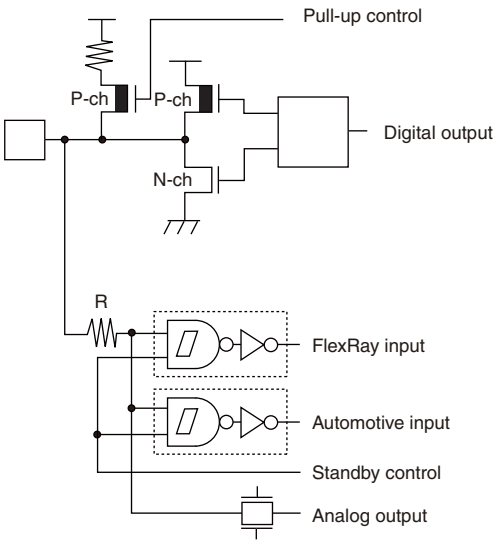
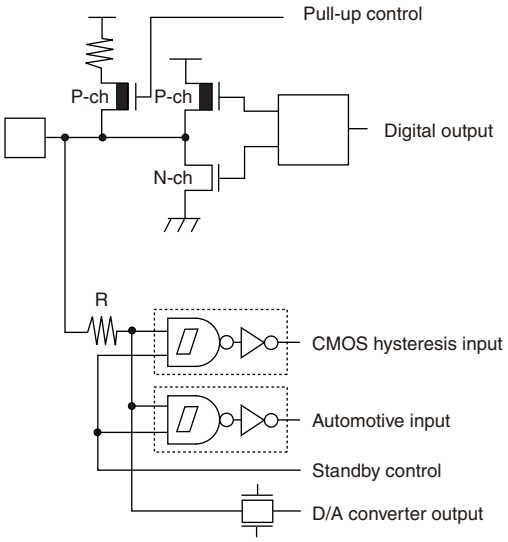
- +5.0V power supply (Pin name) VCC5 (Pin No.) 32, 64
- GND (Pin name) VSS (Pin No.) 1, 33, 55, 59
- External capacity connection output (Pin name) C (Pin No.) 60
- Main clock oscillation output (Pin name) X1 (Pin No.) 54
- Main clock oscillation input (Pin name) X0 (Pin No.) 53
- Mode pin 0 (with high-voltage control) (Pin name) MD0 (Pin No.) 51
- Mode pin 1 (with high-voltage control) (Pin name) MD1 (Pin No.) 52
- NMI interrupt input (Pin name) NMIX (Pin No.) 44
- DEBUG I/F (Pin name) DEBUGIF (Pin No.) 49
- External reset input (Pin name) RSTX (Pin No.) 58

3.4 I/O circuit type

Classifi - cation	Circuit	Overview
A		• Oscillation feedback resistor: Approx. 1 MΩ
B		• CMOS hysteresis input • With 50 kΩ pull-up resistor
C		Schmitt input • With high withstand voltage control
D		• General-purpose I/O port • CMOS level output I_{OH}=-2/-5mA, I_{OL}=2/5mA • With 50kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) • Automotive input (0.8Vcc/0.5Vcc)

Classifi - cation	Circuit	Overview
E		• General-purpose I/O port • CMOS level output I_{OH}=-2/-5mA, I_{OL}=2/5mA • With 50 kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) During standby, the input value retains the previous value. • Automotive input (0.8Vcc/0.5Vcc) During standby, the input value retains the previous value.
F		• With analog input, general-purpose I/O port • CMOS level output I_{OH}=-2/-5mA, I_{OL}=2/5mA • With 50 kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) • Automotive input (0.8Vcc/0.5Vcc)
G		• With analog input, general-purpose I/O port • CMOS level output I_{OH}=-2/-5mA, I_{OL}=2/5mA • With 50 kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) During standby, the input value retains the previous value. • Automotive input (0.8Vcc/0.5Vcc) During standby, the input value retains the previous value.

Classifi - cation	Circuit	Overview
H		• With 1^2C, general-purpose I/O port • CMOS level output I_{OH}=-3mA, I_{OL}=3mA (at I^2C output) I_{OH}=-2/-5mA, I_{OL}=2/5mA (other than above) • With 50 kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) • Automotive input (0.8Vcc/0.5Vcc)
I		Open drain I/O
K		• With analog output, general-purpose I/O port • CMOS level output I_{OH}=-2/-4mA, I_{OL}=2/4mA • With 50 kΩ pull-up resistor control • FlexRay input (0.65Vcc/0.35Vcc) • Automotive input (0.8Vcc/0.5Vcc)

Classifi - cation	Circuit	Overview
L		• With analog output, general-purpose I/O port • CMOS level output I_{OH}=-2/-4mA, I_{OL}=2/4mA • With 50 kΩ pull-up resistor control • FlexRay input (0.65Vcc/0.35Vcc) During standby, the input value retains the previous value. • Automotive input (0.8Vcc/0.5Vcc) During standby, the input value retains the previous value.
M		• With D/A converter output, general-purpose I/O port • CMOS level output I_{OH}=-2/-5mA, I_{OL}=2/5mA • With 50 kΩ pull-up resistor control • CMOS hysteresis input (0.7Vcc/0.3Vcc) • Automotive input (0.8Vcc/0.5Vcc)

CHAPTER 4 CPU

4.1 Overview

The FR81 architecture adopts the FR family instruction set with improved floating point functionality, memory protection functionality, and on-chip debugging system.

The integer family instruction set is compatible with the FR80 family.
For details, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

4.2 Features

The FR family is a CPU core for 32-bit RISC-based controllers equipped with a custom Spansion 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 (4 GB)
- 16-bit fixed instruction length (excluding immediate data transfer instructions)
- High-speed processing of basic instructions at one instruction per cycle using a 5-stage pipeline architecture
- 32-bit × 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 calculation command
- Floating point register (32 bit × 16 units)
- Privilege mode and user mode
- Some address-mapped registers handled as system registers to protect against their access in user mode
- Some instructions handled as privileged instructions to protect against their execution in 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

Code: FR81S09-1v0-91580MS-1-E

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 page address and page size
  - Page size: Can be specified as 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 (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
- Supports single precision
- Six exception sources are supported.
  - Underflow
  - Overflow
  - Division-by-zero
  - Invalid operation
  - Inexact
  - Denormalized operand
- The only rounding mode supported is nearest value
- Denormalized numbers are truncated to 0 or generate an exception
- Floating-point register: 32 bits × 16 sets
- Multiply and Add, Multiply and Sub instructions supported
- Division and square root operations supported

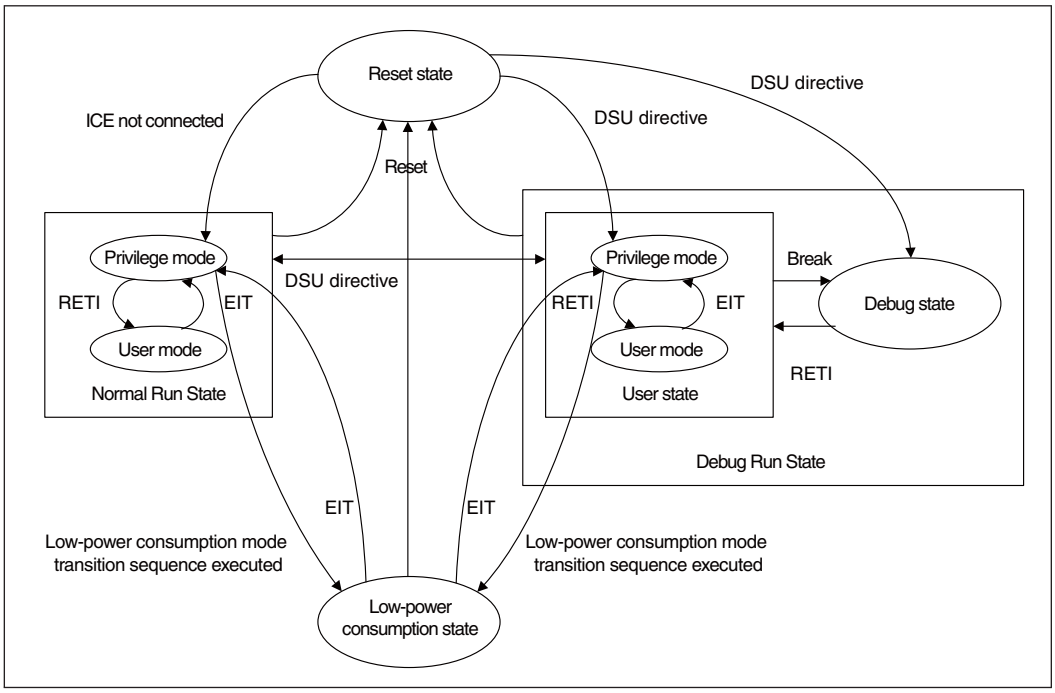
4.3 CPU Operating Description

4.3.1 CPU Operating States

The CPU has four operating states: reset state, normal run state, low-power consumption state, and debug run state.

The operating state transitions are shown below.

Figure 4.3-1 CPU operating state transition diagram



4.3.1.1 Reset State

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

4.3.1.2 Normal Run State

The normal run state is the state where 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 destinations, 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 normal run state is triggered by reset or EIT execution, and the transition from privilege mode to user mode is triggered by RETI execution.

4.3.1.3 Low-power Consumption State

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

4.3.1.4 Debug Run State

The debug run state is the state when the CPU is connected to ICE and debugging related functions are enabled. The debug run state consists of a user state and a debug state. The transition between debug run state and other states is basically performed via the reset state. However, it is also possible to forcibly transit from normal run state into debug run state.

The user state has a privilege mode and user mode the same as the normal run state. However, the CPU enters the debug state when a debug break is performed. In the debug state, instructions are executed in privilege mode with the memory protection functions, etc. all disabled so that all of the registers and memory can be accessed. The transition from debug state to user state is performed by executing the RETI instruction.

4.4 Pipeline Operation

In the FR81, the processing up to the decode stage is shared, but from the execution stage onwards there are two pipelines, an integer pipeline and a floating point pipeline. Although the completion of instructions in each pipeline differs from the order in which the instructions are issued, the processing results are guaranteed to following the order of the program sequence.

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

4.5 Floating Point Operation Processing

This product has a built-in FPU (floating point operations).

For details on floating point operation processing, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

4.6 Data Structures

The FR81 family of CPUs can handle the single-precision floating point data type in addition to the integer data type supported by the models up to the FR80 family.

For the integer data types, little endian bit ordering and big endian byte ordering have been adopted.

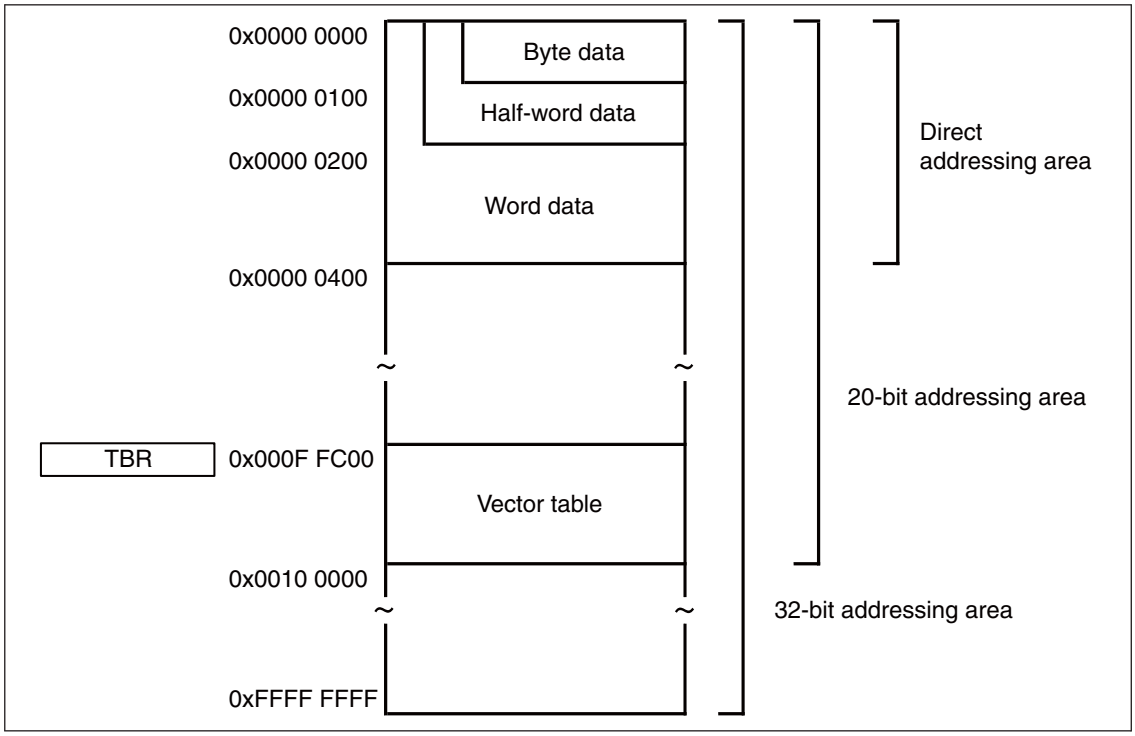
For details, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

4.7 Addressing

The memory space is 32-bit linear.

The CPU manages the address space in bytes. From the CPU, an address on the address space is accessed with a 32-bit value specified. Figure 4.7-1 shows the address space.

Figure 4.7-1 Memory space



The address space is also called the memory space. The address space is a logical address space as viewed from the CPU. Addresses are not exchanged. A logical address as viewed from the CPU is the same as a physical address that has memory and I/O assignments. For details, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

4.8 Programming Model

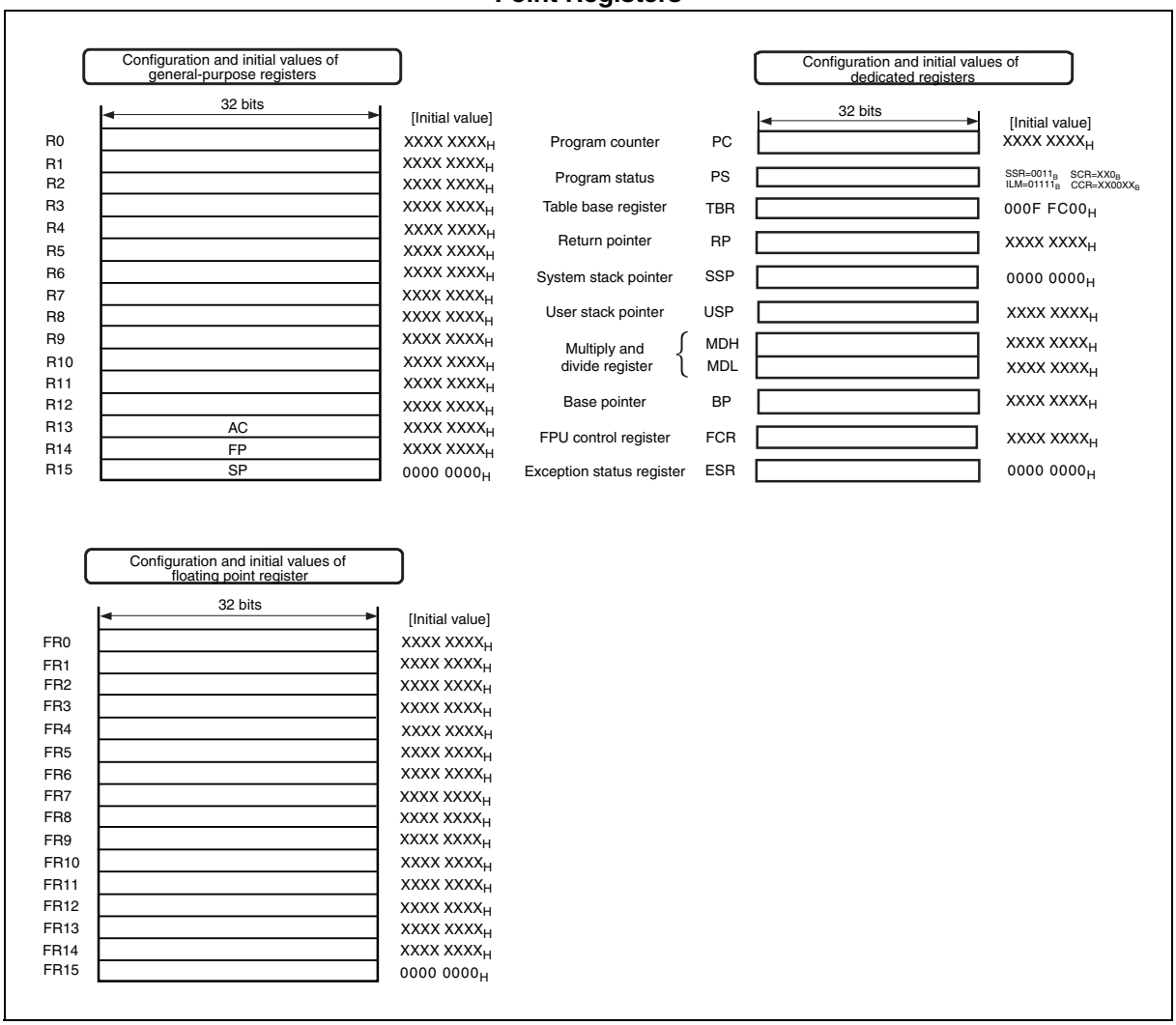
The FR81 CPU has general-purpose registers, dedicated registers, and floating point registers. There are also separate system registers which are address-mapped into the FR81 core.

■ General-purpose Registers, Dedicated Registers, Floating Point Registers

Figure 4.8-1 shows the initial values in this model.

For details on each register, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL".

Figure 4.8-1 Initial Values of General-purpose Registers, Dedicated Registers, and Floating Point Registers



■ System Registers

The system registers are address-mapped registers for system control. These registers can only be accessed in privilege mode. The system registers consist of the following.

- Clock control related registers
- Reset control related registers
- Debug control related registers
- Memory protection related registers
- DMA related registers

- Watchdog timer registers
- Wild register control registers
- FLASH control registers

If a read or write is performed to these registers in user mode, an illegal instruction exception (data access error) is generated.

Access protection of system registers has precedence over the memory protection function. Therefore, even if user access to the system register area is permitted and access in privilege mode is forbidden using the memory protection function, these settings are ignored and the system register area is able to be read and written only in privilege mode, and is not able to be read or written in user mode.

4.9 Reset and EIT Processing

Reset and EIT processing are processing that is executed by a different program from normal when a reset, exception, interrupt, or trap is detected.

4.9.1 Reset

A reset forcibly stops the current processing being executed, initializes the device, and restarts the program from the reset vector entry address.

(Notes) In this model, the FixedVector function returns the flash memory starting address + 0x0024 instead of the value written at address 0xF_FFFC in the flash memory to the reset vector. See "CHAPTER 32 FIXEDVECTOR FUNCTION" for details.

4.9.2 EIT Processing

EIT processing stops the current processing being executed and, before transferring control to a predefined handler program, saves information to memory so that the processing can resume.

4.9.3 Vector Table

Table 4.9-1 Vector Table (1 / 2)

Interrupt source	Interrupt vector number		Interrupt level	Offset	Address on 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
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

Table 4.9-1 Vector Table (2 / 2)

Interrupt source	Interrupt vector number		Interrupt level	Offset	Address on TBR initial value
	Decimal	Hexadecimal			
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
Delayed interrupt	63	3F	ICR47	0x300	0x000FFF00
System reserved (used by REALOS)	64	40	-	0x2FC	0x000FFEFC
System reserved (used by REALOS)	65	41	-	0x2F8	0x000FFEF8
Used by INT instruction	66	42		0x2F4	0x000FFEF4
	|	|	-	|	|
	255	FF		0x000	0x000FFC00

4.10 Memory Protection Function (MPU)

4.10.1 Overview

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 page address and 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: Permitted / forbidden
  - Data read: Permitted / forbidden
  - Data write: Permitted / forbidden
- Access attributes are specified for each area
  - Buffer: Permitted / forbidden
- The access rights and attributes of undefined areas are controlled as a default area.
- Protection violation exceptions occur when a protection violation occurs
- The registers for the memory protection function can only be accessed in privilege mode as system registers
  - Data access error notification function
  - The I/O area (00000000_H to $0000FFFF_H$) is fixed unbufferable

4.10.2 Register List

Table 4.10-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0310	Reserved		MPUCR		MPU Control Register
0x0314	Reserved				Reserved
0x0318	Reserved				Reserved
0x031C	Reserved		Reserved		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

4.10.3 Register Description

■ MPU Control Register (MPUCR)

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)**

15	14	13	12	11	10	9	8	bit
PIE	PRE	PWE	UIE	URE	UWE	Reserved	BE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved				PAN[1:0]		DEE	MPE	
0	0	0	0	0	1	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,WX	R1,WX	R/W	R/W	Attribute

[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 permitted in user mode (initial value)
1	Instruction fetch permitted in user mode

[bit11] URE (User Mode Read Access Enable)

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

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

[bit10] UWE (User Mode Write Access Enable)

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

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

[bit9] reserved

This bit is reserved. Always write "0" to this bit. 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 have not been 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 synchronized data access. Illegal instruction exceptions occur when there is an error during data access only if buffering is forbidden. When buffering is permitted, data access errors can be notified as interrupts.

BE	Bufferable specification for the default area
0	Buffering forbidden (initial value)
1	Buffering permitted

(Note) Set the area where the vector table is allocated to buffering forbidden.
There is no limitation concerning the buffer setting about the area where the EIT processing routine is allocated.

[bit7 to bit4] Reserved

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

[bit3, bit2] PAN (Protection Area Number)

These bits indicate the number of the protection areas that can be specified. These bits are read-only and indicate the number of areas implemented in hardware.

PAN	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 interrupts disabled (initial value)
1	Data access error interrupts enabled

[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

■ Data Access Protection Violation Address Register (DPVAR)

This register stores the address where a data access protection violation occurred.

- **DPVAR: Address 0320_H (Access: Word)**

31	30	· · ·	2	1	0	bit
DPVA[31:0]						
X	X	· · ·	X	X	X	Initial value
R,WX	R,WX	· · ·	R,WX	R,WX	R,WX	Attribute

[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.

■ Data Access Protection Violation Status Register (DPVSR)

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)**

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
RW[1:0]		SZ[1:0]		MD	Reserved		DPV	
0	0	0	0	0	-	-	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R0,W0	R0,W0	R,W	Attribute

[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 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[1:0]=01_B read (read-modify-write) even if the violation occurs in the write part of the read-modify-write.

RW	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

This bit 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.

Only "0" can be written to this bit. Writing "1" is disabled.

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

■ Data Access Error Address Register (DEAR)

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

- **DEAR: Address 0328_H (Access: Word)**

31	30	· · ·	2	1	0	bit
DEA[31:0]						
X	X	· · ·	X	X	X	Initial value
R,WX	R,WX	· · ·	R,WX	R,WX	R,WX	Attribute

[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.

■ Data Access Error Status Register (DESR)

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)**

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
RW[1:0]		SZ[1:0]		MD	Reserved		DAE	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R0,W0	R0,W0	R,W	Attribute

[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 indicates that data access error has occurred. To retain the new data error content, clear this bit. If data access error interrupt is enabled, clearing this bit cancels the interrupt request.

Only 0 can be written to this bit. Writing 1 is disabled.

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

■ Protection Area Base Address Registers 0 to 7 (PABR0 to PABR7)

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

- **PABR0 to PABR7: Address 0330_H, 0338_H, 0340_H, 0348_H, 0350_H, 0358_H, 0360_H, 0368_H (Access: Word)**

31	30	· · ·	10	9	8	bit
PABR[31:8]						
X	X	· · ·	X	X	X	Initial value
R/W	R/W	· · ·	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
PABR[7:0]								
X	X	X	X	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[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 "0".

■ Protection Area Control Registers 0 to 7 (PACR0 to PACR7)

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

- **PACR0 to PACR7: Address 0336_H, $033E_H$, 0346_H, $034E_H$, 0356_H, $035E_H$, 0366_H, $036E_H$ (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
PIE	PRE	PWE	UIE	URE	UWE	Reserved	BE	
0	0	0	0	0	0	-	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	Attribute

7	6	5	4	3	2	1	0	bit
ASZ[4:0]					Reserved		PAE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0	R/W	Attribute

[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

This bit is 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 synchronized data access. 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	Buffering forbidden (initial value)
1	Buffering permitted

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

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

ASZ[4:0]	Size of the specified protection area
00000_B	Reserved
00001_B	Reserved
00010_B	Reserved
00011_B	16B
00100_B	32B
00101_B	64B
00110_B	128B

ASZ[4:0]	Size of the specified protection area
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
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" to these bits. These bits read out "0".

[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

4.10.4 Operating Description of Memory Protection Function (MPU)

■ Setting Up Memory Protection Areas

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.

■ Instruction Access Protection Violation

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.

■ Data Access Protection Violation

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

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

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

■ Data Access Errors

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 data access error occurred while accessing a bufferable area, the PC value that are saved during the data access error interrupt cannot be the same as the PC value of instruction that ordered data access, since the CPU is already executing the subsequent instructions.

If an illegal instruction exception (data access error) occurs during the execution of an instruction that performs multiple data accesses, the data accesses that had executed up until the error occurred are not canceled. If an illegal instruction exception (data access error) occurs during the LDM0, LDM1, STM0, STM1, FLDM, or FSTM instructions, the list of remaining registers is stored in the exception status register ESR:RL, and the bit indicating a data access error ESR:INV[6] 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.

■ Memory protection operation at delay slot

Instructions placed at delay slots are executed as 16-bit.

Therefore, if a 32-bit instruction is placed at a delay slot, and if its lower 16-bit contains instruction access protection violation factors and/or instruction access error factors, an illegal instruction exception (that the instructions cannot be applied to the delay slot) is raised.

■ DEAR/DESR update

Data access error address register (DEAR) and data access error status register (DESR) are updated under the following circumstances:

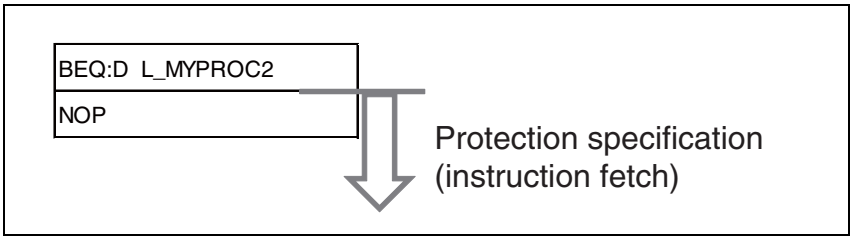
- System register access under user mode (illegal instruction exception)
- Bus error while accessing unbufferable area (illegal instruction exception)
- Bus error while accessing bufferable area (data access error interrupt)

In the case that invalid instruction exception is generated, DEAR and DESR are updated while executing an instruction that performed the corresponding access, while in the case that data access error interrupt is generated, these two registers are updated in an asynchronous manner to the instruction operation.

If factors occur simultaneously, invalid instruction exception factor takes precedence.

4.10.5 Note

- An access protection violation exception occurs during execution of an instruction that will become an access protection violation. For details, see the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL". For details on instruction access protection violations and instruction access protection violation exceptions, see also "■ Instruction Access Protection Violation".
- If the delay slot and instruction access protection area have overlapping boundaries, as shown in the following figure, an instruction access protection violation occurs regardless of whether the branch is taken. The exception occurs on the PC executing the delayed branch instruction.



CHAPTER 5 OPERATION MODE

5.1 Overview

This chapter explains the device operation mode that is determined after release from the reset state. For power consumption control modes and each clock selection mode, see "CHAPTER 9 POWER CONSUMPTION CONTROL".

5.2 Features

This device supports the following operation modes.

■ User Mode

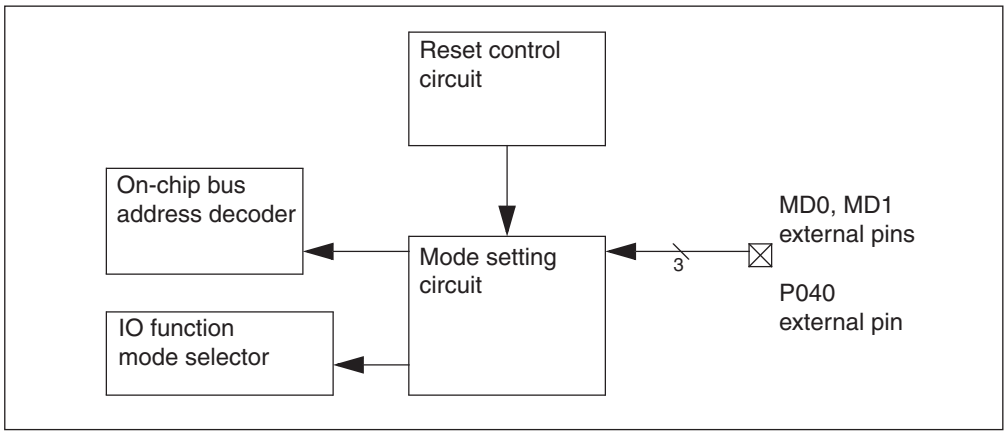
They operate by starting the internal flash memory.

■ Serial Writer Mode

The flash memory can be programmed using the serial writer.

5.3 Configuration

Figure 5.3-1 Block Diagram



Code: BMODED-0v7-91580MS-1-E

5.4 Registers

Table 5.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x07FC	BMODR	Reserved	Reserved	Reserved	Bus mode data register

■ Bus Mode Register: BMODR (Bus MODe Register)

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

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

7	6	5	4	3	2	1	0	bit
BMOD[7:0]								
*	*	*	*	*	*	*	*	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

* Depends on the operation mode.

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

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

BMOD[7:0]	Operation mode
0101xxxx	User mode
0111xxx1	Serial writer mode

5.5 Operation

■ MD0, MD1, P040 Pin Settings

Table 5.5-1 Pin Settings

Operation mode	MD1	MD0	P040
User mode	0	-	-
Serial writer mode	1	0	1

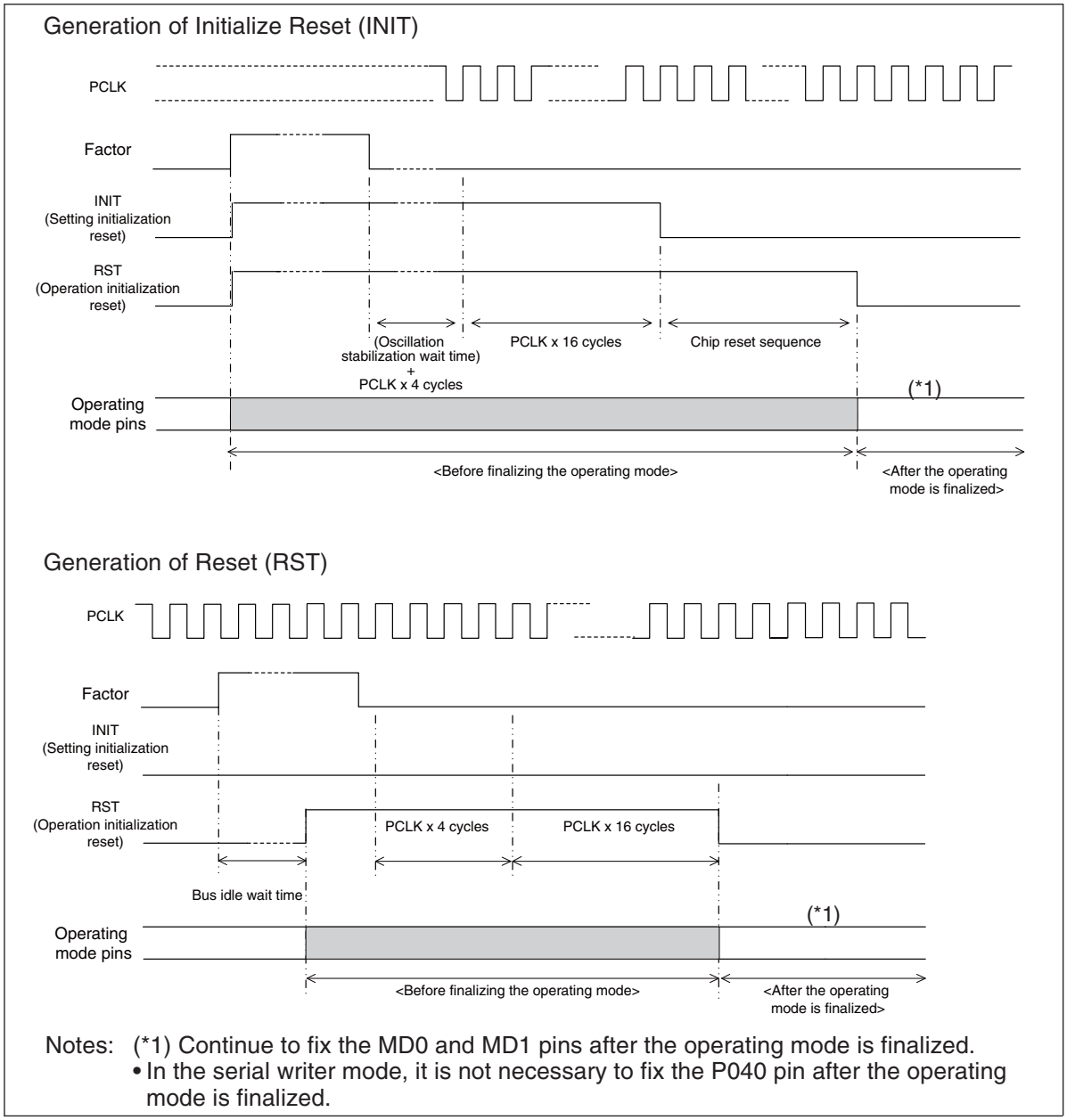
Settings other than those in the table are prohibited.

■ Fetching the Operation Mode

Fetching the operation mode is sampled by the RST (Reset) instruction. During the time when an RST instruction is issued and when it is released, the MD0, MD1, and P040 pin inputs must be determined. (The P040 pin needs not be determined in the User mode.)

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

Figure 5.5-1 Operation Mode Fetch Timing Chart



Notes: (*1) Continue to fix the MD0 and MD1 pins after the operating mode is finalized.
- In the serial writer mode, it is not necessary to fix the P040 pin after the operating mode is finalized.

■ Explanation of Each Operation Mode

The following details each operation mode.

● User Mode

The internal I/O, built-in RAM, and built-in Flash are effective in this mode, but the access to the other area is disabled. The external pins function as peripheral resources or general-purpose ports.

● Serial Writer Mode

Contact the Spansion support engineer.

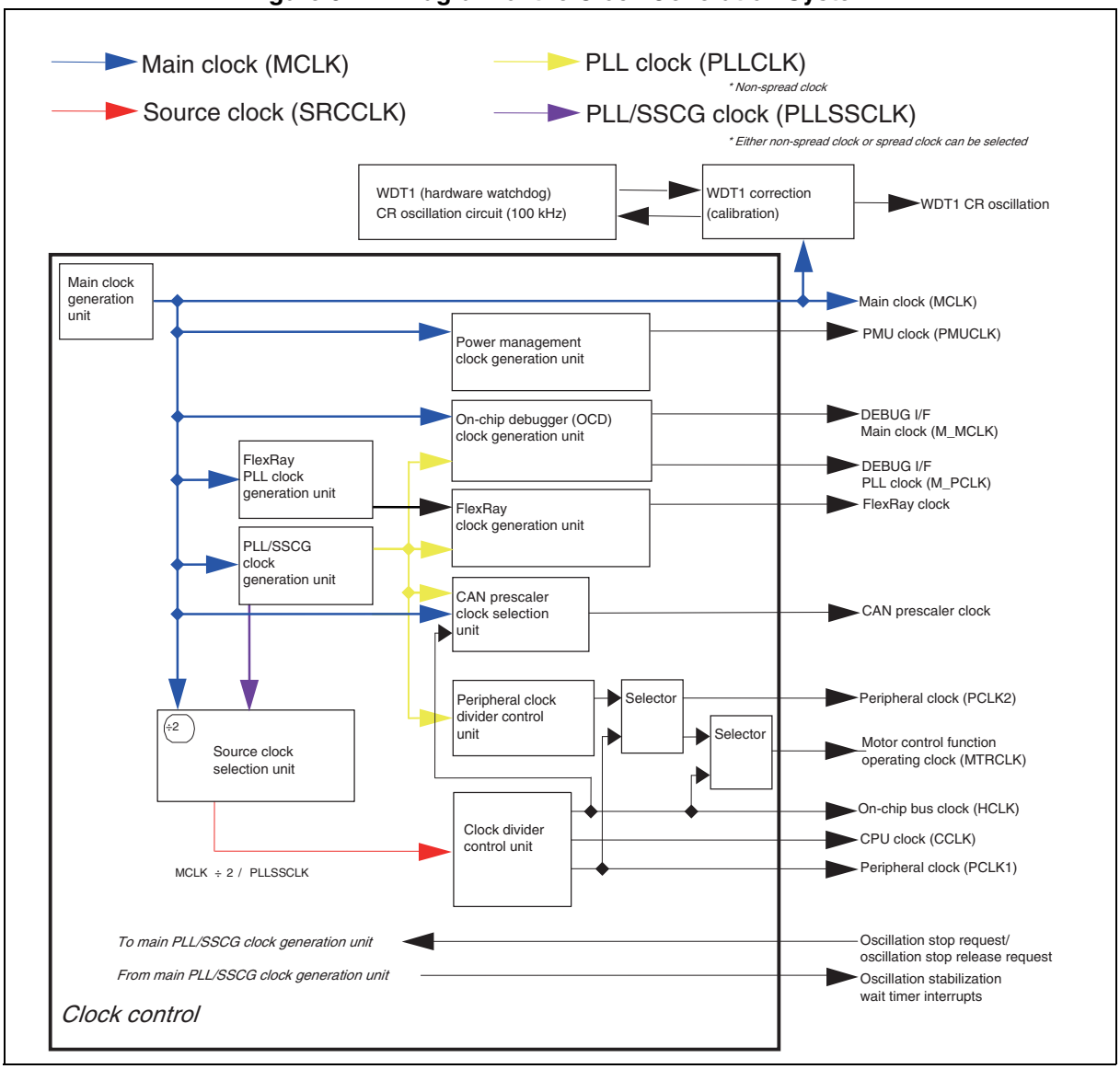
CHAPTER 6 CLOCK

6.1 Overview

Implemented oscillation circuit generates a single clock product, which generates individual clock systems on the chip. This product also implements CR oscillation circuit for watchdog timer 1.

- External pins for the implemented oscillation circuit:
  Main clock: Connects to the crystal oscillator.
- Generation of source clocks: Selects from the clocks which are multiplied by PLL of main clock (MCLK), or divided by 2 of main clock.
- Division of source clock: Divides the source clock and generates operating clocks for supplying to each unit.
- For details on the motor control function operating clock (MTRCLK), see "CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION".

Figure 6.1-1 Diagram of the Clock Generation System



Code:BG4ACCTL-0v4-91580MS-1-E

6.2 Features

- 1 system on-chip oscillator is implemented.
- The main clock (MCLK) is multiplied by on-chip PLL/SSCG.
- Multiplied clock is supplied with independent PLL/SSCG.
- Each clock has been forced not to supply by using the timer until it becomes stabilized (oscillation stabilization wait timer).
- Oscillation stabilization wait end interrupt can be generated.
- Main clock oscillation stabilization wait timer (main timer) can be used as a multi-purpose interrupt interval timer after oscillation of main / PLL/SSCG clocks are stabilized.
- Implements a CR oscillation circuit for 100KHz WDT1 clock.
- Clock for CAN prescaler is generated. Selectable from PLL clock (PLLCLK) [non-spread clock] and main clock (MCLK). If PLL stops when PLL clock is selected, on-chip bus clock (HCLK) is used.
- In order to reduce noise, SSCG clock [spread clock] can be selected as the clock for the CPU and resources.

6.3 Configuration

Figure 6.3-1 Connection Diagram of Clock (1) Main/PLL Clock Generation Unit

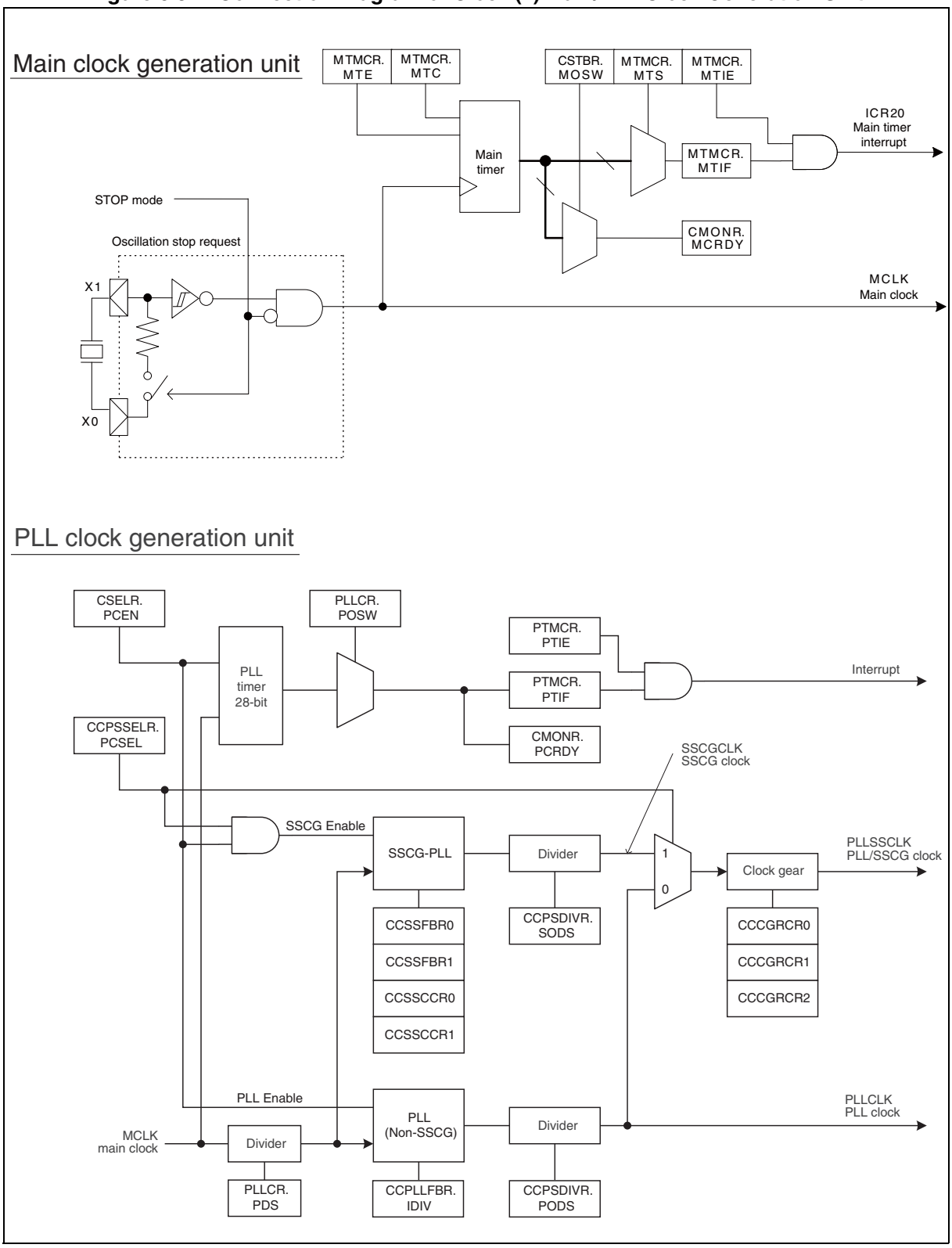


Figure 6.3-2 Connection Diagram of Clock (2) Source Clock Selection Unit

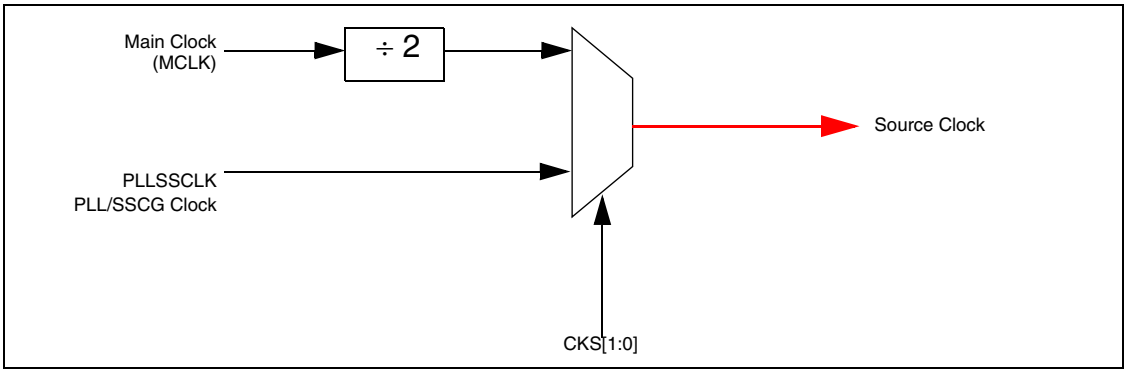


Figure 6.3-3 Connection Diagram of Clock (3) Division Control

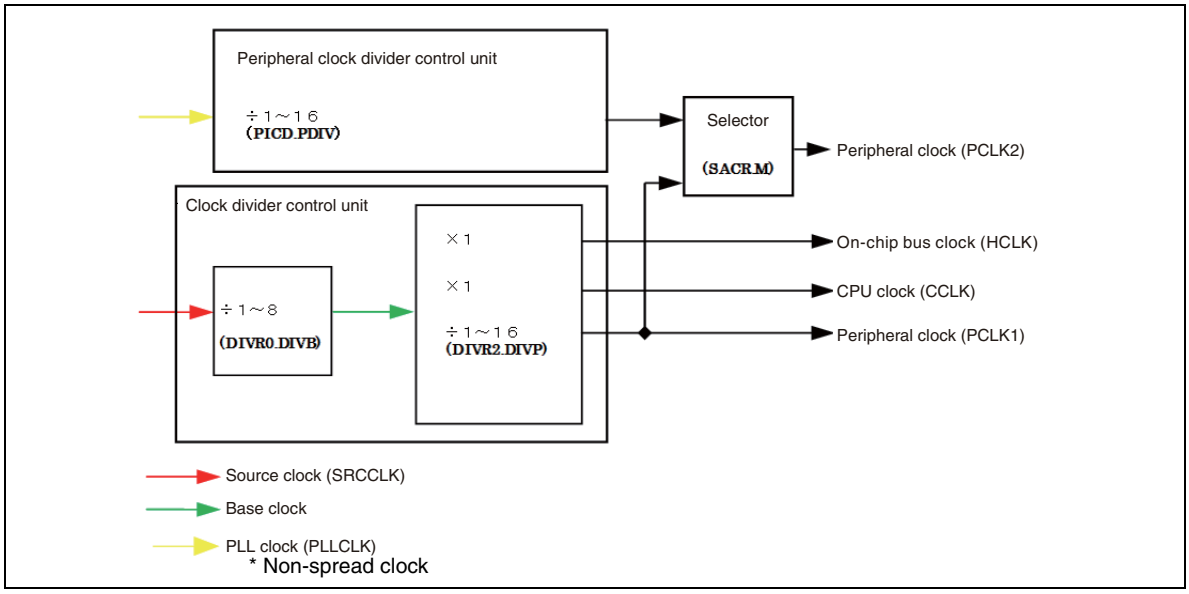


Figure 6.3-4 Connection Diagram of Clock (4) CAN Prescaler Clock Generation

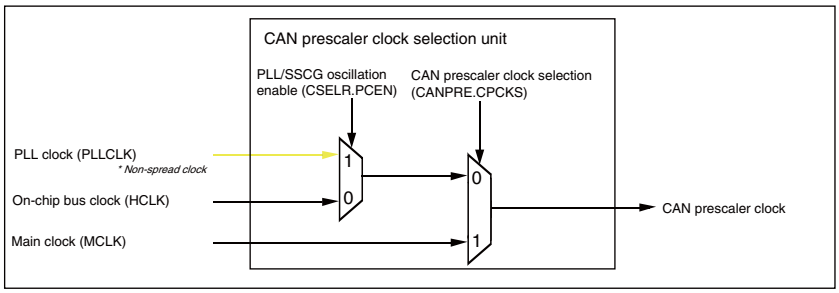


Figure 6.3-5 Connection Diagram of Clock (5) Power Management Clock Generation Unit

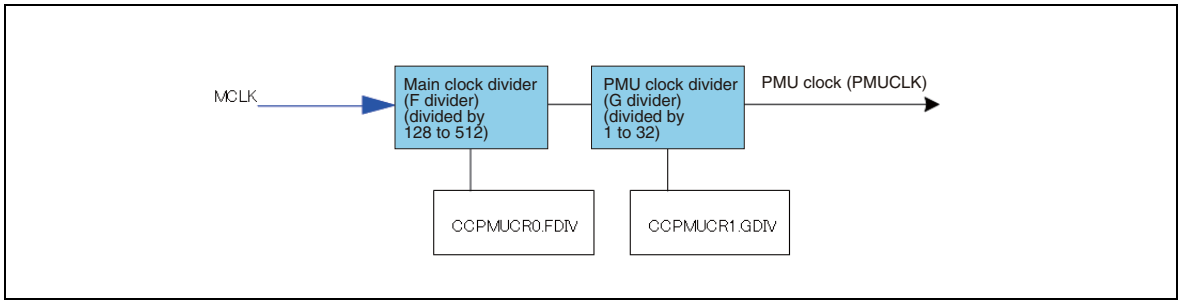


Figure 6.3-6 Diagram of Clock System (MB91F583AM/F584AM/F585AM)

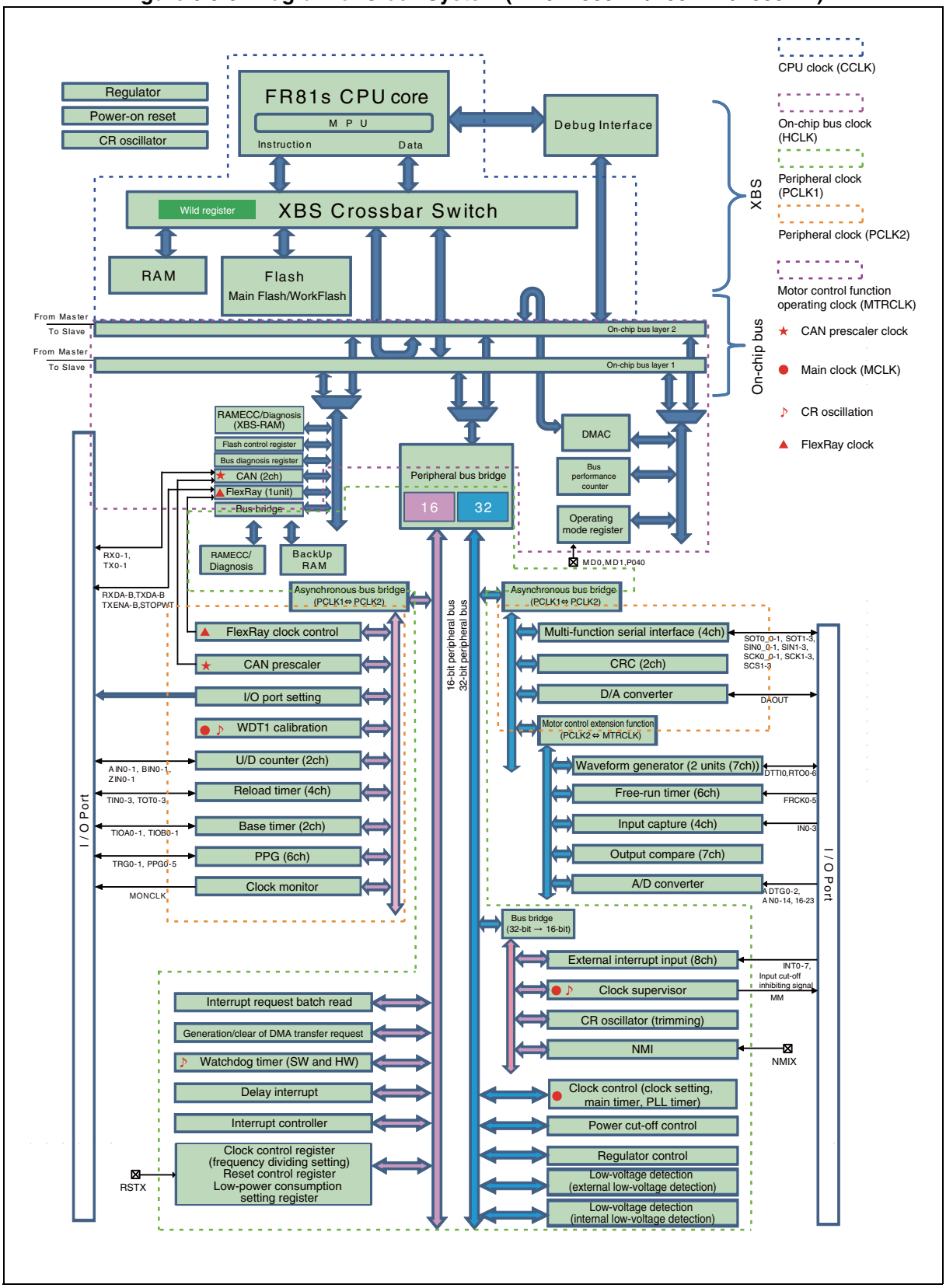
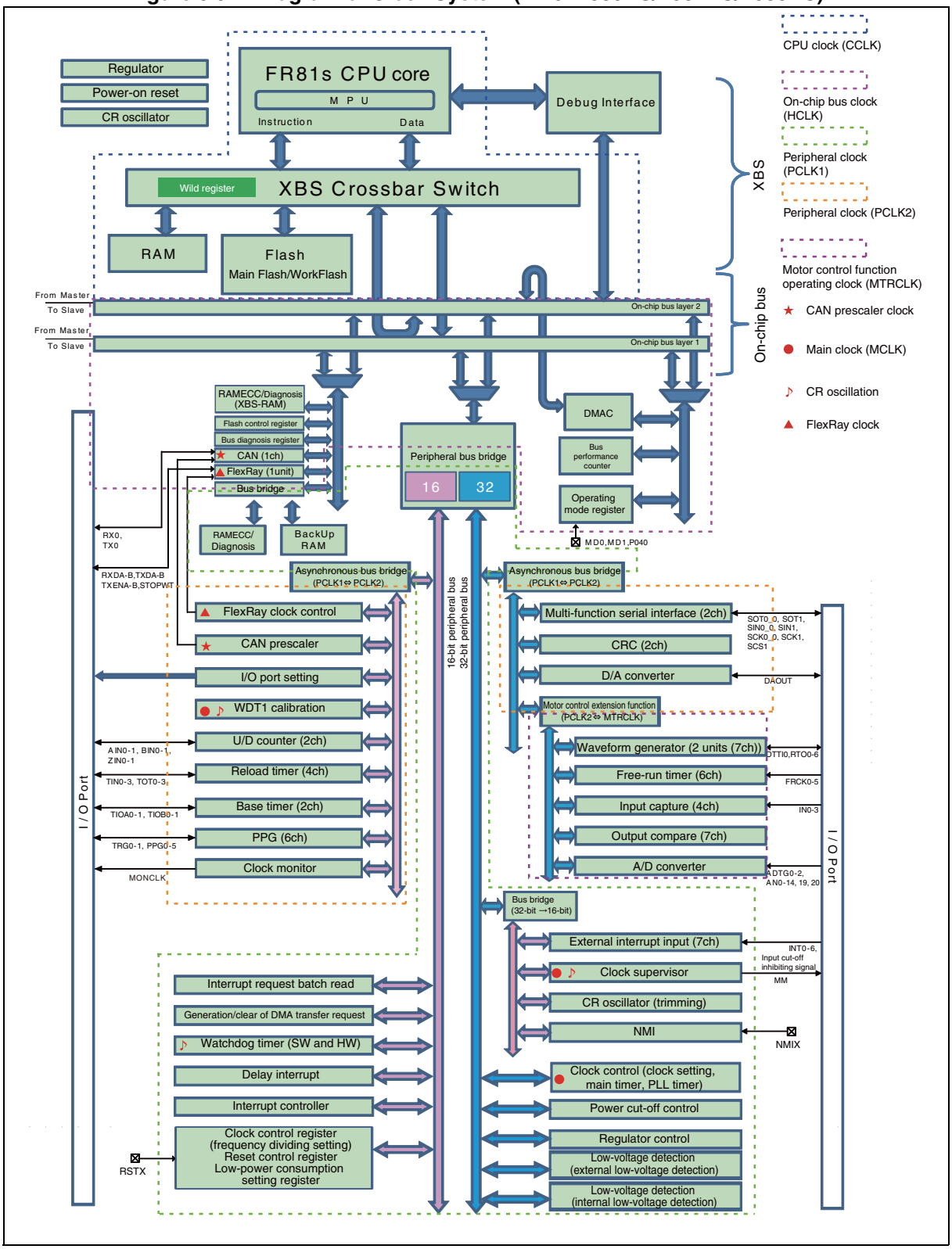


Figure 6.3-7 Diagram of Clock System (MB91F583AS/F584AS/F585AS)



6.4 Registers

Table 6.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x0488	DIVR0	Reserved	DIVR2	Reserved	Divide Configuration Register 0 Divide Configuration Register 2
0x0510	CSELR	CMONR	MTMCR	Reserved	Clock Source Configuration Register Clock Source Monitor Register Main Timer Control Register
0x0514	PLLCR		CSTBR	PTMCR	PLL Setting Register Oscillation Stabilization Wait Setting Register PLL Clock Oscillation Stabilization Wait Timer Control Register
0x0520	CCPSSELR	Reserved	Reserved	CCPSDIVR	PLL/SSCG clock selection register PLL/SSCG output clock divide setting register
0x0524	Reserved	CCPLLFBR	CCSSFBR0	CCSSFBR1	PLL feed back divide setting register SSCG feed back divide setting register 0 SSCG feed back divide setting register 1
0x0528	Reserved	CCSSCCR0	CCSSCCR1		SSCG config setting register 0 SSCG config setting register 1
0x052C	Reserved	CCCGRCR0	CCCGRCR1	CCCGRCR2	clock gear config setting register 0 clock gear config setting register 1 clock gear config setting register 2
0x0530	Reserved	Reserved	CCPMUCR 0	CCPMUCR 1	PMU clock divide setting register 0 PMU clock divide setting 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

6.4.1 DIVR0: (DIVide clock configuration Register 0) Divide Configuration Register 0

This register controls division of clocks.

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

7	6	5	4	3	2	1	0	bit
DIVB[2:0]			Reserved					
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W0	R/W0	R/W0	R/W0	R/W0	Attribute

[bit7 to bit5] DIVB[2:0] (DIVide ratio of Base clock) Base clock divide configuration

These bits configure a division in the area where the base clock is generated from the source clock (SRCCLK) as follows.

CPU clock (CCLK) and on-chip bus clock (HCLK) have the same frequency as that of the base clock.

DIVB[2:0]	Division ratio
000	Do not divide (initial value)
001	2 division
010	3 division
011	4 division
100	5 division
101	6 division
110	7 division
111	8 division

[bit4 to bit0] Reserved

Always write "0" to these bits.

6.4.2 DIVR2: (DIVide clock configuration Register 2) Divide Configuration Register 2

This register controls division of clocks.

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

7	6	5	4	3	2	1	0	bit
DIVP[3:0]				Reserved				
0	0	1	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W0	R/W0	R/W0	R/W0	Attribute

[bit7 to bit4] DIVP[3:0] (DIVide ratio of PCLK) Peripheral clock divide configuration

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

DIVP[3:0]	Base clock→PCLK division ratio
0000	Do not divide.
0001	2 division
0010	3 division
0011	4 divide (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 limit the peripheral clock (PCLK1) under 40MHz.

[bit3 to bit0] Reserved

Always write "0" to these bits.

6.4.3 CSELR: (Clock source SELect Register) Clock Source Configuration Register

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

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

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

7	6	5	4	3	2	1	0	bit
Reserved	PCEN	-				CKS[1:0]		
0	0	1	0	0	0	0	0	Initial value
R,W0	R,W	R1,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	Attribute

[bit7] Reserved

Always write "0" to this bit.

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

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

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

This bit cannot be rewritten when a PLL/SSCG clock (PLLSSCLK) is selected as the source clock (SRCCLK). Also, this bit cannot be rewritten when the main clock oscillation is stopped or during the main clock oscillation stabilization wait time (CMONR:MCRDY=0).

Set this bit to "0" before switching to the stop mode.

(Note) During MDI high-speed communication, PLL becomes oscillation-enabled, regardless of the value of this bit.

[bit5] Undefined

This bit always reads "1" . Writing has no effect on operation.

[bit4 to bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[bit1, bit0] CKS1, CKS0 (ClocK Select) Source clock selection

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

CKS1, CKS0	Source selection
00	Division of the main clock (MCLK) by 2 (initial value).
01	Division of main clock (MCLK) by 2
10	PLL/SSCG clock (PLLCLK)
11	Disable (writing does not affect on the operation)

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

Possible combinations for changing these bits are shown below.

CKS value before change	Eligible values	Rewritten conditions	Ineligible values
00	00, 01	MCRDY=1	11
	10	PCRDY=1	
01	00, 01	MCRDY=1	10
10	00	MCRDY=1	01, 11
	10	PCRDY=1	
11	Disable	Disable	Disable

Do not write the values which cannot be rewritten.

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

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

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

(Note) If you have changed the value of CSELR, read this register and make sure that it is the same value as CSELR you set. After that do not change the next value set in CSELR.

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

7	6	5	4	3	2	1	0	bit
-	PCRDY	MCRDY	-			CKM[1:0]		
0	0	1	0	0	0	0	0	Initial value
R0,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	Attribute

[bit7] Undefined

This bit always reads "0". Writing has no effect on operation.

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

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

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

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

(Note)
- PCRDY=1 may be read immediately after changing PCEN=1 to 0.
- During MDI high-speed communication, PLL becomes oscillation-enabled, regardless of the value of this bit.

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

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

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

This bit cannot select a main clock (MCLK) or a PLL/SSCG clock (PLLSSCLK) as the source clock (SRCCLK) when this bit is set to "0".

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) During MDI high-speed communication, the main clock becomes oscillation-enabled, regardless of the value of this bit.

[bit4 to bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[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)

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

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

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

7	6	5	4	3	2	1	0	bit
MTIF	MTIE	MTC	MTE	MTS[3:0]				
0	0	0	0	1	1	1	1	Initial value
R(RM1), W	R/W	R(RM0), W	R/W	R1,WX	R/W	R/W	R/W	Attribute

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.

Main timer is cleared when it is in the stop mode. When the operation of main timer is not allowed (MTE=0), the main timer stops except that the main clock is waiting for an 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 state (MCRDY=0) is not effective.

When 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 main timer stops (MTE=0).

When rewritten MTE=1 with 0, the main timer will continue to operate until MTC 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 MTC 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 indicates 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	- Write "0". - 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 (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 will not be effective.
When the MTIE bit is set to "0", this bit will not be cleared by DMA transfer.
Read-modify-write instruction makes "1" be read out.
If a set factor and a clear factor occur at the same time, the set factor will take precedence.

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

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

This bit controls interrupts by main timer overflow as follows.

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

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

This bit clears the main timer.

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

MTC	Read
0	Operating normally
1	Clearing the main timer

This bit automatically returns to "0" after writing "1".
Read-modify-write instruction makes "0" 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 enabled

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

MTE	Operation of the main timer
0	Operation disabled (Initial value)
1	Operation enabled

At the time of MTC=1, MTE=1 write is prohibited.
When you perform a PLL/SSCG clock oscillation stabilization wait, make sure to set this bit to "0" and stop the main timer.

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

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

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

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

6.4.6 PLLCR: (PLL Configuration Register) PLL Setting Register

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)**

15	14	13	12	11	10	9	8	bit
Reserved		-	Reserved					
0	0	0	0	0	0	0	0	Initial value
RX, W0	RX, W0	R0,WX	RX, W0	RX, W0	RX, W0	RX, W0	RX, W0	Attribute

7	6	5	4	3	2	1	0	bit
POSW[3:0]				PDS[3:0]				
1	1	1	1	0	0	0	0	Initial value
R1,WX	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

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

When PLL/SSCG clock oscillation is allowed (CSELR:PCEN=1), writing to this register will be disabled.

[bit15, bit14] Reserved

Always write "0" to these bits.

[bit13] Undefined

This bit always reads "0". Writing has no effect on operation.

[bit12 to bit8] Reserved

Always write "0" to these bits.

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

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

POSW[3:0]	PLL oscillation stabilization wait time	At 4MHz	At 20MHz
1000	2^9 × main clock cycle	128.0[μs]	25.6[μs]
1001	2^{10} × main clock cycle	256.0[μs]	51.2[μs]
1010	2^{11} × main clock cycle	512.0[μs]	102.4[μs]
1011	2^{12} × main clock cycle	1024.0[μs]	204.8[μs]
1100	2^{13} × main clock cycle	2048.0[μs]	409.6[μs]
1101	2^{14} × main clock cycle	4096.0[μs]	819.2[μs]
1110	2^{15} × main clock cycle	8192.0[μs]	1638.4[μs]
1111	2^{16} × main clock cycle (initial value)	16384.0[μs]	3276.8[μs]

POSW[3] always reads "1".

(Note) PLL/SSCG clock lock up time wait time specification in this model is 200[μs]. Reserve the 200[μs] wait time or more by either of the following methods.
- Select 256[μs] POSW[3:0] or more
- Reserve the 200[μs] wait time or more by software processing, regardless of POSW[3:0] settings

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

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

PDS[3:0]	PLL input clock division 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

(Notes)
- Follow the configuration steps for your appropriate PLL and system specifications.
- See "■ PLL/SSCG Clock (PLLSSCLK)" for configuration samples.
- The setting value has limitations. Prior to setting, refer to "■ Limitations while using the PLL/SSCG clock".

6.4.7 CSTBR: (Clock STaBilization select Register) Clock Stabilization Select Register

This register configures the oscillation stabilization wait for each clock source.

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 has not been selected as the source clock (SRCCLK) is allowed until the ready state (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 (MCRDY=0).

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

7	6	5	4	3	2	1	0	bit
-	Reserved			MOSW[3:0]				
0	0	0	0	0	0	0	0	Initial value
R0,WX	R,W0	R,W0	R,W0	R,W	R,W	R,W	R,W	Attribute

[bit7] Undefined

This bit always reads"0". Writing has no effect on operation.

[bit6 to bit4] Reserved

Always write "0" to these bits.

[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 main clock (MCLK) as follows.

MOSW[3:0]	Main clock oscillation stabilization wait time	At 4MHz	At 20MHz
0000	2^{15} × main clock cycle (initial value)	8[ms]	1.6[ms]
0001	2^{1} × main clock cycle	500[ns]	100[ns]
0010	2^{5} × main clock cycle	8[µs]	1.6[µs]
0011	2^{6} × main clock cycle	16[µs]	3.2[µs]
0100	2^{7} × main clock cycle	32[µs]	6.4[µs]
0101	2^{8} × main clock cycle	64[µs]	12.8[µs]
0110	2^{9} × main clock cycle	128[µs]	25.6[µs]
0111	2^{10} × main clock cycle	256[µs]	51.2[µs]
1000	2^{11} × main clock cycle	512[µs]	102.4[µs]
1001	2^{12} × main clock cycle	1[ms]	200[µs]
1010	2^{13} × main clock cycle	2[ms]	400[µs]
1011	2^{14} × main clock cycle	4[ms]	800[µs]
1100	2^{17} × main clock cycle	33[ms]	6.6[ms]
1101	2^{19} × main clock cycle	131[ms]	26.2[ms]
1110	2^{21} × main clock cycle	524[ms]	104.8[ms]
1111	2^{23} × main clock cycle	2[s]	400[ms]

(Note) Note that if the clock supervisor function is enabled and the failure detection cycle is shorter than the cycle set in this register, a failure is detected during the oscillation stabilization wait time.The period of the failure detection cycle is as follow.

$2^{12} \times$ CR Oscillation time = approx. 40.96ms

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

This register controls the timer operating with the main clock which performs PLL/SSCG clock oscillation stabilization waits.

PLL/SSCG clock oscillation stabilization wait timer is used only for oscillation stabilization wait time for the PLL/SSCG clock (PLLSSCLK).

PLL/SSCG clock oscillation stabilization wait time is the one set for PLLCR:POSW[3:0].

PLL/SSCG clock oscillation stabilization wait timer starts on being PLL/SSCG clock oscillation enabled (CSELR:PCEN="1") and ends after having run for the period of time set by the oscillation stabilization wait time. It is cleared after a PLL/SSCG clock oscillation stop (CSELR:PCEN="0").

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

7	6	5	4	3	2	1	0	bit
PTIF	PTIE	-						
0	0	0	0	0	0	0	0	Initial value
R(RM1), W	R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

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

The flag indicates that an overflow occurred after the time set for the PLL clock oscillation stabilization wait time select (PLLCR:POSW[3:0]). When PTIE bit is "1" and this bit is set, a PLL/SSCG clock oscillation stabilization wait timer interrupt request is generated.

Clear factor	- "0" write A DMA transfer triggered by PLL/SSCG clock oscillation stabilization wait timer
Set factor	- The end of oscillation stabilization wait time of the PLL/SSCG clock oscillation stabilization wait clock after PCEN=0 to 1.

Writing "1" to this bit is ineffective.

When PTIE bit is set to "0", clearing of this bit by a DMA transfer will not take place.

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

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

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

This bit controls interrupts by PLL/SSCG clock oscillation stabilization wait timer overflows as follows.

PTIE	Operations
0	Interrupt disabled (initial value)
1	Interrupt enabled (output the interrupt request at the time PTIF bit is "1")

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

6.4.9 CCPSSELR (CCtl PII/Sscg clock SELect Register) PLL/SSCG clock select register

This register determines the selection of clock source to feed the system.

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

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

7	6	5	4	3	2	1	0	bit
-							PCSEL	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] PCSEL (PII Clock source SELect) PLL/SSCG Clock source select

PLL/SSCG clock source is selected.

PCSEL	PLL or SSCG
0	Select PLL
1	Select SSCG

(Notes)
- When PCSEL="0", SSCG is always in reset status (because it is unused).
- When PCSEL="1", PLL clock is supplied to CAN and OCDU.

6.4.10 PLL/SSCG output clock divide setting register: CCPSDIVR (CCtl PIl/Sscg clock DIVide Register)

This register sets the PLL/SSCG clock divide ratio.

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

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

7	6	5	4	3	2	1	0	bit
-	PODS[2:0]			-	SODS[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W	Attribute

[bit7] Undefined

This bit always reads"0". Writing has no effect on operation.

[bit6 to bit4] PODS (PIl Oscillator Divider Select): PLL macro oscillation clock divide ratio select

PLL clock divide ratio is configured.

PODS[2:0]	Divide 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

(Notes) This bit can only be set as even divide. Odd divide cannot be set.
The duty of clock output is 50 %.
PLL clock must always be set to run under 128 MHz. (Operation above 128 MHz is not guaranteed.)

[bit3] Undefined

This bit always reads"0". Writing has no effect on operation.

[bit2 to bit0] SODS (Sscg Oscillator Divider Select): SSCG macro oscillation clock divide ratio select

SSCG clock divide ratio is set.

SODS[2:0]	Divide 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

(Notes) This bit can only be set as even divide. Odd divide cannot be set.
The duty of clock output is 50 %.
SSCG clock must always be set to run under 128 MHz. (Operation above 128 MHz is not guaranteed.)
This setting value has limitations. Prior to setting, refer to "■ Limitations while using the PLL/SSCG clock".

6.4.11 PLL feed back divide setting register: CCPLLFBR (CCtl PLL FB clock divide Register)

This register sets the multiplication rate of PLL.

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

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

7	6	5	4	3	2	1	0	bit
-	IDIV[6:0]							
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

[bit7] Undefined

This bit always reads"0". Writing has no effect on operation.

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

The PLL's multiplication rate is set.

IDIV[6:0]	Divide ratio setting
0000000-0001011	Setting disabled
0001100	13
0001101	14
0001110	15
•••	••••••
1100010	99
1100011	100
1100100-1111111	Setting disabled

(Note) This setting value has limitations. Prior to setting, refer to "■ Limitations while using the PLL/SSCG clock".

6.4.12 SSCG feed back divide setting register 0: CCSSFBR0(CCtl SScg FB clock divide Register 0)

This register sets SSCG's multiplication ratio N. Together with the setting of CCSSFBR1, the multiplication ratio of SSCG is P × N.

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

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

7	6	5	4	3	2	1	0	bit
-		NDIV[5:0]						
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] Undefined

These bits always read "0". Writing has no effect on operation.

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

This register sets SSCG's multiplication ratio N.

NDIV[5:0]	Divide ratio setting
000000	Setting disabled
000001	2
000010	3
000011	4
•••	••••••
111101	62
111110	63
111111	Setting disabled

(Note) This setting value has limitations. Prior to setting, refer to "■ Limitations while using the PLL/SSCG clock".

6.4.13 SSCG feed back divide setting register 1: CCSSFBR1(CCtl SScg FB clock divide Register 1)

This register sets SSCG's multiplication ratio P. Together with the setting of CCSSFBR0, the multiplication ratio of SSCG is P × N.

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

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

7	6	5	4	3	2	1	0	bit
-			PDIV[4:0]					
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit5] Undefined

These bits always read "0". Writing has no effect on operation.

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

This register sets SSCG's multiplication ratio P.

PDIV[4:0]	Divide ratio setting
00000	1
00001	2
00010	3
00011	4
•••	••••••
11101	30
11110	31
11111	Setting disabled

(Note) This setting value has limitations. Prior to setting, refer to "■ Limitations while using the PLL/SSCG clock".

6.4.14 SSCG config setting register 0: CCSSCCR0 (CCtl SSCg Config Register 0)

Various SSCG settings can be configured.

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

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

7	6	5	4	3	2	1	0	bit
Reserved				SFREQ[1:0]		SMODE	SSEN	
0	0	0	1	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R/W1	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit5] Reserved

Writing has no effect.

[bit4] Reserved

Writing has no effect on operation.

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

Spread modulation frequency of SSCG is configured.

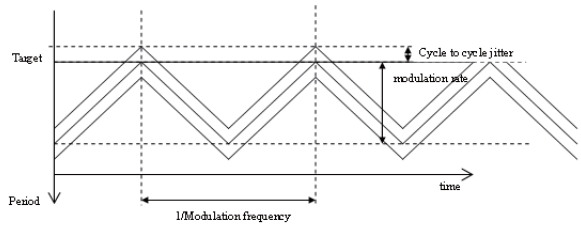
SFREQ[1:0]	Modulation frequency
00	1/1024
01	1/2048
1x	1/4096

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

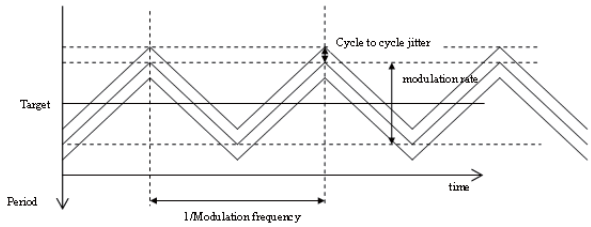
SSCG's spread modulation mode is configured.

SMODE	Modulation mode
0	Down Spread
1	Center Spread

●Down Spread



●Center Spread



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

This enables SSCG's spread spectrum

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

6.4.15 SSCG config setting register 1: CCSSCCR1 (CCtl SSCg Config Register 1)

Various SSCG settings can be done.

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

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

15	14	13	12	11	10	9	8	bit
RATESEL[2:0]			Reserved					
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R/W0	R/W0	Attribute

7	6	5	4	3	2	1	0	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	Attribute

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

Spread modulation rate of SSCG is configured.

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

[bit12 to bit10] Reserved

Writing has no effect.

[bit9 to bit0] Reserved

Always write "0" to these bits.

6.4.16 Clock gear config setting register 0: CCCGRCR0 (CCtl Clock Gear Config Register 0)

Various clock gear settings can be done.

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

7	6	5	4	3	2	1	0	bit
GRSTS[1:0]		-				GRSTR	GREN	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM0), W1	R/W	Attribute

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

Clock gear's status is shown.

GRSTS[1:0]	Status
00	Stopped at low-speed oscillation status of the clock gear, or clock gear is not used (CCCGRCR0.GREN="0"), or PLL/SSCG is in reset status (CSELR.PCEN="0")
01	GEAR UP is in operation
10	Stopped at high-speed oscillation status of the clock gear
11	GEAR DOWN is in operation

[bit5 to bit2] Undefined

These bits always read "0". Writing has no effect on operation.

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

Writing of "1" to this bit initiates clock gear operation.
Operation behavior of clock gear differs depending on the GRSTS bit value. (gear-up, or gear-down)
GRSTS=00

GRSTR	Operation
"0" write	No affect in operation
"1" write	Gear-up operation is starting

GRSTS=01/11

GRSTR	Operation
"0" write	No affect in operation
"1" write	No affect in operation

GRSTS=10

GRSTR	Operation
"0" write	No affect in operation
"1" write	Gear-down operation is starting

(Notes) This bit can be written only when CSELR.CKS=10 (PLL/SSCG clock (PLLSSCLK) is selected) and CCCGRCR0.GREN=1 (clock gear enabled).
This bit is automatically cleared to "0" when clock gear-up (or down) is completed. It is also cleared to "0" when CSELR.PCEN=0 (PLL/SSCG clock oscillation is stopped).
For read-modify-write instruction, this bit is always read as "0". When this bit is "1" and writing is initiated, writing after second time is ignored.

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

Clock gear operation is enabled.

GREN	Operation
0	Clock gear is not used
1	Clock gear is used

(Note) This bit is write enabled only while PLL/SSCG clock oscillation is stopped (CSELR.PCEN=0).

6.4.17 Clock gear config setting register 1: CCCGRCR1 (CCtl Clock Gear Config Register 1)

Various clock gear settings can be made.

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

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

7	6	5	4	3	2	1	0	bit
GRSTP[1:0]		GRSTN[5:0]						
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

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

Step count while performing clock gear-up/down (increment/decrement count) is selected.

GRSTP	Number of steps
00	1
01	2
10	3
11	4

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

Step at where clock gear operation started is selected. A step between 0 to 63 can be selected.

GRSTIN	Number of steps
000000	0
000001	1
000010	2
•••	••••••
111101	61
111110	62
111111	63

(Note) If the setting is GRSTN = 111111 (63 steps), the gear does not operate.

6.4.18 Clock gear config setting register 2: CCCGRCR2 (CCtl Clock Gear Config Register 2)

Various settings for clock gear can be performed.

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

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

7	6	5	4	3	2	1	0	bit
GRLP[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

[bit7 to bit0] GRLP (clock GeaR LooP number select): Clock gear repeat time selection

Repeat time of 1 step is selected. Configurable number can be between 1 to 256. After configured repeat times are over, step is incremented or decremented.

GRLP	Number of steps
0000_0000	1
0000_0001	2
0000_0010	3
•••	••••••
1111_1101	254
1111_1110	255
1111_1111	256

6.4.19 PMU clock divide setting register 0: CCPMUCR0 (CCtl PMU Clock divide Register 0)

Divide setting of PMU clock is configured.

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

7	6	5	4	3	2	1	0	bit
FST	-					FDIV[1:0]		
0	0	0	0	0	0	0	0	Initial value
R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	Attribute

[bit7] FST (F-divider STatus monitor): F divider status monitor

After writing to FDIV[1:0] register, time lag is caused by clock transfer until written value is reflected. This bit can be used to monitor whether the setting value is properly reflected.

Registered value is normally reflected after the main clock × 4 cycles + PCLK1 × ca. 4 cycles.

FST	Monitor
0	Written value has reflected
1	Written value is not yet reflected

[bit6 to bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[bit1, bit0] FDIV (F-DIVide ratio setting): F divide ratio setting

This sets the divide ratio of F divider. PMU must be supplied with clock of 32kHz or less. Set up these bits to use the F divider with 32kHz or less.

FDIV[1:0]	Divide ratio	Target main oscillation frequency
00	128 division (Initial value)	At 4MHz
01	256 division	At 8MHz
10	384 division	At 12MHz
11	512 division	At 16MHz

(Notes)
- While CCPMUCR0.FST bit is "1", writing to these bits are ignored.
- If the main oscillation frequency used is at 20 MHz, set the division ratio to 640 divisions or more (= PMUCLK 32 kHz or less) by combining CCPMUCR0.FDIV[1:0] and CCPMUCR1.GDIV[4:0].
  Setting example: CCPMUCR0.FDIV[1:0]=00, CCPMUCR1.GDIV[4:0]=00100

6.4.20 PMU clock divide setting register 1: CCPMUCR1 (CCtl PMU Clock divide Register 1)

Divide setting of PMU clock is configured.

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

7	6	5	4	3	2	1	0	bit
GST	-		GDIV[4:0]					
0	0	0	0	0	0	0	0	Initial value
R,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] GST (GF-divider STatus monitor): G divider status monitor

After writing to GDIV[4:0] register, time lag is caused by clock transfer until written value is reflected. This bit can be used to monitor whether the setting value is properly reflected.

Registered value is normally reflected after the main clock × 4 cycles + PCLK1 × ca. 4 cycles.

GST	Monitor
0	Written value has reflected
1	Written value is not yet reflected

(Note) While this bit is "1", writing to the CCPMUCR1.GDIV[4:0] bit is ignored.

[bit6, bit5] Undefined

These bits always read "0". Writing has no effect on operation.

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

G divider's divide ratio is configured. The cycle of PMU clock must be 4 times or more of the APB bus clock cycle to be supplied to the PMU. Use this divider accordingly to set the divide ratio of PMU clock so as to meet the relation described above.

GDIV[4:0]	Prescale ratio
00000	No dividing (initial value)
00001	2
00010	3
•••	••••••
11101	30
11110	31
11111	32

(Notes)
- While CCPMUCR1.GST bit is "1", writing to these bits are ignored.
- If the main oscillation frequency used is at 20 MHz, set the division ratio to 640 divisions or more (= PMUCLK 32 kHz or less) by combining CCPMUCR0.FDIV[1:0] and CCPMUCR1.GDIV[4:0].
  Setting example: CCPMUCR0.FDIV[1:0]=00, CCPMUCR1.GDIV[4:0]=00100

6.4.21 Sync/Async Control Register: SACR (Sync/Async Control Register)

Peripheral clock (PCLK2) is selected.

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

7	6	5	4	3	2	1	0	bit
-							M	
1	1	1	1	1	1	1	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	Attribute

[bit7 to bit1] Undefined

These bits always read "1". Writing has no effect on operation.

[bit0] M : Synchronous/asynchronous setting of peripheral clock (PCLK2)

While SSCG clock is selected by CPU, the peripheral clock (PCLK2) can be switched.

M	Synchronous/asynchronous setting
0	Synchronous (CPU/peripheral uses PLL/SSCG clock)
1	Asynchronous (PLL/SSCG clock is used for CPU, and PLL clock for peripherals)

6.4.22 Peripheral Clock Divider Register: PICD (Peripheral Interface Clock Divider register)

Generated from PLL clock (PLLCLK), the divide setting of peripheral clock is configured.

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

7	6	5	4	3	2	1	0	bit
-				PDIV[3:0]				
1	1	1	1	0	0	1	1	Initial value
R1,WX	R1WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

These bits always read "1". Writing has no effect on operation.

[bit3 to bit0] PDIV : Peripheral clock divide ratio setting

While SACR.M=1, PLL clock (PLLCLK) [non-spread clock] is used to configure the divide ratio of peripheral clock (PCLK2).

PDIV[3:0]	PLL clock (PLLCLK) [non-spread clock] → PCLK2 divide ratio
0000	Not divided
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) Configure this register to set the peripheral clock (PCLK2) to always run under 40 MHz or less.

6.5 Operation

6.5.1 Oscillation Control

■ Main Clock (MCLK)

When the OCD tool is connected, the main clock always continues oscillations except the power-on reset (SINIT).

When the OCD tool is not connected, the main clock stops oscillations in one of the following conditions.

- SINIT reset (see "CHAPTER 8 RESET" for details)
- An interval in stop mode

After canceling all the oscillation stop conditions above, start supplying the clock once the oscillation stabilization wait time set for CSTBR:MOSW[3:0] has elapsed. For returning from reset input, the oscillation stabilization wait time set by the initial value will always be used because CSTBR:MOSW[3:0] will be initialized.

■ PLL/SSCG Clock (PLLSSCLK)

This LSI has PLL and SSCG (PLL that generates spread clock). SSCG can be used for noise reduction purpose. Clock combination that the CPU and peripherals can select are as follows.

Table 6.5-1 Clock Mode

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

Selection of CPU/peripheral (timer/communication) clock can be configured by using CCPSSELR.PCSEL. Also, while CPU operates under SSCG clock, peripheral (timer/communication) can be set to operate under PLL clock. In this case, the clock selection of peripheral can be done by using SACR.M, and divide setting can be done by using PICD.PDIV[3:0].

(Note) If CPU is operated by the SSCG clock while the peripherals is operated by PLL clock, access cycle penalty of up to 5 to 8 PCLK2 due to asynchronous clock transfer between CPU/peripheral is added. Set the same frequency for PCLK1 and PCLK2 for this time. In order to have both the CPU and peripherals operate under PLL clock, synchronization must be selected at SACR:M.

In one of the following conditions, PLL/SSCG clock (PLLSSCLK) stops oscillations.

- After a reset (Bus idle wait time should be taken into account before a stop. See "CHAPTER 8 RESET")
- An interval for which main clock oscillation stops (PCEN=0)
- An interval of main clock oscillation stabilization wait time (PCEN=0)
- An interval in stop mode
- An interval in watch mode
- An interval for which a clock other than PLL/SSCG clock (PLLSSCLK) is selected for the source clock (SRCCLK) and "0" is set for CSELR:PCEN

After the cancellation of all the above oscillation stop conditions, supply of the clock begins once the PLL/SSCG clock wait time set for PLLCR.POSW[3:0] has elapsed. To return from a reset input or an INIT state, PLL clock oscillation is stopped until "1" is set for CSELR.PCEN because CSELR.PCEN is initialized to "0".

Clock supply starts after a moment of PLL/SSCG clock lock standby time. When recovering from reset input or INIT status, CSELR.PCEN is initialized to "0", and until "1" is set, PLL/SSCG clock oscillation is stopped.

The calculation method of clock frequency and multiplication rate pertinent with PLL/SSCG is as follows.

(PLL/SSCG setting of microcontroller section)

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

Figure 6.5-1 Microcontroller section PLL peripheral block diagram

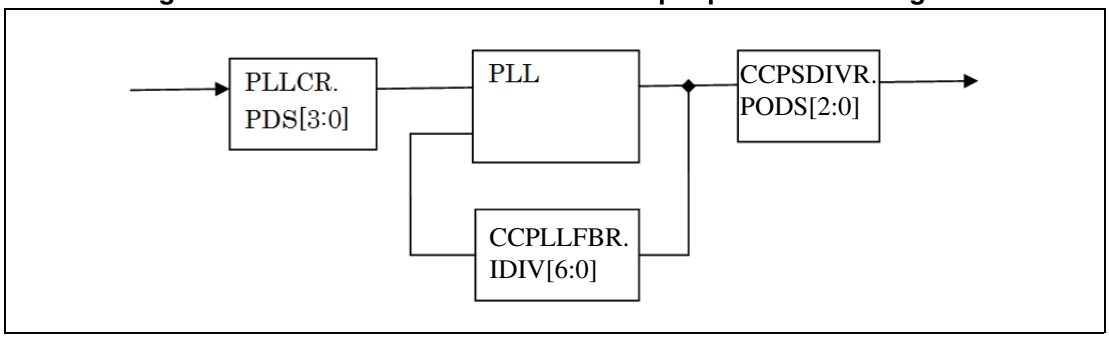
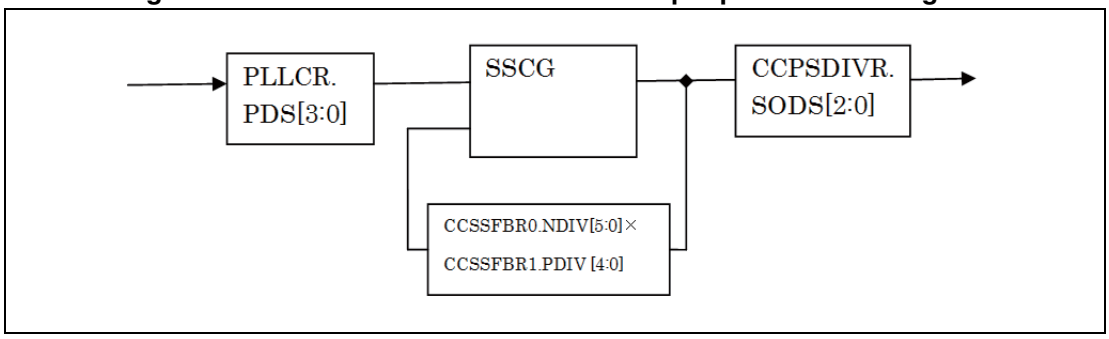


Figure 6.5-2 Microcontroller section SSCG peripheral block diagram



PLL/SSCG input clock, PLL/SSCG multiplication rate, and PLL/SSCG macro oscillating clock must be configured within the following operating condition of PLL/SSCG embedded in this series. Refer to the data sheet for the range of PLL/SSCG operating conditions.

(Notes)
- In debug operation, PLL cannot be stopped because MDI communication requires constant PLL clock supply.
- During PLL-SSCG switching, interrupt cannot be handled properly. Therefore, when switching the PLL-SSCG/synchronous-asynchronous status, you must block interrupt attempts from other resources.
- PLL-SSCG macro oscillating clock frequency has its upper/lower limit. Set up the PLL/SSCG multiplication rate not to exceed the following range.

  200MHz ≤ PLL macro oscillating clock frequency ≤ 333MHz

  200MHz ≤ SSCG macro oscillating clock frequency ≤ 333MHz (DownSpread)

■ Limitations while using the PLL/SSCG clock

When using the PLL / SSCG clock, observe the following limitations.

Clock control PLL clock frequency

Frequency (max)	FCTLR.FAW	CCPSSELR.PCSEL	Remarks
128MHz	01	0	
80MHz	00	0	

Notes:
- PLLCR, or CCPSDIVR and CCPLLFBR, must be configured properly not to exceed the frequency (max).
- Set DIVR2, PICD for the peripheral clock so as not to exceed 40MHz.

Microcontroller section clock control SSCG clock frequency

Frequency (max)	FCTLR. FAW	CCPSSELR. PCSEL	CCSSCCR0. SSEN	CCSSCCR0. SMODE	CCSSCCR1. RATESEL	Remarks
128MHz	01	1	1	0/1	000 to 110	
72MHz	00	1	1	0	000 to 110	DownSpread
72MHz	00	1	1	1	000	CenterSpread(0.5%)
72MHz	00	1	1	1	010	CenterSpread(1%)
72MHz	00	1	1	1	011	CenterSpread(2%)
71MHz	00	1	1	1	100	CenterSpread(3%)
71MHz	00	1	1	1	101	CenterSpread(4%)
70MHz	00	1	1	1	110	CenterSpread(5%)

Notes:
- CCPSDIVR, or CCSSFBR0 and CCSSFBR1, must be configured properly not to exceed the frequency (max).
- Set DIVR2 for the peripheral clock so as not to exceed 40MHz.

Relationship between the modulation rate and division ratio when using the SSCG

CCSSCCR1.RATESEL[2:0]		CCSSFBR0.NDIV[5:0]		
Modulation rate	Setting value	Division ratio range	Setting value lower limit	Setting value upper limit
0.50%	00x	8-60	7h	3Bh
1.00%	010	8-60	7h	3Bh
2.00%	011	8-48	7h	2Fh
3.00%	100	8-31	7h	1Eh
4.00%	101	8-23	7h	16h
5.00%	110	8-18	7h	11h

6.5.2 Oscillation Stabilization Wait

This section explains oscillation stabilization waits for individual clock input.

■ Stabilization Wait Time Generation Conditions

When the oscillation stop control of each clock is released, enter the oscillation stabilization wait state. After the oscillation stabilization wait time set for each clock has elapsed, the oscillation stabilization wait state is released to resume supplying the clock.

Because the configuration register will be initialized by a reset, the main clock (MCLK) goes into oscillation stabilization wait state if it has already stopped oscillation before removing the reset. If the main clock have started oscillation by reset at INIT level and RST level, oscillation stabilization wait time will not be taken because the main clock oscillation will not be stopped by reset at INIT level and RST level.

■ Stabilization Wait Time Select

Oscillation stabilization wait time for each clock can be changed by the setting of CSTBR, PLLCR and PTIMER.

- The initial values after resetting clock oscillation stabilization wait time
- Main clock CSTBR:MOSW[3:0] bit 2^{15} × main clock cycle
- PLL/SSCG clock PLLCR:POSW[3:0] bit 2^{16} × main clock cycle

Because a reset (INIT or RST) initializes CSTBR:MOSW[3:0], the main clock oscillation stabilization wait time specified by the initial value will always be used. Otherwise you can change the main clock oscillation stabilization wait time by setting CSTBR:MOSW[3:0].

Because a reset (INIT or RST) initializes PLLCR:POSW[3:0], the PLL/SSCG lock wait time specified by this initial value will always be used. Otherwise you can change the PLL/SSCG lock wait time by setting PLLCR:POSW[3:0]. After setting PLLCR:POSW[3:0], set CSELR:PCEN to "1". For details of POSW, see Section "6.4.6 PLLCR: (PLL Configuration Register) PLL Setting Register".

■ The End of Stabilization Wait Time

The operation is stopped if the clock selected for the source clock (SRCCLK) is in the oscillation stabilization wait state. The operation restarts at the end of oscillation stabilization wait time. The ready bits corresponding to each clock in the CMONR register at enabling clock indicate whether a clock that has not been selected for the source clock is waiting for oscillation stabilization or not.

Show the display oscillation stabilization of the clock oscillation stabilization wait state.

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

6.5.3 Source Clock (SRCCLK) Select

This section explains the selection control for the source clock (SRCCLK) which will become the running clock.

■ Source Clock Selection at Initialization

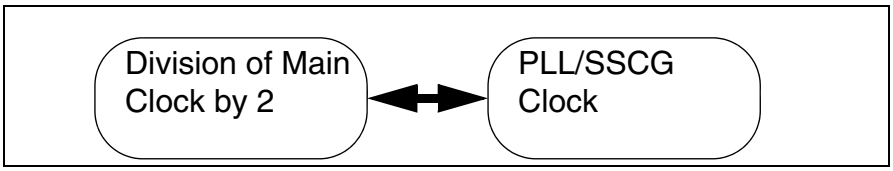
The source clock (SRCCLK) will select the division of main clock (MCLK) by 2 at the time of a trigger of reset (RST).

After that, the source clock can be changed by the setting of CSELR:CKS[1:0] after the program has started.

■ Source Clock Switching Steps

The value of oscillation enable bit (CSELR:xCEN) will be held even after switching the source clock. Set the oscillation stop as required.

Figure 6.5-3 Source Clock Switching Steps



1. Division of main clock by 2 → PLL/SSCG clock

Selecting division of main clock by 2 for the source clock (CMONR:CKM[1:0]=00)
↓
PLL/SSCG multiplication rate, SSCG modulation, PLL/SSCG select, PLL/SSCG lock wait time setting (Setting PLLCR/ CCPSSELR/CCPSDIVR/ CCPLLFBR/ CCSSFBR0/ CCSSFBR1/ CCSSCCR0/ CCSSCCR1)-- when PLL/SSCG oscillation is disabled.
↓
Clock gear setting (CCCGRCR0.GREN/CCCGRCR1/CCCGRCR2)
↓
PLL/SSCG timer interrupt factor clear (PTIF=0)
↓
(as required) PLL/SSCG timer interrupt enable setting (PTIE=1)
↓
PLL/SSCG oscillation start (PCEN=0 → 1)
↓
PLL/SSCG lock wait loop (loop until it turns to PCRDY=1) or interrupt wait
↓
PLL/SSCG timer interrupt clear (PTIF=0, PTIE=0)
↓
Switch source clock to a PLL/SSCG clock (CSELR:CKS[1:0]=00 → 10)
↓
Clock gear starts (CCCGRCR0.GRSTR=1)
↓
Clock gear is confirmed to be Stopped at high-speed oscillation status (CCCGRCR0.GRSTS[1:0]=10)
↓
Selecting PLL/SSCG clock for the source clock (CMONR:CKM[1:0]=10)

2. PLL/SSCG clock → division of main clock by 2

Selecting PLL/SSCG clock for the source clock (CMONR:CKM[1:0]=10)
↓
Clock gear starts (CCCGRCR0.GRSTR=1)
↓
Switch the source clock to division of main clock by 2 (CSELR:CKS[1:0]=10 → 00)
↓
Clock gear starts (CCCGRCR0.GRSTR=1)
↓
Clock gear is confirmed to be Stopped at low-speed oscillation status (CCCGRCR0.GRSTS[1:0]=00)
↓
Selecting a main clock for the source clock (CMONR:CKM[1:0]=00)

Figure 6.5-4 PLL/SSCG mode setting example main → PLL/SSCG

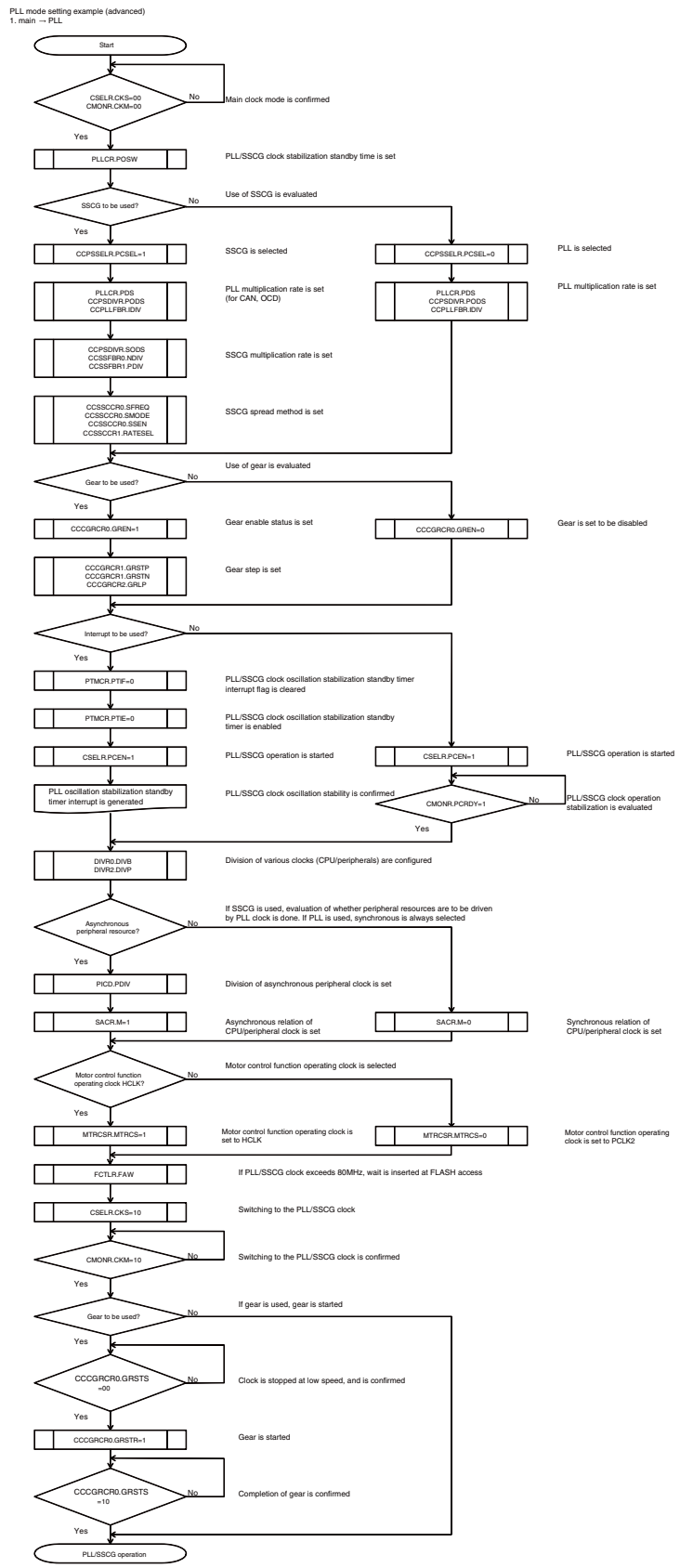
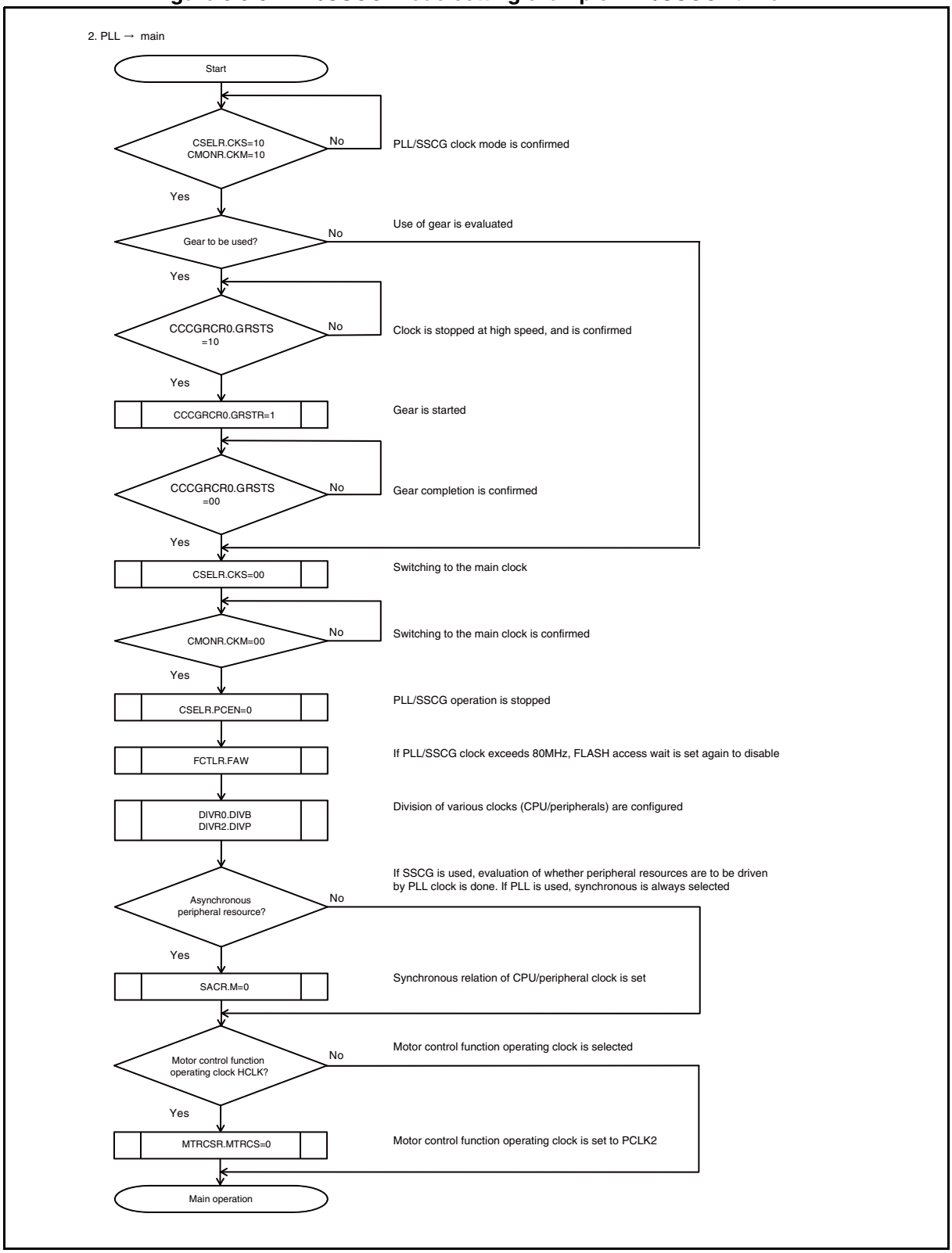


Figure 6.5-5 PLL/SSCG mode setting example PLL/SSCG → main



6.5.4 Timer

■ Main Clock Oscillation Stabilization Wait Timer (Main Timer)

Main timer is the timer operating with the main clock (MCLK). It is used for generating a main clock oscillation stabilization wait time and can be used as the timer generating an interrupt for a certain interval for other main clock stabilization states.

■ PLL/SSCG Clock Oscillation Stabilization Wait Timer (PLL Timer)

PLL timer is the timer operating with the main clock and exclusively used for generating PLL/SSCG clock oscillation stabilization wait time. This timer cannot be used as a general-purpose timer.

■ Configuration

The count of main timer starts when enabling main timer operation (MTMCR:MTE=1). The count of main timer stops and the main timer will be cleared when disabling main timer operation (MTMCR:MTE=0). The main timer is cleared when clearing main timer (MTMCR:MTC=1).

MTMCR:MTC=1 is read out while clearing. You can configure interrupt interval using MTMCR:MTS[3:0]. When MTMCR:MTIE=1, setting MTMCR:MTIF to "1" generates a main timer interrupt. MTMCR:MTIF is cleared by writing "0".

(Note) Configure PCLK1 × 5 clocks or more for the cycle of timer interrupt (MTS). Interrupt factor may not be set when you set extremely shorter interval for the timer interrupt.

■ Timer Interrupt Configuration Steps

This section explains the configuration steps for interrupts. Examples of interrupt configuration are as follows.

Timer interrupt disable setting (MTMCR:MTIE=0), interrupt flag clear setting (MTMCR.MTIF=0)
↓
Timer operation disable setting (MTMCR:MTE=0)
↓
Check for MTC=0
↓
Timer cycle setting (MTMCR:MTS=1000 to 1111)
↓
Timer interrupt enable setting (MTMCR:MTIE=1)
↓
Timer operation enable setting (MTMCR:MTE=1)
↓
An interrupt occurred after the set time elapsed
↓
To interrupt routine
↓
Interrupt flag clear setting (MTMCR:MTIF=0)
↓
Interrupt flag verify (MTMCR:MTIF=0)
↓
Program operation
↓
RETI

(Note) To set the interrupt flag clear, repeat reading until "0" is really reflected because writing "0" will not take effect immediately.

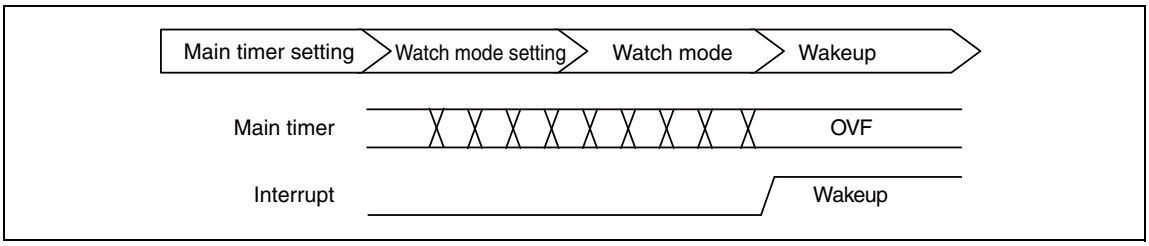
■ Operation of Timer

The main timer increments for main clock (MCLK) while MTMCR:MTE=1. When an overflow occurs in the interval selected by MTMCR:MTS[3:0], MTMCR:MTIF=1 will be set.

■ Watch Mode and Timer Interrupt

In watch mode, all operations but for some functions and the timers will be stopped. (See "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.) You can wake up from the watch mode by using a main timer interrupt.

Figure 6.5-6 Waking Up from the Watch Mode



6.5.5 Notes on Conflict of Clocks

Note that there would be a possibility that two interrupt handlers are called twice if an interrupt of a peripheral which is running with extremely far lower frequency than that of CPU clock (CCLK) is cleared within the interrupt handler and the interrupt handler ends immediately, the peripheral failing to finish its internal processes within the interrupt handler interval.

6.5.6 Clock gear circuit

Due to drastic frequency difference, switching from/to the main clock to/from PLL/SSCG clock causes huge fluctuation of power supply current. By utilizing the clock gear circuit equipped in the clock switch section, you can gradually shift the operating frequency from/to high-frequency to/from low-frequency, reducing sudden surge of power supply current.

■ Gear up procedure

1. After completion of oscillation stabilization standby timer, the initial step clock, selected to clock gear start step, is output.
2. When clock gear start (CCCGRCR0.GRSTR, PGRCR0.PGRSTR, and SGRCR0.SGRSTR) is set to 1 and its rising is detected, the clock gear status flag (CCCGRCR0.GRSTS[1:0], PGRCR0.PGRSTS[1:0], and SGRCR0.SGRSTS[1:0]) shifts from "00" to "01". (Gear-up start)
3. Following the clock gear step selection and repeat time selection, gear-up is initiated.
   Give smaller steps and more repetition time for smoother change.
4. When the clock reaches the maximum step, the clock gear status flag (CCCGRCR0.GRSTS[1:0], PGRCR0.PGRSTS[1:0], and SGRCR0.SGRSTS[1:0]) shifts from "01" to "10". (Gear-up completed, gear stopped)
   After this, the clock is output with the maximum steps (64 steps).
5. After the gear stops, clock gear start (CCCGRCR0.GRSTR, PGRCR0.PGRSTR, and SGRCR0.SGRSTR) is cleared to "0" by hardware.

■ Gear down procedure

1. When clock gear start (CCCGRCR0.GRSTR, PGRCR0.PGRSTR, and SGRCR0.SGRSTR) is set to 1 and its rising is detected, the clock gear status flag (CCCGRCR0.GRSTS[1:0], PGRCR0.PGRSTS[1:0], and SGRCR0.SGRSTS[1:0]) shifts from "10" to "11". (Gear-down start)
2. Following the clock gear step selection and repeat time selection, gear-down is initiated.
   Give smaller steps and more repetition time for smoother change.
3. When the clock reaches the minimum step, the clock gear status flag (CCCGRCR0.GRSTS[1:0], PGRCR0.PGRSTS[1:0], and SGRCR0.SGRSTS[1:0]) shifts from "11" to "00". (Gear-down completed, gear stopped)
   After this, the initial step clock selected for the clock gear start step is output.
4. After the gear stops, clock gear start (CCCGRCR0.GRSTR, PGRCR0.PGRSTR, and SGRCR0.SGRSTR) is cleared to "0" by hardware.

6.5.7 Operation under MDI communication

During MDI communication, control is in effect so that main oscillation does not stop while shifting to the STOP mode.

Also, during MDI high-speed communication, and when CSELR.PCEN is cleared, control is in effect so that PLL reference clock continues to be supplied.

PLL-related register values will be retained, and not be updated. If, however, CSELR.PCEN=0 is assigned by software, PLL-related register value can be freely updated (written).

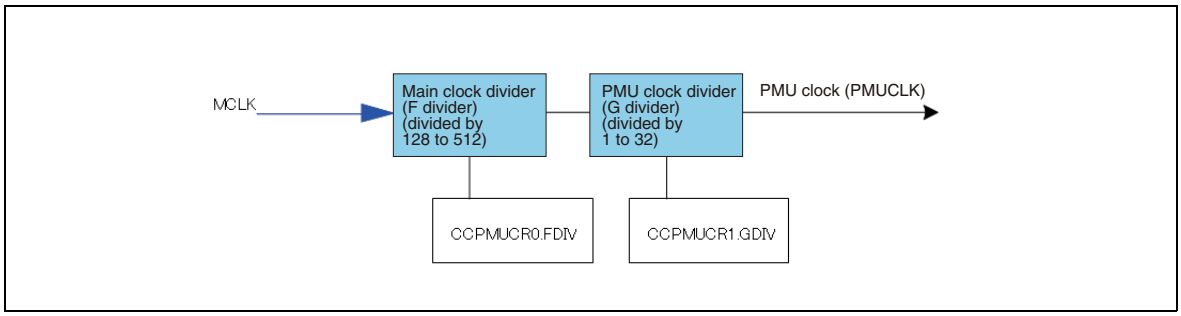
If value different from previous is written to a PLL-related register, and PLL/SSCG clock oscillation is enabled (CSELR.PCEN=1), PLL clock frequency will not be updated (PLL retains its lock status).

Normally, same values should always be written to the PLL-related registers.

(Notes) PLL-related registers are as follows.
- CCPSDIVR.PODS
- CCPLLFBR.IDIV
- PLLCR.PDS

6.5.8 About PMU clock (PMUCLK)

PMU clock is utilized for operating the power management unit (PMU). Prior to controlling standby mode, complete configuring of this clock.



Frequency of PMU clock is calculated as follows.

PMU clock frequency = main clock frequency ÷ CCPMUCR0.FDIV[1:0] divide ratio ÷ CCPMUCR1.GDIV[4:0] divide ratio

The following usage limitation must be followed by the PMU clock. (In case these limitations are violated, situations may be encountered that normal shutdown procedure cannot be completed.)

(1) F and G dividers must be set up for the PMU clock to run under 32kHz.

(2) G divider must be set up to 4 times or more of the peripheral clock (PCLK1) period.

Each usage limitation is explained as follows.

(1) F and G dividers must be set up to run the PMU clock under 32 kHz.

PMU clock is used to control the power switch. Due to stabilization of rising voltage at power-ON state, clock frequency of 32 kHz or less is recommended.

The main clock is selected as the source of PMU clock. Configure the CCPMUCR0.FDIV and CCPMUCR1.GDIV registers to set the PMU clock frequency below 32 kHz.

FDIV[1:0]	GDIV[4:0]	Divide ratio	Target main oscillation frequency
00	00000	128 division (Initial value)	At 4MHz
01	00000	256 division	At 8MHz
10	00000	384 division	At 12MHz
11	00000	512 division	At 16MHz
00	00100	640 division	At 20MHz

(2) G divider must be set up to 4 times or more of the peripheral clock (PCLK1) period.

Switching between peripheral clock (PCLK1) and PMU clock (PMUCLK) requires PMU clock × 4 cycles.

While clock source of the peripheral clock (PCLK1) is set to two cycles of the main clock (CMONR.CKM=00 or CMONR.CKM=01), if, under the configuration of DIVR0.DIVB or DIVR2.DIVP, peripheral clock (PCLK1) runs below 128kHz (32kHz × 4), CCPMUCR1.GDIV register must be set.

GIV[4:0]	Divide ratio
00000	No division (Initial value)
00001	2 division
•••	•••
11110	31 division
11111	32 division

[For reference]

The peripheral clock (PCLK1) frequency can be calculated as follows.

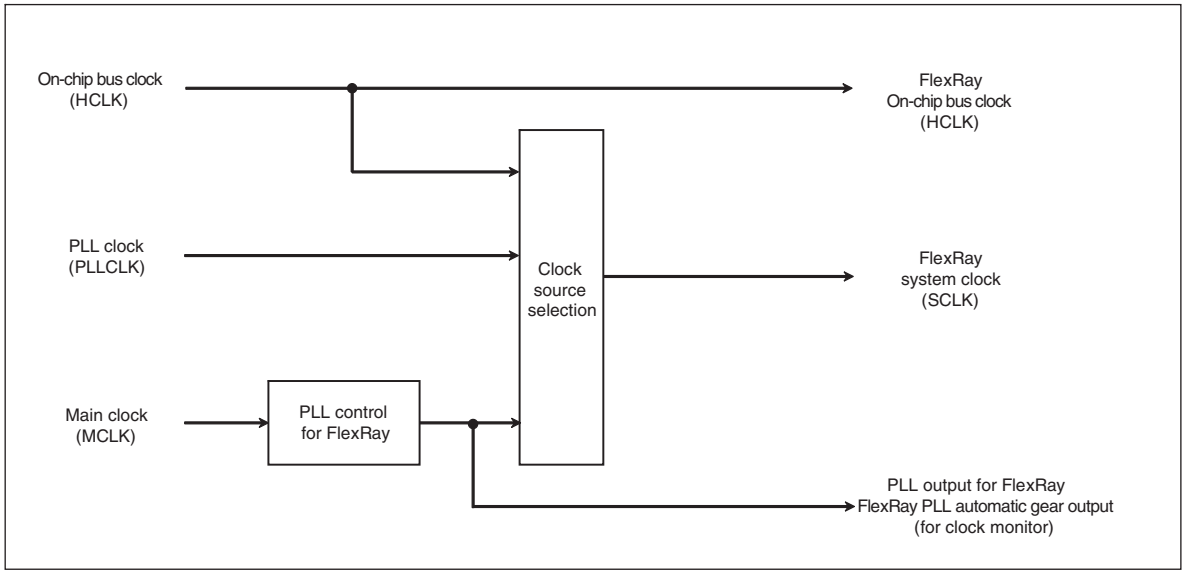
Clock (PCLK1) frequency = clock frequency selected by CMONR.CKM ÷ DIVR0.DIVB[2:0] divide ratio ÷ DIV2.DIVP[3:0]

CHAPTER 7 FLEXRAY DEDICATED CLOCK

7.1 Overview

This product includes a PLL for FlexRay separately from the PLL for the CPU core source clock.
This module controls PLL oscillation and the clock for FlexRay.
For FlexRay, the MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ/F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ has corresponding functions.

Figure 7.1-1 Block Diagram

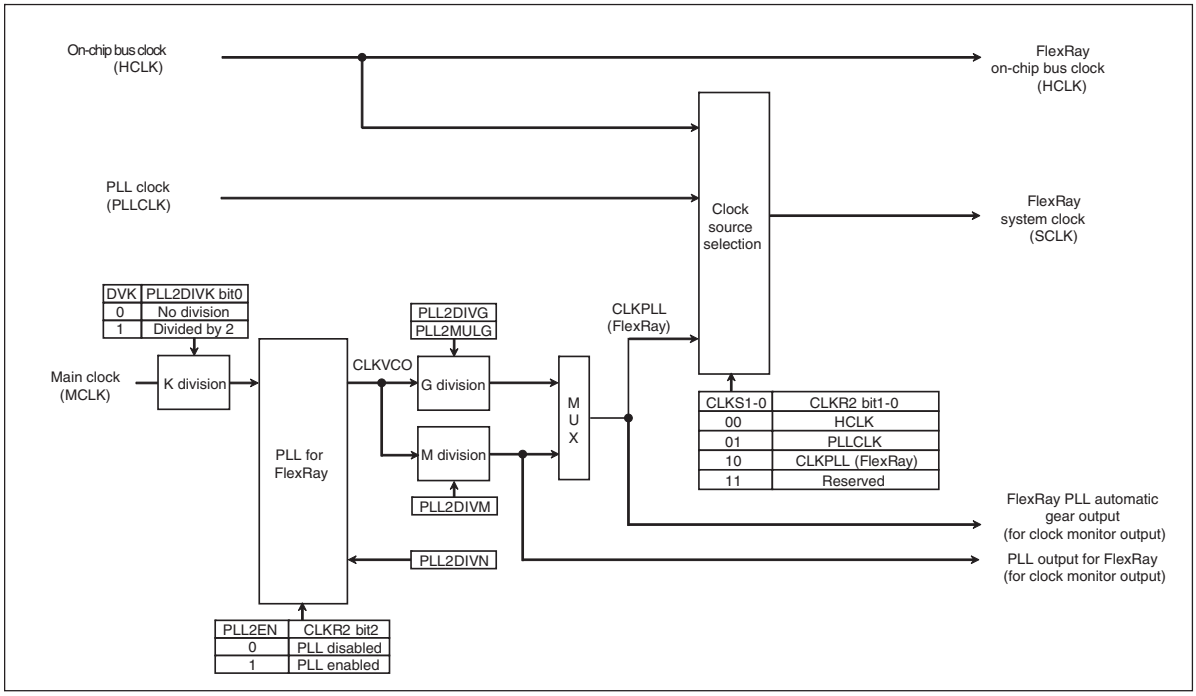


7.2 Features

- PLL multiplication rate can be set freely
- Clock automatic gear up/down function for preventing voltage drops and voltage surges
- FlexRay system clock (SCLK) source selection function
- Interrupt generation function triggered by the detection of the deadlock state of the FlexRay PLL macro

Code:BERAYPLL-0v6-91580MS-1-E

7.3 Configuration



(Notes) If "FlexRay PLLCLK" (CLKR2:CLKS[1:0]=10) is selected as the clock source, the register values of PLL2DIVM, PLL2DIVN, PLL2DIVG, and PLL2MULG cannot be changed.

7.4 Registers

Table 7.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0004E8	PLL2DIVM	PLL2DIVN	PLL2DIVG	PLL2MULG	FlexRay PLL multiplication rate (M division) selection register FlexRay PLL multiplication rate (N division) selection register FlexRay PLL automatic gear multiplication rate (G division) selection register Step multiplication rate selection register for FlexRay PLL G division
0x0004EC	PLL2CTRL	PLL2DIVK	CLKR2	Reserved	Automatic gear control register FlexRay PLL multiplication rate (K division) selection register FlexRay PLL clock output control register

7.4.1 PLL2DIVM:FlexRay PLL multiplication rate (M division) selection register

Address 0000_04E8$_H$ (Access: Byte, Half-word, Word)

This register selects the FlexRay PLL clock multiplication rate.

7	6	5	4	3	2	1	0	bit
-				DVM3	DVM2	DVM1	DVM0	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

These bits always read "0". Writing has no effect on operation.

[bit3 to bit0] DVM3 to DVM0: M division selection for CLKVCO

DVM3 to DVM0	CLKVCO M division (Generated ϕ: CLKPLL)
0000	CLKVCO: 1 (No division)
0001	CLKVCO: 2 (Divided by 2)
0010	CLKVCO: 3 (Divided by 3)
0011	CLKVCO: 4 (Divided by 4)
0100	CLKVCO: 5 (Divided by 5)
0101	CLKVCO: 6 (Divided by 6)
0110	CLKVCO: 7 (Divided by 7)
0111	CLKVCO: 8 (Divided by 8)
• • • •	• • • •
1111	CLKVCO: 16 (Divided by 16)

Notes:

- No division (:1) can be selected for the M division counter. However this setting is not recommended. The generated output clock should be an odd number of the clock duty ratio (PLL direct output).
  Always select a division ratio that is one or higher, and an even number of division ratio (2, 4, 6, etc.).
- An odd number of division ratio (3, 5, 7, etc.) can be selected for the M division counter. However this setting is not recommended.
  The generated output clock should be an odd number of the clock duty ratio.
  Always select an even number of division ratio (2, 4, 6, etc.).
- If CLKPLL (FlexRay) is selected as the clock source, the register value cannot be changed (CLKR2:CLKS[1:0]=10).
- When the PLL2DIVM and PLL2DIVN registers are changed, it is recommended to disable PLL (CLKR2: PLL2EN=0) and enable PLL (CLKR2: PLL2EN=1) later.

7.4.2 PLL2DIVN:FlexRay PLL multiplication rate (N division) selection register

Address 0000_04E9$_H$ (Access: Byte, Half-word, Word)

This register selects the multiplication rate from the PLL input clock to the FlexRay PLL clock.

7	6	5	4	3	2	1	0	bit
-	DVN6	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0	
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

[bit7] Undefined

This bit always reads "0". Writing has no effect on operation.

[bit6 to bit0] DVN6 to DVN0: N division selection for main clock

DVN6 to DVN0	Main clock N division
000000	Main clock: 1 (No division)
000001	Main clock: 2 (Divided by 2)
000010	Main clock: 3 (Divided by 3)
000011	Main clock: 4 (Divided by 4)
000100	Main clock: 5 (Divided by 5)
000101	Main clock: 6 (Divided by 6)
000110	Main clock: 7 (Divided by 7)
000111	Main clock: 8 (Divided by 8)
• • • •	• • • •
111111	Main clock: 128 (Divided by 128)

Notes:
- If CLKPLL (FlexRay) is selected as the clock source, the register value cannot be changed (CLKR2:CLKS[1:0]=10).
- When the PLL2DIVM and PLL2DIVN registers are changed, it is recommended to disable PLL for FlexRay (CLKR2: PLL2EN=0) and enable PLL for FlexRay (CLKR2: PLL2EN=1) later.

7.4.3 PLL2DIVG:FlexRay PLL automatic gear multiplication rate (G Division) selection register

Address 0000_04EA$_H$ (Access: Byte, Half-word, Word)

This register selects the FlexRay PLL clock gear multiplication rate.

7	6	5	4	3	2	1	0	bit
-				DVG3	DVG2	DVG1	DVG0	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

These bits always read "0". Writing has no effect on operation.

[bit3 to bit0] DVG3 to DVG0 (G division selection for PLL automatic gear start/end)

DVG3 to DVG0	PLL output start/end G division
0000	Automatic gear disabled (initial value)
0001	CLKVCO: 2 (Divided by 2)
0010	CLKVCO: 3 (Divided by 3)
0011	CLKVCO: 4 (Divided by 4)
0100	CLKVCO: 5 (Divided by 5)
0101	CLKVCO: 6 (Divided by 6)
0110	CLKVCO: 7 (Divided by 7)
0111	CLKVCO: 8 (Divided by 8)
• • • •	• • • •
1111	CLKVCO: 16 (Divided by 16)

Notes:
- For details on using this function, see Section "7.6 Clock Automatic Gear Up/Down".
- An odd number of division ratio (3, 5, 7, etc.) can be selected for the G division counter. However this setting is not recommended.
  Always select an even number of division ratio (2, 4, 6, etc.).
- If CLKPLL (FlexRay) is selected as the clock source, the register value cannot be changed (CLKR2:CLKS[1:0]=10).

7.4.4 PLL2MULG:FlexRay PLL step multiplication rate (G division) selection register

Address 0000_04EB$_H$ (Access: Byte, Half-word, Word)

This register selects the automatic gear step multiplication rate.

7	6	5	4	3	2	1	0	bit
MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0	
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 to bit0] MLG7 to MLG0: G division selection for PLL automatic gear step multiplication rate

MLG7 to MLG0	G division step multiplication rate
00000000	G division step × 1 (Multiplied by 1)
00000001	G division step × 2 (Multiplied by 2)
00000010	G division step × 3 (Multiplied by 3)
00000011	G division step × 4 (Multiplied by 4)
00000100	G division step × 5 (Multiplied by 5)
00000101	G division step × 6 (Multiplied by 6)
00000110	G division step × 7 (Multiplied by 7)
00000111	G division step × 8 (Multiplied by 8)
• • • •	• • • •
11111111	G division step × 256 (Multiplied by 256)

Notes:
- For details on using this function, see Section "7.6 Clock Automatic Gear Up/Down".
- If CLKPLL (FlexRay) is selected as the clock source, the register value cannot be changed (CLKR2:CLKS[1:0]=10).

7.4.5 PLL2CTRL: Automatic gear control register

Address 0000_04EC$_H$ (Access: Byte, Half-word, Word)

This register sets the automatic gear operation control.

7	6	5	4	3	2	1	0	bit
-				IEDN	GRDN	IEUP	GRUP	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R/W	R(RM1),W	R/W	R(RM1),W	Attribute

[bit7 to bit4] Undefined

These bits always read "0". Writing has no effect on operation.

[bit3] IEDN: Interrupt enable gear down

IEDN	Function description
0	Gear down interrupt disabled (initial value)
1	Gear down interrupt enabled

- To receive interrupt after switching the gear down, enable interrupt.

[bit2] GRDN: Interrupt flag gear down

GRDN	Function description
0	Gear down interrupt inactive (initial value)
1	Gear down interrupt active

- Once the G division counter reaches the programmed end value, this flag is set when the clock source: CLKPLL (FlexRay) is changed to the clock source: HCLK.
- In read-modify-write instructions, "1" is read from this bit.
  There is no influence by writing "1".

[bit1] IEUP: Interrupt enable gear up

IEUP	Function description
0	Gear up interrupt disabled (initial value)
1	Gear up interrupt enabled

- To receive interrupt after switching the gear up, enable interrupt.

[bit0] GRUP: Interrupt flag gear up

GRUP	Function description
0	Gear up interrupt inactive (initial value)
1	Gear up interrupt active

- Once the G division counter reaches the end value defined in the M division counter, this flag is set when the clock source: HCLK is changed to the clock source: CLKPLL (FlexRay).
- In read-modify-write instructions, "1" is read from this bit.
  There is no influence by writing "1".

7.4.6 PLL2DIVK: FlexRay PLL multiplication rate (K division) selection register

Address 0000_04ED$_H$ (Access: Byte, Half-word, Word)

This register selects the FlexRay PLL clock division.

7	6	5	4	3	2	1	0	bit
-							DVK	
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

[bit7 to bit1] Undefined

This bit always reads "0". Writing has no effect on operation.

[bit0] DVK: K division selection for MCLK

This bit selects the main clock (MCLK) division for the FlexRay PLL input clock as follows.

DVK	MCLK (PLL input clock) K division
0	MCLK/ 1 (No division)
1	MCLK/ 2 (Divided by 2)

(Notes)
- If CLKPLL (FlexRay) is selected as the clock source, the register value cannot be changed (CLKR2:CLKS[1:0]=10).
- To set the main clock (MCLK) for the FlexRay PLL input clock to 20 MHz, set this bit to "1". For a setting example, see "7.5 Recommended Setting".

7.4.7 CLKR2:FlexRay PLL clock output control register

Address 0000_04EE$_H$ (Access: Byte, Half-word, Word)

This register sets FlexRay operation control.

7	6	5	4	3	2	1	0	bit
FPOVF	FPOVIR	FPOVIE	Reserved		PLL2EN	CLKS1	CLKS0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,W	R/W	R/W0	R0,W0	R/W	R/W	R/W	Attribute

[bit7] FPOVF: FlexRay PLL alarm flag

When a deadlock state of FlexRay PLL macro is detected, this flag is raised.

FPOVF	Detection of FlexRay PLL deadlock state
0	Ordinary lock state
1	Deadlock state

[bit6] FPOVIR: FlexRay PLL alarm interrupt request flag

FlexRay PLL macro alarm interrupt request is indicated with this flag.

While this bit is set to "1" and FlexRay PLL alarm interrupt request (FPOVIE) is set to "1", FlexRay PLL alarm interrupt is generated.

FPOVIR	Read	Write
0	Ordinary lock state	Flag to be cleared
1	Deadlock state	Invalid

[bit5] FPOVIE: FlexRay PLL alarm interrupt request enabled

When FlexRay PLL alarm interrupt request flag bit becomes "1", this bit determines whether to generate FlexRay PLL alarm interrupt request.

FPOVIE	FlexRay PLL alarm interrupt request enabled
0	Interrupt request disabled
1	Interrupt request enabled

[bit4, bit3] Reserved

Always write "0" to these bits.

[bit2] PLL2EN (FlexRay PLL selection enabled)

This bit sets FlexRay PLL operation as follows.

PLL2EN	Function description
0	FlexRay PLL disabled (initial value)
1	FlexRay PLL operation enabled

- It is prohibited to change the FlexRay PLL operation enable bit (PLL2EN) when CLKPLL (FlexRay) is selected as the clock source (CLKS[1:0]=10).
- Set "0" to this bit before activating stand-by mode.

[bit1, bit0] CLKS1, CLKS0: SCLK output selection

These bits set the selection of SCLK output from FlexRay PLL-I/F as follows.

CLKS1, CLKS0	Function description (SCLK output selection)
00	HCLK (initial value)
01	PLLCLK
10	CLKPLL (FlexRay)
11	Reserved

- To use FlexRay, set CLKS[1:0]=10.

7.5 Recommended Setting

■ Recommended PLL-I/F settings for FlexRay

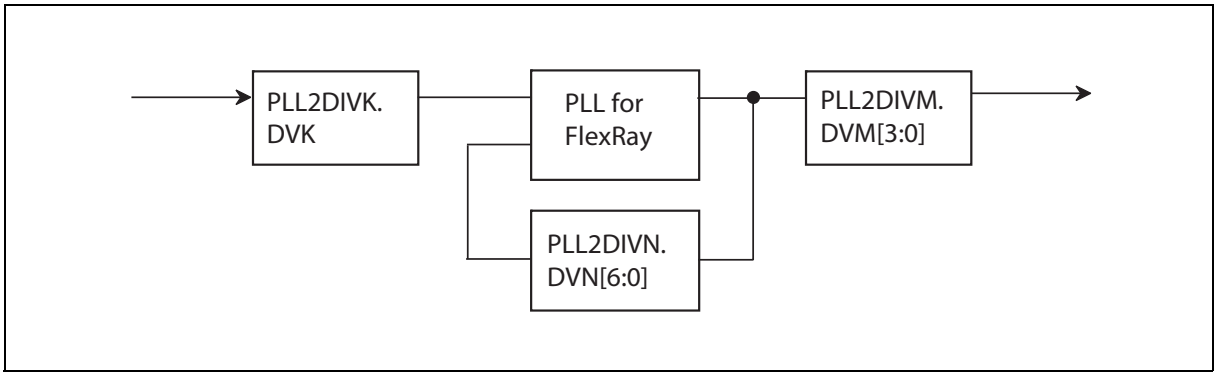
Main clock (MCLK) [MHz]	Frequency parameter			Clock gear parameter		PLL output for FlexRay (CLKVCO) [MHz]	FlexRay clock (SCLK) [MHz]
	DIVK	DIVM	DIVN	DIVG	MULG		
4	0	0011	100_1111	0000	0000_0000	320	80
20	1	0011	001_1111	0000	0000_0000	320	80

- To use FlexRay, it is recommended to set the values as shown in the above table.

(Note) Set the FlexRay dedicated clock to 80 MHz.

■ Calculating frequency

- PLL input frequency for FlexRay = (Main clock frequency)/(PLL2DIVK.DVK division ratio)
- PLL multiplication rate for FlexRay = (PLL2DIVN.DVN[6:0] multiplication rate)
- PLL output frequency for FlexRay = (PLL input clock frequency for FlexRay) × PLL multiplication rate for FlexRay
- FlexRay clock frequency = (PLL macro oscillating clock frequency for FlexRay)/(PLL2DIVM.DVM[3:0] division ratio)



(Note) The PLL macro oscillating clock frequency for FlexRay has upper and lower limits. Set the PLL multiplication rate within the following range.
200 MHz ≤ PLL macro oscillating clock frequency ≤ 400 MHz

7.6 Clock Automatic Gear Up/Down

In order to avoid voltage drops or surges when the clock source is switched from oscillation to high-frequency PLL output (or vice versa), a circuit is mounted in the PLL interface for the FlexRay so that clock gear-up and gear-down can be executed smoothly.

Main functions are implemented using two division counters (M division counter and G division counter).

In the M division counter, the target frequency is designated for PLL feedback.

In the G division counter, the frequency increases from the programmable frequency division designated in the G division setting (DIVG) to the target frequency designated in the M division setting (DIVM), and decreases from M division setting (DIVM) to the programmable end frequency (DIVG).

When the system clock is changed from a lower frequency to a higher frequency (gear up), or from a higher frequency to a lower frequency (gear down), only the DIVG > DIVM setting becomes the effective clock gear specification.

Frequency step is conducted using the PLL output frequency multiplication value as shown below.

Oscillator = 4 MHz, M = 4, N = 80 (Ex.: If PLL output = 320 MHz and frequency output to C unit = 80MHz, the frequency multiplication value will be N = 80)

The gear divider can be set to any even number divider.

■ Setting example

If DIVG=4 and MULG=20, the following gear up is executed when switching from oscillation to PLL.

1. Step: 1-cycle 16.0 MHz (16.0 MHz will become 20-cycle PLL output)
2. Step: 2-cycle 16.8 MHz (16.8 MHz will become 19-cycle PLL output)
3. Step: 3-cycle 17.8 MHz (17.8 MHz will become 18-cycle PLL output)

:

16. Step: 16-cycle 64.0 MHz (64.0 MHz will become 5-cycle PLL output)
17. Step: 17-cycle 80.0 MHz (80.0 MHz will become 4-cycle PLL output)
18. Step: 18-cycle 106.7 MHz (106.7 MHz will become 3-cycle PLL output)
19. Step: 19-cycle 160.0 MHz (160.0 MHz will become 2-cycle PLL output)

→ The target frequency reached in the final transition step (16. to 17. in this example)

When a multiplication value is set in the gear multiplication rate register, each step will be multiplied.

The time after the start frequency is generated until the target frequency is reached can be calculated by the following formula.

$$\text{duration} = \text{mul} \cdot t \cdot \left[\sum_{k=1}^{i} k \cdot (i-k+1) - \sum_{k=j+1}^{i} k \cdot (i-k+1) \right]$$

This formula is the same as the following formula (the finite arithmetic series in the first sum term comes down to the following).

$$\text{duration} = \text{mul} \cdot t \cdot \left[\frac{i \cdot (i+1) \cdot (i+2)}{6} - \sum_{k=j+1}^{i} k \cdot (i-k+1) \right]$$

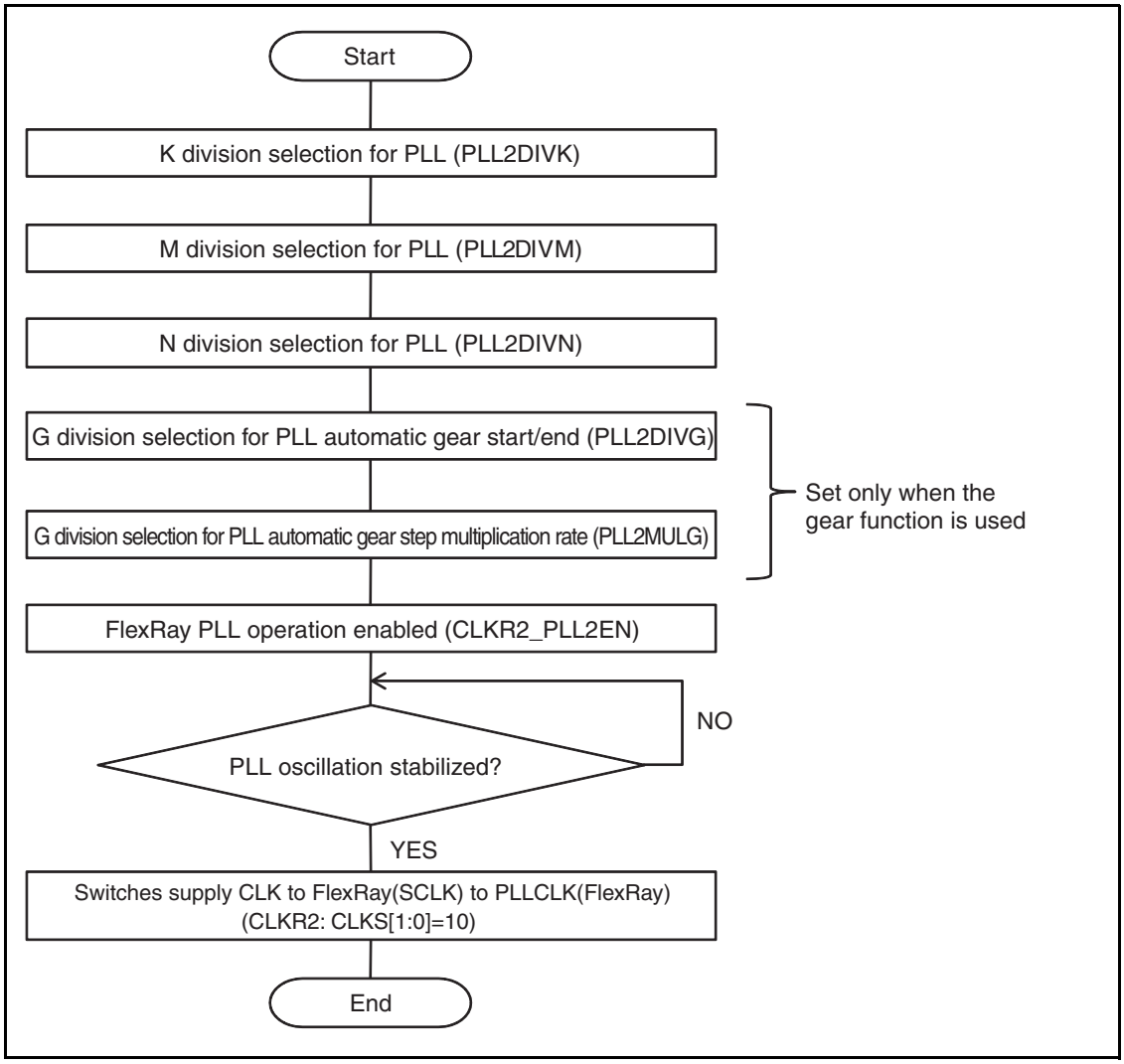
i = G, j = G - M, Mul = MULG, t = 1/f (PLLOUT)

The setting above becomes equal to the 1483PLL output clock cycle, for which the time from the start frequency to the target frequency is 9,262,500ps (approximately 9.3 μs).

7.7 Operations

This section explains FlexRay PLL clock settings.

■ Clock Setting Procedures (Example)



- Use the main timer for the stabilization wait time of FlexRay PLL, and wait for the switching time.

7.8 Notes

■ Clock Automatic Gear

When using the clock automatic gear function, it is recommended to assess the current state of this function by using the gear -up and gear-down flags (PLL2CTRL: GRUP, PLL2CTRL: GRDN). In this way, it is possible to avoid a malfunction that could occur in the clock system by a setting change before completion.

Procedure example:

(1) Set the FlexRay PLL interface register (PLL2DIVN, PLL2DIVM, PLL2DIVG, PLL2MULG) according to the selected frequency and gear time.

(2) Set the FlexRay PLL to ON (CLKR2: PLL2EN=1).

(3) When an interrupt was received after switching the gear up or down, enable the relevant interrupt (PLL2CTRL: IEUP, PLL2CTRL: IEDN).

(4) Wait for the PLL stabilization wait time. The Stabilization wait time is 200us.

(5) Switch the clock source to PLLCLK2 (CLKR2: CLKS[1:0] "00" → "10").

(6) Wait for the PLL2CTRL: GRUP gear up flag (polling or interrupt) before switching the clock source back to HCLK, or check the PLL2CTRL: GRUP=1 setting before changing the bit in the CLKR2 register.

(7) Switch the clock source to HCLK (CLKR2: CLKS[1:0] "10" → "00").

(8) Wait for the PLL2CTRL: GRDN gear down flag (polling or interrupt) before switching the clock source back to PLLCLK2, or check the PLL2CTRL: GRDN=1 setting before changing the bit in the CLKR2 register.

(9) Set the FlexRay PLL to OFF (CLKR2: PLL2EN=0).

■ FlexRay PLL Control

After initialization, FlexRay PLL oscillation is stopped.While stopped, FlexRay PLL output cannot be selected as the clock source.

After starting the program, first set the multiplier for FlexRay PLL for the clock source, and wait until FlexRay PLL is locked. Then change the clock source. While waiting for FlexRay PLL to lock, it is recommended to use the main timer interrupt.

When FlexRay PLL output is selected as the clock source, FlexRay PLL cannot be stopped.

Writing to the register has no affect. Select the on-chip bus clock (HCLK) as the clock source before operations such as stopping the FlexRay PLL for changing to the stop mode.

■ FlexRay PLL Multiplier

Changing the FlexRay PLL multiplier setting to a value other than the initial value must be performed during or before enabling FlexRay PLL after the program is started.

After changing the multiplier setting, wait for FlexRay PLL lock time, and then switch the clock source.

While waiting for FlexRay PLL to lock, it is recommended to use the main timer interrupt.

To change the FlexRay PLL multiplier setting during normal operation, first change the clock source to one other than FlexRay PLL. As with the above, after changing the multiplier setting, wait for the FlexRay PLL to lock, and then switch the clock source.

CHAPTER 8 RESET

8.1 Overview

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.

8.2 Features

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 the standby (power shutdown)

Other than the case of irregular reset (see Section "8.4.1 RSTRR (ReSeT Result Register) Reset Source Register" for details), 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.

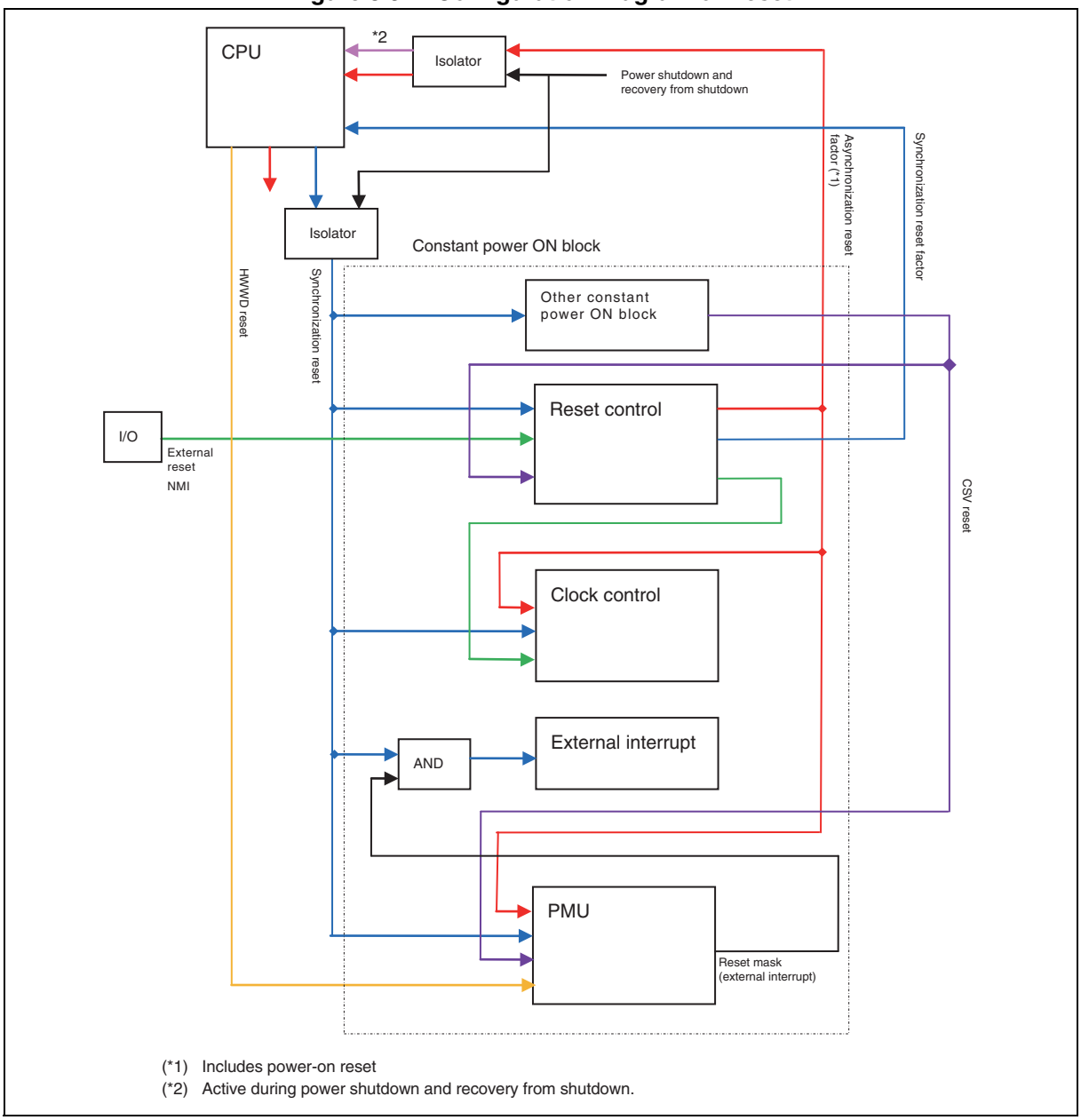
To issue a forced reset in case the bus does not return the response within a certain time frame, the device waits for the reset issue delay counter. If there is no response within the specified time frame, a reset will be issued whether or not the bus has responded. (Reset timeout)

For details on the clock supervisor reset, see "CHAPTER 11 CLOCK SUPERVISOR".

Code:ZBINIT-0v6-91580MS-1-E

8.3 Configuration

Figure 8.3-1 Configuration Diagram of Reset



(*1) Includes power-on reset
(*2) Active during power shutdown and recovery from shutdown.

Figure 8.3-2 Configuration Diagram of Reset

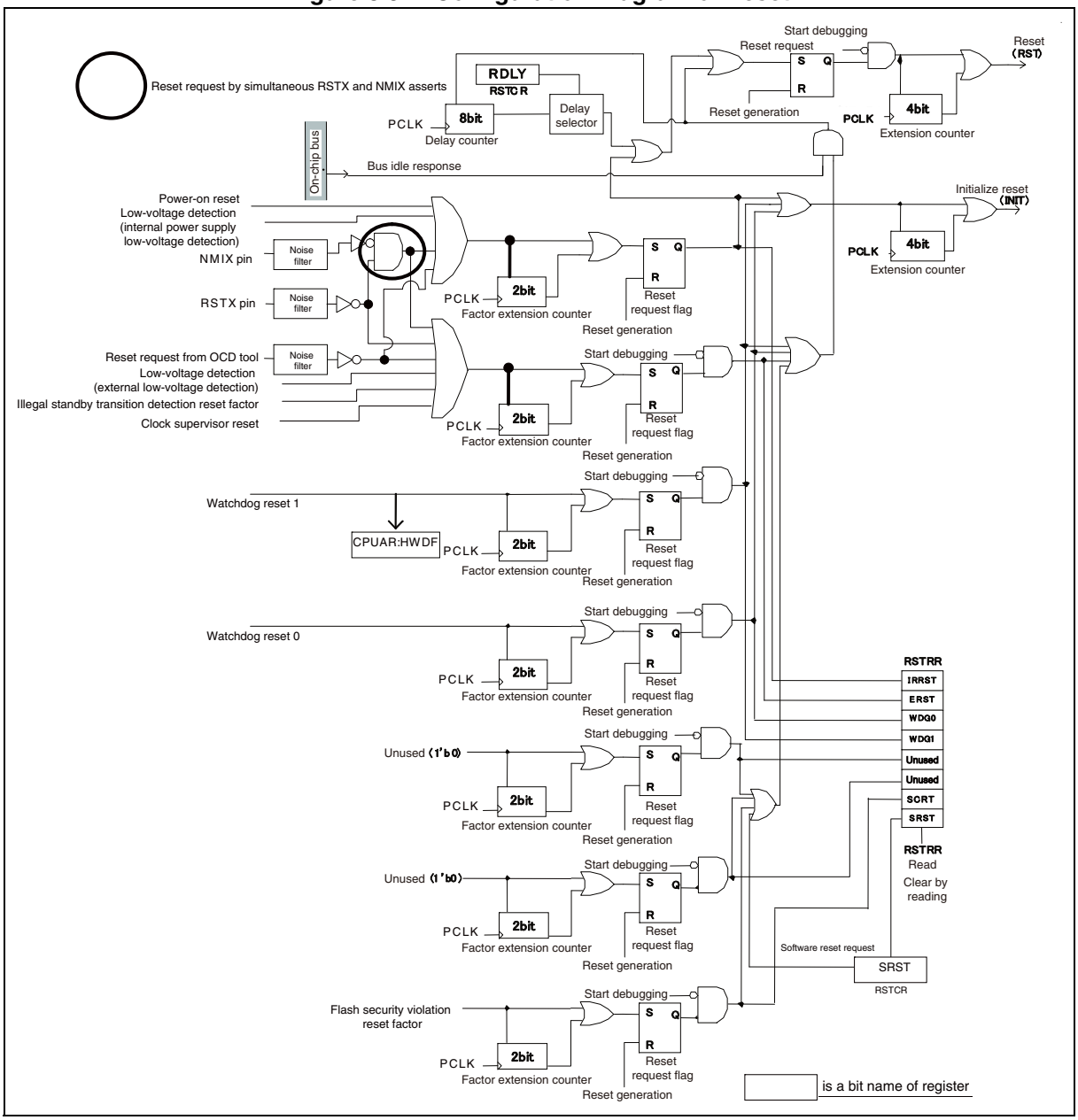
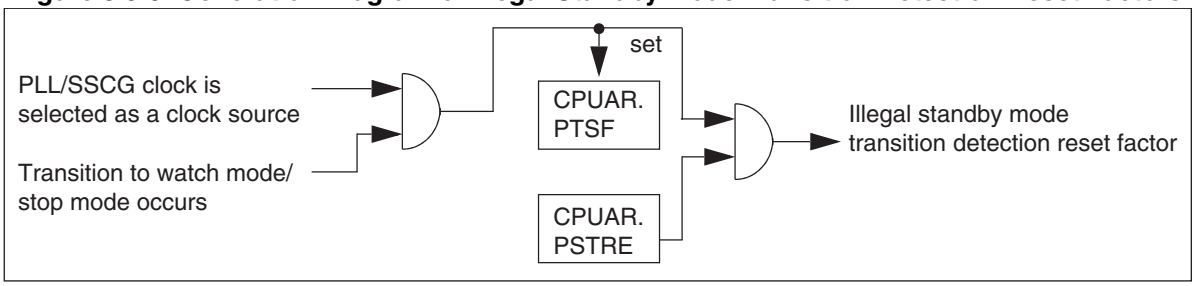


Figure 8.3-3 Generation Diagram of Illegal Standby Mode Transition Detection Reset Factors



8.4 Registers

Table 8.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x0480	RSTRR	RSTCR	Reserved	Reserved	Reset Source Register Reset Control Register
0x518	Reserved	Reserved	CPUAR	Reserved	CPU abnormal operation register
0x590	PMUSTR	Reserved	Reserved	Reserved	PMU Status Register

(Note) See "CHAPTER 9 POWER CONSUMPTION CONTROL" for address 0x0482, 0x0591 and 0x0592.

8.4.1 RSTRR (ReSeT Result Register) Reset Source Register

This register displays various reset factors generated until just before.

(Note) When this register is read out, all bits will be cleared.
Reading during debug state will not clear this register.
During debug state, each reset factor is masked. This register will not detect reset factors as well.

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

7	6	5	4	3	2	1	0	bit
IRRST	ERST	WDG1	WDG0	-		SCRT	SRST	
*	*	*	*	X	X	*	*	Initial value
R,WX	R,WX	R,WX	R,WX	RX,WX	RX,WX	R,WX	R,WX	Attribute

* Due to a reset factor

[bit7] IRRST (IRregular ReSeT) Irregular reset

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

IRRST	Irregular reset detected
0	Irregular reset undetected
1	Irregular reset detected

This bit will be cleared when it is read out.

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

This bit indicates the occurrence of a reset input from RSTX pin input, an illegal standby mode transition detection reset, external low-voltage detection, a clock supervisor reset, or simultaneous asserts of the RSTX and NMIX external pins.

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

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

This bit will be cleared when it is read out.

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

This bit indicates a reset from the watchdog timer 1.

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

WDG1	Watchdog timer 1 reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

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

[bit3, bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[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.

8.4.2 RSTCR (ReSeT Control Register): Reset Control Register

This register controls various types of reset issuance.

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

7	6	5	4	3	2	1	0	bit
RDLY[2:0]			Reserved				SRST	
1	1	1	0	0	0	0	0	Initial value
R,W	R,W	R,W	R/W0	R/W0	R/W0	R/W0	R,W	Attribute

[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 these bits after a reset factor has been detected. (The latter is a case of irregular reset). These bits can be written for only once after the reset.

RDLY[2:0]	Reset timeout value
000	PCLK × 2 cycles
001	PCLK × 4 cycles
010	PCLK × 8 cycles
011	PCLK × 16 cycles
100	PCLK × 32 cycles
101	PCLK × 64 cycles
110	PCLK × 128 cycles
111	PCLK × 256 cycles (initial value)

[bit4 to bit1] Reserved

Always write "0" to these bits.

[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.

Reset is not generated by RSTCR read during debug state.

SRST	Software reset
0	No output (initial value)
1	Reset request is output by RSTCR read

8.4.3 CPUAR (CPU Abnormal operation Register): CPU abnormal operation register

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

- **CPUAR: Address 051A$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
PSTRE	-			PDLF	PMDF	PSTF	HWDF	
0	0	0	0	*	*	*	*	Initial value
R/W	R0,WX	R0,WX	R0,WX	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	Attribute

* It will be initialized to "0" by RSTX pin assert. (This includes simultaneous asserts of RSTX and 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

(Notes) - 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 PSTF bit after the power-on reset is indefinite.

[bit6 to bit4] Undefined

These bits always read "0". Writing has no effect on operation.

[bit3] PDLF (Pll mode pll Dead-Lock detection Flag): PLL mode PLL dead-lock detection flag

If PLL output is selected as clock source, and PLL Dead-Lock detection is initiated, this bit will be set. In that case, source clock is automatically rewritten to the main mode (CKS=CKM=00).

For read-modify-write type instruction, "1" is read.

PDLF	Read	Write
0	No PLL Dead-Lock detection under PLL mode (default)	Clear this bit
1	PLL Dead-Lock to be detected under PLL mode	No effect

If set factor and clear factor occurs simultaneously, set factor takes precedence.

(Note) Dead-Lock detection is enabled only to the source selected at CCPSSELR.PCSEL.

[bit2] PMDF (PII mode Main clock Down detection Flag): PLL mode main oscillation failure detection flag

While PLL output is selected as clock source, this bit is set as clock supervisor detects main oscillation failure. Under such circumstance, source clock is automatically rewritten to the main mode (CKS=DKM=00), and reset (RST level) is generated at once.

For a read-modify-write type instruction, "1" is read.

PMDF	Read	Write
0	No main oscillation failure detection under PLL mode (default)	Clear this bit
1	Main oscillation failure to be detected under PLL mode	No effect

If set factor and clear factor occurs simultaneously, set factor takes precedence.

[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. Under such circumstance, source clock is automatically written to the main mode (CKS=CKM=00). If 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

(Notes) Before using this bit after the reset of a pin other than the RSTX pin, make sure to clear the bit.

[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

If set factor and clear factor occurs simultaneously, the former takes precedence.

(Note) Although RSTRR.WDG1 is also equipped with detection flag, it is read-clear; once read is completed, the factor is erased. Meanwhile, CPUAR.HWDF and its related factor is retained until cleared.

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

This register indicates the status of PMU.

- **PMUSTR: Address 0590_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
PMUST	Reserved					PONR_F	RSTX_F	
0	0	0	0	0	0	1	*	Initial value
R,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	R,W	Attribute

* It is initialized to "1" by an RSTX pin assert (including a simultaneous NMIX assert). It is not initialized by other reset factors.

[bit7] PMUST (Power Management Unit STatus)

Information of whether the previous state was power shutdown mode, is indicated.

PMUST	PMU Status
0	Initial state
1	Operation recovery from power shutdown mode

This bit is cleared as "0" is written. Writing "1" is invalid.

This bit is just initialized by the power-on reset, the low-voltage detection, and the simultaneous assert of RSTX and NMIX only, is not initialized by the other reset factors. Therefore, confirm the reset factors before judging whether the status is the retuning from the power shut-down mode or not.

[bit6 to bit2] Reserved

These bits always read "0". Writing has no effect on operation.

[bit1] PONR_F (Power ON Reset Flag)

This flag is used to detect power-on reset.

PONR_F	Power-on reset
0	Not detected
1	Detected

This bit is cleared when "0" is written. Writing "1" is invalid.

This bit is not initialized by reset factors other than a power-on reset.

[bit0] RSTX_F (ReSeTX input Flag)

This flag is used to detect external reset.

RSTX_F	RSTX input reset
0	Not detected
1	Detected

This bit is cleared when "0" is written. Writing "1" is invalid.

This bit is not initialized by a power-on reset. Before using this bit, make sure to clear it.

8.5 Operation

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

8.5.1 Reset Level

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

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

■ Initialize Reset (INIT)

It initializes all register settings and the entire hardware. It terminates the CPU programs running, and the program counter will be initialized. All peripheral circuits will be initialized. Main oscillation circuit continues to run. If it was inactive, it starts running again. In this case, PLL becomes inactive.

This reset level is applied at a reset by the following reset factors.

- Irregular reset
- Watchdog reset 0, 1

Only the following register will be initialized by this reset level.

- Register of the debug interface (OCDU)

■ Reset (RST)

It initializes the entire hardware and all registers except the ones initialized only by the initialize reset (INIT). It terminates the CPU programs running, and the program counter will be initialized. All peripheral circuits will be initialized. Main oscillation circuit continues to run. If it was inactive, it starts running again. In this case, PLL becomes inactive.

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.

8.5.2 Reset Factor

This section explains each reset factor of this product.

■ Power-on Reset

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).

■ RSTX Pin Input

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

A reset due to this reset factor is detected as an irregular reset at the time of a reset timeout or simultaneous assert of the NMIX pin.

Other than the irregular reset detection, a reset (RST) will be issued.

■ Watchdog Reset 0

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.

■ Watchdog Reset 1

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.

■ External Low-Voltage Detection Reset

Low-voltage detection (external supply 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 14 LOW-VOLTAGE DETECTION (EXTERNAL LOW-VOLTAGE DETECTION)" for the details of voltage detected.

■ Internal Low-Voltage Detection Reset

Low-voltage detection (internal supply voltage) is a hardware reset input from the low-voltage detection circuit located inside of the device. A reset due to this reset factor is detected as an irregular reset, and an initialize reset (INIT) is then issued.

See "CHAPTER 13 LOW-VOLTAGE DETECTION (INTERNAL LOW-VOLTAGE DETECTION)" for the details of voltage detected.

■ Illegal Standby Mode Transition Detection Reset

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.

■ Flash Security Violation Reset

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.

■ Clock Supervisor Reset

It is a reset issued when a violation of the main clock oscillation 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.

■ Software Reset (RSTCR: SRST)

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
```

■ Recovery from standby (power shut-off)

The major portion of the blocks, including the microcontroller section, performs operations corresponding to a power-on reset by activation from the standby mode. The power-on reset factor, however, is located always in the power-ON block, and reset factor register (RSTRR) does not indicate its detection. Instead, factors are indicated at PMU status register (PMUSTR). Upon restarting of microcontroller, confirm this register.
Reset caused by this reset factor issues initialize reset (INIT).

8.5.3 Reset Acceptance

This section explains the acceptance processing of each reset factor.

■ Generation of Reset Request

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

■ Acceptance of Reset Request

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.

■ Reset Issue Delay Counter

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.

■ Irregular Reset

If a reset is issued without confirming the completion of reset request processing, it will generate an irregular reset.

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

- Regardless of the type of reset factor, initialize reset (INIT) will be issued.
- Set the bit7: IRRST bit of RSTRR register to "1".

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

8.5.4 Reset Issue

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

■ Power-on Reset (SINIT)

This power-on reset (SINIT) is issued first for a power-on reset, internal low-voltage detection, or simultaneous RSTX/NMIX asserts. This reset is used exclusively for initializing the indefinite state of division circuits.

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

When this reset is issued, an initialize reset (INIT) and a reset (RST) will be always issued at the same time.

This reset initializes the clock control register.

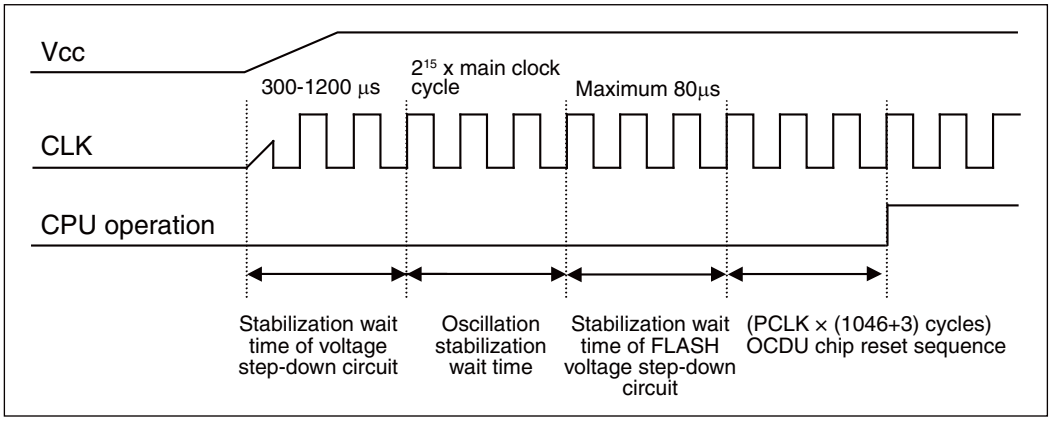
This reset involves the wait time of main clock oscillation to be stabilized. Along with the control register initialization, the oscillation stabilization wait time is 2^{15} main clock.

Table 8.5-1 Oscillation Stabilization Wait Time

Type	Main clock oscillation stabilization wait time
Power-on reset	2^{15} × main clock cycle
Internal low-voltage detection	2^{15} × main clock cycle
Simultaneous RSTX/ NMIX asserts	2^{15} × main clock cycle

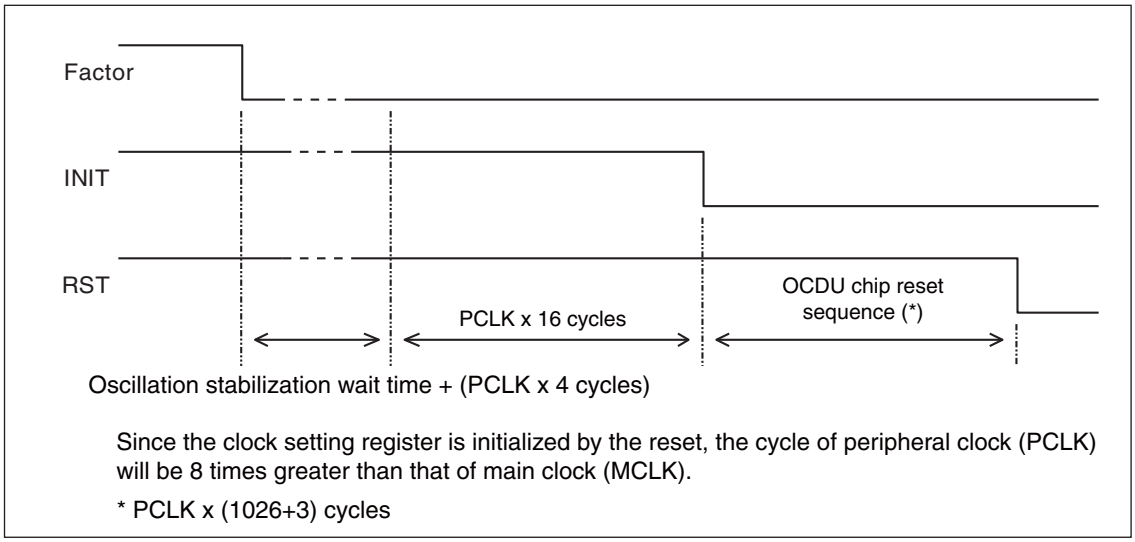
(Note) The regulator stabilization wait time and FLASH stabilization wait time that are related to power-ON and voltage recovery are not incorporated in the oscillation standby time of the above table. Under power-ON reset, this stabilization wait time (300-1200 μs or 80 μs) will be required.

Figure 8.5-1 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 8.5-2 Power-on Reset (SINIT) Sequence



■ Initialize Reset (INIT)

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, oscillation of the clock does not change while the main clock (MCLK) is oscillating.

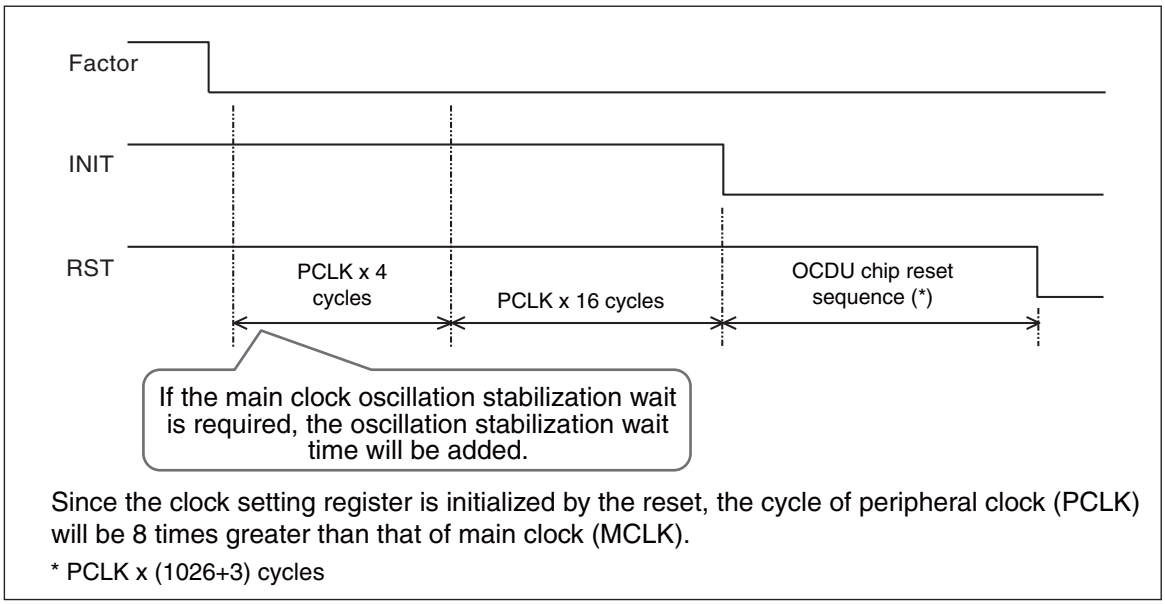
If the main clock is inactive such as in a stop mode, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} × main clock cycle).

Table 8.5-2 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 8.5-3 Initialize Reset (INIT) Sequence



■ Reset (RST)

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 "■ Initialize Reset (INIT)").

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

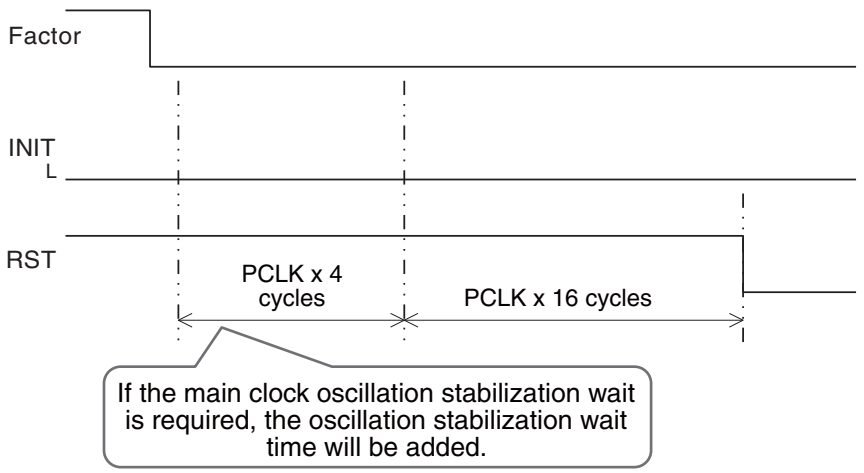
If the main clock is inactive such as in a stop mode before the reset, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} × main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

Table 8.5-3 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 8.5-4 Reset (RST) Sequence

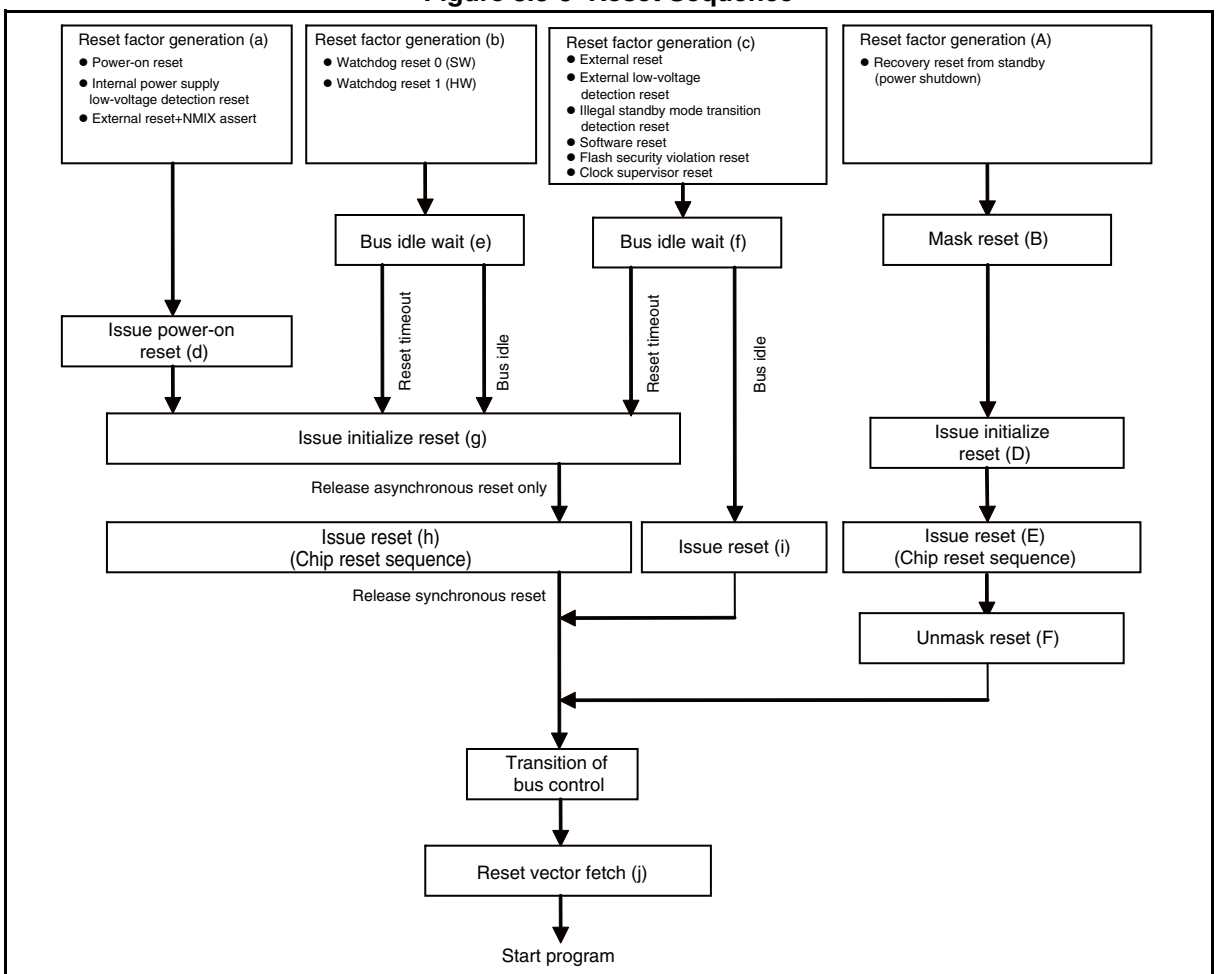


Since the clock setting register is initialized by the reset, the cycle of peripheral clock (PCLK) will be 8 times greater than that of main clock (MCLK).

8.5.5 Reset Sequence

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 8.5-5 Reset Sequence



(Notes)

- If (a) occurs after (g) or during (e)/(f), start over from (a).
- If (a) occurs after (B), the sequence starts over from (a).

See Figure 10.2-1 in "CHAPTER 10 CLOCK RESET STATE TRANSITIONS" for details.

- During (d), it takes the main clock oscillation stabilization wait time.
- During (g), (h), and (i), if necessary (CMONR: MCRDY=0), it takes the main clock oscillation wait time.
- (j) See "CHAPTER 32 FIXEDVECTOR FUNCTION" for details.
- In the case of an illegal standby mode transition detection reset, there is an immediate transition to (h) since it is already in the bus idle state after a reset factor occurs.
- In recovery (A) from standby (power shutdown), a power-on reset is issued. However, to suppress a reset to the following blocks, all resets except the power-on reset to this block are masked (B) during the resetting.

(1) External interrupt block

(2) Power supply management unit

■ Reset Cycle

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 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 main clock oscillation stabilization wait time, the cycle will be extended for the time required.

■ Reset Release

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.

■ Operating Mode Fix

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. Them, it will release the bus control of on-chip bus.

■ Transition of Bus Control

After the mode control circuit releases the bus control of on-chip bus, CPU acquires the bus control and starts running bus operations.

■ Reset Vector Fetch

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

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

■ Reset and Forced Break

If a forced break has occurred during the reset release, it accepts the forced break upon completion of the reset vector fetch. Thus, the PC value by the reset vector acquired will be saved at the emulator space side (stored at the E_BPCHR or E_BPCLR register).

■ Note

An internal reset is issued during recovery from the standby watch mode (power shutdown) and the standby stop mode (power shutdown). For this reason, reset factors other than a power-on reset, internal low-voltage detection reset, and reset by simultaneous RSTX/NMIX asserts are not accepted.

CHAPTER 9 POWER CONSUMPTION CONTROL

9.1 Overview

This product has a variety of low-power consumption modes and can perform the power consumption control feature accordingly for situations.

9.2 Features

■ Clock control

- **Clock division**

You can change the division ratio for each running clock and lower the running frequency accordingly. For details, see "CHAPTER 6 CLOCK".

■ Sleep mode

- **CPU sleep mode**

You can stop only the operation of CPU.

- **Bus sleep mode**

You can stop the operation of CPU and on-chip buses.

■ Standby mode

- **Watch mode**

You can stop all operations except for some clock oscillations and the timer.

- **Stop mode**

You can stop all clock oscillations and operations except for CR oscillation.

■ Standby mode (power shutdown)

- **Watch mode (power shutdown)**

You can shut down the power of the microcontroller unit and stop all operations except for some clock oscillations and the timer at the same time.

- **Stop mode (power shutdown)**

You can shut down the power of the microcontroller unit and stop all clock oscillations and operations at the same time.

Code:ZBPMU-0v4-91580MS-1-E

9.3 Configuration

Figure 9.3-1 Block Diagram of Overall Control

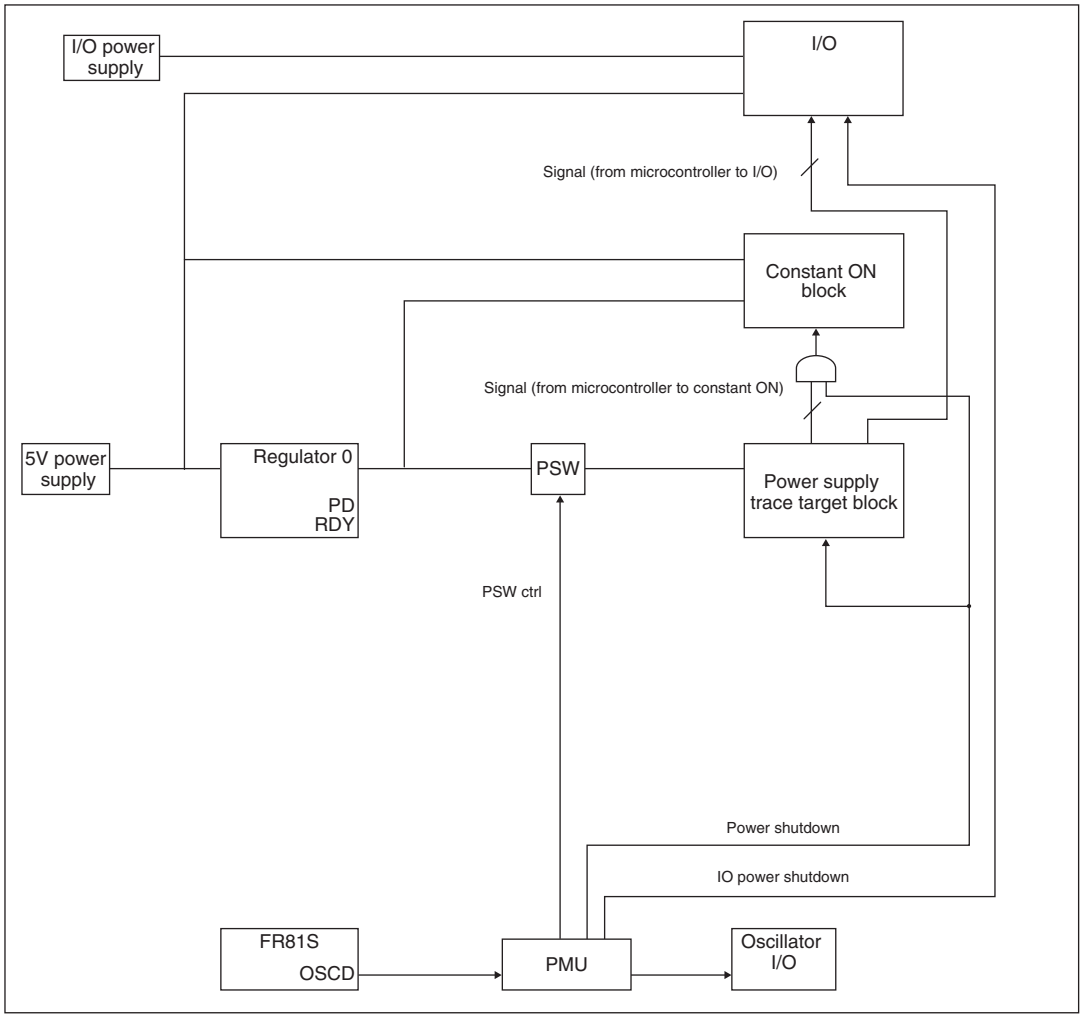
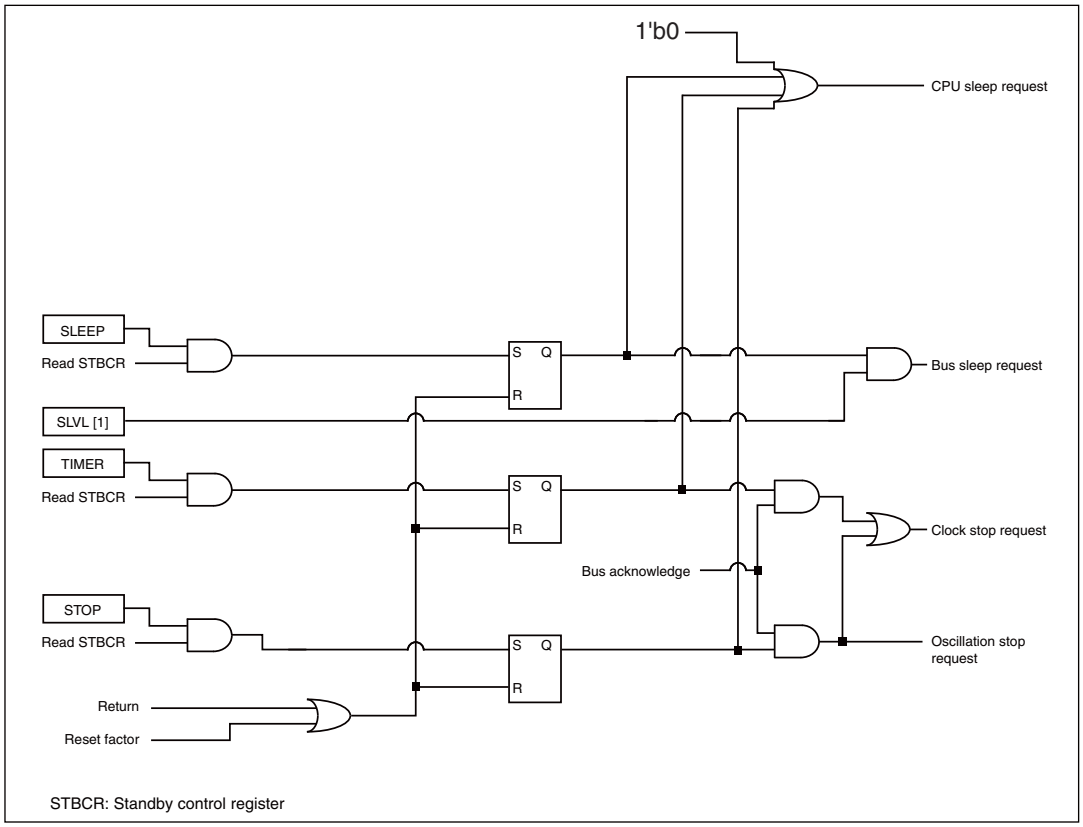


Figure 9.3-2 Block Diagram of Power Consumption Control



9.4 Registers

Table 9.4-1 Register Map

Address	Register				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	Reserved	PMUINTF1	PMUINTF2	Reserved	PMU interrupt flag register 1 to 2
0x059C	Reserved	Reserved	Reserved	Reserved	Reserved

(Note) Note that a "reset" register is assigned to each address from 0x0480 to 0x0481 and address 0x0590 (see "CHAPTER 8 RESET").

(Note) This register group (except for STBCR) is initialized only when any of the following factors occurs.

1. Power-on reset
2. Internal low-voltage detection
3. Simultaneous assert of RSTX and NMIX external pins
4. Hardware watchdog reset

* This register group is not initialized by an INIT-level or RST-level reset (except for STBCR).

9.4.1 STBCR: (STandby mode Control Register) Standby Control Register

This register configures low-power consumption modes.

(Note) Write operations to this register by DMA are disabled.

- **STBCR: Address 0482_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
STOP	TIMER	SLEEP	Reserved	Reserved		SLVL[1:0]		
0	0	0	0	0	0	1	1	Initial value
R,W	R,W	R,W	R0,W0	R0,W0	R0,W0	R/W	R/W	Attribute

[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 enabled. The bit transits to each mode by reading STBCR after writing the values shown below to this bit.

STOP	TIMER	SLEEP	Transition to each standby mode enable
0	0	0	Does not transit (initial value).
0	0	1	Transit to the sleep mode by reading STBCR
0	1	X	Transits to the watch mode by reading STBCR
1	X	X	Transits to the stop mode by reading STBCR

The read values of each bit are as follows regardless of what you have written.

STOP	TIMER	SLEEP	Transition to each standby mode enable
0	0	0	Does not transit.
0	0	1	Transit to sleep mode.
0	1	0	Transits to watch mode.
1	0	0	Transits to stop mode.

Generation of a wake-up factor from each low-power consumption mode clears this register to an initial value.

[bit4 to bit2] Reserved

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

[bit1, bit0] SLVL1, SLVL0 (Standby LeVeL) Standby level setting

These bits control the operations in standby mode and sleep mode as follows.

Mode	SLVL1, SLVL0	Operation controls
Stop mode	0x	Pins are not used for high impedance.
	1x	Pins are used for high impedance.
Watch mode	0x	Pins are not used for high impedance.
	1x	Pins are used for high impedance.
Sleep mode	0x	CPU sleep mode (stop CPU only)
	1x	Bus sleep mode (stops CPU and on-chip buses)*

* On-chip bus will run when DMA transfer is in progress.
For information on pins with high impedance, see "APPENDIX D. Pin Statuses in CPU Status".

9.4.2 PMU control register: PMUCTLR (Power Management Unit ConTroL register)

This register is used to control the PMU.

- **PMUCTLR: Address 0591_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
SHDE	Reserved	IOCTMD	IOCT	Reserved				
0	0	0	0	0	0	0	0	Initial value
R/W	R0,W0	R/W	R/W	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

This register is initialized by power-on reset, internal low-voltage reset, reset by RSTX/NMIX simultaneous assert, and hardware watchdog timer reset.

[bit7] SHDE (SHut Down Enable)

This flag is used to set whether to establish the power shutdown mode or not when the mode of the CPU transits to the standby (watch/stop) mode.

SHDE	SHDE mode enable
0	No power shutdown in process at transition to the standby mode
1	Power shutdown in process at transition to the standby mode

[bit6] Reserved

The read value is always "0". Always write "0" to this bit.

[bit5] IOCTMD (I/O Clear Timing MoDe)

This flag is used to select timing for I/O status retention during recovery from the standby (power shutdown) mode (hardware processing).

IOCTMD	I/O retention release request mode
0	Retains the I/O status until recovery from standby (watch, stop) mode.
1	Retains the I/O status until the ICOT register is cleared.

[bit4] IOCT (I/O Clear Timing)

This flag is used to release I/O status retention during recovery from power shutdown at IOCTMD=1 (software processing).

IOCT	I/O retention release request
0	No request
1	Requesting

This bit is automatically cleared to "0" after "1" is written and I/O retention is released by an I/O retention release request. Furthermore, writing except during I/O retention is invalid. In this bit, writing "0" has no effect on operation.

[bit3 to bit0] Reserved

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

9.4.3 Power on Timing control register: PWRTMCTL (PoWeR on TiMing ConTroL register)

This register is used to control timing for power-on.

- **PWRTMCTL: Address 0592_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved					PTC			
0	0	0	0	0	0	1	1	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	Attribute

This register is initialized by power-on reset, internal low-voltage reset, reset by RSTX/NMIX simultaneous assert, and hardware watchdog timer reset.

[bit7 to bit3] Reserved

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

[bit2 to bit0] PTC (Power on Timing Cycle setting)

This flag is used to set start-up time of the PSW.

PTC[2:0]	Start-up time	Remarks (for 32 kHz)
000	1CLK	30 µs
001	3CLK	90 µs
010	5CLK	150 µs
011	9CLK	270 µs
100	-	-
101	2CLK	60 µs
110	4CLK	120 µs
111	7CLK	210 µs

9.4.4 PMU interrupt flag register1: PMUINTF1(Power Management Unit INTerrupt Flag1 register)

This register is used to indicate that an interrupt request was triggered by an external input at power shutdown.

- **PMUINTF1: Address 0595_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
EIF7	EIF6	EIF5	EIF4	EIF3	EIF2	EIF1	EIF0	
0	0	0	0	0	0	0	0	Initial value
R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	Attribute

This register is initialized by power-on reset, internal low-voltage reset, reset by RSTX/NMIX simultaneous assert, and hardware watchdog timer reset.

[bit7 to bit0] EIF7 to EIF0 (External Interrupt Flag7-0)

This flag is used to indicate that an interrupt request was triggered by an external input at power shutdown.

EIFxx*	External interrupt request
0	No request
1	Request

*: xx is supplied with any number from 7 to 0.

This register is valid only at power shutdown.

This register is cleared by writing "0". Writing "1" has no effect on operation.

(Note) On the MB91F583AS/F584AS/F585AS , EIF7 is a reserved bit. So the bit should be written "0".

9.4.5 PMU interrupt flag register 2: PMUINTF2 (Power Management Unit INTerrupt Flag2 register)

This register is used to indicate that an interrupt request was triggered by an external input at shutdown.

- **PMUINTF2: Address 0596_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	NIF	MTIF	Reserved	Reserved				
0	0	0	0	0	0	0	0	Initial value
R0(RM1) ,W0	R(RM1), W	R(RM1), W	R0(RM1) ,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

This register is initialized by power-on reset, internal low-voltage reset, reset by RSTX/NMIX simultaneous assert, and hardware watchdog timer reset.

[bit7] Reserved

The read value is always "0". Always write "0" to this bit.

[bit6] NIF (Nml Flag):

This flag is used to indicate that an interrupt request was triggered by an NMI at power shutdown.

NIF	NMI interrupt request
0	No request
1	Request

This bit is valid only at power shutdown.

This bit is cleared by writing "0". Writing "1" has no effect on operation.

[bit5] MTIF (Main Timer Interrupt Flag)

This flag is used to indicate that an interrupt request was triggered by the main timer at power shutdown.

MTIF	Main timer interrupt request
0	No request
1	Request

This bit is valid only at power shutdown.

This bit is cleared by writing "0". Writing "1" has no effect on operation.

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

[bit4 to bit0] Reserved

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

9.5 Operation

This section explains the power consumption control features of this product.

9.5.1 Clock Control

This product can perform optimization of power consumption and processing ability by adjusting each operating clock.

■ Division Setting

See "CHAPTER 6 CLOCK".

9.5.2 List of Clock Supply in Low-power Consumption Mode

Table 9.5-1 List of Clock Supply in Low-power Consumption Mode

Clock	Standby		Sleep	
	Stop	Watch	Bus	CPU
CPU clock (CCLK)	❍	❍	❍	×
CAN prescaler clock	❍	❍	*1	×
On-chip bus clock (HCLK)	❍	❍	❍	×
Peripheral clock (PCLK)	❍	❍	×	×
PLL clock (PLLCLK)	❍	❍	×	×
Main clock (MCLK)	❍	×	×	×
CR oscillation	*2	*2	×	×

❍: Stops.

×: Does not stop. (If the main/PLL let them stop in each clock setting register, follow those.)

*1: Stops if on-chip bus clock (HCLK) is selected for the CAN prescaler clock. Follow the PLL outputs if PLL output is selected. Otherwise, the clock does not stop.

*2: The operation differs depending on the model.

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH

MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH: × (Does not stop)

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK

MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK: ❍ (Stops)

9.5.3 Sleep Mode

Sleep mode is the mode in which CPU and on-chip bus stop and only the peripherals run. In sleep mode, there are several modes for different stop ranges:

- CPU sleep mode : Stops CPU only.
- Bus sleep mode : Stops CPU and on-chip bus.

The mode continues the stop state until a wake up request occurs. A return to the program operation within several clock times is possible by generating a wake up request.

Following are the explanation of operation of different modes.

■ CPU Sleep Mode

CPU sleep mode is the mode to stop CPU operation.

In this mode, DMA controller and on-chip bus can continue their operations, while more power will be consumed than that of bus sleep mode.

■ Bus Sleep Mode

Bus sleep mode is the mode to stop CPU and on-chip bus operations. In this mode, 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 on-chip bus clock (HCLK) again.

In this mode, you can decrease the amount of power consumption more than that of CPU sleep mode, but the response time to the DMA transfer request will be somewhat degraded.

■ Configuration of Sleep Mode

Before activating sleep mode, select the level of sleep mode with the values set to the bit1:SLVL[1] in STBCR register.

- When setting "bit1:SLVL[1]= 0" in the STBCR register, goes into CPU sleep mode.
- When setting "bit1:SLVL[1]= 1" in the STBCR register, goes into bus sleep mode.

■ Activation of Sleep Mode

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. Therefore, 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)
MOV     R0, R0               ; Dummy processing for pipeline adjustment
NOP                          ; Dummy processing for pipeline adjustment
```

■ Waking Up from the Sleep Mode

Sleep mode exits on the following conditions.

- Reset
- Generation of interrupt request whose value of corresponding ICR register (See "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".) is a value other than "0x1F"
- Generation of NMI request
- Generation of tool break while connected to ICE

For wake up from interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from the next instruction which activates sleep mode.

For bus sleep mode, resume on-chip bus clock (HCLK) supply temporarily by the DMA transfer request and perform DMA transfer. After the DMA transfer, stop on-chip bus clock (HCLK) again.

■ Effect of Sleep Mode

You can reduce power consumption on the peripheral or external input event wait state drastically by using sleep mode. This mode does not decrease power consumption as much as that of in watch mode or stop mode because 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.

9.5.4 Standby Mode: Watch Mode

Watch mode is the mode to continue the main clock oscillation and counts the main timer.

(Note) Enter the standby mode only when main RUN is in progress. See Section "9.5.9 Transition to Illegal Standby Mode" for the operation of transition to standby mode from PLL run.

(Note) It is prohibited to transit to the standby status during FLASH program/erase.

■ Configuration of Watch Mode

Before activating watch mode, set the state of external pins in watch mode with the bit1:SLVL[1] in the STBCR register.

- When setting "bit1:SLVL[1]= 0" in the STBCR register, the external pins hold previous state.
- When setting "bit1:SLVL[1]= 1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. For details, see "APPENDIX D. Pin Statuses in CPU Status".

■ Activation of Watch Mode

To activate watch mode, follow the steps below.

- Write "0" to bit7:SHDE in the PMUCTLR register.
- When performing PLL RUN, the mode goes into main RUN state.
- Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read 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. Therefore, 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
```

■ Waking Up from Watch Mode

Watch mode exits on the following conditions.

- Reset
- Generation of an interrupt request whose value of corresponding ICR register value (See "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)") is the values other than "0x1F".
- Generation of NMI request
- Generation of tool break while connected to ICE

For wake up from interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from the next instruction which activates watch mode.

■ Effect of Watch Mode

You can reduce power consumption on 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 within a period of time shorter than the time required to return from stop mode is possible by generating a wake up request.

9.5.5 Standby Mode: Watch Mode (Power Shutdown)

Watch mode (power shutdown) is the mode to continue the main clock oscillation after the power of the microcontroller unit is shut down and count the main timer.

(Note) Enter the standby mode only when main RUN is in progress. See Section "9.5.9 Transition to Illegal Standby Mode" for the operation of transition to standby mode from PLL run.

(Notes) It is prohibited to transit to the standby status during FLASH program/erase.

■ Configuration of Watch Mode (Power Shutdown)

Before activating watch mode, set the state of external pins in watch mode with the bit1:SLVL[1] in the STBCR register.

- When setting "bit1:SLVL[1]= 0" in the STBCR register, the external pins hold previous state.
- When setting "bit1:SLVL[1]= 1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. For details, see "APPENDIX D. Pin Statuses in CPU Status".

■ Activation of Watch Mode (Power Shutdown)

To activate watch mode (power shutdown), follow the steps below.

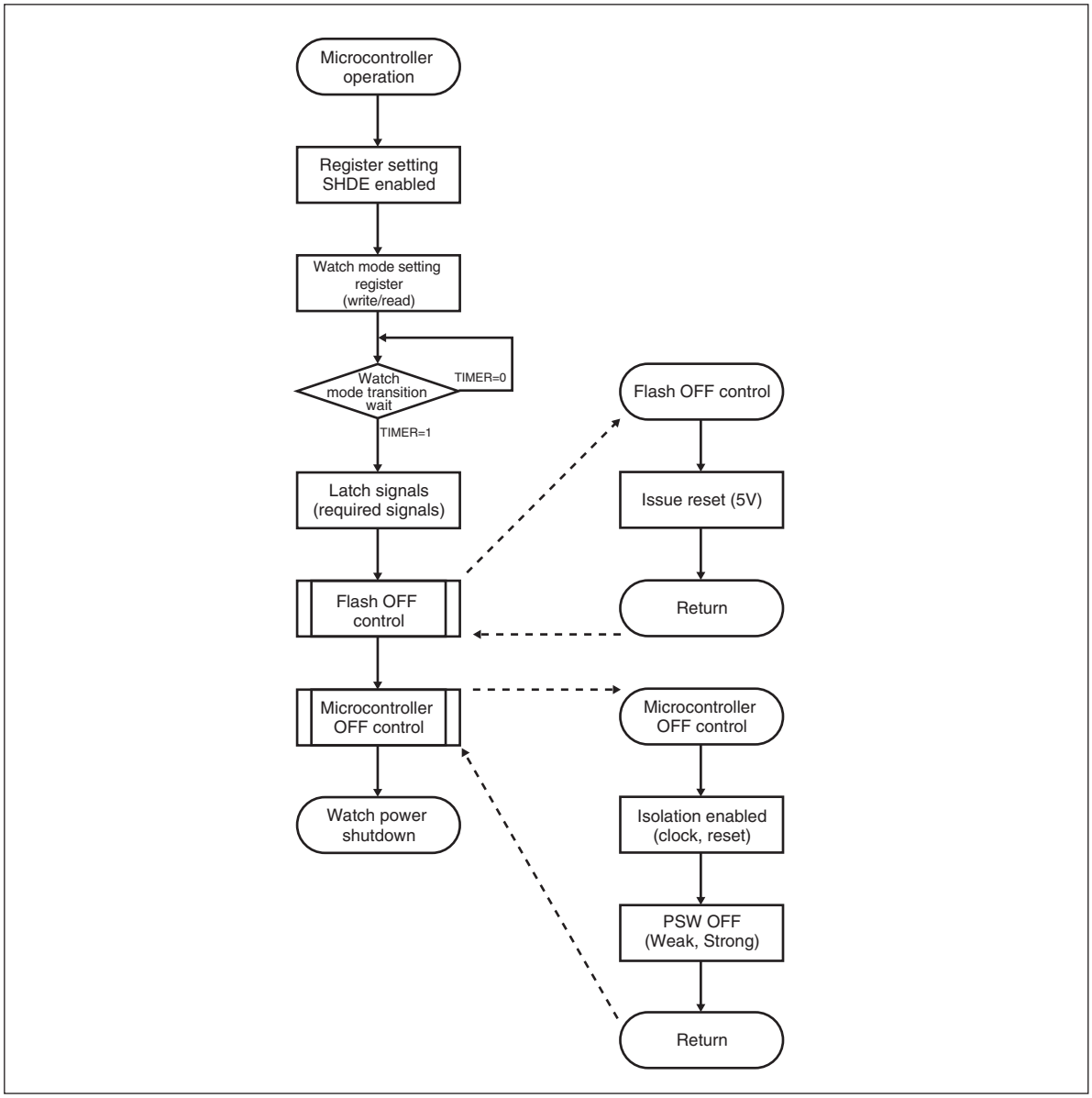
- Write "1" to bit7:SHDE in the PMUCTLR register.
- When performing PLL RUN, the mode goes into main RUN state.
- Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Therefore, perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering watch mode (power shutdown).

[Example] Sample program of watch mode (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_timer, R0 ; TIMER bit="1", SLVL setting
LDI     #_STBCR, R12        ;
STB     R0, @R12            ; Write
LDUB    @R12, R0            ; Read (activation of watch mode (power shutdown))
MOV     R0, R0              ; Dummy processing for pipeline adjustment
NOP                         ; Dummy processing for pipeline adjustment
```

Figure 9.5-1 Transition Sequence to Watch (Power Shutdown) Mode



■ Waking Up from Watch Mode (Power Shutdown)

Watch mode (power shutdown) exits on the following conditions.

- Reset
- Generation of an external interrupt request
- Generation of NMI request
- Generation of a main timer interrupt request

For wake up from interrupt request, the CPU and the interrupt controller do not have to accept this interrupt request. The CPU always starts operation from a reset state.

External interrupt input (IOCTMD=1) register is not initialized.

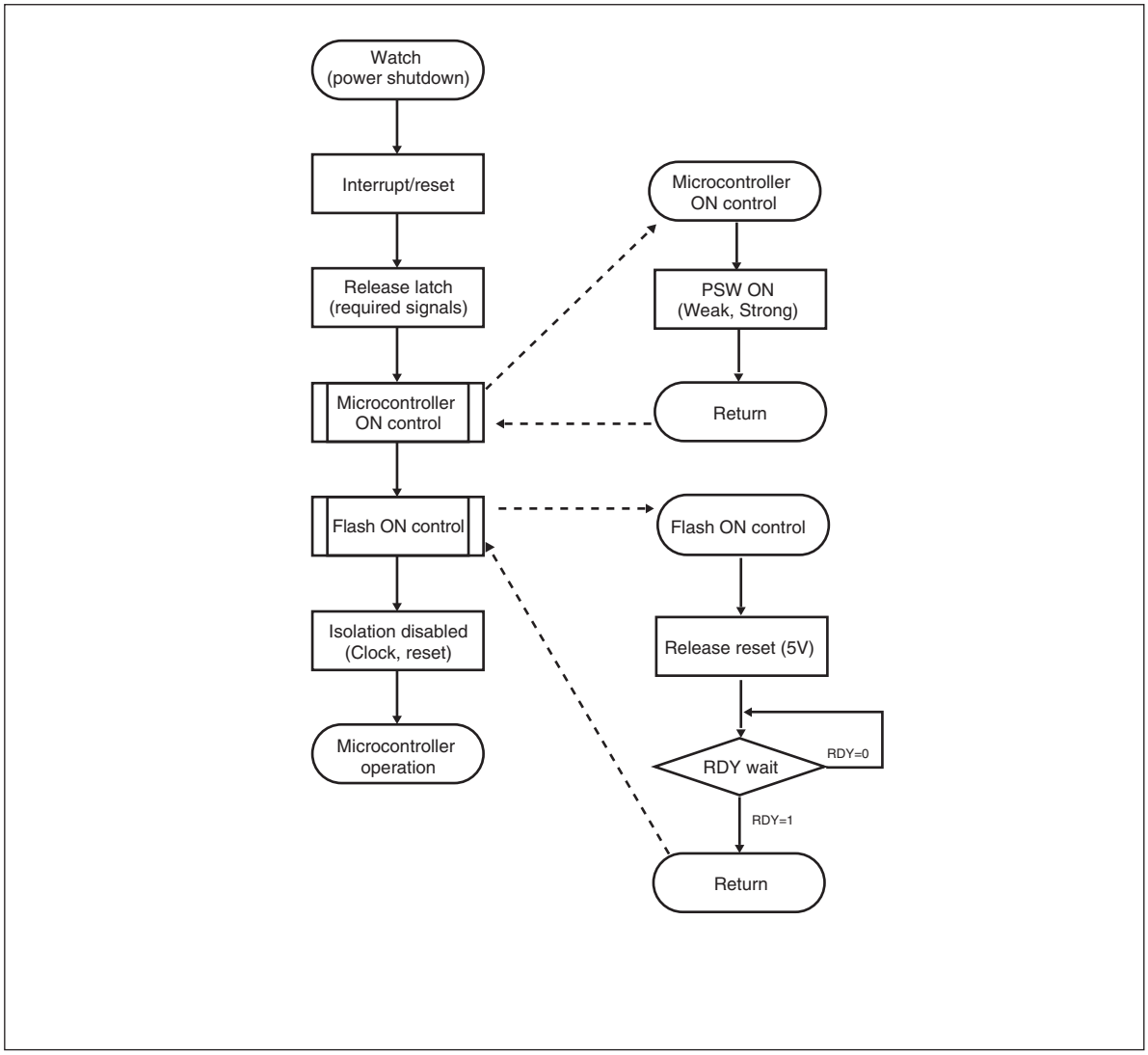
In addition, during wakeup, reset factors other than power-on reset, internal low-voltage reset, and reset by RSTX/NMIX simultaneous assert are not accepted.

At this time, the external interrupt input (IOCTMD=1) register is not initialized. Therefore, if a reset input from RSTX pin input or internal or external low-voltage detection flag is set after start-up, use it after initializing the external interrupt input register.

After recovering from power shutdown reset, the I/O latch may not be released.

Make sure to release the I/O retention (IOCT=1) with the PMUCTLR.IOCT bit by using software after recovering from power shutdown reset.

Figure 9.5-2 Recovery Sequence from Watch (Power Shutdown) Mode



■ Effect of Watch Mode (Power Shutdown)

You can reduce standby current of unwanted circuits drastically by using watch mode (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 waiting for clock oscillation to be stabilized is possible by generating a wake up request.

9.5.6 Standby Mode: Stop Mode

Standby mode is the mode to stop all clock oscillations and minimize the power consumption of this product .

(Note) Enter the standby mode only when main RUN is in progress. See Section "9.5.9 Transition to Illegal Standby Mode" for the operation of transition to standby mode from PLL run.

■ Configuration of Stop Mode

Before activating stop mode, set the external pins states in stop mode with the bit1:SLVL[1] in the STBCR register.

- When setting "bit1:SLVL[1]= 0" in the STBCR register, the external pins hold previous state.
- When setting "bit1:SLVL[1]= 1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. For details, see "APPENDIX D. Pin Statuses in CPU Status".

■ Activation of Stop Mode

To activate stop mode, follow the steps below.

- Write "0" to bit7:SHDE in the PMUCTLR register.
- When performing PLL RUN, the mode goes into main RUN state.
- Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read 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. Therefore, 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
```

■ Waking Up from Stop Mode

Stop mode exits on the following conditions.

- Reset
- Generation of an interrupt request whose value of corresponding ICR register value (See "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)") is the values other than "0x1F".
- Generation of NMI request
- Generation of tool break while connected to ICE

For wake up from interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from the next instruction which activates stop mode.

■ Effect of Stop Mode

You can minimize the power consumption in external input event wait state by using stop mode. On the other hand, this mode needs oscillation stabilization wait time which is the interval from a generation of wake up request to a return of program operation.

9.5.7 Standby Mode: Stop Mode (Power Shutdown)

Stop mode (power shutdown) is the mode to stop all clock oscillations and minimize the power consumption of this product.

(Notes) Enter the standby mode only when main RUN is in progress. See Section "9.5.9 Transition to Illegal Standby Mode" for the operation of transition to standby mode from PLL run.

(Notes) It is prohibited to transit to the standby status during FLASH program/erase.

■ Configuration of Stop Mode (Power Shutdown)

Before activating stop mode (power shutdown), perform the following settings/controls.

(1) Set the external pins states in stop mode (power shutdown) with bit1:SLVL[1] in the STBCR register.

- When setting "bit1:SLVL[1]=0" in the STBCR register, the external pins hold previous state.
- When setting "bit1:SLVL[1]=1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. For details, see "APPENDIX D. Pin Statuses in CPU Status".

■ Activation of Stop Mode (Power Shutdown)

To activate stop mode (power shutdown), follow the steps below.

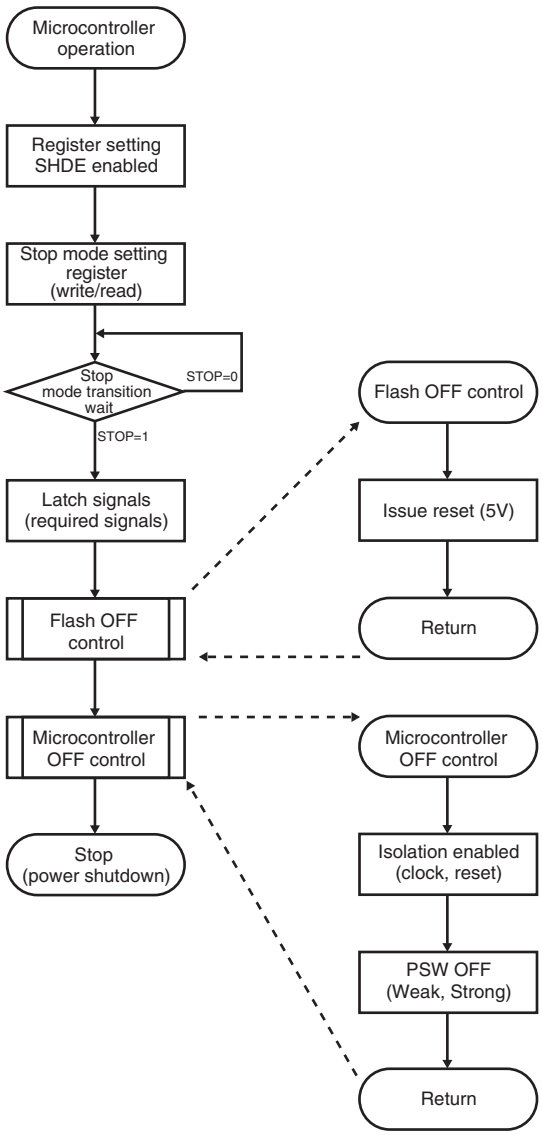
- Write "1" to bit7:SHDE in the PMUCTLR register.
- When performing PLL RUN, the mode goes into main RUN state.
- Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Therefore, 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 (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 (power shutdown))
MOV     R0, R0              ; Dummy processing for pipeline adjustment
NOP                         ; Dummy processing for pipeline adjustment
```

Figure 9.5-3 Transition Sequence to Stop (Power Shutdown) Mode



■ Waking Up from Stop Mode (Power Shutdown)

Stop mode (power shutdown) exits on the following conditions.

- Reset
- Generation of an external interrupt request
- Generation of NMI request

For wake up from interrupt request, the CPU and the interrupt controller do not have to accept this interrupt request. The CPU always starts operation from a reset state.

External interrupt input (IOCTMD=1) register is not initialized.

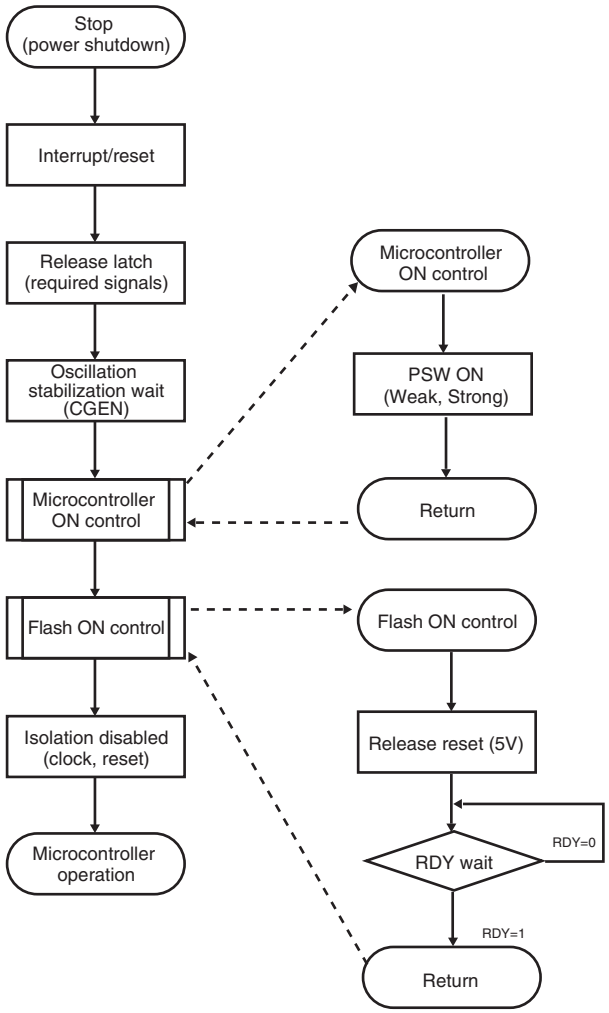
In addition, during wakeup, reset factors other than power-on reset, internal low-voltage reset, and reset by RSTX/NMIX simultaneous assert are not accepted.

At this time, the external interrupt input (IOCTMD=1) register is not initialized. Therefore, if a reset input from RSTX pin input or internal or external low-voltage detection flag is set after start-up, use it after initializing the external interrupt input register.

After recovering from power shutdown reset, the I/O latch may not be released.

Make sure to release the I/O retention (IOCT=1) with the PMUCTLR.IOCT bit by using software after recovering from power shutdown reset.

Figure 9.5-4 Transition Sequence from Stop (Power Shutdown) Mode



■ Effect of Stop Mode (Power Shutdown)

You can minimize standby current of unwanted circuits by using stop mode (power shutdown).

On the other hand, this mode needs oscillation stabilization wait time which is the interval from a generation of wake up request to a return of program operation.

9.5.8 Microcontroller Stop State

When an attempt is made to control the transition to the standby mode from the standby mode (watch mode/ watch mode (power shutdown)/stop mode/stop mode (power shutdown) transition disable state, the transition is not completed.

<Standby transition disable state>
(1) While connected to OCD
(2) While operating PLL

<Standby controls that are not performed during the microcontroller stop state>
(1) Flash memory power saving control
(2) Oscillation stop (in the stop mode/stop mode (power shutdown))

However, if the standby mode transition control is performed while operating PLL, an illegal standby mode transition is detected and oscillation stop operation is performed. For details on the illegal standby mode transition, see "9.5.9 Transition to Illegal Standby Mode".

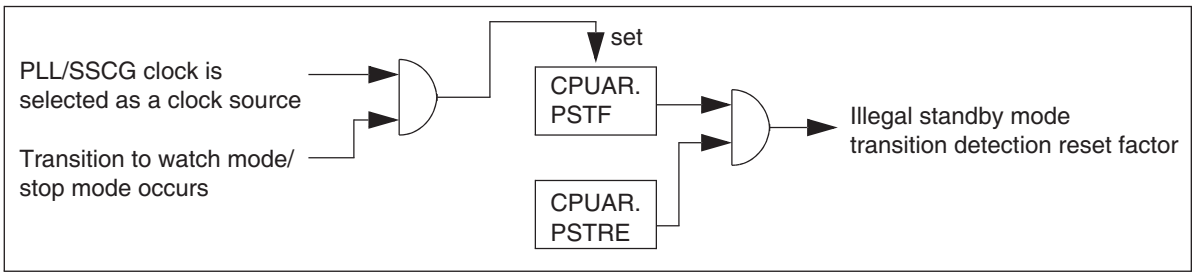
9.5.9 Transition to Illegal Standby Mode

When transiting from PLL run state to standby mode (watch mode or stop mode), standby mode will be set and PLL oscillation stabilization will be removed. (Transition to illegal standby mode)

After returning from the standby mode, CSELR:CKS[1:0]=00 and CMONR:CKM[1:0]=00 (division of main clock by two) will be set.

PSTF flag in CPUAR register will also be set on the transition to standby mode. When the PSTRE bit in CPUAR register is set, a reset will be generated by the illegal standby mode transition detection reset factor. See Section "8.4.3 CPUAR (CPU Abnormal operation Register): CPU abnormal operation register" in "CHAPTER 8 RESET" for CPUAR register.

Figure 9.5-5 Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor



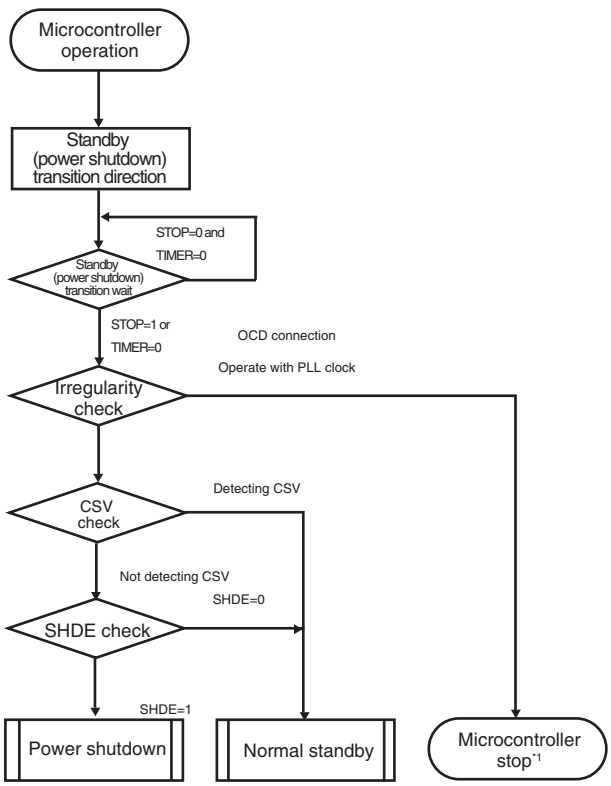
9.5.10 Restrictions on Power Shutdown/Normal Standby Control

This microcontroller does not perform standby control under the following conditions:

- The CPU is operating in PLL
- OCD operation is enabled
- The CSV function detects clock loss *2, *3

Although standby control does not operate under the above states, the CPU enters standby state.

Figure 9.5-6 Restrictions on Power Shutdown/Normal Standby Control



*1 This is not recognized as power shutdown, and the CPU is in a state of transition to standby mode.
*2 This is the state in which an RC clock is operating because loss was detected when a main clock was operating.
*3 If standby (power shutdown) transition is directed after clock loss is detected, normal standby processing applies.

Because the power is not supplied to most of the blocks in the chip in the standby (power shutdown) mode, only some registers are retained when recovering from the standby mode. Table 9.5-2 lists the registers that are retained when recovering from the standby (power shutdown).

Table 9.5-2 Registers Retained When Recovering from Standby Mode (Power Shutdown)

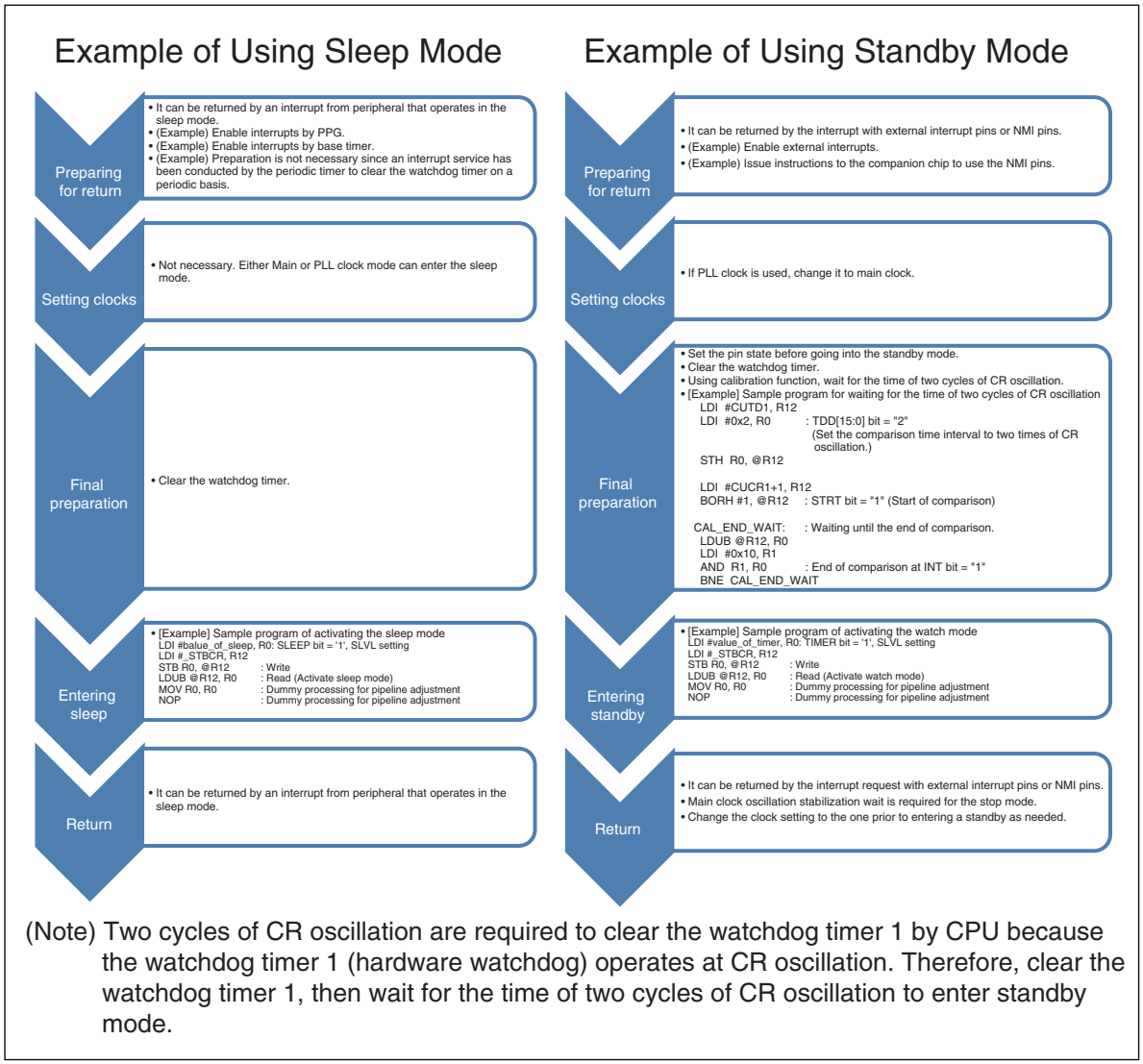
Register group	Register/Flag name	Type	Address	Remarks
PMU register	PMUSTR.PMUST	Flag	0590H bit7	
	PMUSTR.PONR_F	Flag	0590H bit1	
	PMUSTR.RSTX_F	Flag	0590H bit0	
	PMUCTLR	Register	0591H	
	PWRTMCTL	Register	0592H	
	PMUINTF1	Flag	0595H	
	PMUINTF2	Flag	0596H	
Reset factor register	CPUAR.PDLF	Flag	051AH bit3	
	CPUAR.PMDF	Flag	051AH bit2	
	CPUAR.PSTF	Flag	051AH bit1	
	CPUAR.HWDF	Flag	051AH bit0	
	LVD5R.LVD5R_F	Flag	0584H bit0	
	LVD5F.LVD5F_F	Flag	0585H bit0	
	LVD.LVD_F	Flag	0586H bit0	
Low-voltage detection register	LVD.LVD_PD	Register	0586H bit7	
	LVD.LVD_OE	Register	0586H bit3	
	LVD5F.LVD5F_PD	Register	0585H bit7	
	LVD5F.LVD5F_OE	Register	0585H bit3	
CSV register	CSVCR	Register	056DH	
External interrupt register	EIRR0	Register	0550H	*1
	ENIR0	Register	0551H	*1
	ELVR0	Register	0552H	*1

*1 This register is initialized when PMUCTLR.IOCTMD is 0.

9.6 Usage Example

The following is the examples of using in sleep mode and standby mode.

Figure 9.6-1 Usage Examples of using in Sleep Mode and Standby Mode



(Note) Two cycles of CR oscillation are required to clear the watchdog timer 1 by CPU because the watchdog timer 1 (hardware watchdog) operates at CR oscillation. Therefore, clear the watchdog timer 1, then wait for the time of two cycles of CR oscillation to enter standby mode.

CHAPTER 10 CLOCK RESET STATE TRANSITIONS

10.1 Overview

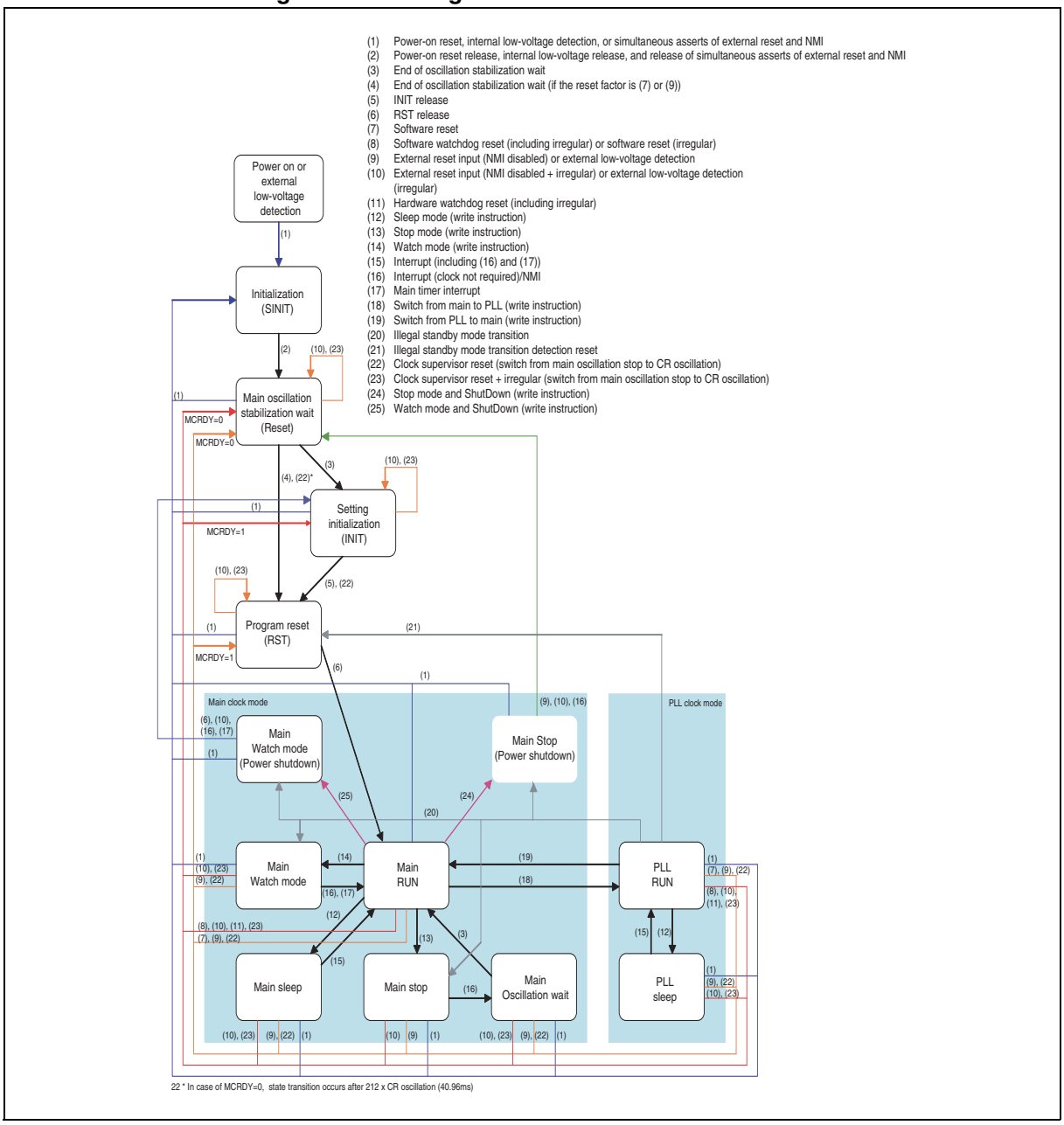
This chapter explains state transitions for clocks and resets. See "CHAPTER 9 POWER CONSUMPTION CONTROL" for state features and configuration of power consumption control. For operation of resets, see "CHAPTER 8 RESET".

10.2 Device States and Transitions

10.2.1 Diagram of State Transitions

The device state transitions for this product is shown below.

Figure 10.2-1 Diagram of Device State Transitions



Code:CRST-1v0-91580MS-1-E

(Note) There are some registers that are not reset when recovering from the watch mode (power shutdown) and the stop mode (power shutdown). For details, see Section "9.5.10 Restrictions on Power Shutdown/Normal Standby Control" in "CHAPTER 9 POWER CONSUMPTION CONTROL".

(Note) While connected to the OCD tool, transition may differ from the above diagram. Refer to "CHAPTER 49 ON-CHIP DEBUGGER (OCD)".

10.2.2 Explanation of States

Device states for this product 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 external pins in stop state and watch mode state will be deleted.

■ Sleep Mode

The program is not running. The state transits by program operations. There are some settings; one to stop program execution of CPU only (CPU sleep mode) and the other to stop CPU, on-chip bus (on-chip bus) and on-chip bus clock (HCLK) driven peripheral (bus sleep mode). See "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.

■ Watch Mode State

The devices are not running. The state transits by program operations. Internal circuits other than oscillation circuits (a main clock generation unit) stop. Stop PLL oscillation before going into 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 RUN state by some specific (no clock required) effective interrupts, main timer interrupts and watch counter interrupts. See "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.

■ Watch mode (power-shutdown) state

This is device-stop state which unnecessary section of the watch mode is shut off its power supply. Transition is controlled by program operations. As internal circuit power is shut off, internal circuit other than oscillation circuit (main clock generating section) is stopped. Prior to shifting to the watch mode (power shut-off), stop the PLL oscillation.

Also, external pins can uniformly be configured to high impedance (except for a handful of pins).

Use of certain valid interrupts (that do not require clock presence) and main timer interrupt, shifting to the setting initialization (INIT) state can be done. Refer to "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.

■ Stop State

The devices are not running. The state transits by program operations. All internal circuits will stop. Stop PLL oscillation before going into stop 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 interrupts. See "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.

■ Stop (power shut-off) state

This is device-stop state which unnecessary section of the stop mode is shut off its power supply. Transition is controlled by program operations. As internal circuit power is shut off, all circuit is stopped. Prior to shifting to the stop mode (power shut-off), stop the PLL oscillation.

Also, external pins can uniformly be configured to high impedance (except for a handful of pins).

By NMI interrupt, transition to the main oscillation stabilization wait (reset) state is initiated. Refer to "CHAPTER 9 POWER CONSUMPTION CONTROL" for details.

■ Main Oscillation Stabilization Wait (RUN) State

The devices are not running. Transits after returning from the stop state. All the internal circuits except for the timer operations for oscillation stabilization wait will stop. All internal clocks stop but the enabled oscillation circuits will still be running. After the elapse of the oscillation stabilization wait time interval set, transits to 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 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 state (INIT). Outputs a 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 RUN state (normal operation) after removing the operation initialization reset (RST) request. See "CHAPTER 8 RESET" for details.

■ Setting Initialization (INIT) State

All settings are initialized. Transits after accepting a setting initialization (INIT) request. The main oscillation circuit continues to run but 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 removed. See "CHAPTER 8 RESET" for details.

10.2.3 Priority of State Transition Requests

State transition requests will be processed with the priority below, no matter what states they are in. However, some requests will be generated in some specific situations. So those requests would only be effective in those situations.

The 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)

The lowest priority, sleep mode request (register write) (generates RUN state only)

10.3 Device state and corresponding regulator mode

The regulator mode corresponding to each device state is shown in Table 10.3-1. To learn more about regulator mode, refer to "CHAPTER 12 REGULATOR CONTROL".

Table 10.3-1 Relationship between device state and regulator mode

Device state	Main clock	Regulator mode
Main RUN	Active	Main mode
Main sleep	Active	Main mode
Main watch mode	Active	Standby mode
Main watch mode (Power shutdown)	Active	Standby mode
Main stop	Inactive	Standby mode
Main stop (Power shutdown)	Inactive	Standby mode
Main oscillation wait	Active	Main mode
PLL RUN	Active	Main mode
PLL sleep	Active	Main mode

(Note) While OCD tool is connected, regulator mode becomes the main mode, as seen in the tables above.

CHAPTER 11 CLOCK SUPERVISOR

11.1 Overview

The clock supervisor is a function that monitors the main clock oscillation.

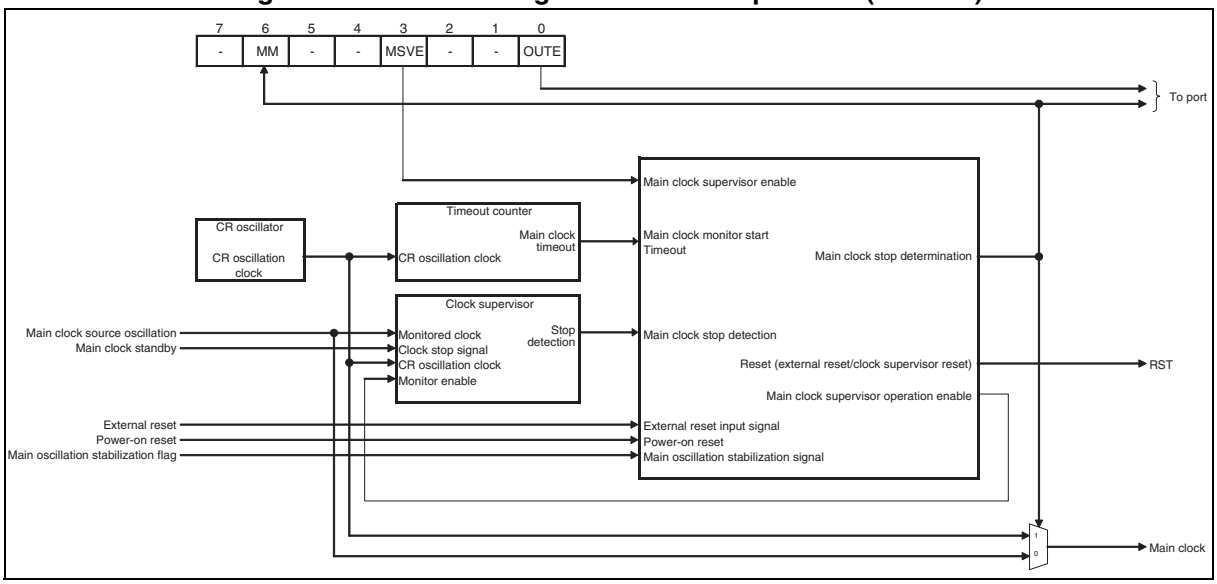
If some kind of problem occurs in the clock and it stops unintentionally, the built-in CR oscillator can substitute for the clock.

11.2 Configuration

The blocks that make up the clock supervisor are shown below.

- Clock supervisor
- Timeout counter
- Control logic
- CR oscillator

Figure 11.2-1 Block Diagram of Clock Supervisor (detailed)



* External reset: On assert of RSTX pin (This includes simultaneous asserts of RSTX and NMIX.)

(Note) On models that have a clock monitor, the clock can be output from a port.
The MM bit can be output from a port (P046) by setting CSVCR.OUTE=1.

Code:FJ58-8v0-91580MS-1-E

11.3 Registers

■ Register map

Table 11.3-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x056C	Reserved	CSVCR	Reserved	Reserved	Clock supervisor control register

11.3.1 Clock Supervisor Control Register (CSVCR)

The clock supervisor control register (CSVCR) is used to enable and check the status of various functions.

- **CSVCR: Address 056D$_H$ (Access: Byte)**

7	6	5	4	3	2	1	0	bit
Reserved	MM	Reserved	Reserved	MSVE	Reserved	Reserved	OUTE	
0	0	0	1	1	1	0	0	Initial values
R/W0	R,W	R,W0	R/W1	R/W	R/W1	R/W0	R/W	Attributes

[bit7] Reserved.

"0" should be written to this bit.

[bit6] MM (Main clock Missing)

This bit will be set if any problem is found in the main oscillation clock.

When this bit is "0", there are no problems in the main clock.

While the main clock is not recovered, "0" writing is ignored.

This bit is cleared to "0" by power-on reset or external reset. Other types of resets have no effect on this bit.

MM	Read	Write
0	Main oscillation clock missing undetected	While the main clock is recovered, this bit is cleared.
1	Main oscillation clock missing detected	No effect

(Note) Do not enable the PLL/SSCG oscillation operation when this bit is "1".

[bit5] Reserved.

"0" should be written to this bit.

[bit4] Reserved.

"1" should be written to this bit.

[bit3] MSVE (Main clock SuperVisor Enable)

When this bit is set to "1", the main clock supervisor is enabled.

This bit is initialized to "1" at power-on reset.

Other types of resets have no effect on this bit.

MSVE	Description
0	Main clock supervisor disabled
1	Main clock supervisor enabled (Initial value)

[bit2] Reserved.

"1" should be written to this bit.

[bit1] Reserved.

"0" should be written to this bit.

[bIT0] OUTE (Output Enable)

This bit enables the output of the MM bit from a port.
When this bit is set to "1", the MM bit output is enabled.
This bit is cleared to "0" by power-on reset or external reset.
Other types of resets have no effect on this bit.

OUTE	Description
0	MM output from the port is disabled (initial value)
1	MM output from the port is enabled

11.4 Operation Description

This section explains clock supervisor operations.

If failure occurs in the main clock while the main clock is operating, clock can be replaced by on-chip CR oscillation clock according to the configuration.

The clock is determined that failure occurs if there is no clock input for a period of 36 to 44 μs and MCU is reset.

The state of the main oscillation clock missing detect bit is retained after reset allowing the software to determine that a problem occurred.

If the main clock was stopped intentionally, the main clock supervisor automatically stops.

The main clock stopped bit in the clock supervisor control register can be output from a port.

After the main clock has stopped, the bit indicating the stop can be cleared by software.

If the main oscillation has been restored when this bit is cleared, operation using the main oscillation starts, and if it has not been restored, operation using the CR oscillator starts.

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the standby mode, CR oscillator continues to be oscillated.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
While in transition to the standby mode, CR oscillator automatically stops. Upon recovering from the standby mode, CR oscillator is restarted automatically.

(Note) Please do not enable the PLL/SSCG oscillation operation when the main clock is replaced the CR oscillator and it operates after detecting the main clock stop.

11.4.1 Initial State

In the initial settings, the oscillation of the CR oscillator and main clock supervisor function are enabled.

■ CR oscillator

The oscillation is enabled when the power is turned on.

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the standby mode, CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
This is stopped only when shifted to the standby mode. When the standby mode is released, oscillation is automatically resumed.

■ Main clock supervisor

Main clock supervisor is enabled after the main oscillation stabilization wait time has elapsed.

When the main clock supervisor is enabled, if the main clock stops, the main clock is replaced by the CR oscillation clock.

Moreover, the MM bit of the CSVCR register is set to "1" and an RST level reset is generated.

[Note]

Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.

In this case, after the timeout time measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

[Note]

If the main clock is stopped at time of power ON, main supervisor function is enabled after timeout time measured by internal RC oscillator regardless of oscillation stabilization standby time, and the stopping of main clock is detected.

11.4.2 Stopping CR Oscillator and the Clock Supervisor Function

■ CR oscillator

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH

MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:

While in transition to the standby mode, CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK

MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:

CR oscillator is stopped during standby mode. While main clock has issues, shifting to the standby mode is prohibited. You can check the CSVCR register's MM bit to see if the clock has issues.

[Note]

If clock has issues, operation clock stops as CR oscillation is stopped, since operation clock is already switched to the CR oscillation clock.

■ Main clock supervisor

Set the MSVE bit of the CSVCR register to "0".

11.4.3 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".

[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 time measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

11.4.4 Stop Mode

■ CR oscillator

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the stop mode, CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
By shifting to the stop mode, oscillation stops. After the stop mode is released, oscillation automatically resumes.

■ 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 time measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

(Note) If shifting to stop mode is initiated while main clock supervisor function is disallowed, it will remain disallowed even after the stop mode is released.

11.4.5 Stop mode (power shut-off)

■ CR oscillator

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the stop mode (power shut-off), CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
By shifting to the stop mode (power shut-off), oscillation stops. After the stop mode (power shut-off) is released, oscillation automatically resumes.

■ Main clock supervisor

If shifting to stop mode (power shut-off) is initiated while main clock supervisor function is allowed, it will stop automatically.
Main clock supervisor enable bit (CSVCR.MSVE) will not be set as "0".
After the stop mode (power shut-off) is released, oscillation automatically resumes after a moment of 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 time measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

(Note) If shifting to stop mode (power shut-off) is initiated while main clock supervisor function is disallowed, it will remain disallowed even after the stop mode (power shut-off) is released.

11.4.6 Watch Mode

■ CR oscillation

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the watch mode, the CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
By transiting to the watch mode, oscillation stops. After the watch mode is released, oscillation automatically resumes.

■ Main clock supervisor

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
A transition to the watch mode does not affect the main clock supervisor function.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
Since CR oscillation is stopped, even if the main oscillation is stopped, it is not detected.

11.4.7 Watch Mode (Power Shutdown)

■ CR oscillation

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
While in transition to the watch mode (power shutdown), CR oscillator continues to oscillate.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
By transiting to the watch mode (power shutdown), oscillation stops. After the watch mode (power shutdown) is released, oscillation automatically resumes.

■ Main clock supervisor

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH
MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH:
A transition to the watch mode (power shutdown) does not affect the main clock supervisor function.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK:
Since CR oscillation is stopped, even if the main oscillation is stopped, it is not detected.

11.4.8 Checking the Reset Source Using the Clock Supervisor

The method for checking whether or not the clock supervisor detected a clock problem and generated a reset is shown below.

First, read the RSTRR register (see "CHAPTER 8 RESET") to check the reset source.

If the ERST bit in the RSTRR register is "1", this indicates that any of reset input from RSTX external pin, illegal standby mode transition detection reset, external low-voltage detection, clock supervisor reset or simultaneous asserts of RSTX and NMIX external pins was occurred.

In this case, read the CSVCR register and check the MM bit.

The reset source can be checked as follows.

ERST	MM	Reset source
1	1	Clock supervisor reset
1	0	Reset input from the RSTX external pin or Illegal standby mode transition detection reset or External low-voltage detection or Simultaneous asserts of RSTX and NMIX external pins

(Note) Since the MM bit is not cleared by resets other than power-on reset and external reset, the RSTRR register (see "CHAPTER 8 RESET") must be read and resets as well as other reset factors need to be checked.

11.4.9 Switchback from CR Clock

■ Main clock supervisor

When the MPU detects that the MM bit was set after recovering from a reset, the MPU can interpret that the main clock is stopped and it is switched to the CR oscillation clock. At this time, if the recovery of the main clock can be checked, the clock can be switched back to the main clock by writing "0" to the MM bit. If the main clock is not recovered, then writing "0" to the MM bit has no effect. The MM bit continues holding "1". If the main clock is operating when "0" is written to the MM bit, the MM bit is cleared and the clock is switched back to the main clock via the synchronization stage. The MM bit can be polled until the main clock is recovered.

```
ldi #_csvcr,r1
clear_CSV_loop:
bandh #0b1011,@r1 ;; Clear MM
btsth #0b0100,@r1 ;; Check MM
bne clear_CSV_loop
```

CHAPTER 12 REGULATOR CONTROL

12.1 Overview

The operation of regulator which generates internal voltage is automatically changed according to the device state transition. It is automatically changed to the following two regulator modes.

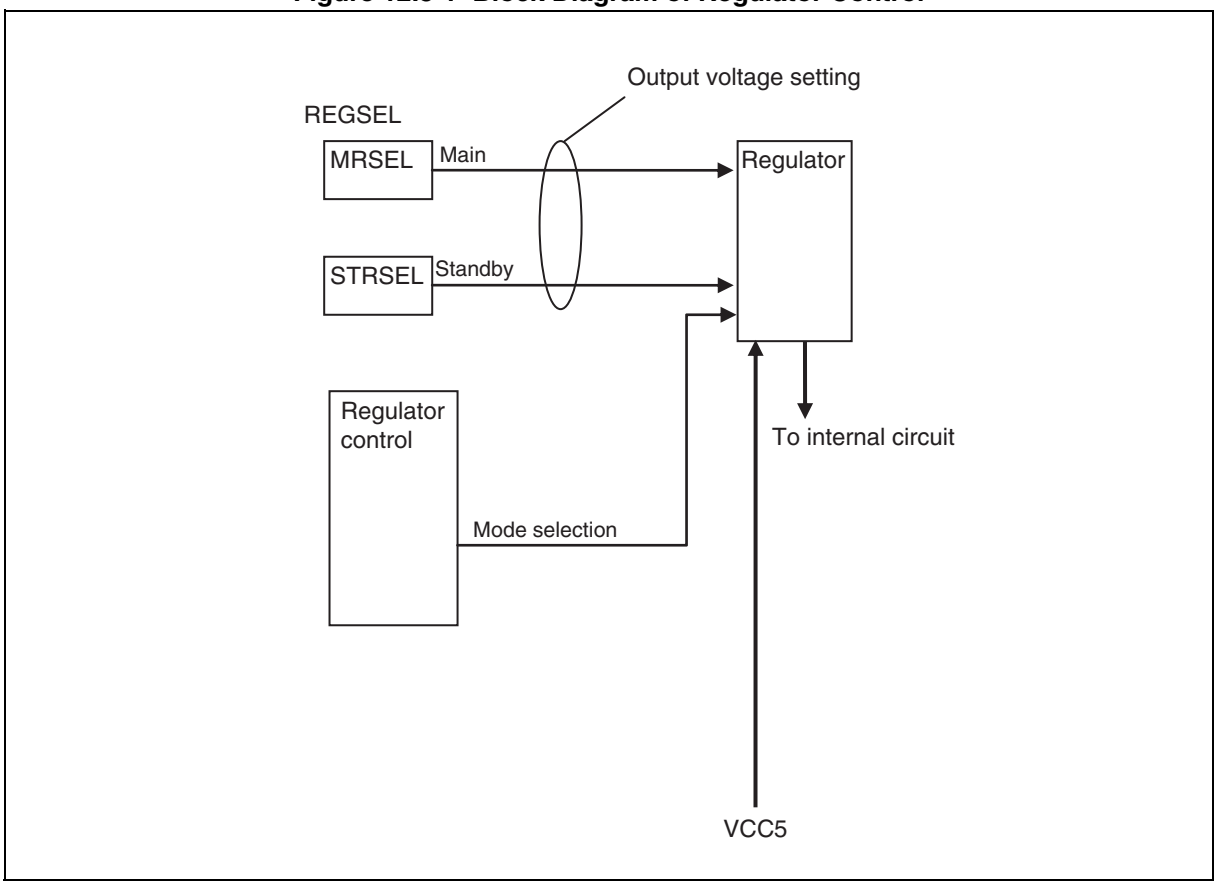
- Main mode (at normal operation)
- Standby mode (in stop and watch modes)

12.2 Features

The regulator mode can automatically be changed according to the device state transition.

12.3 Configuration

Figure 12.3-1 Block Diagram of Regulator Control



Code: ZBLVDR_REG-0v5-91580MS-1-E

12.4 Registers

Table 12.4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0580	REGSEL	Reserved	Reserved	Reserved	Regulator output voltage selection register

■ Regulator Output Voltage Select Register : REGSEL

This register selects the output voltage level of each regulator mode of main, and standby.

- **REGSEL: Address 0580_H (Access : Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
MRSEL[1:0]		Reserved		STRSEL[2:0]			-	
0	1	1	0	1	1	0	0	Initial value
R/W0	R/W1	R/W1	R/W0	R/W1	R/W1	R/W0	R0,WX	Attribute

[bit7, bit6] MRSEL[1:0] (Main Regulator voltage SELect)

These bits set the output voltage level of main regulator (regulator mode: main mode).

MRSEL[1:0]	Main regulator output voltage
00	Reserved
01	1.2 ± 0.1V
10	Reserved
11	Reserved

[bit5] Reserved

This bit must always be written to "1".

[bit4] Reserved

This bit must always be written to "0".

[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: Always set this setting value to 1.2V (STRSEL[2:0]=110) before using it.

[bit0] Undefined

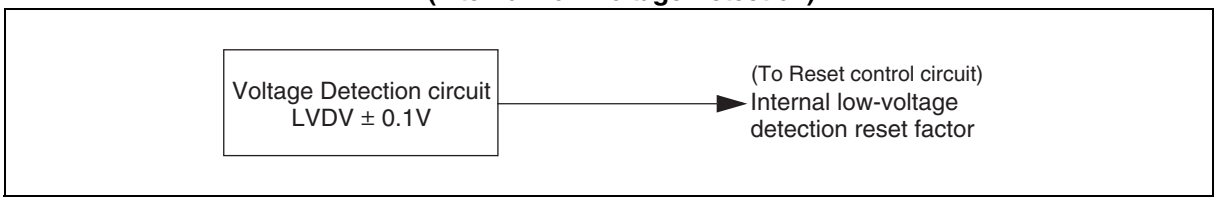
This bit always reads "0". Writing has no effect on operation.

CHAPTER 13 LOW-VOLTAGE DETECTION (INTERNAL LOW-VOLTAGE DETECTION)

13.1 Overview

The internal low-voltage detection is a function that monitors the internal power supply voltage and detects when the internal power supply voltage falls below the detection voltage value. It sets the detection flag if low-voltage is detected. When the detection flag is set, it is in the reset state by low-voltage detection reset.

Figure 13.1-1 Block Diagram of Low-Voltage Detection (Internal Low-voltage Detection)

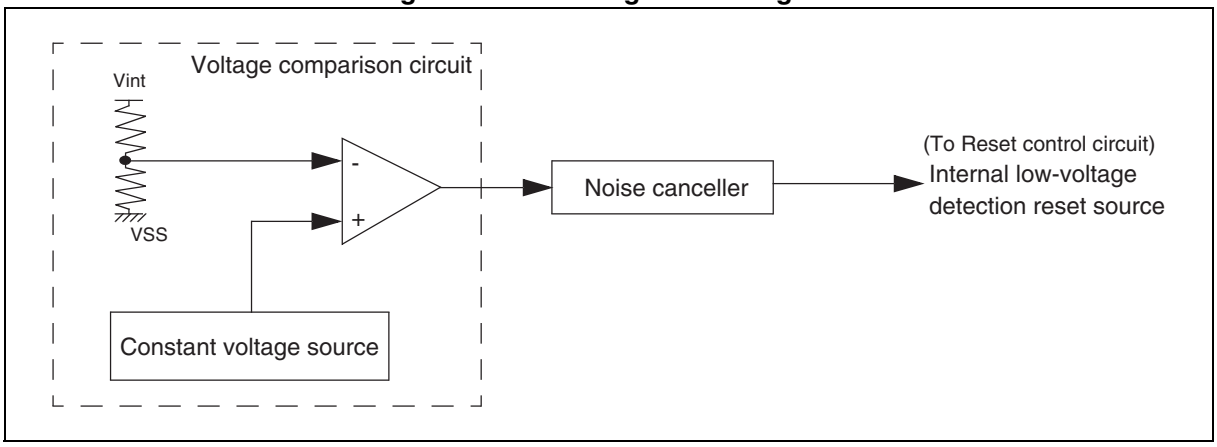


13.2 Features

- Method: Generates a settings initialization reset if a voltage lower than LVDV ± 0.1V is detected. (LVDV:0.9V)
- Number of units: 1
- Operation: Continues to operate in sleep mode, stop mode, and watch mode.
- Voltage comparison circuit: Compares the internal power supply voltage to the detection voltage, and changes the output from "H" to "L" if low-voltage is detected.

13.3 Configuration

Figure 13.3-1 Configuration diagram



Code: ZBLVDR_INT-0v5-91580MS-1-E

13.4 Registers

Table 13.4-1 Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0584	LVD5R*	LVD5F*	LVD	Reserved	Internal low-voltage detection register

*: Refer to "CHAPTER 14 LOW-VOLTAGE DETECTION (EXTERNAL LOW-VOLTAGE DETECTION)" for LVD5R, LVD5F.

13.4.1 Internal Low-Voltage Detection Register : LVD

This register has internal low-voltage detection flag (LVD_F).

- **LVD : Address 0586_H (Access : Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
LVD_PD	LVD_SEL[2:0]			LVD_OE	Reserved		LVD_F	
0	1	0	0	0	0	0	0	Initial value
R/W	R/W1	R/W0	R/W0	R/W	R0,WX	R0,WX	R(RM1), W	Attribute

[bit7] LVD_PD (Low-Voltage Detect fall Power Down)

This bit sets whether to detect the internal voltage fall.

LVD_PD	Internal power supply fall power down setting
0	Disabled (Detection is executed.)
1	Enabled (Detection is stopped.)

This bit is initialized only by power-on reset.

(Note) When power down is changed from enabled to disabled (operation start) in this bit setting, enable the detection (OE=0) after 100 μs elapse. If the detection is enabled before 100 μs elapse, a detection flag may be set.

[bit6 to bit4] LVD_SEL (Low-Voltage Detect internal power fall SELect)

This bit is a detection level of the internal voltage fall detection.

LVD_SEL[2:0]	Internal power supply fall detection voltage setting
100	0.9V ± 0.1V
Other than above	Setting prohibited

Note: These bits can be rewritten only when LVD_OE=1.

[bit3] LVD_OE (Low-Voltage Detect internal power fall Output Enable)

This bit is an output enable signal of the internal power supply voltage fall detection.

LVD_OE	Internal power supply fall detection output enable setting
0	Enable
1	Stop

This bit is initialized only by power-on reset.

[bit2, bit1] Reserved

[bit0] LVD_F (Low-Voltage Detect internal power fall Flag)

This bit reads an internal voltage fall detection state.

LVD_F	Internal power supply fall detection flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a drop in the internal voltage is detected, the LVD_F bit is set to "1".
It will be initialized only at the external reset input.

13.5 Operation Description

The internal low-voltage detection monitors the internal power supply voltage and detects when the internal power supply voltage falls below the detection voltage value and sets the detection flag. It generates a setting initialization reset if it detects low-voltage and sets the flag.

If the internal power supply voltage falls below the detection voltage, it takes the oscillation stabilization wait time after the internal low-voltage detection voltage is recovered. For details, see "CHAPTER 8 RESET".

Oscillation stabilization wait time	2^{15}× Main clock cycle

13.6 Notes

- Internal low-voltage detection operation
  - If the internal voltage falls and the internal low-voltage detection flag is set (LVD:LVD_F=1), internal reset is generated by the low-voltage detection reset function.
    Thus, writing and reading of the internal low-voltage detection register (LVD) is not allowed.
  - The internal low-voltage detection circuit can operate even in sleep mode, stop mode, and watch mode.
    The current is consumed even in any of these modes.
  - The internal low-voltage detection circuit can be started/stopped by the user setting.
- Oscillation stabilization wait time
  If the internal voltage falls below the detection voltage, it takes the oscillation stabilization wait time after the internal low-voltage detection voltage is recovered. For details, see "CHAPTER 8 RESET".
- Hysteresis of detection/release
  The release voltage becomes set value +0.1V so that detection/release may have the hysteresis of 0.1V. For example, when LVD:0.9V ± 0.1V is set, the release voltage becomes 1.0V ± 0.1V.

CHAPTER 14 LOW-VOLTAGE DETECTION (EXTERNAL LOW-VOLTAGE DETECTION)

14.1 Overview

The external low-voltage detection is a function that monitors the power supply voltage and detects when the power supply voltage falls below the detection voltage value.

Figure 14.1-1 Block diagram for low-voltage detection (external low-voltage detection)

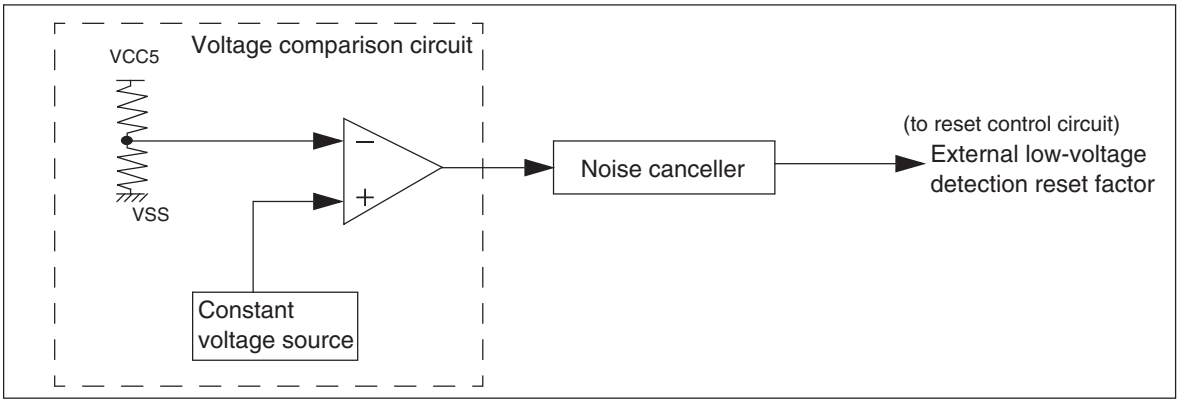
* Rising edge LVDV: 2.3V
Falling edge LVDV: 3.7V - 4.3V (total of 4 ways) variable

14.2 Features

- Method: Voltage under LVDV ± 8% is detected, and resetting of configuration initialization occurred.
  (Rising edge LVDV: 2.3V fixed, falling edge LVDV: 3.7V - 4.3V, variable)
- Number of units: 1
- Operation: Starting/stopping of operation can be configured by the user setting.
  Following the internal RAM write period, low-voltage reset is generated.
- Voltage comparison circuit: Detected voltage and supply voltage are compared. When low-voltage is detected, "L" is set to output.

14.3 Configuration

Figure 14.3-1 Configuration Diagram



Code: ZBLVDR_EXT-0v5-91580MS-1-E

14.4 Registers

Table 14.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0584	LVD5R	LVD5F	LVD*	Reserved	External low-voltage detection rising edge detection register External low-voltage detection falling edge detection register

* For LVD, refer to "CHAPTER 13 LOW-VOLTAGE DETECTION (INTERNAL LOW-VOLTAGE DETECTION)".

14.4.1 LVD5R (Low-Voltage Detect 5v Rise register) External low-voltage detection - rising edge detection register

This register is responsible for clearing of low-voltage detection reset flags.

- **LVD5R: Address 0x0584$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved							LVD5R_F	
0	0	0	0	0	0	0	1	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1), W	Attribute

[bit7 to bit1] Reserved

[bit0] LVD5R_F (Low-Voltage Detect 5v Rise Flag)

External power voltage rising edge detection status can be read.

LVD5R_F	External Power Voltage Rising Edge Detection Flag	
	Read	Write
0	Power supply voltage drop not detected	Clear the flag
1	Power supply voltage drop detected	No effect on operation

When rising edge of external power voltage is detected, LVD5R_F bit is set to "1".

This can be cleared at the time of external reset input.

14.4.2 LVD5F (Low-Voltage Detect 5v Fall register) External low-voltage detection falling edge detection register

This register is responsible for clearing of low-voltage detection reset flags.

- **LVD5F: Address 0x0585$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
LVD5F_PD	LVD5F_SEL[3:1]			LVD5F_OE	LVD5F_SEL[0]	Reserved	LVD5F_F	
0	0	1	1	0	0	0	1	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R(RM1), W	Attribute

[bit7] LVD5F_PD (Low-Voltage Detect 5v Fall Power Down)

Detection of external power voltage falling edge can be configured to initiate and stop.

LVD5F_PD	External power falling edge power down setting
0	Disabled (Detection is executed)
1	Enabled (Detection is stopped)

This bit is initialized only by power-on reset.

(Notes) When power down is changed from enabled to disabled (operation start) in this bit setting, enable the detection (LVD5F_OE=0) 100μs after setting LVD5F_OE=1. If the detection is enabled before 100 μs elapse, a detection flag may be set.

[bit6 to bit4, bit2] LVD5F_SEL (Low-Voltage Detect 5v Fall SELect)

Detection level of external power voltage falling edge can be configured.

LVD5F_SEL[3:0]	External power falling edge detection voltage setting
0100	3.7V ±8%
0110	3.9V ±8%
1000	4.1V ±8%
1010	4.3V ±8%
Other than above	Setting disabled

* Writing is enabled only while LVD5F_OE="1".

[bit3] LVD5F_OE (Low-Voltage Detect 5v Fall Output Enable)

This is output permission signal for external power voltage falling edge detection.

LVD5F_OE	External power falling edge detection output permission setting
0	Enabled
1	Stopped

This bit is initialized only by power-on reset.

[bit1] Reserved

[bit0] LVD5F_F(Low-Voltage Detect 5v Fall Flag)

This is an external power voltage falling edge detection flag.

LVD5F_F	External power voltage falling edge detection flag	
	Read	Write
0	Power supply voltage drop not detected	Clear the flag
1	Power supply voltage drop detected	No effect on operation

When external power voltage decrease is detected, LVD5F_F bit is set to "1".

This can be cleared upon entering external reset.

14.5 Operation Description

■ External low-voltage detection

This function monitors the power supply voltage and generates a settings initialization reset if the external power supply voltage drops below the configured value. The contents of this register cannot be guaranteed if a low-voltage is detected and a settings initialization reset occurs. After the low-voltage reset is released, the reset sequence is executed without delaying for the oscillation stabilization wait time, and then the program is restarted from the address specified by the reset vector.

14.6 Notes

Warnings when using the low-voltage detection reset circuit

- Operation by program
  - The low-voltage detection reset circuit operates as setting without detecting low-voltages from the external power supply.
    The detecting low-voltages from the external power supply can be used as power-on reset.
  - The power current is consumed at the sleep mode, the stop mode, the watch mode. Because the detecting low-voltages from the external power supply operates continuously.

- Operation in stop mode
  Because the low-voltage detection reset continues to operate even in stop mode, when a low-voltage is detected in stop mode, the settings initialization reset is generated and stop mode is cleared.

- Hysteresis of detection/release
  Because detection/release has 0.1V hysteresis, +0.1 is added to the setting value of release voltage. For example, when the setting is LVD5F:4.1V ± 8%, the release voltage is 4.2V ± 8%.

CHAPTER 15 NMI INPUT

15.1 Overview

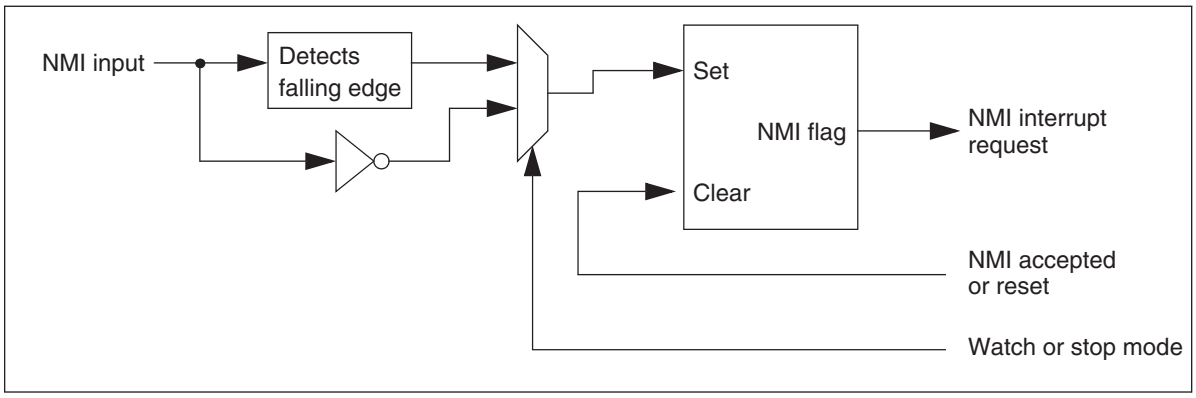
The NMI (non-maskable interrupt) is an ORed signal with a non-maskable interrupt input from the NMIX pin and an interrupt factor not masked for the functional safety. Hereafter in this chapter, a signal ORed with a NMIX pin and other factors is referred to as "NMI input". The NMI can be used as a source for recovering from stop mode.

15.2 Features

This can be utilized under stop mode (incl. power shut-off), or watch mode (incl. power shut-off).

15.3 Configuration

Figure 15.3-1 Block Diagram of NMI Input



15.4 Registers

This function has no registers.

15.5 Operation Description

■ 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 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.

To determine whether the NMI is set by the NMIX external pin or other factors, read IRPR15L register. For details on the IRPR15L register, see "CHAPTER 23 INTERRUPT REQUEST BATCH READ".

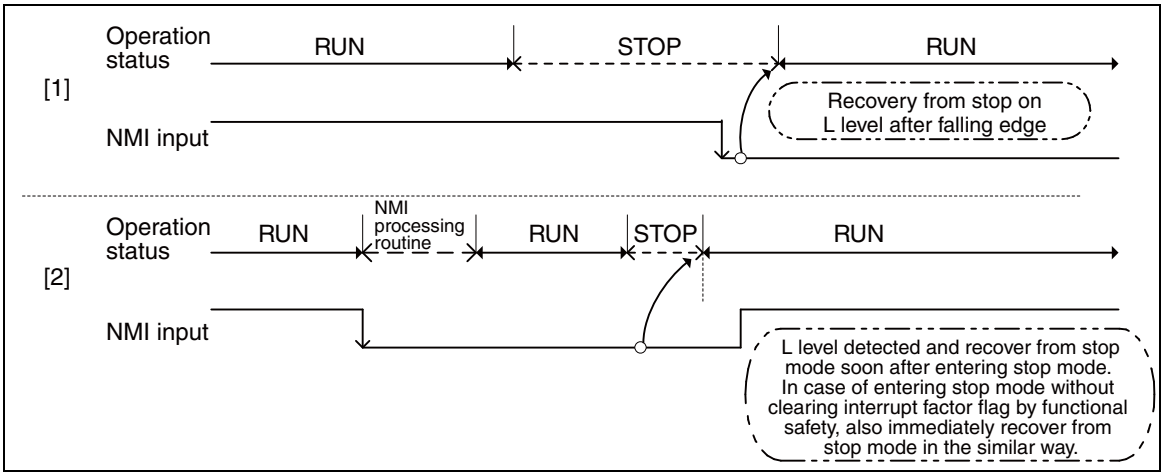
Code: FR81S09_NMI-1v0-91580MS-1-E

■ Recovering From Stop Mode

When switching to stop mode, an "L" level is input to 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 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 Figure 15.5-1). Likewise, if the CPU switches to stop mode (power shutdown) without returning the level of the NMIX pin to the "H" level, the CPU does not control the power shutdown. Return the NMIX pin to the "H" level before entering stop mode so that the NMIX pin is set to the "L" level in stop mode.

The NMI request is not accepted during return from the standby mode (shut-down) because the internal reset is generated.

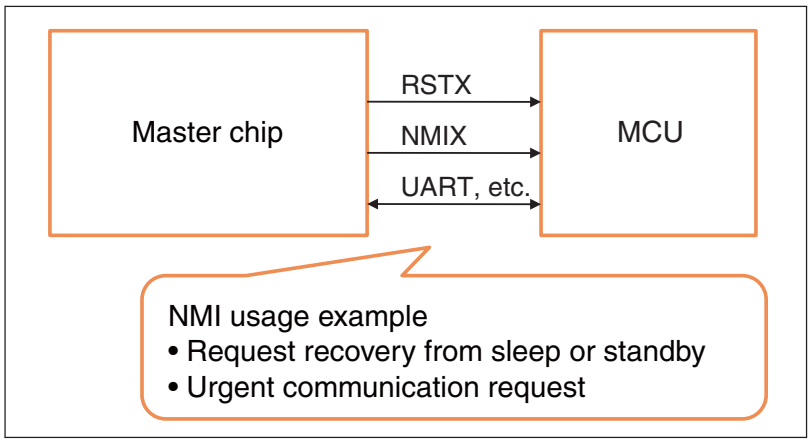
Figure 15.5-1 Recovering from Stop Mode



15.6 Usage Example

This section gives a usage example of the NMI function.

Figure 15.6-1 Usage Example

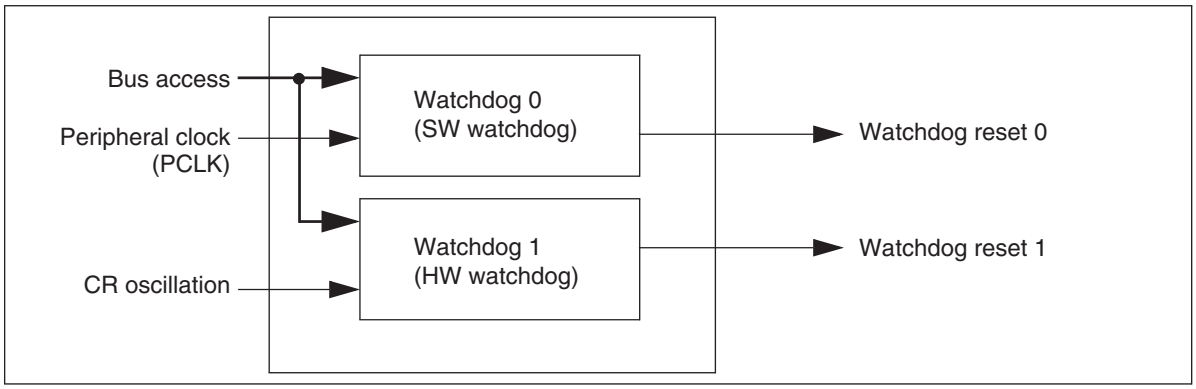


CHAPTER 16 WATCHDOG TIMER

16.1 Overview

This product has two watchdog timers that can detect software and hardware running out of control, and generate reset requests.

Figure 16.1-1 Block Diagram of Watchdog Timer



16.2 Features

■ Watchdog Timer 0 (Software Watchdog)

- Stop mode detection function

  Able to detect the transition to watch mode or stop mode and generate a reset request.
- Watchdog timer clear

  The timer can be 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.
- Counter stop conditions

  The count stops while the CPU is stopped.

■ Watchdog Timer 1 (Hardware Watchdog)

This timer is driven by the clock generated by the built-in CR oscillator circuit immediately after the reset is released.

- 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^{11} cycles.
- Counter stop conditions

  The count stops while using ICE.

  MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH

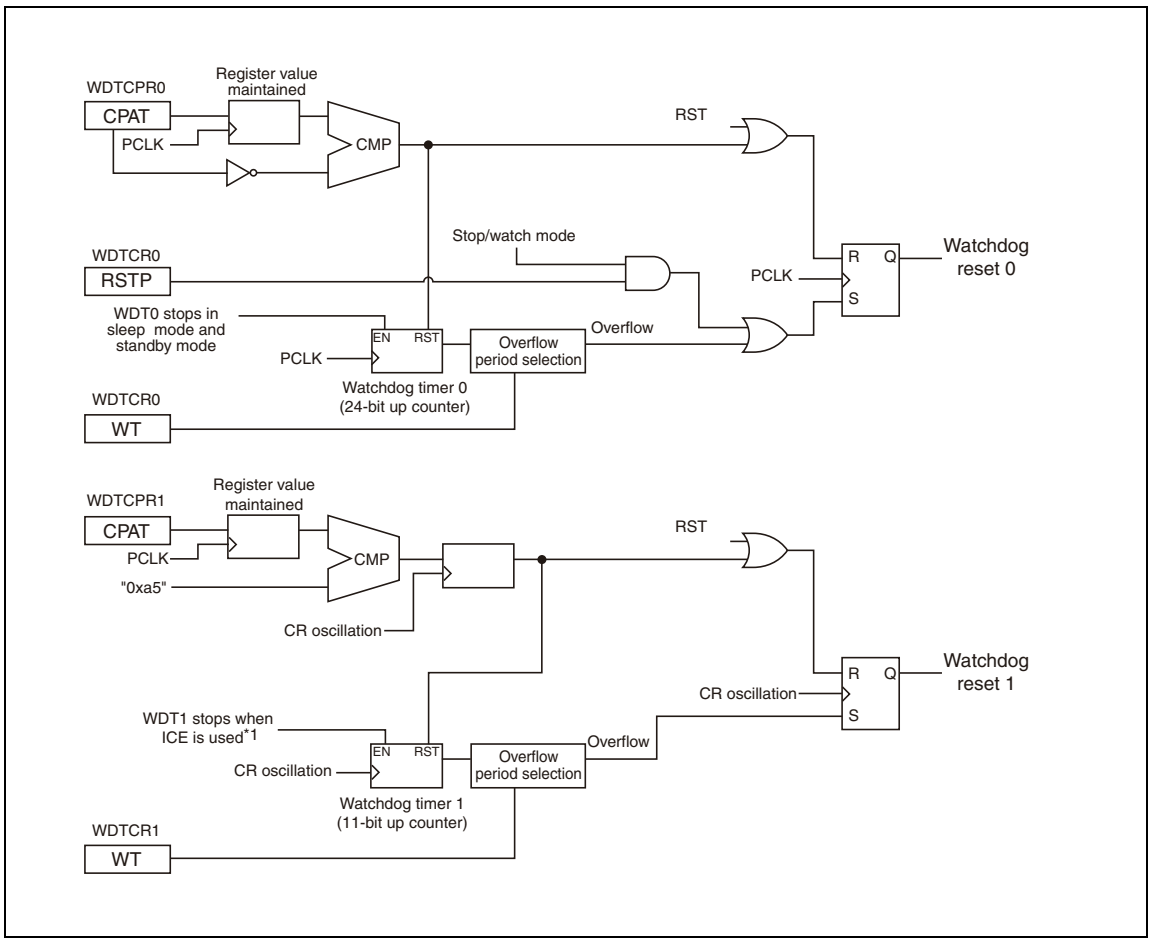
  MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH: While in watch/stop mode, count is continued.

Code: WDT-1v0-91580MS-1-E

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK
MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK: While in watch/stop mode, count is stopped.

16.3 Configuration

Figure 16.3-1 Block Diagram of Watchdog Timer



*1: In MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK, WDT1 is stopped when using ICE and in standby mode.

16.4 Registers

Table 16.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x003C	WDTCR0	WDTCPR0	WDTCR1	WDTCPR1	Watchdog timer 0 control register Watchdog timer 0 clear register Watchdog timer 1 cycle information register Watchdog timer 1 clear register

■ WDTCR0 (WatchDog Timer Configuration Register 0) Watchdog Timer 0 Control Register

This register configures each of the settings of watchdog timer 0.
Writing to this register is ignored after watchdog timer 0 starts.

- **WDTCR0: Address 003C$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	RSTP	Reserved		WT[3:0]				
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W	R/W0	R/W0	R/W	R/W	R/W	R/W	Attribute

[bit7] Reserved

Always write "0" to this bit.

[bit6] RSTP (Reset By SToP) Stop Mode Detection Reset Enabled

This bit configures whether a reset is generated when a transition to watch mode or stop mode is detected while watchdog timer 0 is operating. When this bit is enabled, the watchdog reset 0 occurs when the CPU switches to watch mode or stop mode. When this bit is not enabled, watchdog timer 0 is paused when the CPU switches to watch mode or stop mode, and the count stops until the CPU recovers from watch mode or stop mode.

RSTP	Stop mode detection
0	Not detected (initial value)
1	Generates a reset when detected

Writing to this bit is ignored after watchdog timer 0 starts.

[bit5, bit4] Reserved

Always write "0" to these bits.

[bit3 to bit0] WT[3:0] (Watchdog Timer Interval) Watchdog Timer Period Selection

These bits configure the number of cycles from when watchdog timer 0 was last cleared until watchdog reset 0 is issued as follows.

WT[3:0]	Watchdog timer 0 period
0000	PCLK (peripheral clock) $\times 2^{9}$ cycles
0001	PCLK $\times 2^{10}$ cycles
0010	PCLK $\times 2^{11}$ cycles
0011	PCLK $\times 2^{12}$ cycles
0100	PCLK $\times 2^{13}$ cycles
0101	PCLK $\times 2^{14}$ cycles
0110	PCLK $\times 2^{15}$ cycles
0111	PCLK $\times 2^{16}$ cycles
1000	PCLK $\times 2^{17}$ cycles
1001	PCLK $\times 2^{18}$ cycles
1010	PCLK $\times 2^{19}$ cycles
1011	PCLK $\times 2^{20}$ cycles
1100	PCLK $\times 2^{21}$ cycles
1101	PCLK $\times 2^{22}$ cycles
1110	PCLK $\times 2^{23}$ cycles
1111	PCLK $\times 2^{24}$ cycles

Writing to these bits are ignored after watchdog timer 0 starts.

Watchdog timer 0 is not counted during periods where the CPU is not operating.

Counting is performed while the CPU is operating even if DMA transfers are being performed.

■ WDTCPR0 (WatchDog Timer Clear Pattern Register 0) Watchdog Timer 0 Clear Register

This register starts or clears watchdog timer 0 (delays the issue of a reset).

- **WDTCPR0: Address 003D$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
CPAT[7:0]								
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	Attribute

[bit7 to bit0] CPAT[7:0] (Clear PATtern) Watchdog Timer 0 Clear

Watchdog timer 0 starts by the first write to this register after the reset is released. The watchdog timer is cleared after being started by writing a value with all of the bits inverted from the previous value written. If a value other than the inverse value of the previously written value is written, the watchdog reset 0 is issued at that time.

The value read out from this register is always "0x00" regardless of the value written.

■ WDTCR1 (Watchdog Timer Cycle information Register 1) Watchdog Cycle information Register 1

This register shows a period of watchdog timer 1.

- **WDTCR1: Address 003E$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-				WT[3:0]				
0	0	0	0	0	0	1	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R0,WX	Attribute

[bit7 to bit4] Undefined

These bits always read "0". Writing has no effect on operation.

[bit3 to bit0] WT[3:0] (Watchdog Timer Interval) Watchdog Timer Period Selection

These bits configure the number of cycles from when watchdog timer 1 was last cleared until watchdog reset 1 is issued. Writing to these bits are ignored.

WT[3:0]	Watchdog timer 1 period
0010	CR oscillator × 2^{11} cycles

■ WDTCPR1 (WatchDog Timer Clear Pattern Register 1) Watchdog Timer 1 Clear Register

This register clears watchdog timer 1 (delays the issue of a reset).

- **WDTCPR1: Address 003F$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
CPAT[7:0]								
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	Attribute

[bit7 to bit0] CPAT[7:0] (Clear PATtern) Watchdog Timer 1 Clear

Watchdog timer 1 starts after the reset is released. The watchdog timer is cleared after being started 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.

16.5 Operation Description

This section explains the watchdog timer function.

16.5.1 Software Watchdog Function

■ Setup

Before starting watchdog timer 0, set bits 3 to 0: WT[3:0] of the WDTCR0 register to select the period from clearing the watchdog timer until the reset is issued.

Because watchdog timer 0 is only counted when the CPU is operating, set the period based on the number of program steps and the clock division setting.

Before starting watchdog timer 0, set bit6: RSTP of the WDTCR0 register to select whether or not to generate a reset when a transition to watch mode or stop mode is detected.

- When RSTP=0, the timer stops in watch mode or stop mode.
- When RSTP=1, a reset is generated as soon as the CPU enters watch mode or stop mode.

If you are using watch mode or stop mode, set RTSP=0. Writing to the RSTP bit is ignored after watchdog timer 0 starts.

■ Starting

Watchdog timer 0 starts by the first write of any data to the WDTCPR0 register after reset.

It does not matter what the write data is.

The WDTCPR0 register always reads out "0x00" regardless of the data written.

■ Operation

This section explains the operation of watchdog timer 0 after it has started.

● Counting Conditions

Table 16.5-1 lists the count conditions.

Watchdog timer 0 counts at the rising edges of the peripheral clock (PCLK) while the CPU is operating.

DMA transfers do not effect the operation of the count.

The count only stops while the CPU is stopped, such as in sleep mode. Sampling of the CPU operating state is performed on the peripheral clock (PCLK), with changes in the operating state within the peripheral clock cycle ignored.

The count is stopped in emulator mode when the ICE is connected. The count is also stopped if the watchdog reset suppression function is enabled in the debug interface functions while the ICE is connected.

In all of the circumstances above, because the counter is not cleared but is only paused when the count is stopped, when the count resumes the count continues from the counter value prior to the stop.

Because the peripheral clock is stopped during the oscillator stabilization wait time of the source clock, the watchdog timer count also stops.

Table 16.5-1 Watchdog Timer 0 Counting Conditions

	Standby		Standby (Power shut-off)		Sleep		Run	Main Oscillation Stabilization Wait
	Stop	Watch	Stop	Watch	Bus	CPU		
Peripheral clock (PCLK)	Inactive	Inactive	Inactive	Inactive	Active	Active	Active	Inactive
Watchdog timer 0	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Active	Inactive

See above for the counter stop conditions when the ICE is connected.

● Clearing the Timer

Once the watchdog timer has started, the timer must be cleared before the timer period has elapsed.

Clearing the watchdog timer is performed by writing data to WDTCPR0. The write data must be the value with all bits inverted of the data previously written to WDTCPR0.

When watchdog timer 0 is started, for example, if it is started by writing "0x55" to WDTCPR0, the timer is cleared subsequently by alternatively writing "0xAA" → "0x55" → "0xAA" → "0x55".

Because the read value of WDTCPR0 is always "0x00", the value written previously cannot be determined by reading WDTCPR0. Storing the previously written value in a different location can be avoided by performing two consecutive writes when performing a single clear.

● Reset Request Generation

Watchdog timer 0 generates a watchdog reset request under the following conditions.

- An overflow of the configured watchdog timer period occurs
- There is a transition to watch mode or to stop mode while stop mode detection reset is enabled
- A value other than the inverse value of the previous written value is written to the clear register

16.5.2 Hardware Watchdog Function

■ Setup

Bits 3 to 0: WT[3:0] of the WDTCR1 register of watchdog timer 1 is fixed in hardware.

■ Starting

Watchdog timer 1 starts immediately after the reset is released.

■ Operation

This section explains the operation of watchdog timer 1 after it has started.

● Counting Conditions

Table 16.5-2 lists the count conditions.

Watchdog timer 1 counts at the rising edges of the CR oscillator.

The count is stopped in emulator mode when the ICE is connected. The count is also stopped if the watchdog reset suppression function is enabled in the debug interface functions while the ICE is connected.

MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH

MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH: While in watch/stop mode, count is continued.

MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK

MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK: While in watch/stop mode, count is stopped.

The count does not stop while waiting for the oscillator to stabilize when recovering from standby mode.

Table 16.5-2 Watchdog Timer 1 Counting Conditions (MB91F583AMG/F584AMG/F585AMG/F583AMH/F584AMH/F585AMH/MB91F583ASG/F584ASG/F585ASG/F583ASH/F584ASH/F585ASH)

	Standby		Standby (Power shut-off)*2		Sleep		Run	Main Oscillation Stabilization Wait
	Stop	Watch	Stop	Watch	Bus	CPU		
CR oscillation	Active	Active	Active	Active	Active	Active	Active	Active
Watchdog timer 1*1	Active	Active	Inactive	Inactive	Active	Active	Active	Active

*1: See above for the counter stop conditions when the ICE is connected.

*2: While in standby (power shut-off), CR oscillation functions, but watchdog timer 1 stops counting since microcontroller's power supply is shut off.

Table 16.5-3 Watchdog Timer 1 Counting Conditions (MB91F583AMJ/F584AMJ/F585AMJ/F583AMK/F584AMK/F585AMK/MB91F583ASJ/F584ASJ/F585ASJ/F583ASK/F584ASK/F585ASK)

	Standby		Standby (Power shut-off)		Sleep		Run	Main Oscillation Stabilization Wait
	Stop	Watch	Stop	Watch	Bus	CPU		
CR oscillation	Inactive	Inactive	Inactive	Inactive	Active	Active	Active	Active
Watchdog timer 1*1	Active	Active	Inactive	Inactive	Active	Active	Active	Active

*1: See above for the counter stop conditions when the ICE is connected.

● Clearing the Timer

Once the watchdog timer has started, the timer must be cleared before the timer period has elapsed.

Watchdog timer 1 is cleared by writing "0xa5" to WDTCPR1.

(Notes)
- Clear the timer before going into standby mode or sleep mode.
- Two cycles of CR oscillation are required to clear the watchdog timer 1 by CPU because the watchdog timer 1 (hardware watchdog) operates at CR oscillation. Therefore, clear the watchdog timer 1, then wait for the time of two cycles of CR oscillation to enter standby mode. See the Figure 9.6-1 in "CHAPTER 9 POWER CONSUMPTION CONTROL" for procedure.

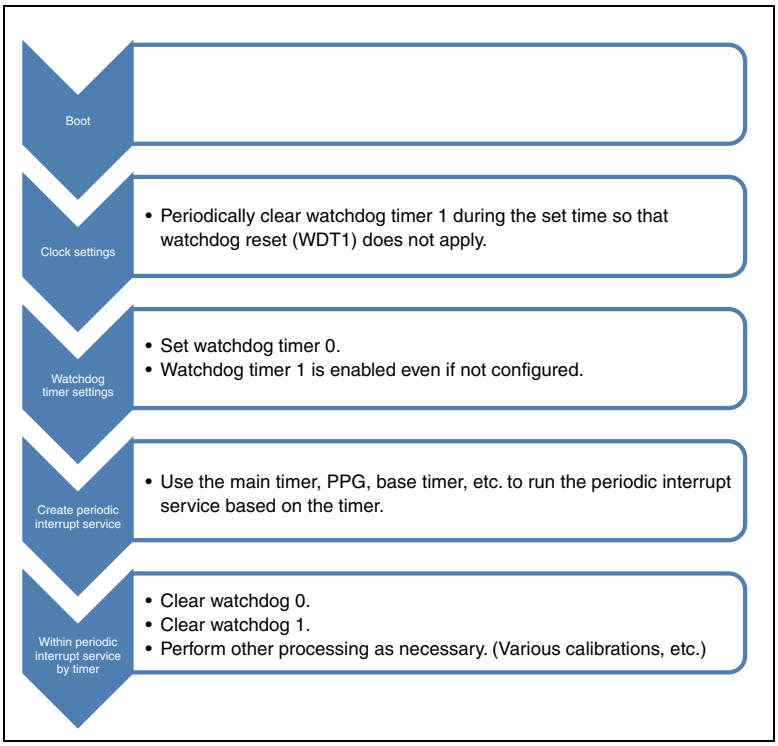
● Reset Request Generation

Watchdog timer 1 generates a watchdog reset request under the following conditions.
- An overflow of the watchdog timer period occurs
- A value other than "0xa5" is written to WDTCPR1

16.6 Usage Example

This example is provided for clearing the watchdog timer.

Figure 16.6-1 Usage Example of Clearing the Watchdog Timers



CHAPTER 17 RAMECC FUNCTION

17.1 Overview

The RAMECC function performs 1-bit error correction and 2-bit error detection in bytes, for data to be read from or written to the RAM.

17.2 Features

■ Target RAM

XBS RAM: 32k bytes (MB91F583)
48k bytes (MB91F584)
48k bytes (MB91F585)

Backup RAM : 8k bytes

■ RAMECC Function

Up to two-bit error detection in bytes is executed for data to be read from or written to the RAM.

If error detection is 1 bit, the error will be corrected.

■ Interrupt Function

Double-bit error will be detected to generate RAM double bit error interrupt signal.

■ Test Mode

In this mode, fake error will be generated for software debugging.

Code: FS26-1v1-91580MS-1-E

17.3 Configuration

Figure 17.3-1 Block Diagram of XBS RAMECC Function (Configuration)

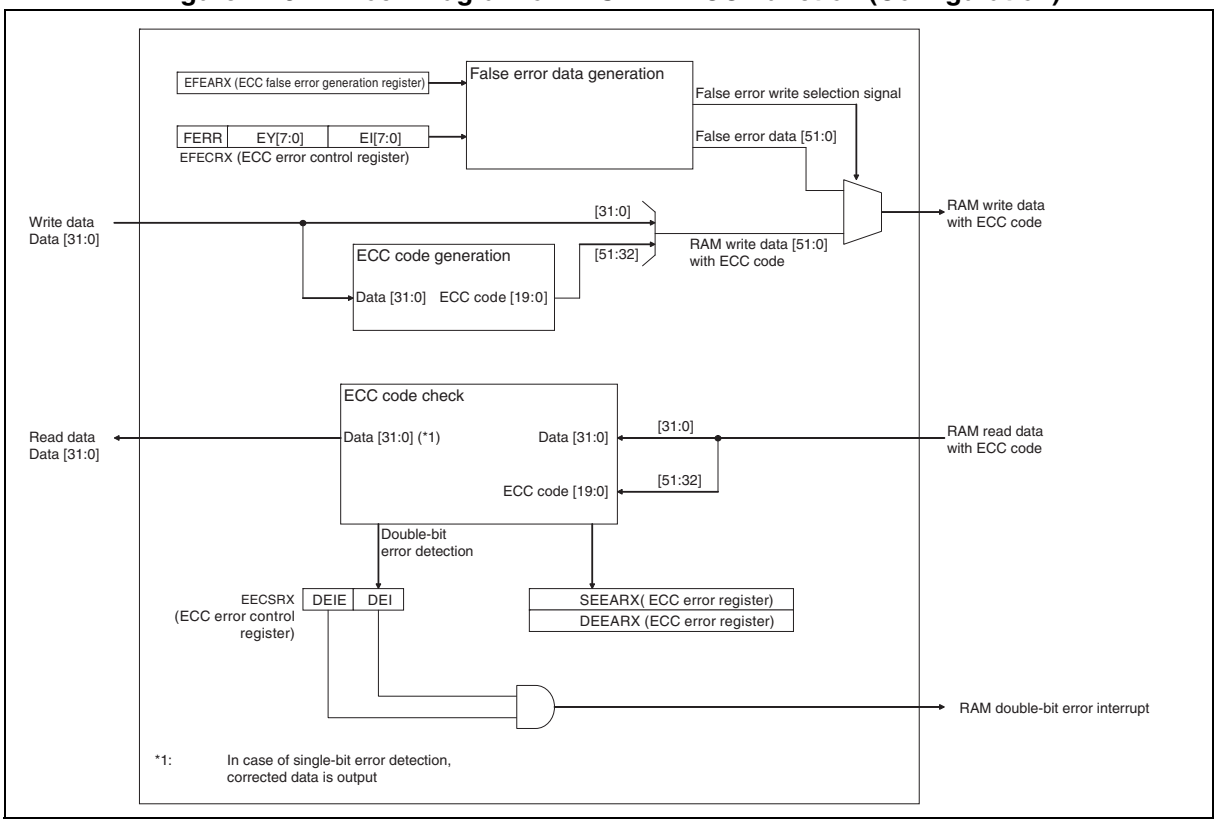
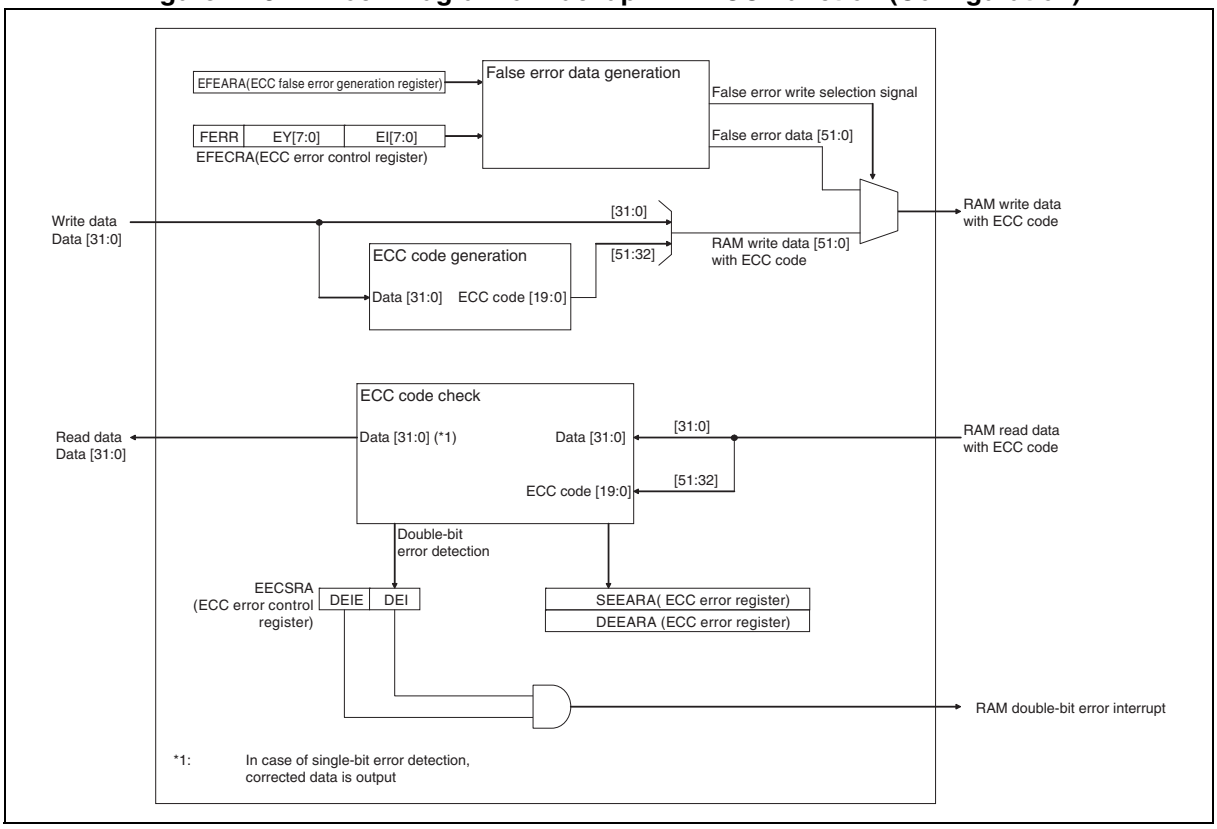


Figure 17.3-2 Block Diagram of Backup RAMECC Function (Configuration)



17.4 Registers

Table 17.4-1 Register Map

Address	Register				Register Function
	+0	+1	+2	+3	
0x3000	SEEARX		DEEARX		Single bit ECC error address register XBS RAM Double bit ECC error address register XBS RAM
0x3004	EECSRX	Reserved	EFEARX		ECC error control register XBS RAM ECC fake error generation address register XBS RAM
0x3008	Reserved	EFECRX			ECC fake error generation control register XBS RAM
0x3024	SEEARA		DEEARA		Single-bit ECC error address register BACKUP-RAM Double-bit ECC error address register BACKUP-RAM
0x3028	EECSRA	Reserved	EFEARA		ECC error control register BACKUP-RAM ECC fake error generation address register BACKUP-RAM
0x302C	Reserved	EFECRA			ECC fake error generation control register BACKUP-RAM

17.4.1 Single bit ECC Error Address Register XBS RAM:SEEARX

If 1-bit error correction is executed during ECC checking for XBS RAM, the ECC error address register holds the relevant address.

- **SEEARX: Address 3000_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15, bit14] Reserved

At writing, write "0".

[bit13 to bit0] D13 to D0: Single bit error generation address bits

If 1-bit error correction is executed during ECC checking, these bits hold the relevant address.

If one of the above events is detected with data already set in this register, this register is not overwritten and holds the first data.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the XBS RAM base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with SEEARX + 2'b00)

17.4.2 Double bit ECC Error Address Register XBS RAM : DEEARX

If 2-bit error correction is executed during ECC checking for XBS RAM, the ECC error address register holds the relevant address.

- **DEEARX: Address 3002_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15, bit14] Reserved

At writing, write "0".

[bit13 to bit0] D13 to D0: Double bit error generation address bits

If 2-bit error correction is executed during ECC checking, these bits hold the relevant address.

If one of the above events is detected with data already set in this register, this register is not overwritten and holds the first data.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the XBS RAM base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with DEEARX + 2'b00)

17.4.3 ECC Error Control Register XBS RAM (EECSRX)

The ECC error control register (EECSRX) holds the status indicating whether 1-bit error correction or 2-bit error detection was executed, or specifies whether an interrupt caused by these events is enabled during ECC checking for XBS RAM.

- **EECSRX: Address 3004_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				DEIE	DEI	Reserved	SEI	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W0	R(RM1),W0	Attributes

[bit7 to bit4] Reserved

At writing, write "0".

[bit3] DEIE: Interrupt enable bit for a double-bit error factor

DEIE	Function
0	Interrupt prohibited
1	Interrupt enabled

[bit2] DEI: Double-bit error generation bit

DEI	Read	Write
0	No double-bit error occurs.	Flag clearing.
1	A double-bit error occurred.	No influence on the operation.

[bit1, bit0] Reserved

At writing, write "0".

[bit0] SEI: Single-bit error generation bit

SEI	Read	Write
0	No single-bit error occurs.	Flag clearing.
1	A single-bit error occurred.	No influence on the operation.

17.4.4 ECC Fake Error Generation Address Register XBS RAM (EFEARX)

ECC fake error generation address register (EFEARX) is used to specify an address to generate a fake error.

- **EFEARX: Address 3006_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15, bit14] Reserved

At writing, write "0".

[bit13 to bit0] D13 to D0: Fake error generation address setting bits

This bit is used to specify an address to generate a fake ECC error for XBS RAM.

Setting with EFECRX.FERR=1 will generate a write access to this address and generate an ECC error by including an intentional error in the data to be written following the setting at the EFECRX.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the XBS RAM base address to the base address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with EFEARX + 2'b00)

17.4.5 ECC Fake Error Generation Control Register XBS RAM (EFECRX)

ECC fake error generation control register (EFECRX) is used to specify a fake error type of XBS RAM to be generated in generation byte and generation bit.

- **EFECRX: Address 3009_H (Access: Byte, Half-word, Word)**

23	22	21	20	19	18	17	16	bit
Reserved							FERR	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	Attributes

15	14	13	12	11	10	9	8	bit
EY7	EY6	EY5	EY4	EY3	EY2	EY1	EY0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit23 to bit17] Reserved

At writing, write "0".

[bit16] FERR: Fake error generation enable bit

FERR	Function
0	Prohibition of a fake ECC error
1	Enabling of a fake ECC error

This bit enables a fake ECC error for XBS RAM.

"0": Prohibits a fake ECC error. (normal operation)
Also, writing "0" from software is ignored.

"1": Enables a fake ECC error. When this bit is set to "1", the following operation sequence is automatically performed.

1. Start writing data including an intentional error to the address specified by EFEARX following EY7-0, EI7-0.
2. Read the same address and detect ECC error.
3. Clear this bit to "0".

[bit15 to bit8] EY7 to EY0: Fake error generation byte setting bits

EY7 to EY0	Target byte on the 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 are used to specify a byte position to generate a fake ECC error of XBS RAM.

For example, when EY2="1" and other bits are "0", the target byte to have fake error generated becomes RAM data [23:16] and no error will be generated to other data.

In addition, when EY2=EY3="1" and other bits are "0", the target byte to have fake error generated becomes RAM data [31:16].

[bit7 to bit0] EI7 to EI0: Fake error generation bit setting bits

EI7 to EI0	Target bit in the byte
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
E17	[7]

These bits are used to specify a bit position to generate a fake ECC error.

For example, when EY2="1", EI4="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [20] and a single-bit error will be corrected.

In addition, when EY2="1", EI4=EI7="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [23] and RAM data [20], and a double-bit error will be detected.

In addition, when EY2=EY3="1", EI4="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [28] and RAM data [20], and a single-bit error will be corrected for each byte.

17.4.6 Single bit ECC Error Address Register BACKUP-RAM: SEEARA

If 1-bit error correction is executed during ECC checking for Backup RAM, the ECC error address register holds the relevant address.

- **SEEARA: Address 3024_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15 to bit11] Reserved

At writing, write "0".

[bit10 to bit0] D10 to D0: Single bit error generation address bits

If 1-bit error correction is executed during ECC checking, these bits hold the relevant address.

If one of the above events is detected with data already set in this register, this register is not overwritten and holds the first data.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the Backup RAM base address to the offset address where lower two bits were added.
(Absolute address) = (0000_4000_H) + (Offset address set with SEEARA + 2'b00)

17.4.7 Single bit ECC Error Address Register BACKUP-RAM: DEEARA

If 2-bit error correction is executed during ECC checking for Backup RAM, the ECC error address register holds the relevant address.

- **DEEARA: Address 3026_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15 to bit11] Reserved

At writing, write "0".

[bit10 to bit0] D10 to D0: Double bit error generation address bits

If 2-bit error correction is executed during ECC checking, these bits hold the relevant address.

If one of the above events is detected with data already set in this register, this register is not overwritten and holds the first data.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the Backup RAM base address to the offset address where lower two bits were added.
(Absolute address) = (0000_4000_H) + (Offset address set with DEEARA + 2'b00)

17.4.8 ECC Error Control Register BACKUP-RAM: EECSRA

The ECC error control register (EECSRX) holds the status indicating whether 1-bit error correction or 2-bit error detection was executed, or specifies whether an interrupt caused by these events is enabled during ECC checking for Backup RAM.

- **EECSRA: Address 3028_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				DEIE	DEI	Reserved	SEI	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W0	R(RM1),W0	Attributes

[bit7 to bit4] Reserved

At writing, write "0".

[bit3] DEIE: Interrupt enable bit for a double-bit error factor

DEIE	Function
0	Interrupt prohibited
1	Interrupt enabled

[bit2] DEI: Double-bit error generation bit

DEI	Read	Write
0	No double-bit error occurs.	This bit is cleared.
1	A double-bit error occurred.	No influence on the operation.

[bit1] Reserved

At writing, write "0".

[bit0] SEI: Single-bit error generation bit

SEI	Read	Write
0	No single-bit error occurs.	Flag clearing.
1	A single-bit error occurred.	No influence on the operation.

17.4.9 ECC Fake Error Generation Address Register BACKUP-RAM: EFEARA

ECC fake error generation address register (EFEARA) is used to specify an address to generate a fake error for Backup RAM.

- **EFEARA: Address $302A_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	Attributes

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

[bit15 to bit11] Reserved

At writing, write "0".

[bit10 to bit0] D10 to D0: Fake error generation address setting bits

This bit is used to specify an address to generate a fake ECC error for Backup RAM.

Setting with EFECRA.FERR=1 will generate a write access to this address and generate an ECC error by including an intentional error in the data to be written following the setting at the EFECRA.

(Note) The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the Backup RAM base address to the base address where lower two bits were added.
(Absolute address) = (0000_4000_H) + (Offset address set with EFEARA + 2'b00)

17.4.10 ECC Fake Error Generation Control Register BACKUP-RAM: EFECRA

ECC fake error generation control register (EFECRA) is used to specify a fake error type of Backup RAM to be generated in generation byte and generation bit.

- **EFECRA: Address 302D$_H$ (Access: Byte, Half-word, Word)**

23	22	21	20	19	18	17	16	bit
Reserved							FERR	
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	Attributes

15	14	13	12	11	10	9	8	bit
EY7	EY6	EY5	EY4	EY3	EY2	EY1	EY0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

- **[bit23 to bit17] Reserved**

At writing, write "0".

- **[bit16] FERR: Fake error generation enable bit**

FERR	Function
0	Prohibition of a fake ECC error
1	Enabling of a fake ECC error

This bit enables a fake ECC error for Backup RAM.

"0": Prohibits a fake ECC error. (normal operation)
Also, writing "0" from software is ignored.

"1": Enables a fake ECC error. When this bit is set to "1", the following operation sequence is automatically performed.

1. Start writing data including an intentional error to the address specified by EFEARA following EY7-0, EI7-0.
2. Read the same address and detect ECC error.
3. Clear this bit to "0".

[bit15 to bit8] EY7 to EY0: Fake error generation byte setting bits

EY7 to EY0	Target byte on the 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 are used to specify a byte position to generate a fake ECC error of Backup RAM.

For example, when EY2="1" and other bits are "0", the target byte to have fake error generated becomes RAM data [23:16] and no error will be generated to other data.

In addition, when EY2=EY3="1" and other bits are "0", the target byte to have fake error generated becomes RAM data [31:16].

[bit7 to bit0] EI7 to EI0: Fake error generation bit setting bits

EI7 to EI0	Target bit in the byte
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
E17	[7]

These bits are used to specify a bit position to generate a fake ECC error of Backup RAM.

For example, when EY2="1", EI4="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [20] and a single-bit error will be corrected.

In addition, when EY2="1", EI4=EI7="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [23] and RAM data [20], and a double-bit error will be detected.

In addition, when EY2=EY3="1", EI4="1" and other bits are "0", the target bit to have fake error generated becomes RAM data [28] and RAM data [20], and a single-bit error will be corrected for each byte.

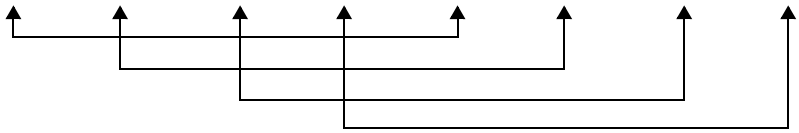
17.5 Operation

17.5.1 RAMECC Function

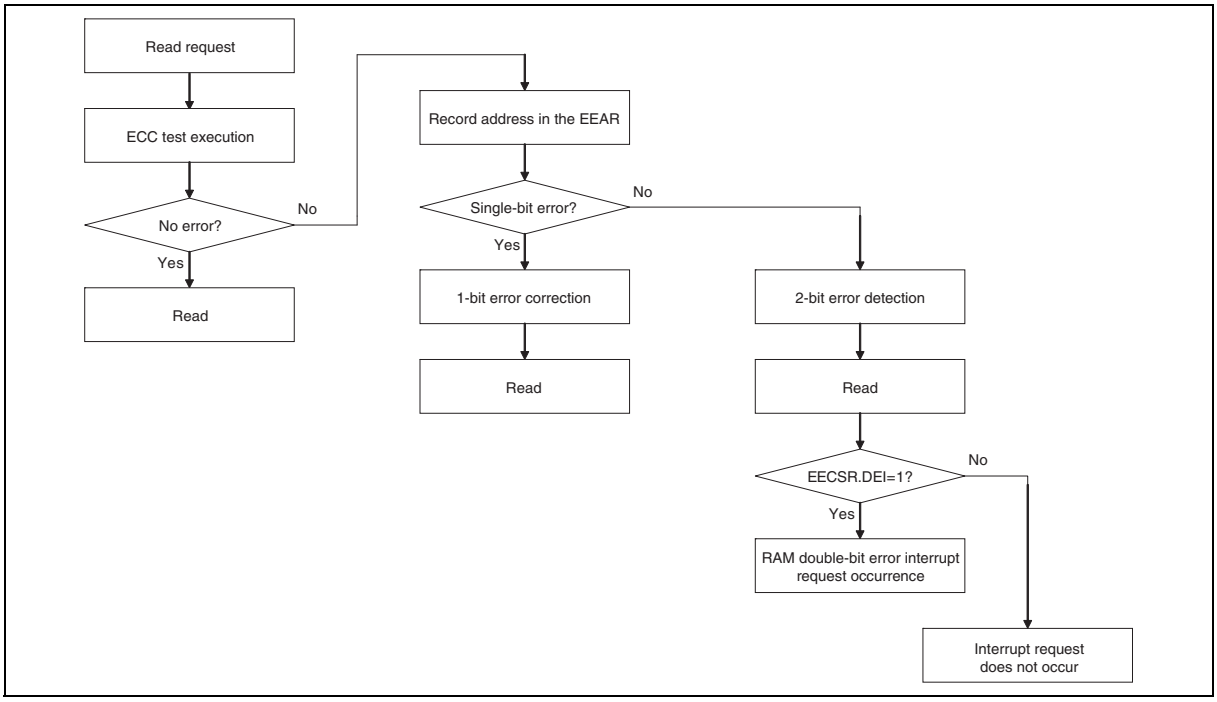
The RAMECC function will always be operated (however, it will be stopped during RAM diagnosis). At error detection, the EEAR register holds an error generation address. If an error is detected with the error generation address held in the EEAR register, the EEAR register is not overwritten and holds the first value.

ECC sign matrix records the redundant 5 bits per byte as an ECC sign.

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]



The operating flowchart is shown below.



17.5.2 Interrupt-Related Register

To generate an interrupt, write "1" to the interrupt generation enabling bits (DEIE) according to the purposes, and set the RAMECC interrupt vector.

Interrupt factor	Interrupt vector	Interrupt level
DEI (RAM double-bit error interrupt)	#15($000FFFC0_H$)	Fixed to 15(F_H)

For details of the interrupt levels and interrupt vectors, see "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

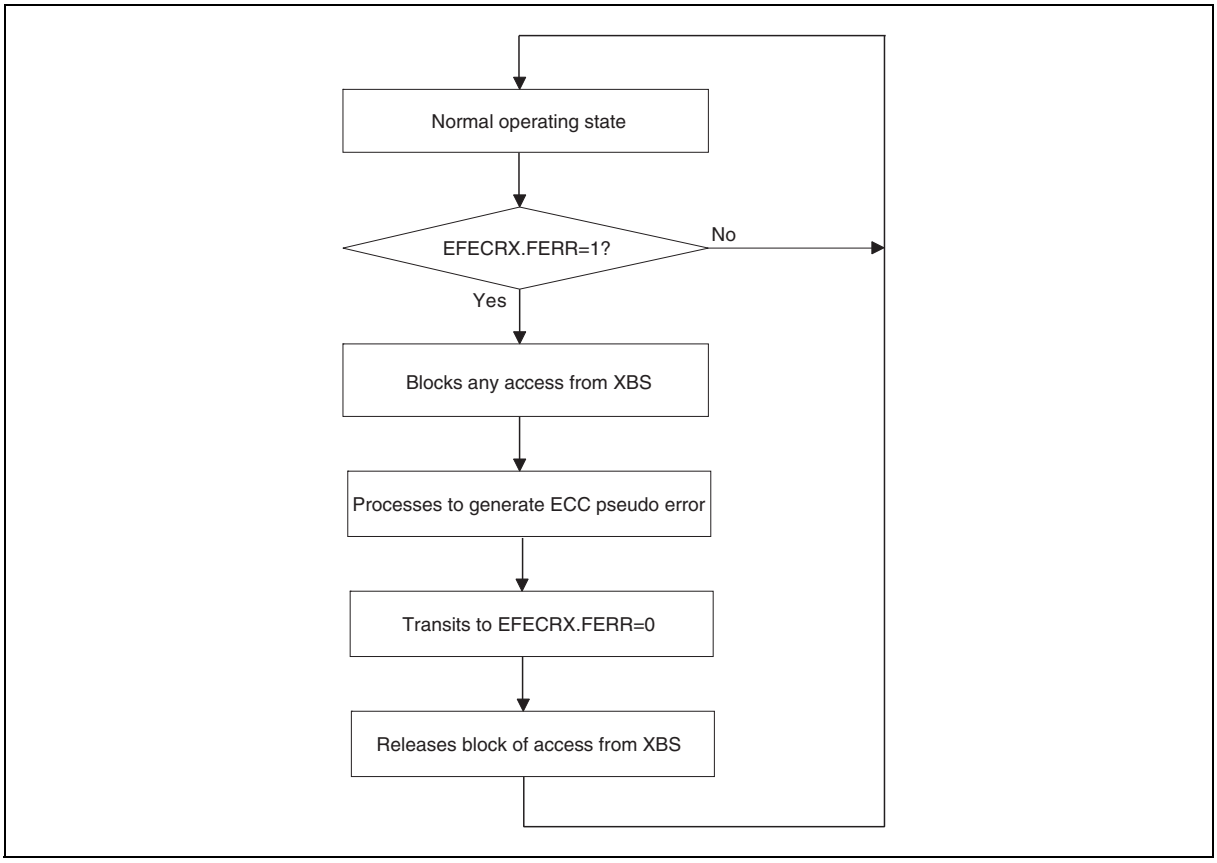
The interrupt request flags (DEI) are not automatically cleared. So, clear them with software before return from interrupt processing. (Write "0" in the DEI bits.)

17.5.3 Test Mode

In this mode, fake error will be generated for software debugging.

The ECC fake error of XBS RAM will be generated with following procedure (same procedure as Backup RAM):

1. Specify an address to generate a fake error to the ECC fake error generation address register (EFEARX).
2. Specify the byte and bit with the ECC fake error generation control register XBS RAM (EFECRX).
   (1) Set a byte position to the EFECRX.EY[7:0] to generate a fake error.
   (2) Set a bit position to the EFECRX.EI[7:0] to generate a fake error.
3. Write "1" to the FERR bit in the ECC fake error generation control register XBS RAM (EFECRX).



Start writing data including an intentional error to the address specified by EFEARX following EY[7:0] and EI[7:0]. Read the data after the write to detect a fake error. Operation after writing "1" to the FERR bit is automatically performed.

17.5.4 Notes

When detected malfunction of 3 bits or more, a single bit error is detected and an unintended correction may be operated.

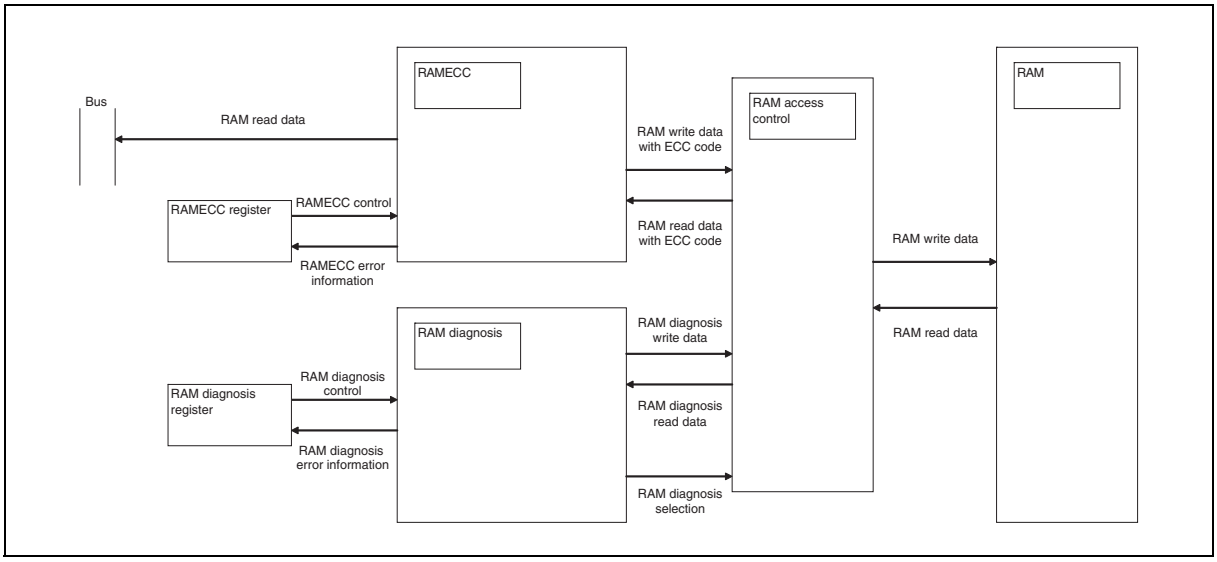
During generation of the ECC fake error of XBS RAM, if an attempt is made to access the XBS RAM, the access is blocked. Therefore, access XBS RAM after the completion of the ECC fake error process.

(Same procedure as Backup RAM.)

CHAPTER 18 RAM DIAGNOSIS FUNCTION

18.1 Overview

The RAM diagnosis function diagnoses and initializes RAM.



For RAMECC functions, see "CHAPTER 17 RAMECC FUNCTION".

18.2 Features

■ Target RAM

The RAM diagnosis targets are listed below. (The diagnosis targets also include ECC areas.)

- XBS RAM
  MB91F583: 32 Kbytes
  MB91F584: 48 Kbytes
  MB91F585: 48 Kbytes
- Backup RAM: 8 Kbytes

■ 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 initialization 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)

Code: FS28-3v1-91580MS-1-E

■ Test mode

In this mode, fake TEST error will be generated for software debugging.

18.3 Configuration

Figure 18.3-1 Block Diagram of XBS RAM Diagnosis Function (Construction)

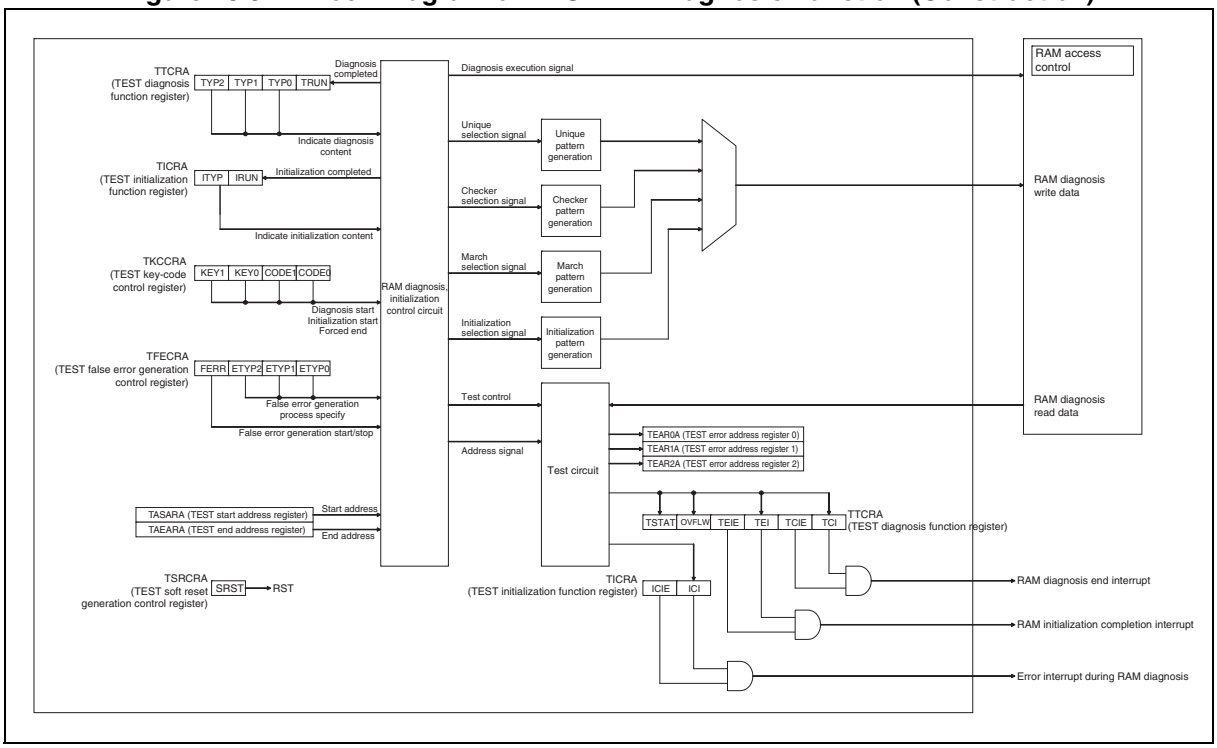
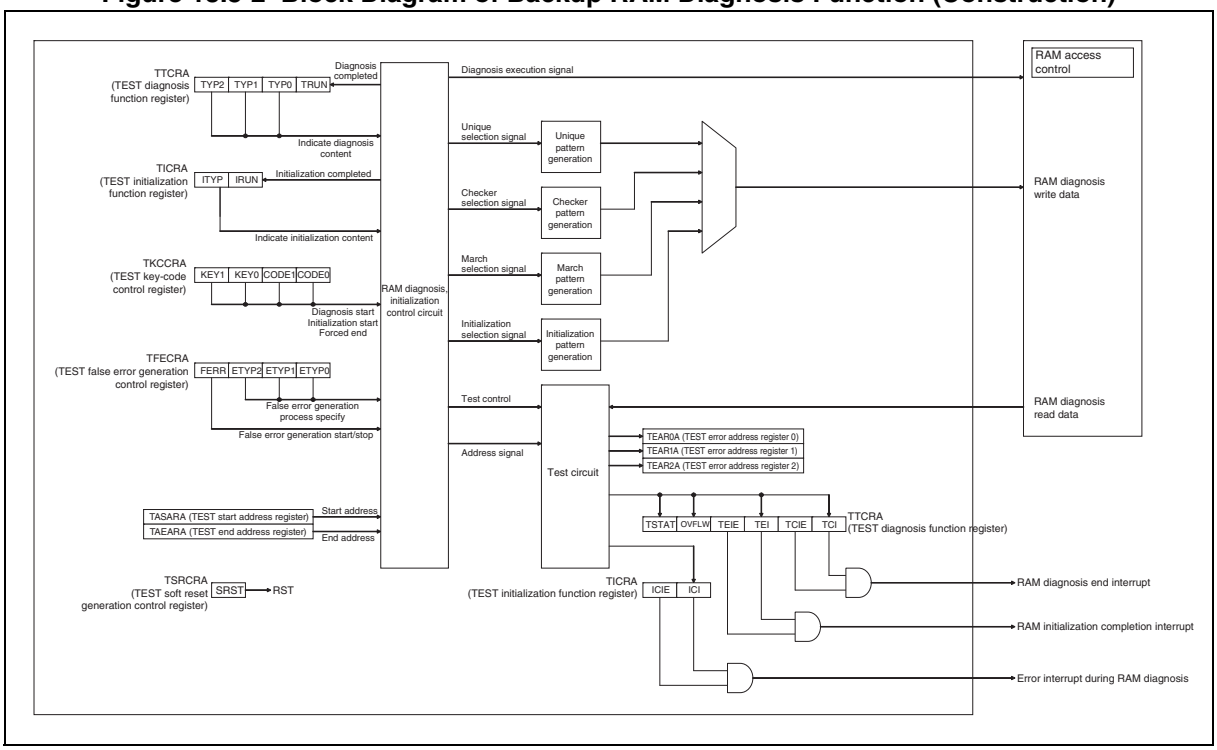


Figure 18.3-2 Block Diagram of Backup RAM Diagnosis Function (Construction)



18.4 Registers

Table 18.4-1 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 soft 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 soft reset generation control register BACKUP-RAM TEST key code control register BACKUP-RAM

18.4.1 TEST Error Address Register 0 XBS RAM : TEAR0X

If an error occurs during RAM diagnosis, TEST error address register 0 (TEAR0X) holds the relevant address.

- **TEAR0X: Address 300C$_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[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. D13 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D13 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 bit14] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[bit13 to bit0] D13 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)

18.4.2 TEST Error Address Register 1 XBS RAM : TEAR1X

Only when an error occurs in the address different from that held in TEAR0X during RAM diagnosis, TEST error address register 1 (TEAR1X) holds that address.

- **TEAR1X: Address 3010_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[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. D13 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D13 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 bit14] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[bit13 to bit0] D13 to D0: Error generation address bits

During RAM diagnosis of 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 TEAR1X + 2'b00)

18.4.3 TEST Error Address Register 2 XBS RAM : TEAR2X

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)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved		D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[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. D13 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D13 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 bit14] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[bit13 to bit0] D13 to D0: Error generation address bits

During RAM diagnosis of 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 TEAR2X + 2'b00)

18.4.4 TEST Start Address Register XBS RAM : TASARX

TEST start address register (TASARX) specifies the start address of RAM diagnosis for XBS RAM and initialization.

- **TASARX: Address 301A$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved		ST13	ST12	ST11	ST10	ST9	ST8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit15, bit14] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[bit13 to bit0] ST13 to ST0: RAM diagnosis start address bits

These bits are used to specify the address from which the RAM diagnosis and initialization start.

(Note) Setting of a value outside the XBS RAM area and a value that sets TASARX.ST13 to ST0 > TAEARX.ED13 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)

18.4.5 TEST End Address Register XBS RAM : TAEARX

TEST end address register (TAEARX) specifies the end address of RAM diagnosis for XBS RAM and initialization.

- **TAEARX: Address 3018_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved		ED13	ED12	ED11	ED10	ED9	ED8	
0	0	1	0	1	1	1	1	Initial values
R0, W0	R0, W0	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0	
1	1	1	1	1	1	1	1	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit15, bit14] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[bit13 to bit0] ED13 to ED0: RAM diagnosis end address bits

These bits are used to specify the address with which the RAM diagnosis for XBS RAM and initialization end.

(Note) Setting of a value outside the XBS RAM area and a value that sets TASARX.ST13 to ST0 > TAEARX.ED13 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'b00)

(Note) The initial value of this register is calculated using the last address (0001_BFFF_H) of MB91F584/F585 (XBUS RAM size: 48 KB). The last address varies with the model. Therefore, before the start of initialization/diagnosis, use the TKCCRX (TEST key code control register) to change it to the specified address. The last address in MB91F583 (XBUS RAM size: 32 KB) is 0001_7FFF_H.

18.4.6 TEST Diagnosis Function Register XBS RAM : TTCRX

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)**

15	14	13	12	11	10	9	8	bit
Reserved						TSTAT	OVFLW	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	Attributes

7	6	5	4	3	2	1	0	bit
TEIE	TEI	TCIE	TCI	TTYP2	TTYP1	TTYP0	TRUN	
0	0	0	0	1	1	0	0	Initial values
R/W	R(RM1),W	R/W	R (RM1), W	R/W	R/W	R/W	R, WX	Attributes

[bit15 to bit10] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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 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.

"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.

18.4.7 TEST Initialization Function Register XBS RAM : TICRX

The TEST initialization function register (TICRX) specifies the RAM initialization content, and holds the initialization result and its status.

- **TICRX: Address 301D$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				ICIE	ICI	ITYP	IRUN	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R/W	R (RM1), W	R/W	R, WX	Attributes

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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.

"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.

"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.

18.4.8 TEST Soft Reset Generation Control Register XBS RAM : TSRCRX

The TEST soft 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)**

7	6	5	4	3	2	1	0	bit
SRST	Reserved							
0	0	0	0	0	0	0	0	Initial values
R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

[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.

This bit reads out "0".

"1": Reset pulses occur for 4 τ only and the internal circuit for RAM diagnosis except this register is reset.

[bit6 to bit0] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

18.4.9 TEST Fake Error Generation Control Register XBS RAM : TFECRX

TEST fake error generation control register (TFECRX) generates a fake error in RAM diagnosis operation for XBS RAM. Error generation can be specified which of RAM diagnosis operations.

- **TFECRX: Address 301C$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				FERR	ETYP2	ETYP1	ETYP0	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	R/W	Attributes

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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.

"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

18.4.10 TEST Key Code Control Register XBS RAM : TKCCRX

The TEST key code control register (TKCCRX) is used to start or forcibly terminate the RAM diagnosis for XBS RAM or initialization.

- **TKCCRX: Address 3023_H (Access: Byte, Half-word, Word))**

7	6	5	4	3	2	1	0	bit
KEY1	KEY0	Reserved				CODE1	CODE0	
0	0	0	0	0	0	0	0	Initial values
R0, W	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	Attributes

[bit7, bit6] KEY1, KEY0: Key code control bits

Key code control bits. Operation is performed when operation indication content is set to CODE[1:0] (no change during 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". At writing, write "0".

[bit1, bit0] CODE1, CODE0: RAM diagnosis/initialization control bits

These bits specify operational direction for the key code procedure above.

CODE[1:0]	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.

18.4.11 TEST Error Address Register 0 BACKUP-RAM : TEAR0A

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)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[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". At writing, write "0".

[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)

18.4.12 TEST Error Address Register 1 BACKUP-RAM : TEAR1A

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)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[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". At writing, write "0".

[bit10 to bit0] D10 to D0: Error generation address bits

During RAM diagnosis of 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)

18.4.13 TEST Error Address Register 2 BACKUP-RAM : TEAR2A

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)**

31	30	29	28	27	26	25	24	bit
TER2	TER1	TER0	Reserved					
0	0	0	0	0	0	0	0	Initial values
R, WX	R, WX	R, WX	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

15	14	13	12	11	10	9	8	bit
Reserved					D10	D9	D8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	R, WX	Attributes

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 values
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attributes

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for of 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". At writing, write "0".

[bit10 to bit0] D10 to D0: Error generation address bits

During RAM diagnosis of 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 TEAR2A + 2'b00)

18.4.14 TEST Start Address Register BACKUP-RAM : TASARA

TEST start address register (TASARX) specifies the start address of RAM diagnosis for XBS RAM and initialization.

- **TASARA: Address 303E$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved					ST10	ST9	ST8	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit15 to bit11] Reserved

Reserved bit. This bit reads out "0". At writing, write "0".

[bit10 to bit0] ST10 to ST0: RAM diagnosis start address bits

These bits are used to specify the address from which the RAM diagnosis of Backup RAM and initialization start.

(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)

18.4.15 TEST End Address Register BACKUP-RAM : TAEARA

TEST end address register (TAEARA) specifies the end address of RAM diagnosis for Backup RAM and initialization.

- **TAEARA: Address 303C$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved					ED10	ED9	ED8	
0	1	0	1	1	1	1	1	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0	
1	1	1	1	1	1	1	1	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit15 to bit11] Reserved

Reserved bit. This bit reads out "0". At writing, write "0".

[bit10 to bit0] ED10 to ED0: RAM diagnosis end address bits

These bits are used to specify the address with which the RAM diagnosis for Backup RAM and initialization end.

(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'b00)

18.4.16 TEST Diagnosis Function Register BACKUP-RAM : TTCRA

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)**

15	14	13	12	11	10	9	8	bit
Reserved						TSTAT	OVFLW	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R, WX	R, WX	Attributes

7	6	5	4	3	2	1	0	bit
TEIE	TEI	TCIE	TCI	TTYP2	TTYP1	TTYP0	TRUN	
0	0	0	0	1	1	0	0	Initial values
R/W	R(RM1),W	R/W	R (RM1), W	R/W	R/W	R/W	R, WX	Attributes

[bit15 to bit10] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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 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 for Backup RAM 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.TSTAT="1" when RAM diagnosis of Backup RAM 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 (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 BackupRAM 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 of Backup RAM status.

"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.

18.4.17 TEST Initialization Function Register BACKUP-RAM : TICRA

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)**

7	6	5	4	3	2	1	0	bit
Reserved				ICIE	ICI	ITYP	IRUN	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R/W	R (RM1), W	R/W	R, WX	Attributes

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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 of Backup RAM initialization.

"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.

18.4.18 TEST Soft Reset Generation Control Register BACKUP-RAM : TSRCRA

The TEST soft 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)**

7	6	5	4	3	2	1	0	bit
SRST	Reserved							
0	0	0	0	0	0	0	0	Initial values
R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	Attributes

[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 of Backup RAM.

This bit reads out "0".

"1": Reset pulses occur for 4 τ only and the internal circuit for RAM diagnosis except this register is reset.

[bit6 to bit0] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

18.4.19 TEST Fake Error Generation Control Register BACKUP-RAM : TFECRA

TEST fake error generation control register (TFECRA) generates a fake error in RAM diagnosis operation for Backup RAM. Error generation can be specified which of RAM diagnosis operations.

- **TFECRA: Address 3040_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				FERR	ETYP2	ETYP1	ETYP0	
0	0	0	0	0	0	0	0	Initial values
R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	R/W	R/W	Attributes

[bit7 to bit4] Reserved

Reserved bits. These bits read out "0". At writing, write "0".

[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

18.4.20 TEST Key Code Control Register BACKUP-RAM : TKCCRA

The TEST key code control register (TKCCRA) is used to start or forcibly terminate the RAM diagnosis for Backup RAM or initialization.

- **TKCCRA: Address 3047_H (Access: Byte, Half-word, Word))**

7	6	5	4	3	2	1	0	bit
KEY1	KEY0	Reserved				CODE1	CODE0	
0	0	0	0	0	0	0	0	Initial values
R0, W	R0, W	R0, W0	R0, W0	R0, W0	R0, W0	R/W	R/W	Attributes

[bit7, bit6] KEY1, KEY0: Key code control bits

Key code control bits. Operation is performed when operation indication content is set to CODE[1:0] (no change during 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". At writing, write "0".

[bit1, bit0] CODE1, CODE0: RAM diagnosis/initialization control bits

These bits specify operational direction for the key code procedure above.

CODE[1:0]	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.

18.5 Operation

18.5.1 RAM Diagnosis

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.

When march ends, the contents of RAM are all1. When any other diagnosis ends, some kind of value remains. If a read access is performed without performing initialization or a write access, an ECC error occurs. Be sure to perform initialization or a write access before performing a read access.

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 "02h" -> "42h" -> "82h" -> "C2h" four times to the TEST key code control register (TKCCRX), then start diagnosing.

Perform the same procedure for RAM diagnosis of Backup RAM.

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 2X) and TEST diagnosis function register (TTCRX). The RAM holds the diagnosis data.

In addition, write continuously "00h" -> "40h" -> "80h" -> "C0h" four times 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 results is not reliable.

RAM diagnosis of Backup RAM is also performed in the same procedure.

18.5.2 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 "01h" -> "41h" -> "81h" -> "C1h" four times to the TEST key code control register (TKCCRX), then start diagnosing.

Perform the same procedure for RAM diagnosis of Backup RAM.

When RAM initialization are completed, IRUN bit of the TEST initialization function register (TICRX) becomes "0" to finish the RAM initialization.

In addition, write continuously "00h" -> "40h" -> "80h" -> "C0h" four times 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 reliable.

RAM initialization of Backup RAM is also performed in the same procedure.

18.5.3 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)
TTCRX.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 22 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.)

18.5.4 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 "02h" -> "42h" -> "82h" -> "C2h" four times to the TEST key code control register (TKCCRX), then start diagnosis pattern (See Section "18.5.1 RAM Diagnosis").

Backup RAM fake errors are also generated in the same procedure.

18.5.5 Number of Required Cycles

	XBS RAM	Backup RAM
MB91F583	8K word address (32 Kbytes)	2K word address (8 Kbytes)
MB91F584	12K word address (48 Kbytes)	
MB91F585	12K word address (48 Kbytes)	

As an example, the following shows the estimation of the cycle count required for RAM diagnosis for XBS RAM:MB91F585 and initialization types.

(1) "RAM diagnosis (unique)"

- Write (1 cycle)
- Read 1 (1 cycle)
- Read 2 (1 cycle)

Processes above have full word address (12k: MB91F585) per a word address and the entire number of cycles is as follows:

$$(\underset{\text{Write}}{\underline{1}} + \underset{\text{Read 1}}{\underline{1}} + \underset{\text{Read 2}}{\underline{1}}) \times \underset{\text{12k words}}{\underline{12288}} + 1 = \underset{\text{Total}}{\underline{36865}}$$

(2) "RAM diagnosis (checker)"

- Write 1 (1 cycle)
- Read 1 (1 cycle)

Processes above have full word address (12k: MB91F585) per a word address. 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)
- Read 2 (2 × 4 cycles)

Moreover, the same processing is repeated with data different from above data. The entire number of cycles is as follows:

$$((\underset{\text{Write 1}}{\underline{1}} + \underset{\text{Read 1}}{\underline{1}}) \times \underset{\text{12k words}}{\underline{12288}} + 1 + \underset{\text{Write 2}}{\underline{5}} + \underset{\text{Read 2}}{\underline{8}}) \times \underset{\text{Repetition}}{\underline{2}} = \underset{\text{Total}}{\underline{49180}}$$

(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 processes above have full word addresses (12k: MB91F585) per a word address. Moreover, the same processing is repeated with data different from above data. So, the entire number of cycles is as follows:

$$(\underset{\text{Write}}{\underline{\quad 3 \quad}} + \underset{\text{Read}}{\underline{\quad 4 \quad}}) \times \underset{\text{12k words}}{\underline{\quad 12288 \quad}} \times \underset{\text{Repetition}}{\underline{\quad 2 \quad}} = \underset{\text{Total}}{\underline{\quad 172032 \quad}}$$

(4) "RAM initialization"

- Write (1 cycle)

Processes above have full word address (12k: MB91F585) per a word address. The entire number of cycles is as follows:

$$\underset{\text{Write}}{\underline{\quad 1 \quad}} \times \underset{\text{12k words}}{\underline{\quad 12288 \quad}} = \underset{\text{Total}}{\underline{\quad 12288 \quad}}$$

The time required for 2MHz and 80MHz operations is obtained as follows.

Table 18.5-1 Time Required for RAM Diagnosis and Initialization

	Unique	Checker	March	Initialization	Total
Number of cycles	36865	49180	172032	12288	270365
2[MHz] (=500[ns])	18432.5[μs]	24590[μs]	86016[μs]	6144[μs]	135[ms]
80MHz (=12.5[ns])	460.8[μs]	614.8[μs]	2150.4[μs]	153.6[μs]	3.4[ms]

Moreover, the time required for the diagnosis with initial register values is obtained as follows.

Table 18.5-2 Time Required for Diagnosis (Initialization) after Power-on Reset Releasing

	Unique	Checker	Total
Number of cycles	36865	49180	86045
2[MHz] (=500[ns])	18432.5[μs]	24590[μs]	43[ms]

18.5.6 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).
- If either diagnosis or initialization is being performed, the start setting is ignored, and the process currently being executed continues. To start any process, use the following procedure to confirm that no process is being executed, and then start it.
  1. Confirm that TTCR:TRUN, TICR, and IRUN are all "0".
  2. Use TKCCR to start a key code operation of either diagnosis or initialization.
- A forcible termination is also available by a key code operation. Use the following operation for forcible termination.
  - Write "8h00","8h40","8h80","8hc0" in succession to TKCCR.

CHAPTER 19 BUS DIAGNOSIS FUNCTION

19.1 Overview

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

19.2 Features

The internal bus of the CPU consists of AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2) and Rbus(PCLK2). The bus diagnosis function is accomplished by adding parity to AHB, APB, and Rbus address and data, and to bus control signals and by checking that parity. 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.

Code: FS44-1v4-91580MS-1-E

19.3 Configuration

Figure 19.3-1 Configuration Diagram

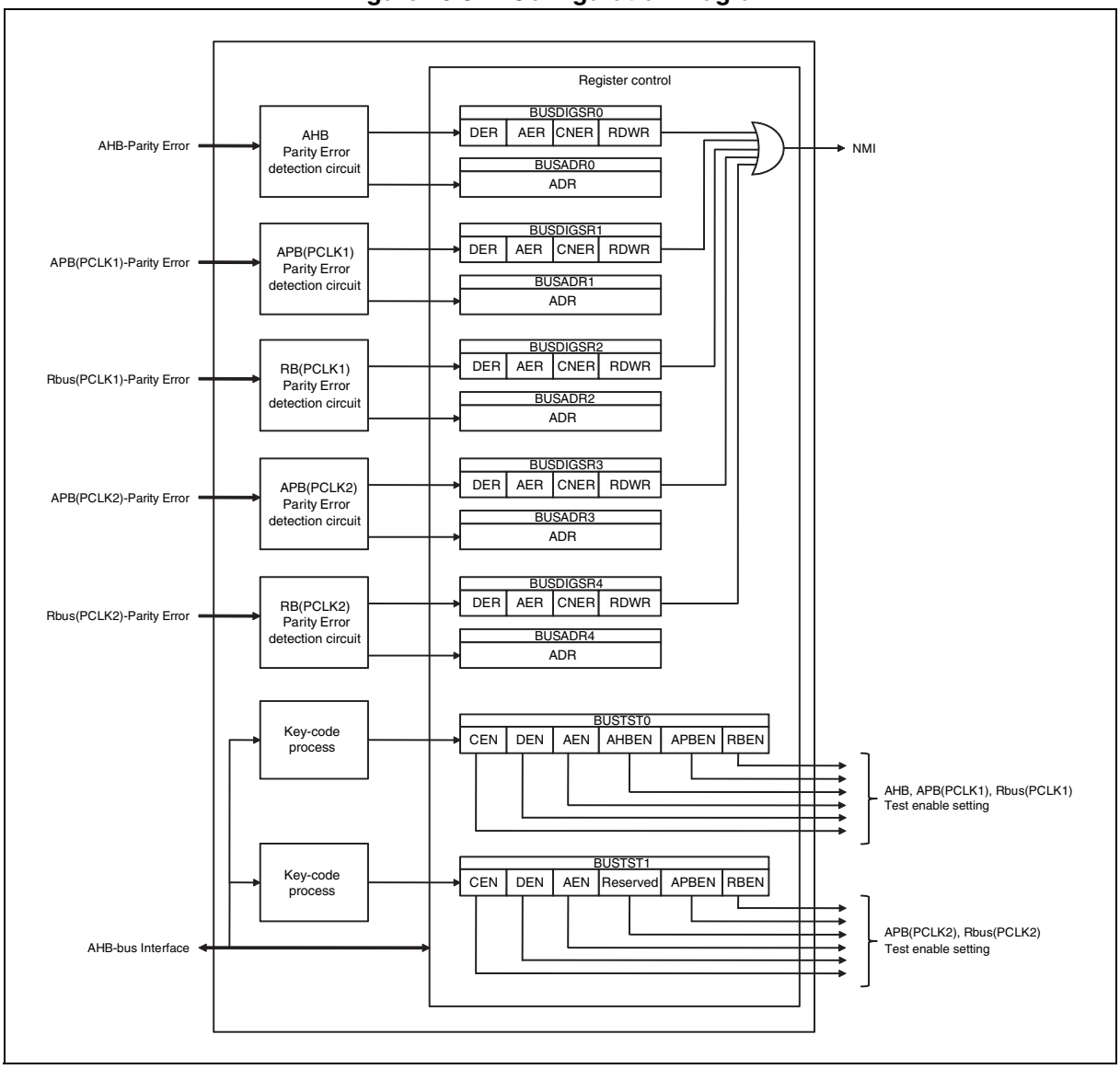


Figure 19.3-2 Bus diagnosis target area diagram (MB91F583AM/F584AM/F585AM)

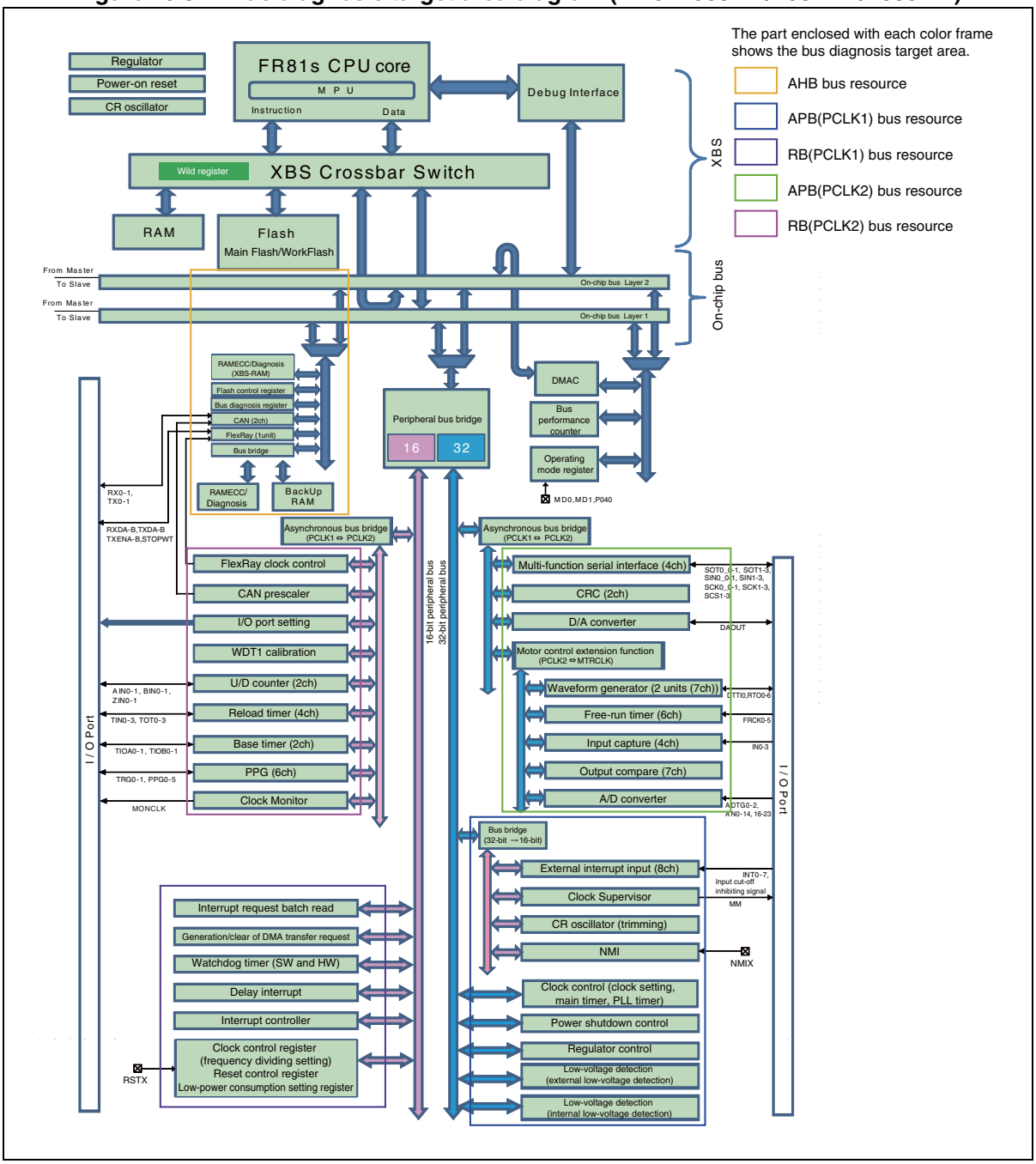
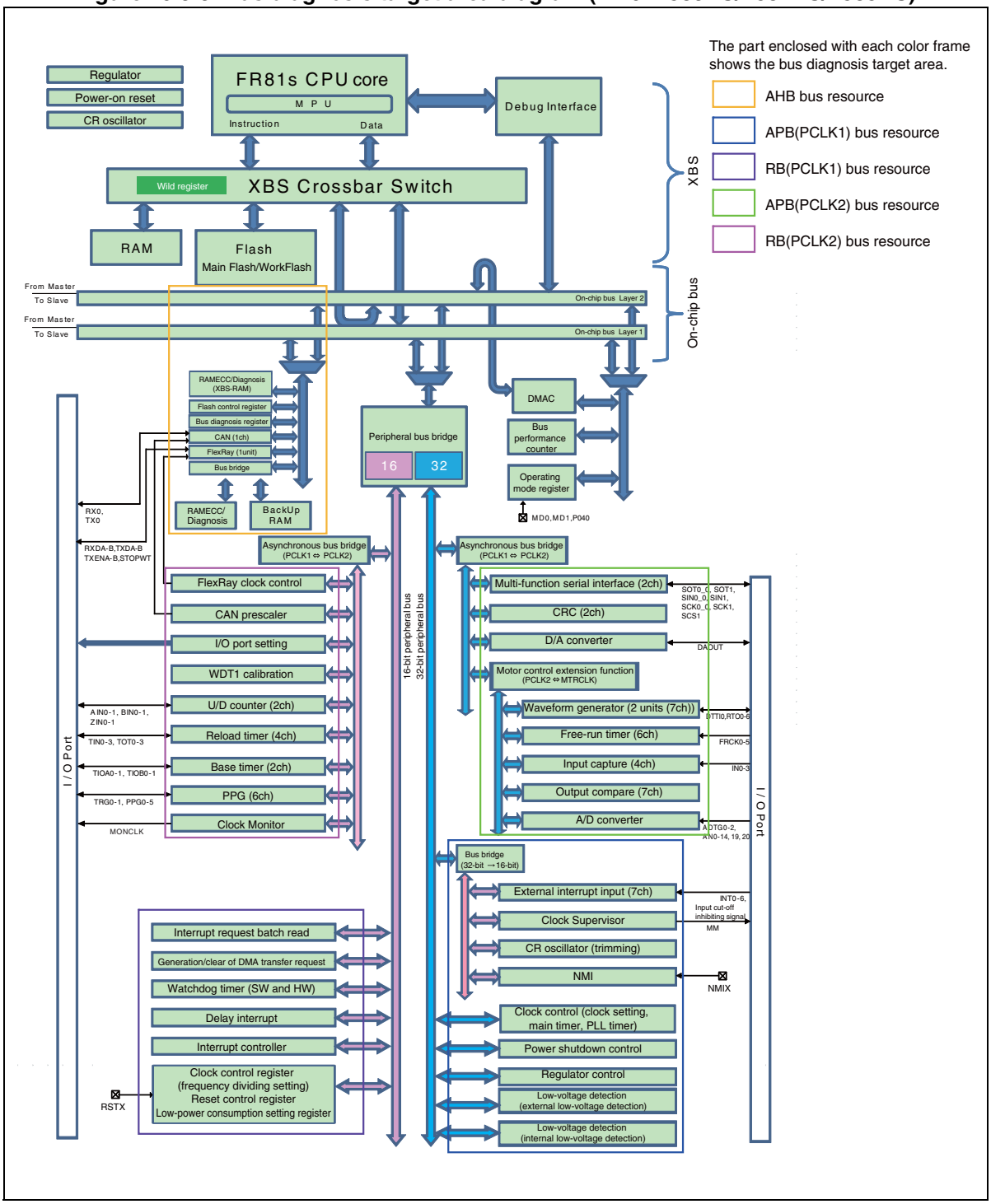


Figure 19.3-3 Bus diagnosis target area diagram (MB91F583AS/F584AS/F585AS)



<Notes>

The following address areas are outside the diagnosis area.

Watchdog timer	0x0000003C - 0x0000003F *
Delay interrupt	0x00000044 *
MPU	0x00000310 - 0x000003AC
Interrupt controller	0x00000440 - 0x0000046F *
Reset control/Low-power consumption control	0x00000480 - 0x00000482 *
Clock control	0x00000488 - 0x0000048A *
DMA transfer request by peripheral	0x00000490 - 0x00000497 *
Bus performance counter	0x00000710 - 0x0000071F
Flash memory register	0x00000840 - 0x00000843
Wild register	0x0000085A - 0x0000085B, 0x00000880 - 0x000008FF
OCDU	0x00000BFC - 0x00000BFF
Synchronous/Asynchronous setting	0x00001000 - 0x00001001
DMA controller	0x00000C00 - 0x00000DFF
Bus diagnosis	0x00003100 - 0x00003127 (Read data is a diagnosis target.)

*: Area outside the diagnosis area connected with R-bus, the error detection might be occurred according to the register setting condition and the trouble condition.
For detail information, see "19.5.3 Notes".

19.4 Registers

Table 19.4-1 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
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

19.4.1 Bus Diagnosis Status Register (BUSDIGSR)

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)**

15	14	13	12	11	10	9	8	bit
DER[3]	DER[2]	DER[1]	DER[0]	AER[3]	AER[2]	AER[1]	AER[0]	
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	Attributes

7	6	5	4	3	2	1	0	bit
PECLR	Reserved					CNER	RDWR	
0	0	0	0	0	0	0	0	Initial values
R0/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	Attributes

[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 one of the AER bits is "1", or the CNER bit is "1".
- Data parity error detects an error for valid data of access size and notifies it.
- 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 set these bits to "0".

[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>

- 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.
- If a diagnosis error occurs during writing to bus diagnosis status register, writing continues and the target error flag is not set to "1".

19.4.2 Bus Diagnosis Test Register (BUSTSTR0/1)

The bus diagnosis test register (BUSTSTR0/BUSTSTR1) sets the bus diagnosis test function.

- **BUSTSTR0: Address 3106_H (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
KEY1	KEY0	Reserved		CEN	RBEN	APBEN	AHBEN	
0	0	0	0	0	0	0	0	Initial values
R0/W	R0/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[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] Reserved

Always set these bits to "0".

[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 and the same operation as when "0" is set in this bit occurs.

[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, and the same operation as when "0" is set in this bit occurs.

[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, and the same operation

as when "0" is set in this bit occurs.

[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, and the same operation as when "0" is set in this bit occurs.

[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 and the same operation as when "0000" is set in these bits occurs.
- 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 and the same operation as when "0000" is set in these bits occurs.

<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.
  This operation can be set only when the test function is used.

- **BUSTSTR1: Address 311A$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
KEY1	KEY0	Reserved		CEN	RBEN	APBEN	Reserved	
0	0	0	0	0	0	0	0	Initial values
R0/W	R0/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W0	Attributes

7	6	5	4	3	2	1	0	bit
DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[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] Reserved

Always set these bits to "0".

[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 and the same operation as when "0" is set in this bit occurs.

[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, and the same operation as when "0" is set in this bit occurs.

[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, and the same operation as when "0" is set in this bit occurs.

[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 and the same operation as when "0000" is set in these bits occurs.
- 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 and the same operation as when "0000" is set in these bits occurs.

19.4.3 Bus Diagnosis Address Register (BUSADR)

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".

19.5 Operation

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.

19.5.1 Error detection

When the error is detected by the bus diagnosis, the direction of access address access when error is detected is maintained in each bus diagnosis address register (BUSADR), RDWR bit in bus diagnosis status register (BUSDISR).

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 DER[3:0] bit to which bus diagnosis status register (BUSDISR) responds 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 by 16bit peripheral bus becomes half word access *2 accesses. Therefore, only the half word access error that does the error detection in the beginning 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 publishing/stop

NMI keeps being published while either error of the address of each bus (AER 3:0) and control error (CNER) or data error (DER) has been detected.

When all of error of the address of each bus (AER 3:0) and control error (CNER) and data error (DER) are cleared, publishing NMI is stopped.

(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 goes runaway after NMI is generated.

19.5.2 Test function

In this function, a pseudo error can occur if bus diagnosis test register (BUSTSTR) is used.

The key code processing is necessary for being set to this register.

If "00", "01" or neither "10" nor "11" is continuously written in KEY1 and the KEY0 bits, the register is not set.

At this time, if it is not the same value as four times, the register value is not updated as for the written data.

However, even if the bus diagnosis test register is set, the pseudo data error is not detected for ALL "1" the reading data.

Note: ALL "1" read might detect the error according to the register setting and the access requirement.
Please refer to "■ Bus 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 responding 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 responding 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 respond 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 respond 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 diagnosed bus error by bus diagnosis test register (BUSTSTR).
   • In the key code, the same error setting is written 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 note in "■ Bus error setting".)

19.5.3 Notes

• When bus diagnosis test function of R-bus is used

When it accesses the following resource area in the state (RBEN was set in "1" by the bus diagnosis test register, and either AEN[3:0] and CEN or DEN[3:0] was set in "1"), the error is detected. At this time, it might seem that the bus was diagnosed.

At this time, the write access to the resource is executed.

• Watchdog timer
• Delay interrupt
• Interrupt controller
• Reset control/power consumption control
• Clock control
• DMA forwarding demand by peripheral
• DMA controller

19.5.4 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

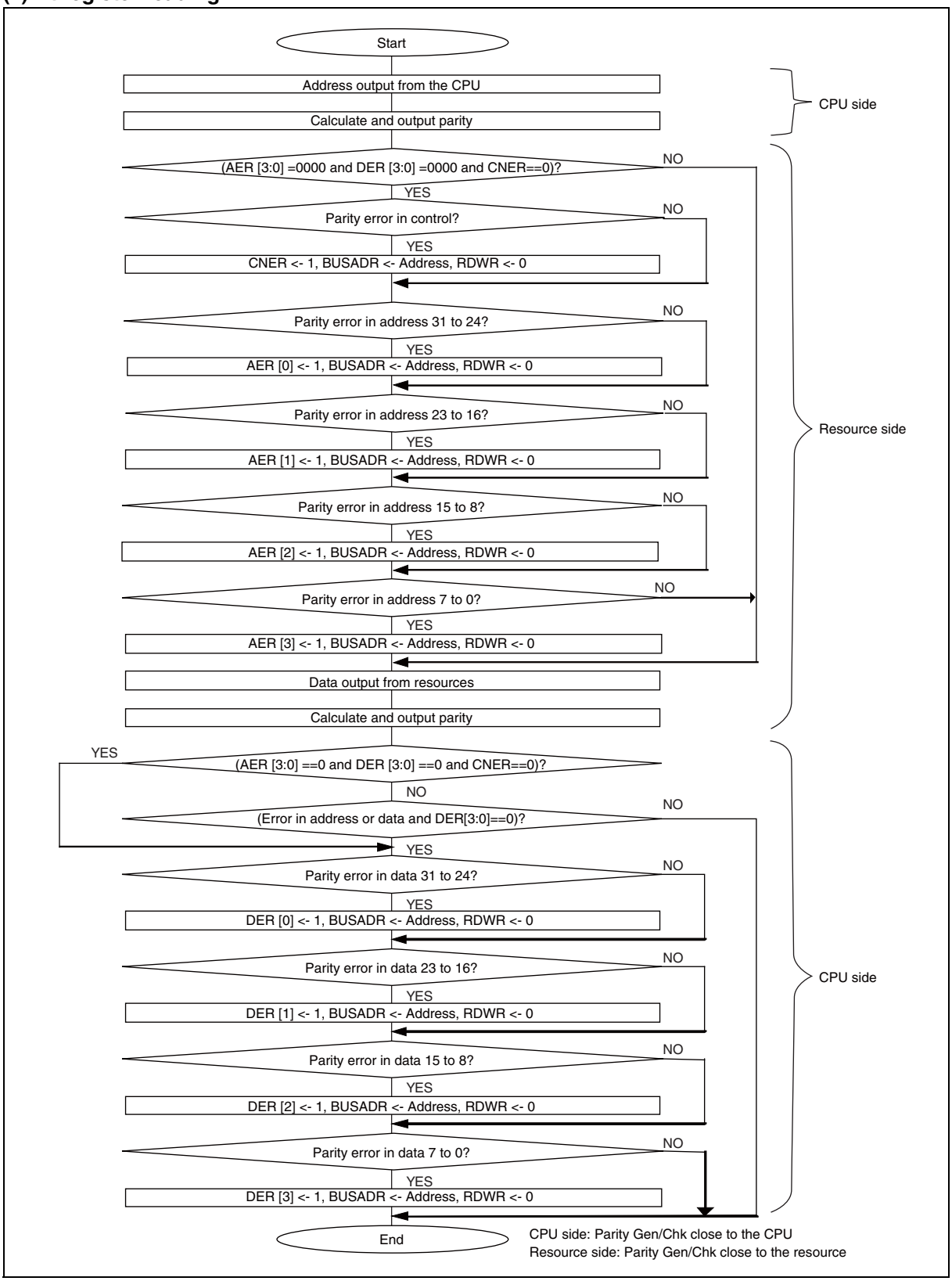
When you want to make the error detected in bus diagnosis status register DER[0] in the test function

⇒ Set "1" to bus error setting and data error setting DEN[0] by the key code access, and do byte access [31:24].
When DEN[3:1] is set, and the byte is accessed [31:24], the error might be detected in bus diagnosis status register DER[3:1].

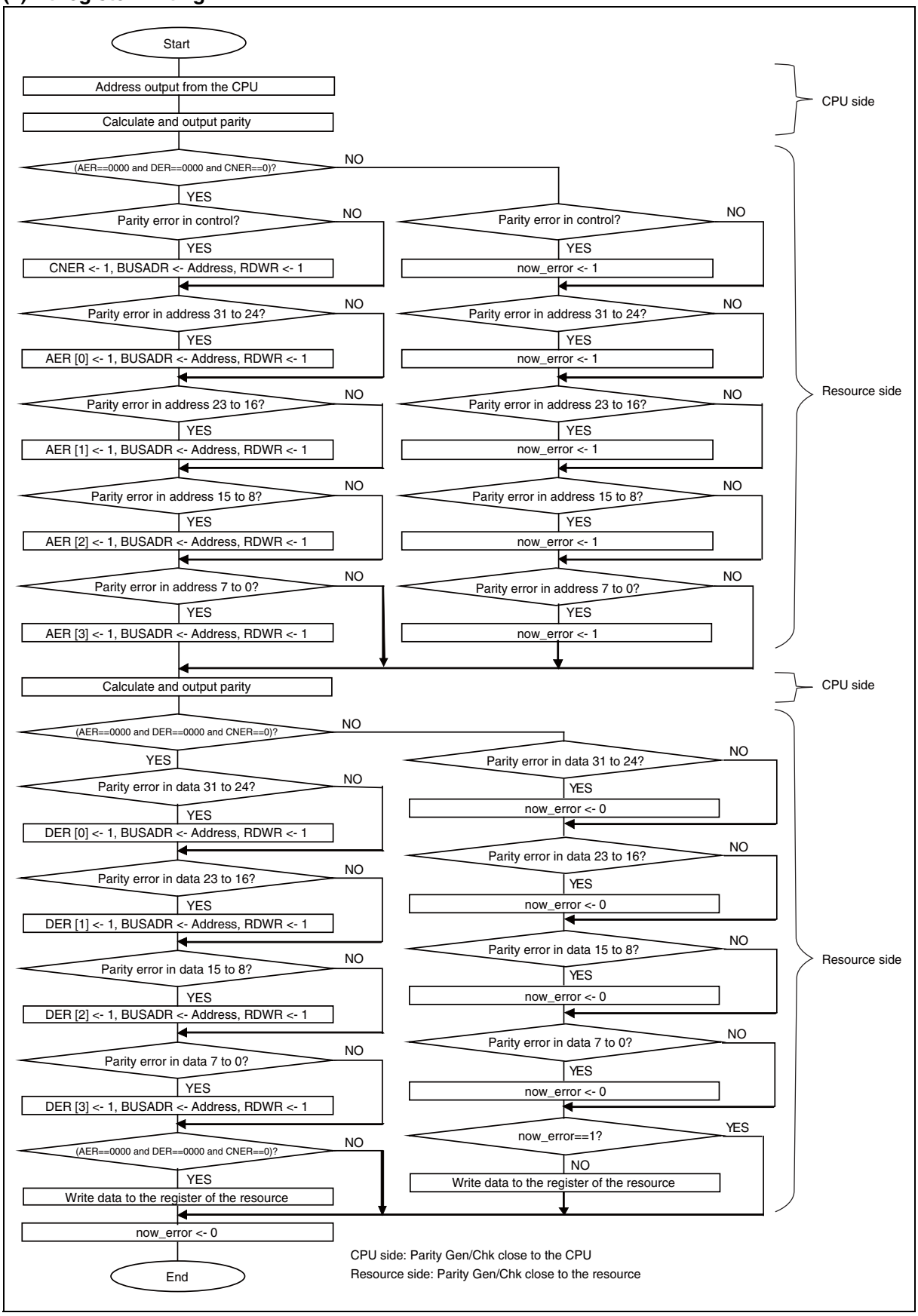
■ Bus Diagnosis Operation Flow

The operation flow of the bus diagnosis is shown below.

(1) At register reading



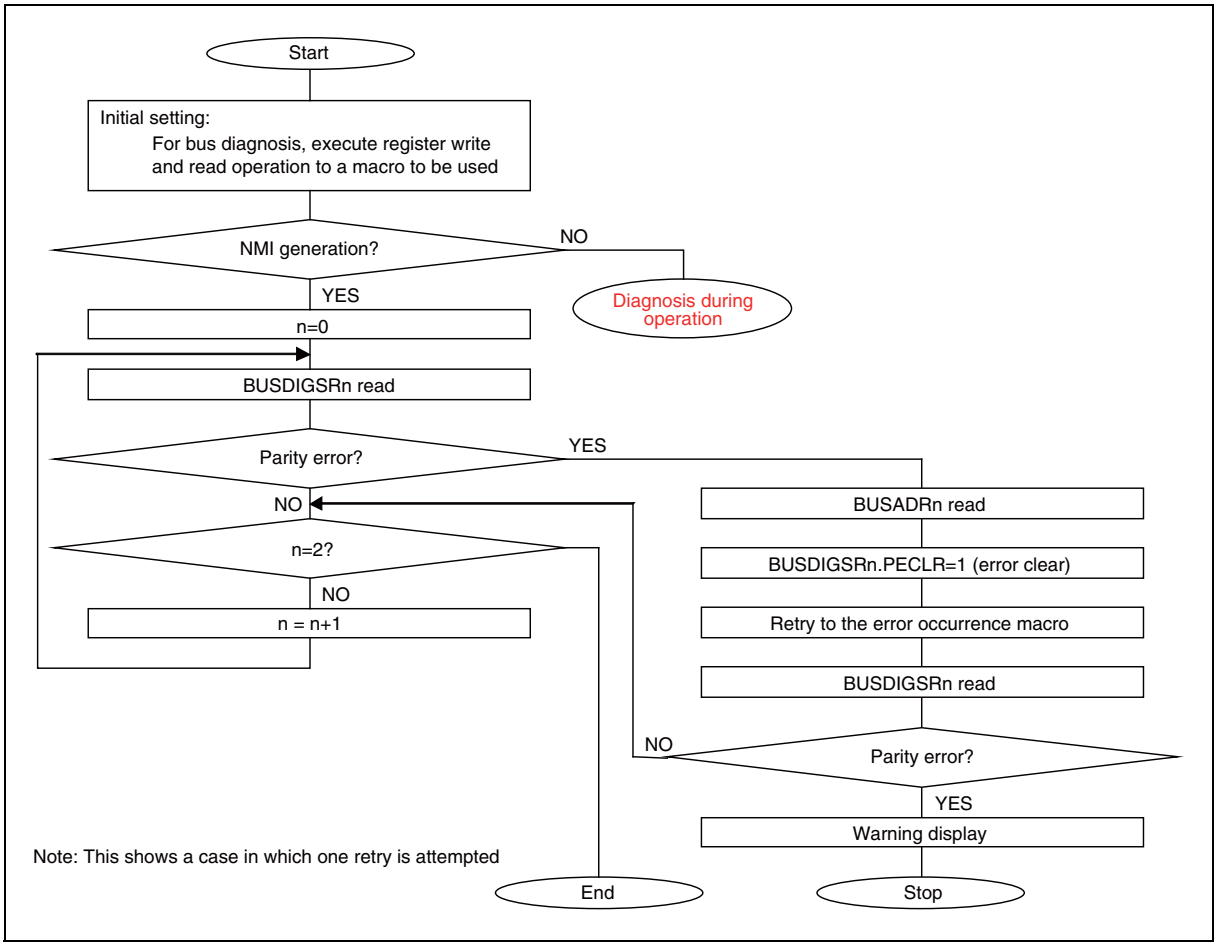
(2) At register writing



(3) Processing at error detection

The following gives an example of processing at error detection.

■ At Initialization

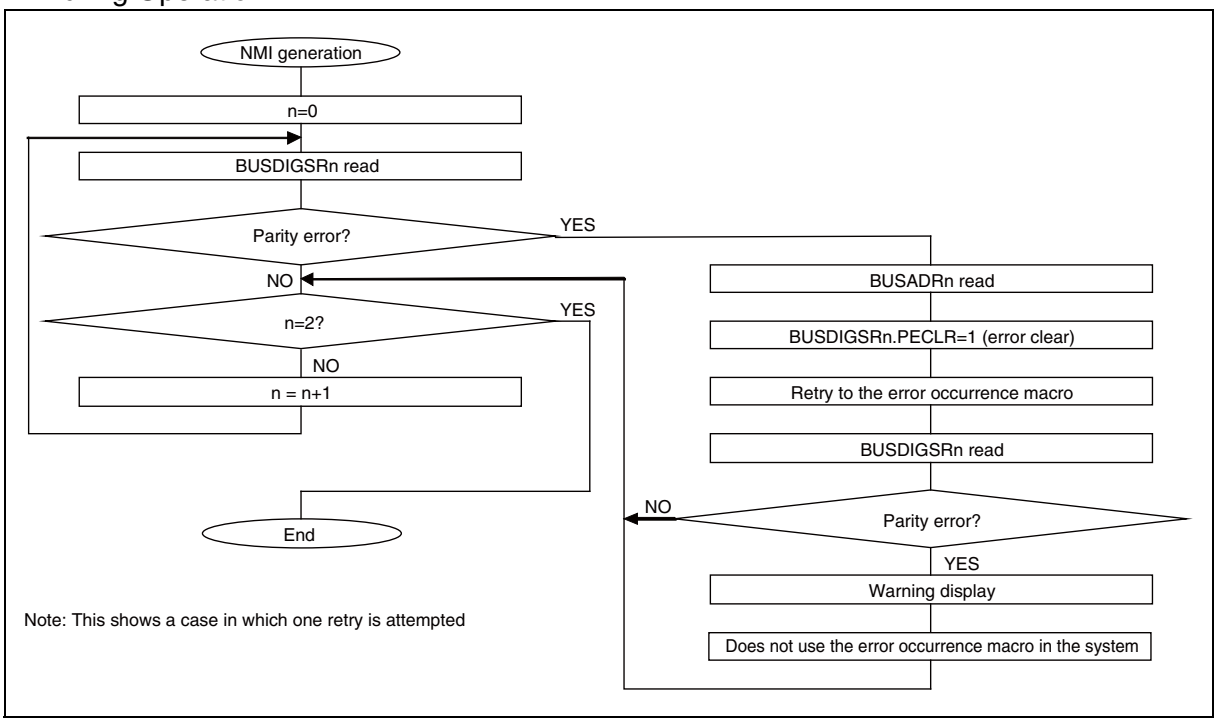


(Note) Before setting the ILM interrupt level mask register to 16 (10000_B) or more, set the address of the interrupt function to the interrupt vector.
If the program is executed in following steps while the bus signal continuously malfunctions, CPU refers to the interrupt function address immediately after the ILM setting.

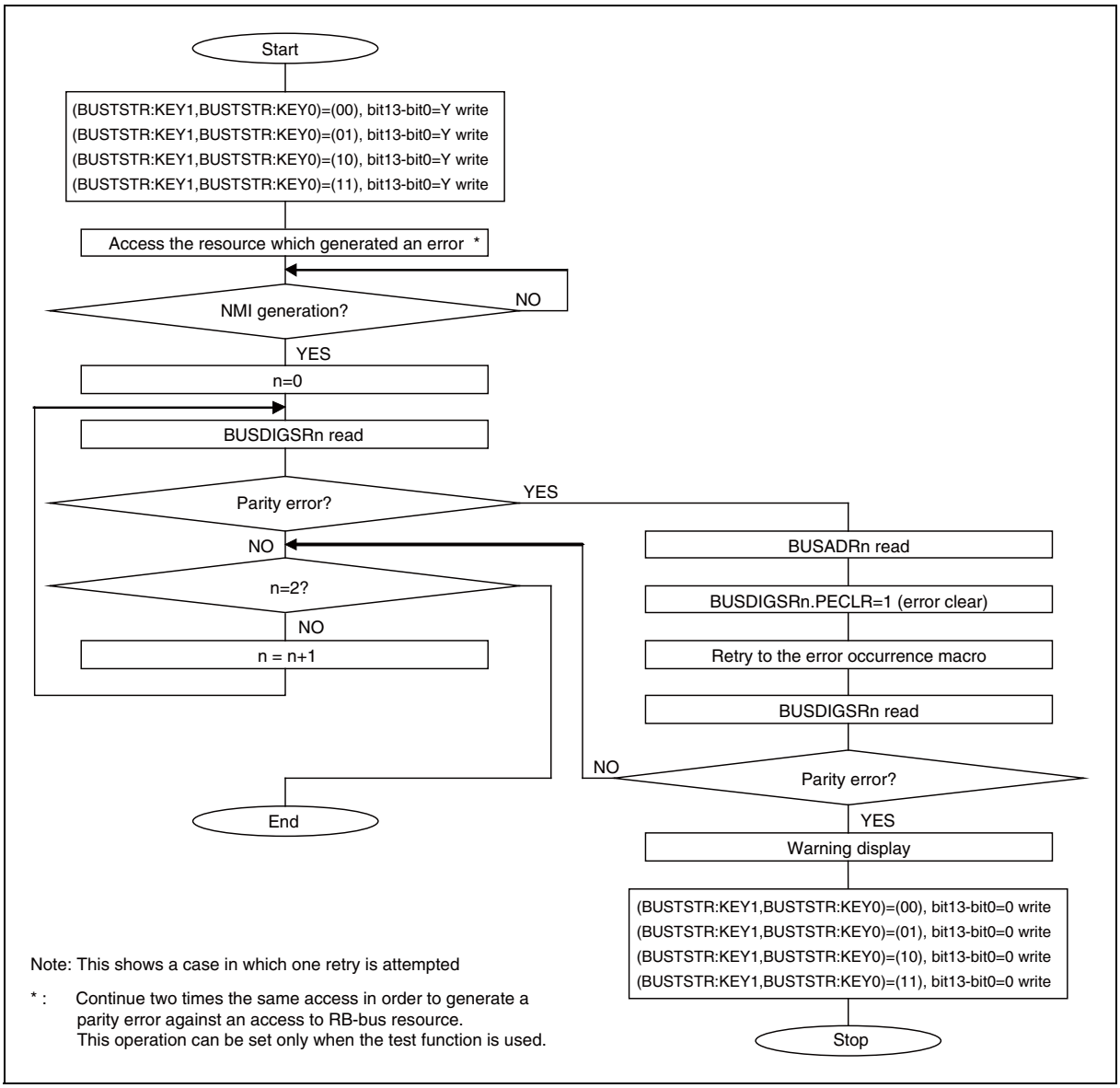
Example:

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 20 DELAYED INTERRUPT

20.1 Overview

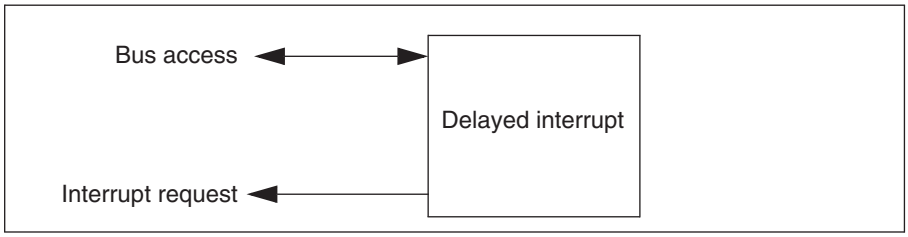
Delayed interrupt is a function for generating interrupts for the OS to switch between tasks.
This function allows interrupt requests to the CPU to be generated and cancelled by software.

20.2 Features

The delayed interrupt can be generated by writing to a register.

20.3 Configuration

Figure 20.3-1 Block Diagram of Delayed Interrupt



20.4 Registers

Table 20.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0044	DICR	Reserved	Reserved	Reserved	Delayed Interrupt Control Register

■ DICR: (Delayed Interrupt Control Register) Delayed Interrupt Control Register

This register controls the delayed interrupts.

- **DICR: Address 0044_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-							DLYI	
1	1	1	1	1	1	1	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	Attribute

[bit7 to bit1] Undefined

These bits always read "1". Writing has no effect on operation.

[bit0] DLYI (DeLaYed Interrupt enable) Delayed Interrupt Enable Bit

This bit generates and clears the delay interrupt source.

DLYI	Description
Write 0	Clears the delayed interrupt source
Write 1	Generates the delayed interrupt source

Code: FR81S09_DINT-1v0-91580MS-1-E

20.5 Operation Description

Delayed 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

Delayed interrupts are allocated to the interrupt sources with the highest interrupt vector number.

In this core, delayed interrupts are allocated to interrupt vector number 63 (0x3F).

■ DLYI Bit of the DICR Register

Writing "1" to this bit generates a delayed interrupt source. Writing "0" to this bit cancels the delayed interrupt source.

This bit functions like a standard interrupt source flag of other interrupt and should be cleared in the interrupt routine at the same time as when switching a task.

20.6 Restrictions

Do not use delayed interrupts in DMA transfer requests.

CHAPTER 21 EXTERNAL INTERRUPT INPUT

21.1 Overview

Interrupt request inputs from external interrupt input pins (INT0 to INT7)

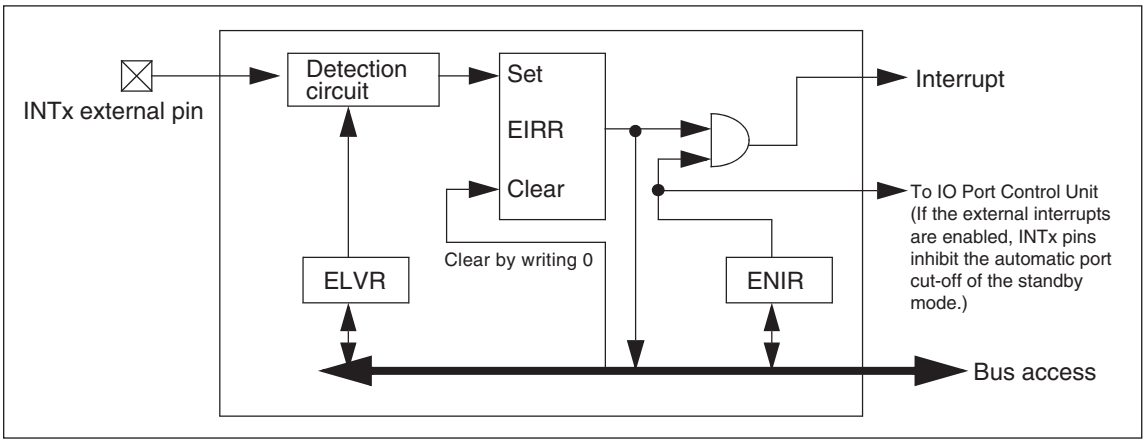
The external interrupt input pins for the models MB91F583AS/F584AS/F585AS are INT0 to INT6.

21.2 Features

- MB91F583AM/F584AM/F585AM: 8 systems external interrupt input pins (INT0 to INT7)
  MB91F583AS/F584AS/F585AS: 7 systems external interrupt input pins (INT0 to INT6)
- Interrupt detection factors: Four types: ("L" level, "H" level, rising edge, falling edge)

21.3 Configuration

Figure 21.3-1 Block Diagram of External Interrupt



21.4 Registers

Table 21.4-1 Register Map

Address	Register				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

Code: BG04-1v0-91580MS-1-E

■ EIRR0: (External Interrupt Request Register 0) External Interrupt Factor Register 0

This register holds information that an external interrupt factor has been generated.

- **EIRR0: Address 0550_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	
X	X	X	X	X	X	X	X	Initial value
R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	Attribute

[bit7 to bit0] ER7 to ER0 (External interrupt Request 7 to 0) External interrupt request bits

A flag to indicate that there is an interrupt request by input of external pin INT. Writing 0 will clear it.

ERx	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.
- Writing "1" to these bits is disabled.
- The values read with read-modify-write 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 interrupt request level register.
- The value after resetting this register depends on the pin state after the reset.
- This register is initialized by any reset factor except a recovery reset from the standby (power shutdown) at PMUCTLR. IOCTMD = 1.
- In models MB91F583AS/F584AS/F585AS, ER7 is a reserved bit. Always write "0" to this bit.

■ ENIR0: (ENable Interrupt request Register 0) External Interrupt Enable Register 0

This register enables external interrupt inputs.

- **ENIR0: Address 0551_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	
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 to bit0] EN7 to EN0 (interrupt ENable) External interrupt enable bits

These bits perform mask controls of interrupt requests from external pin INT inputs.

ENx	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.
- This register is initialized by any reset factor except a recovery reset from the standby (power shutdown) at PMUCTLR. IOCTMD = 1.
- In models MB91F583AS/F584AS/F585AS, EN7 is a reserved bit. Always write "0" to this bit.

■ ELVR0: (External Interrupt LeVel Register 0) External Interrupt Request Level Register 0

This register selects detection conditions for external interrupt requests.

- **ELVR0: Address 0552_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	
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
LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	
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 bit1] LB7 to LB0 (Level select B) Level selection B

[bit14 to bit0] LA7 to LA0 (Level select A) Level selection A

These bits select detection conditions for external interrupt requests. Combination of 2 bits, LA bit and LB bit will be used.

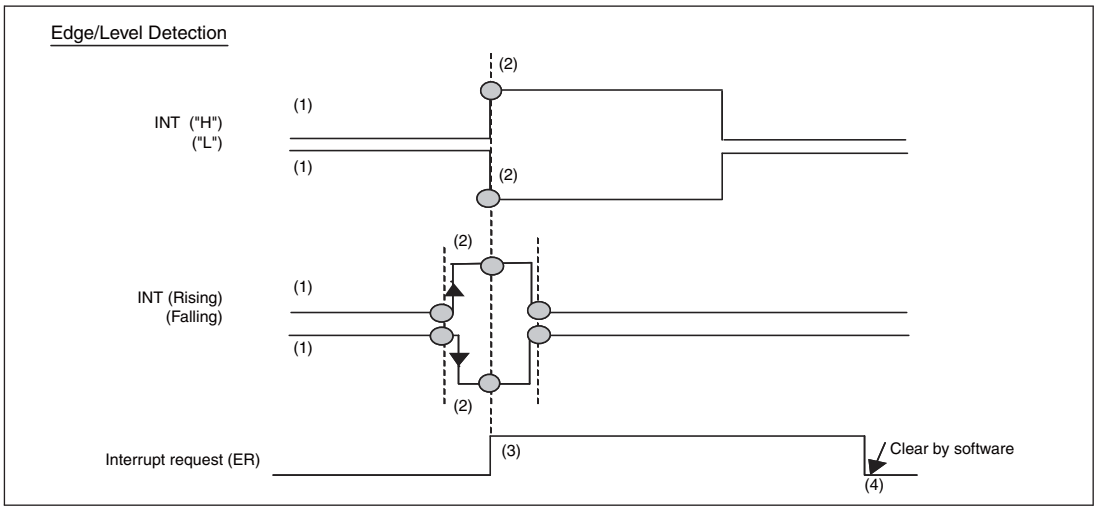
LBx	LAx	Detection conditions
0	0	L level detection (Initial value)
0	1	H level detection
1	0	Rising edge detection
1	1	Falling edge detection

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.
- The factor bit in the interrupt factor register may be set by changing interrupt request level register. Initialize the interrupt factor register after changing interrupt request level register.
- This register is initialized by any reset factor except a recovery reset from the standby (power shutdown) at PMUCTLR. IOCTMD = 1.
- In models MB91F583AS/F584AS/F585AS, LA7 and LB7 are reserved bits. Always write "0" to these bits.

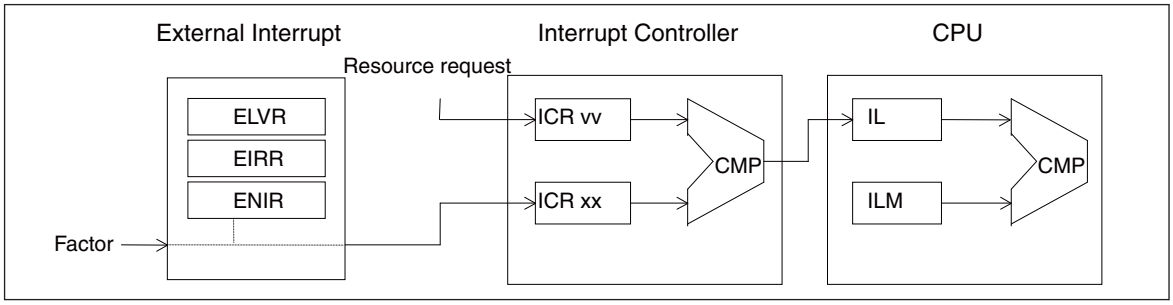
21.5 Operation

Figure 21.5-1 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 21.5-2 Operation of External Interrupt



(1)Operation of external interrupt

This module generates 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. For the enabled channel, the standby mode automatic input/output shutdown feature to the external pin will also be suppressed. See "CHAPTER 9 POWER CONSUMPTION CONTROL" for the automatic input/output shutdown feature.

(3)Setting procedure of external interrupts

When setting registers which reside in the external interrupt unit, follow the steps below.

1. Disable the corresponding bit for the enable register.
2. Set the corresponding bit for the request level setting register.
3. Read the request level register.

4. Clear the corresponding bit for factor register.
5. Enable the corresponding bit for 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 set the registers in this module. The factor register must be cleared before you set the enable register to enable state.

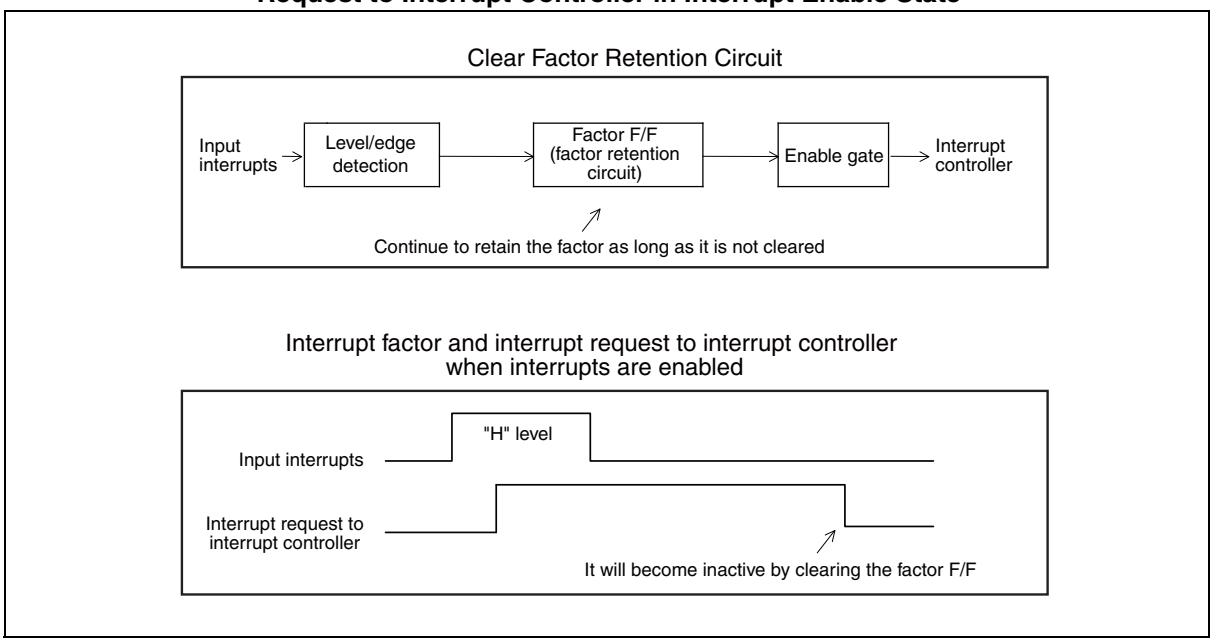
This has to be done to avoid generating erroneous interrupt factors at the time of setting register or in interrupt enable state.

(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 21.5-3 Clearing the Number of Factors Being Held and Interrupt Factor and Interrupt Request to Interrupt Controller in Interrupt Enable State



21.6 Settings

Table 21.6-1 Necessary Settings for Using External Interrupts

Settings	Setting register	Setting method
Detection level settings	External interrupt request level setting register (ELVR0)	See Section "7.1 Overview".
Make external pins to use for input.	See "CHAPTER 24 I/O PORTS".	See "CHAPTER 24 I/O PORTS".
External interrupt	An input from the external pin → Inputs signals at pins INT0 to INT7	-

21.7 Q&A

21.7.1 About Detection Levels and Their Setting Procedures

Four levels: ("L" level, "H" level, rising edge, falling edge)

Set the detection level bits as follows: (ELVR0. LBx, LAx) (x=0 to 7)

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".

21.7.2 How to Make External Pins to Use for Input

See "CHAPTER 24 I/O PORTS".

21.7.3 About Interrupt Related Registers

See "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

21.7.4 About Interrupt Types

Interrupt factors are only for external interrupts. There are no select bits.

21.7.5 How to Enable/Disable/Clear Interrupts

Interrupt request enable flag, interrupt request flag

Interrupt enable setting is done by the interrupt enable bit (ENIR0.EN0 to 7).

Operation	Interrupt enable bit (ENx)
To disable interrupt requests,	Set "0".
To enable interrupt requests,	Set "1".

Interrupt request clear is done by the interrupt request bit (EIRR0.ER0 to 7).

Operation	Interrupt request bit (ERx)
To clear interrupt requests,	Write "0".

21.7.6 How to Use CAN WAKEUP function

See "CHAPTER 36 CAN".

21.8 Notes

The external interrupt input register is not initialized by a recovery from the standby watch mode (power shutdown) at PMUCTLR.IOCTMD = 1 or from the standby stop mode (power shutdown). To maintain the state before and during the recovery, set PMUCTLR.IOCTMD in advance to set the IO retaining state. Then, after setting the IO port, set PMUCTLR.IOCT to release the IO retaining state. For details on the PMUCTLR register, see "CHAPTER 9 POWER CONSUMPTION CONTROL". Also, during a recovery from the standby watch mode (power shutdown) at PMUCTLR.IOCTMD = 1 or from the standby stop mode (power shutdown), an internal reset is issued. Therefore, it does not accept any reset factor except a power-on reset, internal low-voltage reset, and simultaneous asserts of RSTX and NMIX. At this time, the external interrupt input register is not initialized. Therefore, if a reset input from the RSTX pin input or an external low-voltage detection flag is set after start-up, initialize the external interrupt input register before using it.

CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)

22.1 Overview

The interrupt controller performs arbitration of interrupt requests.

22.2 Features

This module is composed of the following parts.

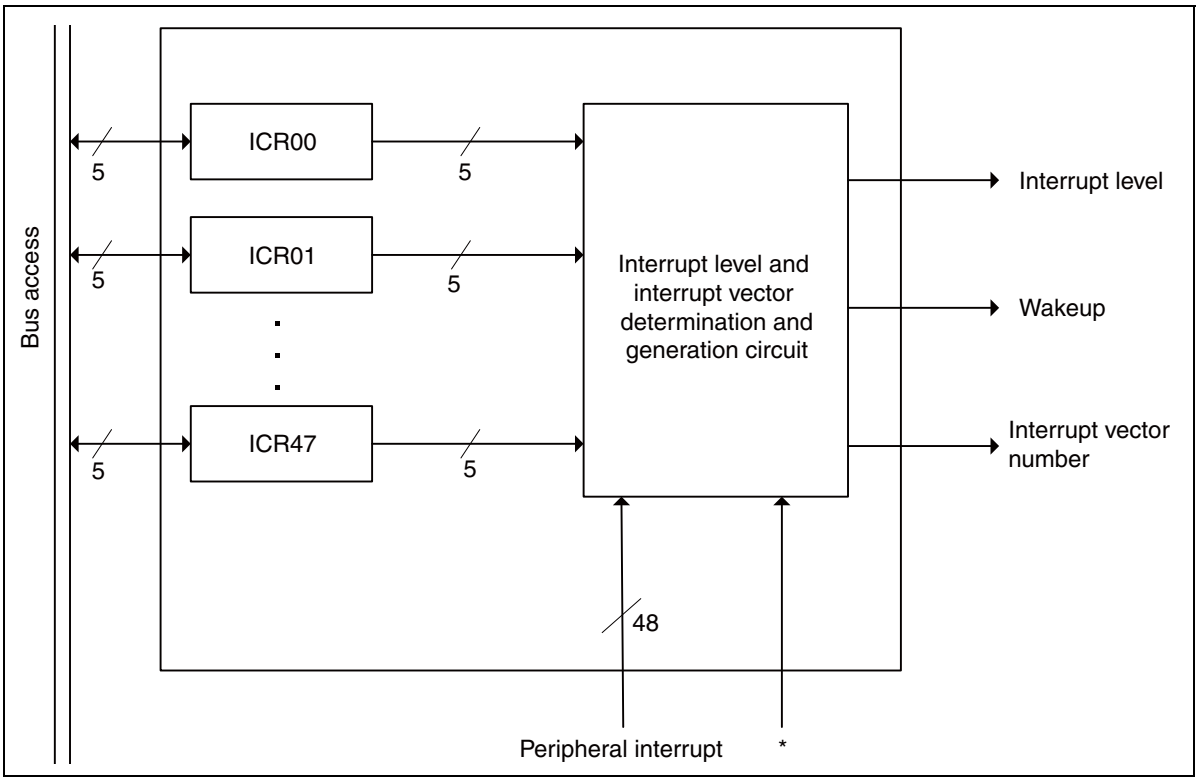
- ICR register
- Interrupt priority determination circuit
- Interrupt level and interrupt vector generation circuit

This module has the following functions.

- Detecting NMI requests and peripheral interrupt requests
- Priority determination (by level and interrupt vector)
- Transmitting the interrupt level of the source with the highest priority to the CPU
- Transmitting the interrupt vector number of the source with the highest priority to the CPU
- Generating wakeup requests by NMI / interrupts that occur with a level other than "11111"

22.3 Configuration

Figure 22.3-1 Block Diagram of Interrupt Control



* NMI, error during internal bus diagnosis, XBS RAM double-bit error, or Backup RAM double-bit error

Code: INTCNT-1v0-91580MS-1-E

22.4 Registers

Table 22.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0440	ICR00	ICR01	ICR02	ICR03	Interrupt controller 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	

22.4.1 ICR00 to ICR47 (Interrupt Control Register 00 to 47) Interrupt Controller Registers 00 to 47

One register is provided for each interrupt input to set the level for the corresponding interrupt request.

- **ICR00 to 47: Address 0x0440 to 0x046F$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
reserved			IL[4:0]					
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1/WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit5] Undefined

These bits always read "1". Writing has no effect on operation.

[bit4 to bit0] IL[4:0] (Interrupt Level control) Interrupt Level Control

The interrupt level setting bits specify the interrupt level for the corresponding interrupt request. An interrupt request is masked in the CPU if the interrupt level set in these registers is greater than or equal to the level mask value in the ILM register of the CPU. These bits are initialized to "5'b11111" on reset. The correspondence between the configurable interrupt level settings bits and the interrupt levels is shown below.

IL[4:0]	Interrupt level	
10000	16	Configurable highest level (High)
10001	17	
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.

22.5 Operation Description

22.5.1 Setup

(1) Configure the ICR register of the interrupt vector number corresponding to the peripheral for which you want to generate the interrupt.

(2) Configure the peripheral where you want to generate the interrupt. (Configure interrupt output as enabled on the peripheral.)

22.5.2 Starting

Start the configured peripheral.

22.5.3 Determining Priorities

This module selects the highest priority interrupt among interrupt sources that occur simultaneously and outputs the interrupt level and interrupt vector number for the interrupt source to the CPU.

The criteria for determining the priority of interrupt sources are as follows.

(1) NMI

(2) Sources that meet the following conditions

- If the numeric value of the interrupt level is not 31 (5’b11111). (31 indicates interrupts disabled)
- The source where the numeric value of the interrupt level is the smallest.
- When the interrupt level is the same (except for 31), the source that has the smallest interrupt vector number from amongst these.

If no interrupt source is selected by the above criteria, 31 (5’b11111) is output as the interrupt level. The interrupt vector number at this time is undefined.

22.5.4 Recovering From Stop Mode

The function for using an interrupt request to recover from stop mode is performed by this module. If an interrupt request (the interrupt level is anything other than "5’b11111") is generated from a peripheral (including NMI), a request is generated to the clock control unit to recover from stop mode.

As the interrupt priority judgment unit restarts operation once the clock supply starts after recovery from stop mode, the CPU is able to execute instructions until the interrupt priority judgment unit produces a result.

For interrupts that are not used as sources for recovering from stop mode, set the interrupt level of the corresponding interrupt control registers (ICR00 to 47) to "5’b11111" (interrupts disabled).

22.5.5 Recovering From Standby Mode (Power Shutdown)

When the interrupt level is higher than ICR = 0x1F (interrupt disabled) and an interrupt factor has been generated, the standby recovery factor is valid. Therefore, the CPU cannot transit to the power shutdown state. It continues executing the instruction.

If the interrupt level is ICR = 0x1F (interrupt disabled) and an interrupt factor has been generated, a standby recovery factor does not apply. In such case, the CPU transits to the power shutdown state once, but because it is in a state in which there is a power shutdown recovery factor, it is recovered immediately in the power shutdown recovery sequence. (This process is executed from the reset vector.)

CHAPTER 23 INTERRUPT REQUEST BATCH READ

23.1 Overview

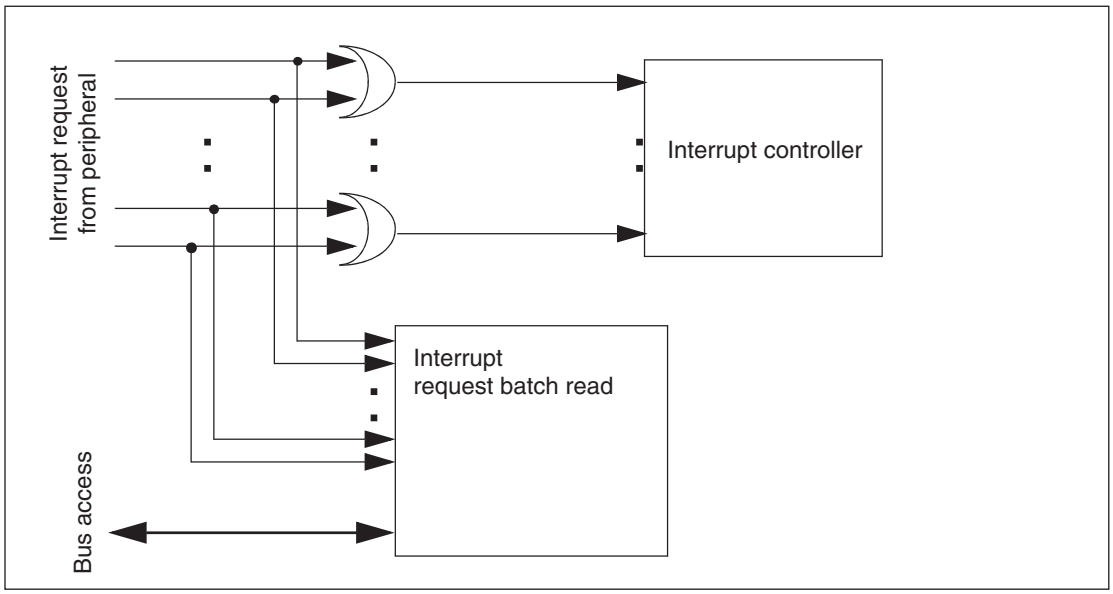
This module can read multiple interrupt requests assigned to one interrupt vector number in a batch. Interrupt requests that have been generated can be identified by using the bit search instruction of the FR80-family CPU.

23.2 Features

Using this module, you can easily check whether interrupts have been generated.

23.3 Configuration

Figure 23.3-1 Block Diagram of Interrupt Request Batch Read



Code: BIRPR-0v4-91580MS-1-E

23.4 Registers

Table 23.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x0120	IRPR0H	IRPR0L	IRPR1H	IRPR1L	Interrupt request batch read register 0 high-order (#17) Interrupt request batch read register 0 low-order (#18) Interrupt request batch read register 1 high-order (#19) Interrupt request batch read register 1 low-order
0x0124	IRPR2H	IRPR2L	IRPR3H	IRPR3L	Interrupt request batch read register 2 high-order Interrupt request batch read register 2 low-order (#36) Interrupt request batch read register 3 high-order (#38) Interrupt request batch read register 3 low-order (#39)
0x0128	IRPR4H	IRPR4L	IRPR5H	IRPR5L	Interrupt request batch read register 4 high-order (#40) Interrupt request batch read register 4 low-order (#41) Interrupt request batch read register 5 high-order (#42) Interrupt request batch read register 5 low-order (#43)
0x012C	IRPR6H	IRPR6L	IRPR7H	IRPR7L	Interrupt request batch read register 6 high-order (#44) Interrupt request batch read register 6 low-order (#45) Interrupt request batch read register 7 high-order (#46) Interrupt request batch read register 7 low-order (#47)
0x0130	IRPR8H	IRPR8L	IRPR9H	IRPR9L	Interrupt request batch read register 8 high-order (#48) Interrupt request batch read register 8 low-order (#49) Interrupt request batch read register 9 high-order (#50) Interrupt request batch read register 9 low-order (#51)
0x0134	IRPR10H	IRPR10L	IRPR11H	IRPR11L	Interrupt request batch read register 10 high-order (#52) Interrupt request batch read register 10 low-order (#53) Interrupt request batch read register 11 high-order (#54) Interrupt request batch read register 11 low-order (#55)
0x0138	IRPR12H	IRPR12L	IRPR13H	IRPR13L	Interrupt request batch read register 12 high-order (#56) Interrupt request batch read register 12 low-order (#57) Interrupt request batch read register 13 high-order (#58) Interrupt request batch read register 13 low-order (#59)
0x013C	IRPR14H	IRPR14L	IRPR15H	IRPR15L	Interrupt request batch read register 14 high-order (#60) Interrupt request batch read register 14 low-order (#61) Interrupt request batch read register 15 high-order (#62) Interrupt request batch read register 15 low-order (#15)
0x0140	IRPR16H	IRPR16L	IRPR17H	IRPR17L	Interrupt request batch read register 16 high-order (#21) Interrupt request batch read register 16 low-order (#23) Interrupt request batch read register 17 high-order (#25) Interrupt request batch read register 17 low-order (#27)
0x0144	IRPR18H	IRPR18L	-	-	Interrupt request batch read register 18 high-order (#31) Interrupt request batch read register 18 low-order (#35) - -

#nn : Interrupt vector number (decimal)

23.4.1 IRPR0H (Interrupt Request Peripheral Read register 0H) Interrupt request batch read register 0 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #17)

- **IRPR0H: Address 0120_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RTIR0	RTIR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[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.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.2 IRPR0L (Interrupt Request Peripheral Read register 0L) Interrupt request batch read register 0 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #18)

- **IRPR0L: Address 0121_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RTIR2	RTIR3	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[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.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.3 IRPR1H (Interrupt Request Peripheral Read register 1H) Interrupt request batch read register 1 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #19)

- **IRPR1H: Address 0122_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RXIR0	ISIR0	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] RXIR0 (Multi-Function-Serial-Interface RX Interrupt Request 0) Multi-function serial interface ch.0 reception completion interrupt request

[bit6] ISIR0 (Multi-Function-Serial-Interface Status Interrupt Request 0) Multi-function serial interface ch.0 status interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.4 IRPR1L (Interrupt Request Peripheral Read register 1L) Interrupt request batch read register 1 low-order

- **IRPR1L: Address 0123_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.5 IRPR2H (Interrupt Request Peripheral Read register 2H) Interrupt request batch read register 2 high-order

- **IRPR2H: Address 0124_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.6 IRPR2L (Interrupt Request Peripheral Read register 2L) Interrupt request batch read register 2 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #36)

- **IRPR2L: Address 0125_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
MTIR	PTIR	GEAR_IRQ	OVF_IRQ	-				
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] MTIR (Main Timer Interrupt Request) Main timer interrupt request

[bit6] PTIR (PLL Timer Interrupt Request) PLL timer interrupt request

[bit5] GEAR_IRQ (PLL Gear Interrupt Request) FlexRay PLL gear interrupt request

[bit4] OVF_IRQ (Over Flow Interrupt Request) FlexRay PLL alarm interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

- **[bit3 to bit0] Undefined**

These bits always read "0". Writing has no effect on operation.

(Note) In models without the FlexRay function (MB91F583AMH/F584AMH/F585AMH/F583AMK/F584AMK/F585AMK/F583ASH/F584ASH/F585ASH/F583ASK/F584ASK/F585ASK), bit5:GEAR_IRQ and bit4:OVF_IRQ are reserved bits. These bits always read "0". Writing has no effect on operation.

23.4.7 IRPR3H (Interrupt Request Peripheral Read register 3H) Interrupt request batch read register 3 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #38)

- **IRPR3H: Address 0126_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
UDIR0	UDIR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

bit7:UDIR0 (Up/Down Counter Interrupt Request 0) U/D counter interrupt request 0

bit6:UDIR1 (Up/Down Counter Interrupt Request 1) U/D counter interrupt request 1

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.8 IRPR3L (Interrupt Request Peripheral Read register 3L) Interrupt request batch read register 3 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #39)

- **IRPR3L: Address 0127_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR0	FRTMIR0	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR0 (Free Run Timer Zero Interrupt Request 0) Free-run timer 0 0 detection interrupt request

[bit6] FRTMIR0 (Free Run Timer Compare Clear Interrupt Request 0) Free-run timer 0 Compare clear interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.9 IRPR4H (Interrupt Request Peripheral Read register 4H) Interrupt request batch read register 4 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #40)

- **IRPR4H: Address 0128_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR1	FRTMIR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR1 (Free Run Timer Zero Interrupt Request 1) Free-run timer 1 0 detection interrupt request

[bit6] FRTMIR1 (Free Run Timer Compare Clear Interrupt Request 1) Free-run timer 1 Compare clear interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.10 IRPR4L (Interrupt Request Peripheral Read register 4L) Interrupt request batch read register 4 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #41)

- **IRPR4L: Address 0129_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR2	FRTMIR2	PPGIR0	PPGIR1	PPGIR2	PPGIR3	-		
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR2 (Free Run Timer Zero Interrupt Request 2) Free-run timer 2 0 detection interrupt request

[bit6] FRTMIR2 (Free Run Timer Compare Clear Interrupt Request 2) Free-run timer 2 Compare clear interrupt request

[bit5] PPGIR0 (PPG Interrupt Request 0) PPG0 interrupt request

[bit4] PPGIR1 (PPG Interrupt Request 1) PPG1 interrupt request

[bit3] PPGIR2 (PPG Interrupt Request 2) PPG2 interrupt request

[bit2] PPGIR3 (PPG Interrupt Request 3) PPG3 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit1, bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.11 IRPR5H (Interrupt Request Peripheral Read register 5H) Interrupt request batch read register 5 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #42)

- **IRPR5H: Address 012A$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR3	FRTMIR3	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR3 (Free Run Timer Zero Interrupt Request 3) Free-run timer 3 0 detection interrupt request

[bit6] FRTMIR3 (Free Run Timer Compare Clear Interrupt Request 3) Free-run timer 3 Compare clear interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.12 IRPR5L (Interrupt Request Peripheral Read register 5L) Interrupt request batch read register 5 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #43)

- **IRPR5L: Address 012B$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR4	FRTMIR4	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR4 (Free Run Timer Zero Interrupt Request 4) Free-run timer 4 0 detection interrupt request

[bit6] FRTMIR4 (Free Run Timer Compare Clear Interrupt Request 4) Free-run timer 4 Compare clear interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.13 IRPR6H (Interrupt Request Peripheral Read register 6H) Interrupt request batch read register 6 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #44)

- **IRPR6H: Address 012C$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
FRTZIR5	FRTMIR5	PPGIR4	PPGIR5	-				
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] FRTZIR5 (Free Run Timer Zero Interrupt Request 5) Free-run timer 5 0 detection interrupt request

[bit6] FRTMIR5 (Free Run Timer Compare Clear Interrupt Request 5) Free-run timer 5 Compare clear interrupt request

[bit5] PPGIR4 (PPG Interrupt Request 4) PPG4 interrupt request

[bit4] PPGIR5 (PPG Interrupt Request 5) PPG5 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit3 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.14 IRPR6L (Interrupt Request Peripheral Read register 6L) Interrupt request batch read register 6 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #45)

- **IRPR6L: Address 012D$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ICUIR0	ICUIR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] ICUIR0 (ICU Interrupt Request 0) ICU0 interrupt request

[bit6] ICUIR1 (ICU Interrupt Request 1) ICU1 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.15 IRPR7H (Interrupt Request Peripheral Read register 7H) Interrupt request batch read register 7 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #46)

- **IRPR7H: Address 012E$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ICUIR2	ICUIR3	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] ICUIR2 (ICU Interrupt Request 2) ICU2 interrupt request

[bit6] ICUIR3 (ICU Interrupt Request 3) ICU3 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.16 IRPR7L (Interrupt Request Peripheral Read register 7L) Interrupt request batch read register 7 low-order

- **IRPR7L: Address 012F$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.17 IRPR8H (Interrupt Request Peripheral Read register 8H) Interrupt request batch read register 8 high-order

- **IRPR8H: Address 0130$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.18 IRPR8L (Interrupt Request Peripheral Read register 8L) Interrupt request batch read register 8 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #49)

- **IRPR8L: Address 0131$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR0	OCUIR1	-						
0	0	0	0	0	0	0	0	Initial value

R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR0 (OCU Interrupt Request 0) OCU0 interrupt request

[bit6] OCUIR1 (OCU Interrupt Request 1) OCU1 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.19 IRPR9H (Interrupt Request Peripheral Read register 9H) Interrupt request batch read register 9 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #50)

- **IRPR9H: Address 0132_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR2	OCUIR3	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR2 (OCU Interrupt Request 2) OCU2 interrupt request

[bit6] OCUIR3 (OCU Interrupt Request 3) OCU3 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.20 IRPR9L (Interrupt Request Peripheral Read register 9L) Interrupt request batch read register 9 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #51)

- **IRPR9L: Address 0133_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR4	OCUIR5	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR4 (OCU Interrupt Request 4) OCU4 interrupt request

[bit6] OCUIR5 (OCU Interrupt Request 5) OCU5 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.21 IRPR10H (Interrupt Request Peripheral Read register 10H) Interrupt request batch read register 10 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #52)

- **IRPR10H: Address 0134_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR6	OCUIR7	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR6 (OCU Interrupt Request 6) OCU6 interrupt request

[bit6] OCUIR7 (OCU Interrupt Request 7) OCU7 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.22 IRPR10L (Interrupt Request Peripheral Read register 10L) Interrupt request batch read register 10 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #53)

- **IRPR10L: Address 0135_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR8	OCUIR9	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR8 (OCU Interrupt Request 8) OCU8 interrupt request

[bit6] OCUIR9 (OCU Interrupt Request 9) OCU9 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.23 IRPR11H (Interrupt Request Peripheral Read register 11H) Interrupt request batch read register 11 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #54)

- **IRPR11H: Address 0136_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
OCUIR10	OCUIR11	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] OCUIR10 (OCU Interrupt Request 10) OCU10 interrupt request

[bit6] OCUIR11 (OCU Interrupt Request 11) OCU11 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.24 IRPR11L (Interrupt Request Peripheral Read register 11L) Interrupt request batch read register 11 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #55)

- **IRPR11L: Address 0137_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
WGIR0	WGIR1	WGIR2	WGRIR0	WGRIR1	WGRIR2	DTTIIR0	-	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	Attribute

[bit7] WGIR0 (Wave Generator Timer Underflow Interrupt Request 0) Waveform generator 0 Dead timer underflow interrupt request

[bit6] WGIR1 (Wave Generator Timer Underflow Interrupt Request 1) Waveform generator 1 Dead timer underflow interrupt request

[bit5] WGIR2 (Wave Generator Timer Underflow Interrupt Request 2) Waveform generator 2 Dead timer underflow interrupt request

[bit4] WGRIR0 (Wave Generator Timer Reload Interrupt Request 0) Waveform generator 0 Dead timer reload interrupt request

[bit3] WGRIR1 (Wave Generator Timer Reload Interrupt Request 1) Waveform generator 1 Dead timer reload interrupt request

[bit2] WGRIR2 (Wave Generator Timer Reload Interrupt Request 2) Waveform generator 2 Dead timer reload interrupt request

[bit1] DTTIIR0 (DTTI Interrupt Request 0) DTTI0 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit0] Undefined

This bit always reads "0". Writing has no effect on operation.

23.4.25 IRPR12H (Interrupt Request Peripheral Read register 12H) Interrupt request batch read register 12 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #56)

- **IRPR12H: Address 0138_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
WGIR3	WGIR4	WGIR5	WGRIR3	WGRIR4	WGRIR5	DTTIIR1	-	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	Attribute

[bit7] WGIR3 (Wave Generator Timer Underflow Interrupt Request 3) Waveform generator 3 Dead timer underflow interrupt request

[bit6] WGIR4 (Wave Generator Timer Underflow Interrupt Request 4) Waveform generator 4 Dead timer underflow interrupt request

[bit5] WGIR5 (Wave Generator Timer Underflow Interrupt Request 5) Waveform generator 5 Dead timer underflow interrupt request

[bit4] WGRIR3 (Wave Generator Timer Reload Interrupt Request 3) Waveform generator 3 Dead timer reload interrupt request

[bit3] WGRIR4 (Wave Generator Timer Reload Interrupt Request 4) Waveform generator 4 Dead timer reload interrupt request

[bit2] WGRIR5 (Wave Generator Timer Reload Interrupt Request 5) Waveform generator 5 Dead timer reload interrupt request

[bit1] DTTIIR1(DTTI Interrupt Request 1) DTTI1 interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit0] Undefined

This bit always reads "0". Writing has no effect on operation.

23.4.26 IRPR12L (Interrupt Request Peripheral Read register 12L) Interrupt request batch read register 12 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #57)

- **IRPR12L: Address 0139_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ADCIR0	ADCIR1	ADCIR2	ADCIR3	ADCIR4	ADCIR5	ADCIR6	ADCIR7	
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] ADCIR0 (A/D Converter Interrupt Request 0) A/D converter 0 Conversion completion interrupt request

[bit6] ADCIR1 (A/D Converter Interrupt Request 1) A/D converter 1 Conversion completion interrupt request

[bit5] ADCIR2 (A/D Converter Interrupt Request 2) A/D converter 2 Conversion completion interrupt request

[bit4] ADCIR3 (A/D Converter Interrupt Request 3) A/D converter 3 Conversion completion interrupt request

[bit3] ADCIR4 (A/D Converter Interrupt Request 4) A/D converter 4 Conversion

completion interrupt request

[bit2] ADCIR5 (A/D Converter Interrupt Request 5) A/D converter 5 Conversion completion interrupt request

[bit1] ADCIR6 (A/D Converter Interrupt Request 6) A/D converter 6 Conversion completion interrupt request

[bit0] ADCIR7 (A/D Converter Interrupt Request 7) A/D converter 7 Conversion completion interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

23.4.27 IRPR13H (Interrupt Request Peripheral Read register 13H) Interrupt request batch read register 13 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #58)

- **IRPR13H: Address 013A$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ADCIR8	ADCIR9	ADCIR10	ADCIR11	ADCIR12	ADCIR13	ADCIR14	-	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	Attribute

[bit7] ADCIR8 (A/D Converter Interrupt Request 8) A/D converter 8 Conversion completion interrupt request

[bit6] ADCIR9 (A/D Converter Interrupt Request 9) A/D converter 9 Conversion completion interrupt request

[bit5] ADCIR10 (A/D Converter Interrupt Request 10) A/D converter 10 Conversion completion interrupt request

[bit4] ADCIR11 (A/D Converter Interrupt Request 11) A/D converter 11 Conversion completion interrupt request

[bit3] ADCIR12 (A/D Converter Interrupt Request 12) A/D converter 12 Conversion completion interrupt request

[bit2] ADCIR13 (A/D Converter Interrupt Request 13) A/D converter 13 Conversion completion interrupt request

[bit1] ADCIR14 (A/D Converter Interrupt Request 14) A/D converter 14 Conversion completion interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit0] Undefined

This bit always reads "0". Writing has no effect on operation.

23.4.28 IRPR13L (Interrupt Request Peripheral Read register 13L) Interrupt request batch read register 13 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #59)

- **IRPR13L: Address 013B$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
ADCIR16	ADCIR17	ADCIR18	ADCIR19	ADCIR20	ADCIR21	ADCIR22	ADCIR23	
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] ADCIR16 (A/D Converter Interrupt Request 16) A/D converter 16 Conversion completion interrupt request

[bit6] ADCIR17 (A/D Converter Interrupt Request 17) A/D converter 17 Conversion completion interrupt request

[bit5] ADCIR18 (A/D Converter Interrupt Request 18) A/D converter 18 Conversion completion interrupt request

[bit4] ADCIR19 (A/D Converter Interrupt Request 19) A/D converter 19 Conversion completion interrupt request

[bit3] ADCIR20 (A/D Converter Interrupt Request 20) A/D converter 20 Conversion completion interrupt request

[bit2] ADCIR21 (A/D Converter Interrupt Request 21) A/D converter 21 Conversion completion interrupt request

[bit1] ADCIR22 (A/D Converter Interrupt Request 22) A/D converter 22 Conversion completion interrupt request

[bit0] ADCIR23 (A/D Converter Interrupt Request 23) A/D converter 23 Conversion completion interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

(Note) On the MB91F583AS/F584AS/F585AS, bit7:ADCIR16, bit6:ADCIR17, bit5:ADCIR18, bit2:ADCIR21, bit1:ADCIR22, and bit0:ADCIR23 are reserved bits. Writing has no effect on operation.

23.4.29 IRPR14H (Interrupt Request Peripheral Read register 14H) Interrupt request batch read register 14 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #60)

- **IRPR14H: Address 013C$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
BT0IR0	BT0IR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] BT0IR0 (Base Timer 0 Interrupt Request 0) Base timer 0 interrupt request 0

[bit6] BT0IR1 (Base Timer 0 Interrupt Request 1) Base timer 0 interrupt request 1

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.30 IRPR14L (Interrupt Request Peripheral Read register 14L) Interrupt request batch read register 14 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #61)

- **IRPR14L: Address 013D$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
BT1IR0	BT1IR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] BT1IR0 (Base Timer 1 Interrupt Request 0) Base timer 1 interrupt request 0

[bit6] BT1IR1 (Base Timer 1 Interrupt Request 1) Base timer 1 interrupt request 1

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.31 IRPR15H (Interrupt Request Peripheral Read register 15H) Interrupt request batch read register 15 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #62)

- **IRPR15H: Address 013E$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DMAC0IR	DMAC1IR	DMAC2IR	DMAC3IR	DMAC4IR	DMAC5IR	DMAC6IR	DMAC7IR	
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] 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.

23.4.32 IRPR15L (Interrupt Request Peripheral Read register 15L) Interrupt request batch read register 15 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #15)

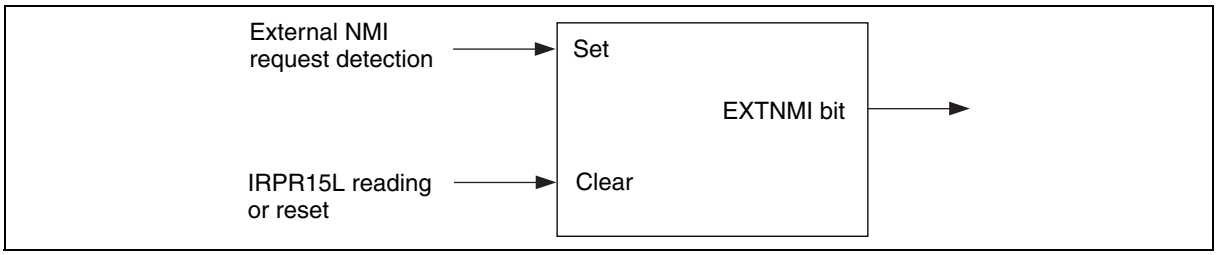
- **IRPR15L: Address 013F$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
EXTNMI	BUS_NMI	BR_ECC_DE	XB_ECC_DE	-				
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] EXTNMI (External Non Maskable Interrupt Request) External NMI request

The EXTNMI bit is set by the detection of an external NMI request, and cleared by the reading of this register.

NMI request is not retained during return from the standby mode (shut-down) because the internal reset is generated.



[bit6] BUS_NMI (Bus Diag Error Interrupt Request): Bus diagnosis error interrupt request

[bit5] BR_ECC_DE (Backup RAMECC Double Bit Error Interrupt Request): Backup RAM double-bit error interrupt request

[bit4] XB_ECC_DE(XBS RAMECC Double Bit Error Interrupt Request): XBS RAM double-bit error interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit3 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.33 IRPR16H (Interrupt Request Peripheral Read register 16H) Interrupt request batch read register 16 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #21)

- **IRPR16H: Address 0140_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RXIR1	ISIR1	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] RXIR1 (Multi-Function-Serial-Interface RX Interrupt Request 1): Multi-function serial interface ch.1 reception completion interrupt request

[bit6] ISIR1 (Multi-Function-Serial-Interface Status Interrupt Request 1): Multi-function serial interface ch.1 status interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.4.34 IRPR16L (Interrupt Request Peripheral Read register 16L) Interrupt request batch read register 16 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #23)

- **IRPR16L: Address 0141_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RXIR2	ISIR2	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] RXIR2 (Multi-Function-Serial-Interface RX Interrupt Request 2): Multi-function serial interface ch.2 reception completion interrupt request

[bit6] ISIR2 (Multi-Function-Serial-Interface Status Interrupt Request 2): Multi-function serial interface ch.2 status interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

(Note) On the MB91F583AS/F584AS/F585AS, this register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.35 IRPR17H (Interrupt Request Peripheral Read register 17H) Interrupt request batch read register 17 high-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #25)

- **IRPR17H: Address 0142_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
RXIR3	ISIR3	-						
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

[bit7] RXIR3 (Multi-Function-Serial-Interface RX Interrupt Request 3): Multi-function serial interface ch.3 reception completion interrupt request

[bit6] ISIR3 (Multi-Function-Serial-Interface Status Interrupt Request 3): Multi-function serial interface ch.3 status interrupt request

Read value of each bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

[bit5 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

(Note) On the MB91F583AS/F584AS/F585AS, this register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.36 IRPR17L (Interrupt Request Peripheral Read register 17L) Interrupt request batch read register 17 low-order

- **IRPR17L: Address 0143_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.37 IRPR18H (Interrupt Request Peripheral Read register 18H) Interrupt request batch read register 18 high-order

- **IRPR18H: Address 0144_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-								
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

This register has undefined bits. These bits always read "0". Writing has no effect on operation.

23.4.38 IRPR18L (Interrupt Request Peripheral Read register 18L) Interrupt request batch read register 18 low-order

This register indicates the peripheral that has issued the interrupt request.(Interrupt vector number #35)

- **IRPR18L: Address 0145_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
XBTC	XBIC	XBTE	BRTC	BRIC	BRTE	-		
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	Attribute

[bit7] XBTC (XBs ram Test Completed interrupt request): XBS RAM test completed interrupt request

[bit6] XBIC (XBs ram Initialization Completed interrupt request): XBS RAM initialization completed interrupt request

[bit5] XBTE (XBs ram Test Error interrupt request): XBS RAM test error interrupt request

[bit4] BRTC (Backup RAM Test Completed interrupt request): Backup RAM test completed interrupt request

[bit3] BRIC (Backup RAM Initialization Completed interrupt request): Backup RAM initialization completed interrupt request

[bit2] 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.

[bit1, bit0] Undefined

These bits always read "0". Writing has no effect on operation.

23.5 Operation

Within each interrupt handler, the pertinent register is read to determine what bits are set and what interrupt requests have been generated.

(Note) This register does not provide a function that can be used to input external interrupts. Read register EIRR0, which is used to input external interrupts.

CHAPTER 24 I/O PORTS

24.1 Overview

This chapter explains the configurations for peripheral bus assignment to external pins and for the use of external pins as I/O ports.

For the MB91F583AS/F584AS/F585AS, there are no pins P000 to P007, P060 to P066, P080 to P087, P090 to P092, P095 to P097, and P100 to P102.

For FlexRay, the MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ/F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ has corresponding functions.

24.2 Features

- I/O multiplexing

If the I/O of multiple peripherals is assigned for one external pin, one of these peripherals is selected to be used.

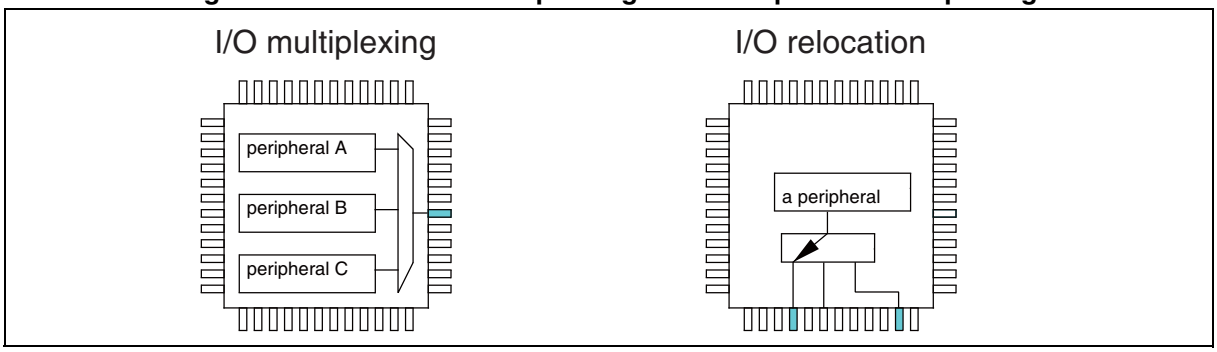
- I/O relocation

If one pin of one peripheral corresponds to the I/O of multiple external pins, 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 24.2-1 Illustration Explaining the Concept of I/O Multiplexing



- Key code function

This function is for error writing protection. If writing is not executed to the key code register (KEYCDR) according to the specified method, writing to the target register will become invalid. Also, word access for the target resister cannot be executed.The following are the key code target registers.

- Data direction register
- Port function register
- Extended port function register
- Port input level register
- Port output drive register
- Port pull-up enable register
- Analog input enable register
- Analog output enable register

Code: ZBPORT-0v3-91580MS-1-E

24.3 Configuration

No configuration diagram is provided.

24.4 Registers

For the MB91F583AS/F584AS/F585AS, there are no pins P000 to P007, P060 to P066, P080 to P087, P090 to P092, P095 to P097, and P100 to P102. The PDR, DDR, PFR, PDDR, PPER, PILR, and PODR of the corresponding pins are reserved bits. Both writing and reading produce no effect.

Table 24.4-1 Register Map (MB91F583AM/F584AM/F585AM)

Address	Register				Register function
	+0	+1	+2	+3	
0x0000	PDR00	PDR01	PDR02	PDR03	Port data registers 0 to 10
0x0004	PDR04	PDR05	PDR06	PDR07	
0x0008	PDR08	PDR09	PDR10	Reserved	
0x0E00	DDR00	DDR01	DDR02	DDR03	Data direction registers 0 to 10 (Key code target register)
0x0E04	DDR04	DDR05	DDR06	DDR07	
0x0E08	DDR08	DDR09	DDR10	Reserved	
0x0E20	PFR00	PFR01	PFR02	PFR03	Port function registers 0 to 10 (Key code target register)
0x0E24	PFR04	PFR05	PFR06	PFR07	
0x0E28	PFR08	PFR09	PFR10	Reserved	
0x0E40	PDDR00	PDDR01	PDDR02	PDDR03	Input data direct registers 0 to 10
0x0E44	PDDR04	PDDR05	PDDR06	PDDR07	
0x0E48	PDDR08	PDDR09	PDDR10	Reserved	
0x0E60	EPFR00	EPFR01	EPFR02	EPFR03	Extended port function registers 0 to 14 (Key code target register)
0x0E64	Reserved	Reserved	EPFR06	EPFR07	
0x0E68	EPFR08	EPFR09	EPFR10	Reserved	
0x0E6C	Reserved	Reserved	EPFR14	Reserved	
0x0EC0	PPER00	PPER01	PPER02	PPER03	Port pull-up enable registers 0 to 10 (Key code target register)
0x0EC4	PPER04	PPER05	PPER06	PPER07	
0x0EC8	PPER08	PPER09	PPER10	Reserved	
0x0EE0	PILR00	PILR01	PILR02	PILR03	Port input level selection registers 0 to 10 (Key code target register)
0x0EE4	PILR04	PILR05	PILR06	PILR07	
0x0EE8	PILR08	PILR09	PILR10	Reserved	
0x0F20	PODR00	PODR01	PODR02	PODR03	Port output drive registers 0 to 10 (Key code target register)
0x0F24	PODR04	PODR05	PODR06	PODR07	
0x0F28	PODR08	PODR09	PODR10	Reserved	
0x0F40	PORTEN	Reserved	Reserved	Reserved	Port input enable register
0x0F44	KEYCDR		Reserved	Reserved	Key code register

Table 24.4-2 Register Map (MB91F583AS/F584AS/F585AS)

Address	Register				Register function
	+0	+1	+2	+3	
0x0000	Reserved	PDR01	PDR02	PDR03	Port data registers 0 to 9
0x0004	PDR04	PDR05	Reserved	PDR07	
0x0008	Reserved	PDR09	Reserved	Reserved	
0x0E00	Reserved	DDR01	DDR02	DDR03	Data direction registers 0 to 9 (Key code target register)
0x0E04	DDR04	DDR05	Reserved	DDR07	
0x0E08	Reserved	DDR09	Reserved	Reserved	
0x0E20	Reserved	PFR01	PFR02	PFR03	Port function registers 0 to 9 (Key code target register)
0x0E24	PFR04	PFR05	Reserved	PFR07	
0x0E28	Reserved	PFR09	Reserved	Reserved	
0x0E40	Reserved	PDDR01	PDDR02	PDDR03	Input data direct registers 0 to 9
0x0E44	PDDR04	PDDR05	Reserved	PDDR07	
0x0E48	Reserved	PDDR09	Reserved	Reserved	
0x0E60	EPFR00	EPFR01	EPFR02	EPFR03	Extended port function registers 0 to 10 (Key code target register)
0x0E64	Reserved	Reserved	EPFR06	EPFR07	
0x0E68	EPFR08	EPFR09	EPFR10	Reserved	
0x0E6C	Reserved	Reserved	Reserved	Reserved	
0x0EC0	Reserved	PPER01	PPER02	PPER03	Port pull-up enable registers 0 to 9 (Key code target register)
0x0EC4	PPER04	PPER05	Reserved	PPER07	
0x0EC8	Reserved	PPER09	Reserved	Reserved	
0x0EE0	Reserved	PILR01	PILR02	PILR03	Port input level selection registers 0 to 9 (Key code target register)
0x0EE4	PILR04	PILR05	Reserved	PILR07	
0x0EE8	Reserved	PILR09	Reserved	Reserved	
0x0F20	Reserved	PODR01	PODR02	PODR03	Port output drive registers 0 to 9 (Key code target register)
0x0F24	PODR04	PODR05	Reserved	PODR07	
0x0F28	Reserved	PODR09	Reserved	Reserved	
0x0F40	PORTEN	Reserved	Reserved	Reserved	Port input enable register
0x0F44	KEYCDR		Reserved	Reserved	Key code register

24.4.1 PDR0 to PDR10 (Port Data Registers 0 to 10) Port Data Registers 0 to 10

These registers hold the output levels of the pins corresponding to individual ports that are in output mode.

- **PDR0 to PDR10: Address 0000_H, 0001_H, ..., $000A_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
P[7:0]								
X	X	X	X	X	X	X	X	Initial value
R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	Attribute

[bit7 to bit0] P (Port) Port data setting bits

These bits set the output levels of external pins P000, P001, ..., when the ports are in output mode.
PDR0.P[7:0] is for external pins P007, P006, ..., P000.
PDR1.P[7:0] is for external pins P017, P016, ..., P010.
PDR2.P[7:0] is for external pins P027, P026, ..., P020.
(A similar process continues)
The assignment is as shown above.

PDR5.P[7], PDR6.P[7], PDR7.P[7:3], and PDR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P	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.

24.4.2 DDR0 to 10 (Data Direction Registers 0 to 10) Data Direction Registers 0 to 10

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.

- **DDR0-10: Address $0E00_H$, $0E01_H$, ..., $0E0A_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
P[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

[bit7 to bit0] P (Port) Data direction selection bits

These bits set the I/O directions of external pins P000, P001, ..., when the ports are in output mode.

DDR0.P[7:0] is for external pins P007, P006, ..., P000.

DDR1.P[7:0] is for external pins P017, P016, ..., P010.

DDR2.P[7:0] is for external pins P027, P026, ..., P020.

(A similar process continues)

The assignment is as shown above.

DDR5.P[7], DDR6.P[7], DDR7.P[7:3], and DDR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P	Operation
0	Input (initial value)
1	Output

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.3 PFR0 to PFR10 (Port Function Registers 0 to 10) Port Function Registers 0 to 10

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 pertinent register must be set for the port function.

- **PFR0-10: Address $0E20_H$, $0E21_H$, ..., $0E2A_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
P[7:0]								
*	*	*	*	*	*	*	*	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

*For the initial values of the registers, see "Appendix B. I/O MAP".

[bit7 to bit0] P (Port) Port function selection bits

These bits are used to set the port function.

PFR0.P[7:0] is for external pins P007, P006, ..., P000.

PFR1.P[7:0] is for external pins P017, P016, ..., P010.

PFR2.P[7:0] is for external pins P027, P026, ..., P020.

(A similar process continues)

The assignment is as shown above.

PFR5.P[7], PFR6.P[7], PFR7.P[7:3], and PFR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P	Operation
0	Port function or peripheral input pin (initial value)
1	Peripheral I/O (bidirectional) pin or peripheral output pin (set by EPFR)

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.4 PDDR0 to PDDR10 (Port Data Direct Registers 0 to 10) Input Data Direct Registers 0 to 10

These registers make it always possible to read the voltage levels of individual external pins. These registers can always be read.

- **PDDR0-10: Address $0E40_H$, $0E41_H$, ..., $0E4A_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
P[7:0]								
X	X	X	X	X	X	X	X	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit7 to bit0] P (Port) Read bits

The value at the external pin can be read.
PDDR0.P[7:0] is for external pins P007, P006, ..., P000.
PDDR1.P[7:0] is for external pins P017, P016, ..., P010.
PDDR2.P[7:0] is for external pins P027, P026, ..., P020.
(A similar process continues)
The assignment is as shown above.

PDDR5.P[7], PDDR6.P[7], PDDR7.P[7:3], and PDDR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P	Operation
0	"L" level
1	"H" level

24.4.5 PPER0 to PPER10 (Port Pull-up Enable Registers 0 to 10) Port Pull-up Enable Registers 0 to 10

In the input state, these registers are used to enable pull-up that functions as ports.

- **PPER0 to PPER10: Address $0EC0_H$, $0EC1_H$, ..., $0ECA_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
P[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

[bit7 to bit0] P (Port) Port Pull-up enable selection bits

PPER0.P[7:0] is for external pins P007, P006, ..., P000.
PPER1.P[7:0] is for external pins P017, P016, ..., P010.
PPER2.P[7:0] is for external pins P027, P026, ..., P020.
(A similar process continues)
The assignment is as shown above.

PPER5.P[7], PPER6.P[7], PPER7.P[7:3], and PPER10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P	Operation
0	Without pull-up (initial value)
1	Pull-up supported

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.6 PILR0 to PILR10 (Port Input Level Registers 0 to 10) Port Input Level Selection Registers 0 to 10

These registers are used to set input levels for individual ports. Glitch input may occur at a pin. Therefore, if the pin is used to supply external input clock or a trigger to a peripheral, the peripheral must be disabled.

- **PILR0 to PILR10: Address $0EE0_H$, $0EE1_H$, ..., $0EEA_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
P[7:0]								
1	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 to bit0] P (Port) Port input level selection bits

PILR0.P[7:0] is for external pins P007, P006, ..., P000.
PILR1.P[7:0] is for external pins P017, P016, ..., P010.
PILR2.P[7:0] is for external pins P027, P026, ..., P020.
(A similar process continues)
The assignment is as shown above.

PILR5.P[7], PILR6.P[7], PILR7.P[7:3], and PILR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P021 to P023, P025 to P027

PILR	Input level	Remarks
0	FlexRay VIL=0.35Vcc VIH=0.65Vcc	
1	Automotive VIL=0.5Vcc VIH=0.8Vcc	Initial value

Pins other than P021 to P023, P025 to P027

PILR	Input level	Remarks
0	CMOS Schmitt VIL=0.3Vcc VIH=0.7Vcc	
1	Automotive VIL=0.5Vcc VIH=0.8Vcc	Initial value

[P040, P041, P063, P064, P080, P081, P083, or P084 pin only]

When the multifunction serial interface has been selected and when I^2C is selected for the operation mode for multifunction serial interface, the I^2C level is applied for the input threshold value no matter what is selected for this bit.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.7 PODR0 to PODR10 (Port Output Drive Registers 0 to 10) Port Output Drive Registers 0 to 10

These registers are used to set drive levels for individual ports.

- **PODR0 to PODR10: Address $0F20_H$, $0F21_H$, ..., $0F2A_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
P[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

[bit7 to bit0] P (Port) Port output drive selection bits

PODR0.P[7:0] is for external pins P007, P006, ..., P000.
PODR1.P[7:0] is for external pins P017, P016, ..., P010.
PODR2.P[7:0] is for external pins P027, P026, ..., P020.
(A similar process continues)
The assignment is as shown above.

PODR5.P[7], PODR6.P[7], PODR7.P[7:3], and PODR10.P[7:3] are reserved bits. Both writing and reading produce no effect.

P021 to P023, P025 to P027

P[n]	Operation
0	4mA
1	2mA

Pins other than P021 to P023, P025 to P027

P[n]	Operation
0	5mA
1	2mA

[P040, P041, P063, P064, P080, P081, P083, or P084 pin only]

When the multifunction serial interface has been selected and when I^2C is selected for the operation mode for multifunction serial interface, 3mA is applied. In other cases, the setting in the above table needs to be followed.

When a pin is set to FlexRay output, set the pin drive level to 4mA.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.8 EPFR00 to 14 (Extended Port Function Registers 00 to 14) Extended Port Function Registers 00 to 14

These registers control switching between the peripheral, and control 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.

■ EPFR00 (Extended Port Function Register 00) Extended Port Function Register 00

CAN output is enabled (I/O multiplexing).

- **EPFR00: Address $0E60_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-						TX1E	TX0E	
1	1	1	1	1	1	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	Attribute

[bit7 to bit2] Undefined

These bits always read "1". Writing has no effect on operation.

[bit1] TX1E CAN channel 1 transmission data output is enabled

[bit0] TX0E CAN channel 0 transmission data output is enabled

TX0E	Operation
0	CAN channel 0 output disabled (initial value)
1	CAN channel 0 output enabled

Same as for TX1E.

(Notes)
- This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.
- On the MB91F583AS/F584AS/F585AS, bit1:TX1E is a reserved bit. So the bit should be written "0".

■ EPFR01 (Extended Port Function Registers 01, 02, 14) Extended Port Function Registers 01, 02, 14

These registers control I/O of the multi-function serial interface (I/O relocation and I/O multiplexing).

- **EPFR01: Address $0E61_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-			SIN0E	SOT0E[1]	SCK0E[1]	SOT0E[0]	SCK0E[0]	
1	1	1	0	0	0	0	0	Initial value
R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	Attribute

- **EPFR02: Address $0E62_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-		SCS2E	SOT2E	SCK2E	SCS1E	SOT1E	SCK1E	
1	1	0	0	0	0	0	0	Initial value
R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- **EPFR14: Address 0E6E$_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-		SCS3E	-	SOT3E	-	SCK3E	-	
1	1	0	1	0	1	0	1	Initial value
R1,WX	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W	R1,WX	Attribute

- **EPFR01**

[bit7 to bit5] Undefined

These bits always read "1". Writing has no effect on operation.

[bit4] SIN0E Multi-function serial interface ch.0 SIN input pin select

[bit3, bit1] SOT0E Multi-function serial interface ch.0 SOT output/input pin select

[bit2, bit0] SCK0E Multi-function serial interface ch.0 SCK output/input pin select

- **EPFR02**

[bit7, bit6] Undefined

These bits always read "1". Writing has no effect on operation.

[bit5] SCS2E Multi-function serial interface ch.2 SCS output enable

[bit4] SOT2E Multi-function serial interface ch.2 SOT output enable

[bit3] SCK2E Multi-function serial interface ch.2 SCK output enable

[bit2] SCS1E Multi-function serial interface ch.1 SCS output enable

[bit1] SOT1E Multi-function serial interface ch.1 SOT output enable

[bit0] SCK1E Multi-function serial interface ch.1 SCK output enable

- **EPFR14**

[bit7, bit6] Undefined

These bits always read "1". Writing has no effect on operation.

[bit5] SCS3E Multi-function serial interface ch.3 SCS output enable

[bit4] Undefined

These bits always read "1". Writing has no effect on operation.

[bit3] SOT3E Multi-function serial interface ch.3 SOT output enable

[bit2] Undefined

These bits always read "1". Writing has no effect on operation.

[bit1] SCK3E Multi-function serial interface ch.3 SCK output enable

[bit0] Undefined

These bits always read "1". Writing has no effect on operation.

SCK0E[1:0]	Operation
00	Input from multi-function serial interface SCK0_0, output disabled (initial value)
01	Input/output from/to multi-function serial interface SCK0_0
10	Input/output from/to multi-function serial interface SCK0_1
11	Reserved (input from SCK0_1, output to SCK0_1)

Same as for SOT0E[1:0].

SIN0E	Operation
0	Input from multi-function serial interface ch.0 SIN0_0 (initial value)
1	Input from multi-function serial interface ch.0 SIN0_1

SCK1E	Operation
0	Multi-function serial interface ch.1 SCK output disabled (initial value)
1	Multi-function serial interface ch.1 SCK output disabled

Same as for SCK2E, SCK3E, SOT1E to SOT3E, and SCS1E to SCS3E.

(Notes)
- This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.
- On the MB91F583AS/F584AS/F585AS, SIN0E, SOT0E[1], SCK0E[1], SCS2E, SOT2E, SCK2E, SCS3E, SOT3E, and SCK3E are reserved bits. So each bit should be written "0".

■ EPFR03 (Extended Port Function Registers 03) Extended Port Function Registers 03

PPG output is enabled (I/O multiplexing).

- **EPFR03: Address 0E63$_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
		PPG5E	PPG4E	PPG3E	PPG2E	PPG1E	PPG0E	
1	1	0	0	0	0	0	0	Initial value
R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] Undefined

These bits always read "1". Writing has no effect on operation.

[bit5] PPG5E PPG ch.5 output enable

[bit4] PPG4E PPG ch.4 output enable

[bit3] PPG3E PPG ch.3 output enable

[bit2] PPG2E PPG ch.2 output enable

[bit1] PPG1E PPG ch.1 output enable

[bit0] PPG0E PPG ch.0 output enable

PPG0E	Operation
0	Output from PPG0 pin disabled (initial value)
1	Output from PPG0 pin enabled

Same as for PPG1E to PPG5E.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

■ EPFR06 (Extended Port Function Register 06) Extended Port Function Register 06

Base timer output is enabled (I/O multiplexing).

- **EPFR06: Address 0E66$_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-						TIA1E	TIA0E	
1	1	1	1	1	1	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	Attribute

[bit7 to bit2] Undefined

These bits always read "1". Writing has no effect on operation.

[bit1] TIA1E Base timer TIOA1 output enable

[bit0] TIA0E Base timer TIOA0 output enable

TIA0E	Operation
0	Base timer TIOA0 output disabled (initial value)
1	Base timer TIOA0 output enabled

Same as for TIA1E.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

■ EPFR07 (Extended Port Function Register 07) Extended Port Function Register 07

Reload timer output is enabled (I/O multiplexing).

- **EPFR07: Address $0E67_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-				TOT3E	TOT2E	TOT1E	TOT0E	
1	1	1	1	0	0	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

These bits always read "1". Writing has no effect on operation.

[bit3] TOT3E Reload timer ch.3 TOT output enable

[bit2] TOT2E Reload timer ch.2 TOT output enable

[bit1] TOT1E Reload timer ch.1 TOT output enable

[bit0] TOT0E Reload timer ch.0 TOT output enable

TOT0E	Operation
0	Reload timer TOT0 output disabled (initial value)
1	Reload timer TOT0 output enabled

Same as for TOT1E to TOT3E.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

■ EPFR08 (Extended Port Function Register 08) Extended Port Function Register 08

FlexRay output is enabled (I/O multiplexing).

- **EPFR08: Address $0E68_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-				TXENBE	TXENAE	TXDBE	TXDAE	
1	1	1	1	0	0	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

These bits always read "1". Writing has no effect on operation.

[bit3] TXENBE FlexRay ch.B TXENBoutput enable

[bit2] TXENAE FlexRay ch.A TXENAoutput enable

[bit1] TXDBE FlexRay ch.B TXDBoutput enable

[bit0] TXDAE FlexRay ch.A TXDAoutput enable

TXDAE	Operation
0	FlexRay ch.0 TXDA output disabled (Initial value)
1	FlexRay ch.0 TXDA output enabled

Same as for TXDBE, TXENAE, and TXENBE.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

■ EPFR09 (Extended Port Function Register 09) Extended Port Function Register 09

Clock monitor output is enabled (I/O multiplexing).

- **EPFR09: Address $0E69_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-							MONCLKE	
1	1	1	1	1	1	1	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	Attribute

[bit7 to bit1] Undefined

These bits always read "1". Writing has no effect on operation.

[bit0] MONCLKE Clock monitor MONCLK output enable

MONCLKE	Operation
0	Clock monitor MONCLK output disabled (Initial value)
1	Clock monitor MONCLK output enabled

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

■ EPFR10 (Extended Port Function Registers 10) Extended Port Function Registers 10

Waveform generator output is enabled (I/O multiplexing).

- **EPFR10: Address $0E6A_H$ (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
-	RTO6E	RTO5E	RTO4E	RTO3E	RTO2E	RTO1E	RTO0E	
1	0	0	0	0	0	0	0	Initial value
R1,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] Undefined

This bit always read "1". Writing has no effect on operation.

[bit6] RTO6E Waveform generator ch.6 output enable

[bit5] RTO5E Waveform generator ch.5 output enable

[bit4] RTO4E Waveform generator ch.4 output enable

[bit3] RTO3E Waveform generator ch.3 output enable

[bit2] RTO2E Waveform generator ch.2 output enable

[bit1] RTO1E Waveform generator ch.1 output enable

[bit0] RTO0E Waveform generator ch.0 output enable

RTO0E	Operation
0	Waveform generator ch.0 RTO output disabled (Initial value)
1	Waveform generator ch.0 RTO output enabled

Same as for RTO1E to RTO6E.

(Note) This register is a key code target register.It is necessary to set the key code in order to execute writing. Word access for this register is prohibited.

24.4.9 PORTEN (PORT ENable register) Port Input Enable Register

This register contains setting for enabling port input. Before a power-on reset, inputs to most pins are disabled in order to avoid pass-through current fluctuations before the ports are configured by software. For information on pins whose inputs are disabled, see "APPENDIX D. Pin Statuses in CPU Status". After each port pin is configured for its function, its input must be enabled with global port enable (PORTEN. GPORTEN).

- **PORTEN: Address $0F40_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-						CPOR-TEN	GPOR-TEN	
1	1	1	1	1	1	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	Attribute

[bit7 to bit2] Undefined

These bits always read "1". Writing has no effect on operation.

[bit1] CPORTEN (Communication PORT ENable) Serial write pin enable for input

CPORTEN	Operation
0	No operation (Initial value)
1	Input to the serial write pin (multi-function serial ch.0) is enabled. Regardless of the GPORTEN bit setting, this bit can be used to enable input to the serial write pin (multi-function serial ch.0).

[bit0] GPORTEN (Global PORT ENable) Global input enable

GPORTEN	Operation
0	Inputs to most pins are disabled. For information on pins whose inputs are disabled, see "APPENDIX D. Pin Statuses in CPU Status". (Initial value)
1	The input is enabled by this bit.

24.4.10 KEYCDR key code register

This is the register for the register writing settings that include 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)**

15	14	13	12	11	10	9	8	bit
KEY1	KEY0	SIZE	RADR12	RADR11	RADR10	RADR9	RADR8	
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	Attribute

7	6	5	4	3	2	1	0	bit
RADR7	RADR6	RADR5	RADR4	RADR3	RADR2	RADR1	RADR0	
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	Attribute

[bit15, bit14] KEY[1:0]: 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 settings will become invalid and it will be necessary to reset them from the beginning.

[bit13] SIZE: Access size

The access size is set 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

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".

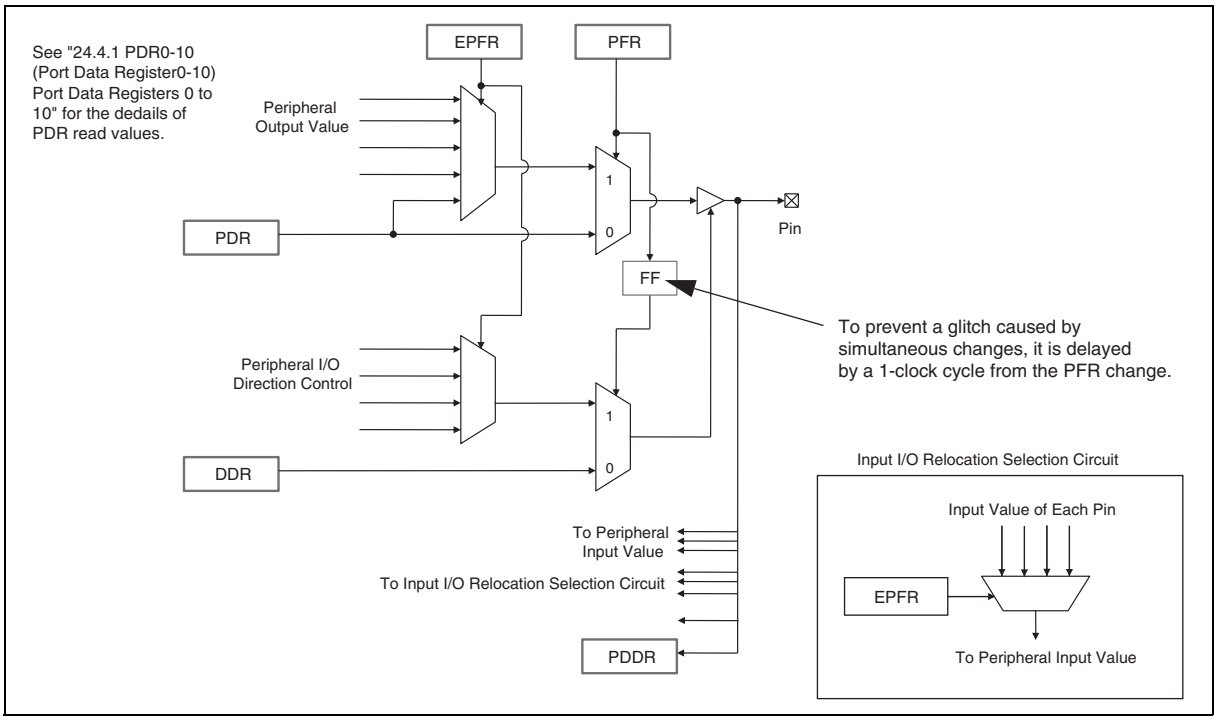
(Note) When different data is written while writing the key code "00", "01", "10", and "11", the key code setting will become invalid and it will be necessary to reset it from the beginning.
Key code setting might be canceled because of the DMA transfer. Read the value of the target register, and check if the value is updated.

24.5 Operation

24.5.1 Pin I/O Assignment

The pin I/O assignment is as follows. The I/O direction of each pin is controlled based on the configuration shown below.

Figure 24.5-1 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 then functions as a port, also set the DDR and PDR values in advance if necessary. When doing this note that the I/O direction of the pin is once set as specified by the DDR. (See Section "24.5.2 Notes on Port Changes".) For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode". For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".

Set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

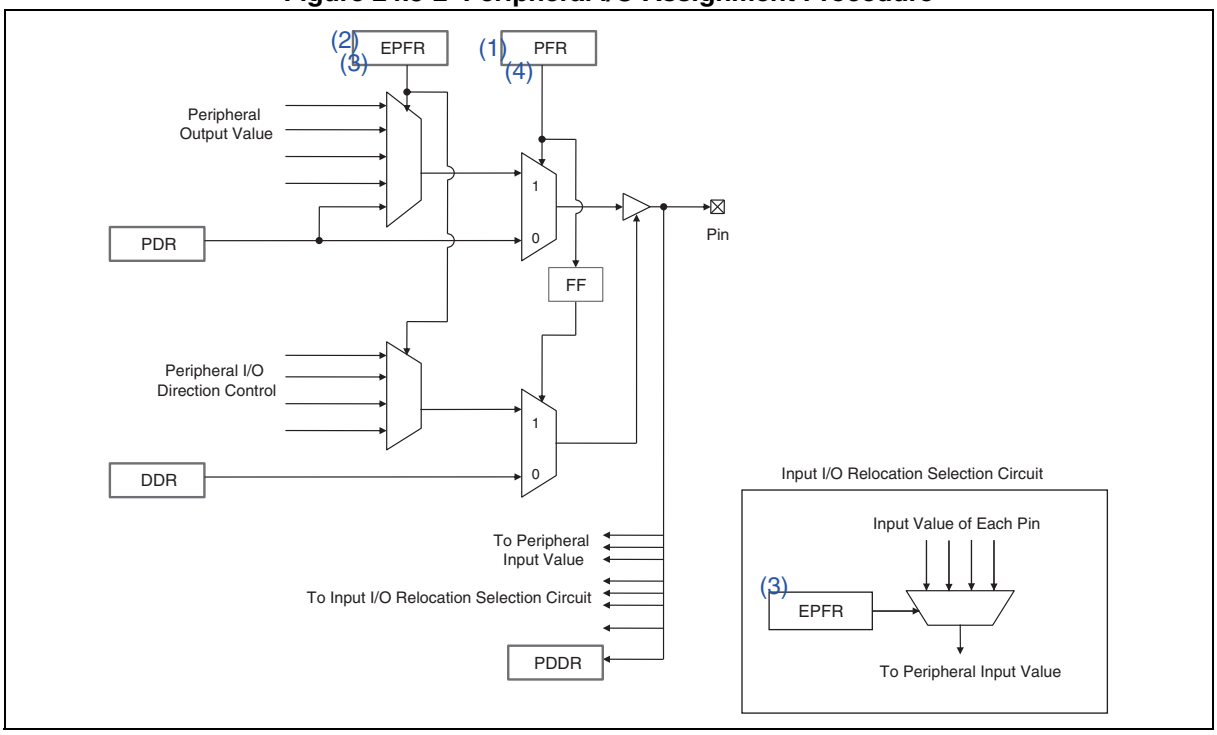
■ Peripheral I/O (Bidirectional) Pin Assignment

(Preparation)

- Since the pin once functions as a port as the result of step (1), also set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode".For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".
- For a pin with the D/A converter function, set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(1) Set the PFR for the pin to enable the port function.
(2) Disable the EPFRs for all other peripherals to be used by the pin.
(3) If the pin is also used for or the pertinent peripheral is one of the targets of I/O multiplexing, set the EPFR of the peripheral. On the other hand, if the pertinent peripheral provides I/O relocation, set a pin to be used for the EPFR of the peripheral.
(4) Set the PFR for the peripheral.

Figure 24.5-2 Peripheral I/O Assignment Procedure



(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

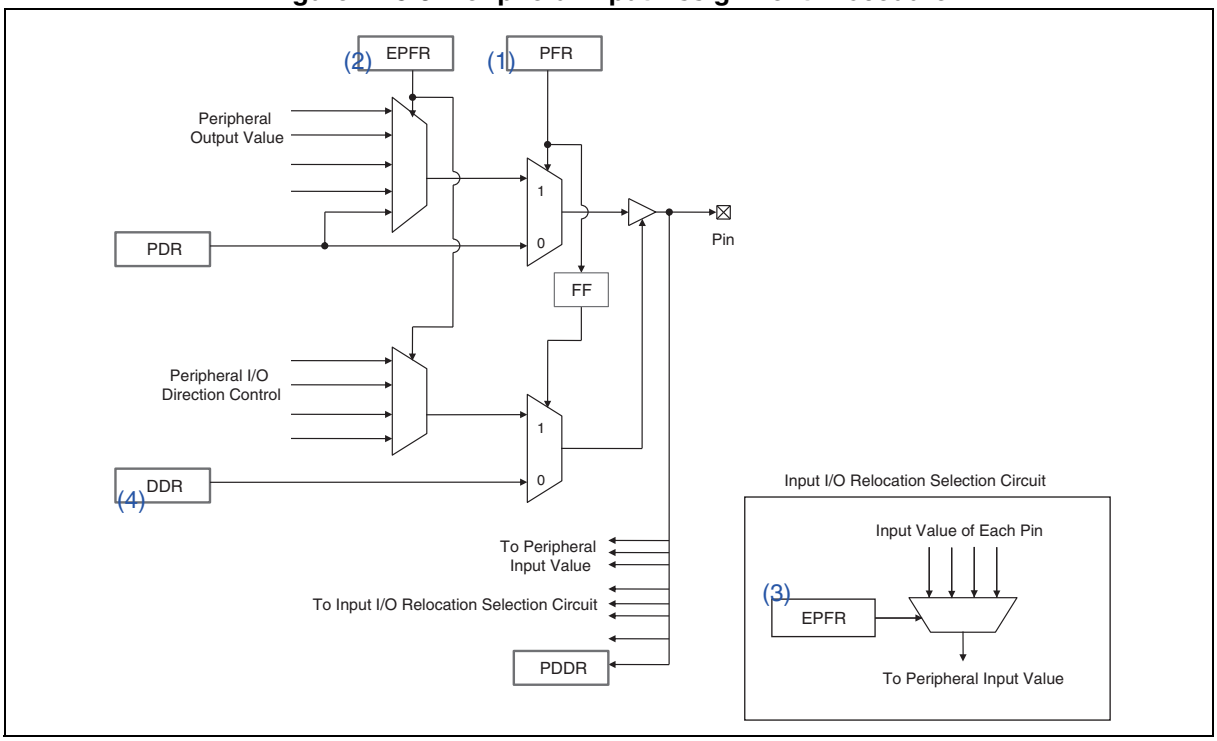
■ Peripheral Input Assignment

(Preparation)

- Since the pin once functions as a port as the result of step (1), also set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode".For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".
- For a pin with the D/A converter function, set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(1) Set the PFR for the pin to enable the port function.
(2) Disable the EPFRs for all other peripherals to be used by the pin.
(3) If the pertinent peripheral provides I/O relocation, set a pin to be used for the EPFR of the peripheral.
(4) Set the DDR for input.

Figure 24.5-3 Peripheral Input Assignment Procedure



(Notes)

- As shown in the figure above, if the pin is set for peripheral output or similar purpose, its output is supplied to the inputs to other peripherals that share the pin.
  Example: Since RX0 pin and INT1 pin are assigned to the same pin (pin number 87*, P045), it is possible to return from the sleep mode or the standby mode by the reception operation of CAN.
  *: The pin number for the MB91F583AS/F584AS/F585AS is 57.
- The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

■ Peripheral Output Assignment

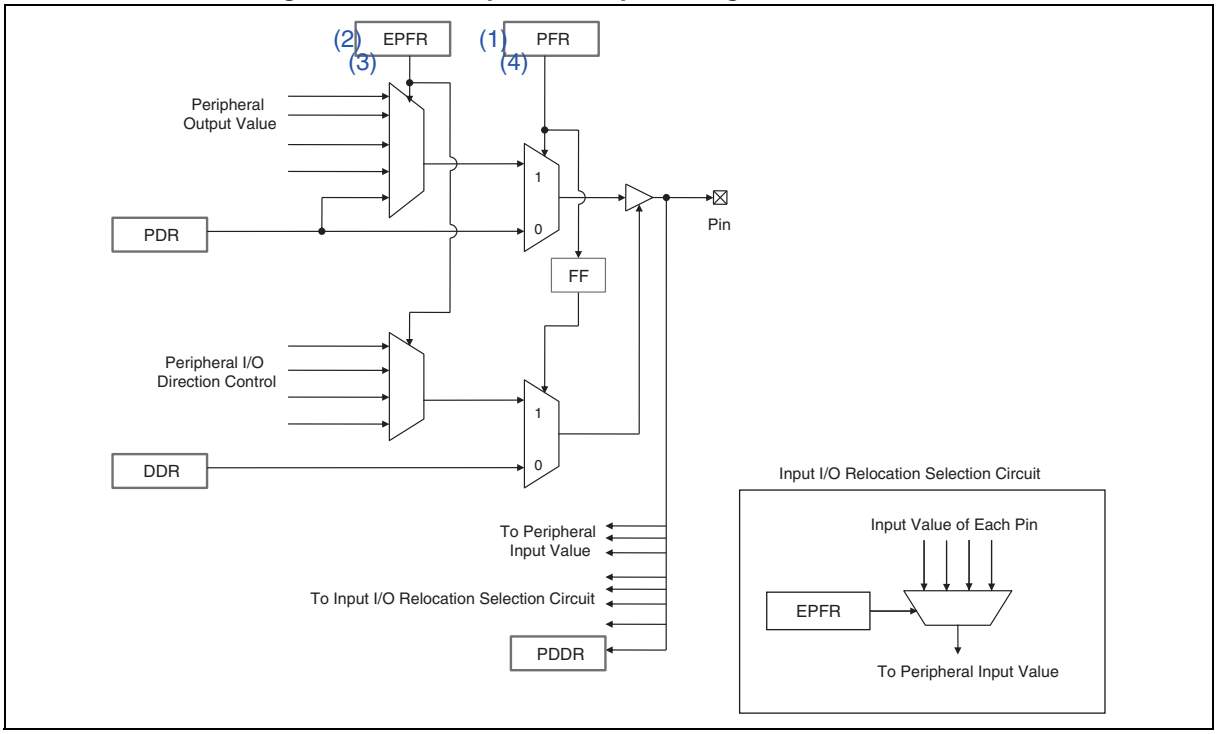
The setting method is the same as that described in "■ Peripheral I/O (Bidirectional) Pin Assignment".

(Preparation)

- Since the pin once functions as a port as the result of step (1), also set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode".For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".
- For a pin with the D/A converter function, set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(1) Set the PFR for the pin to enable the port function.
(2) Disable the EPFRs for all other peripherals to be used by the pin.
(3) If the pertinent peripheral is one of the targets of I/O multiplexing, set the EPFR of the peripheral. On the other hand, if the pertinent peripheral provides I/O relocation, set a pin to be used for the EPFR of the peripheral.
(4) Set the PFR for the peripheral.

Figure 24.5-4 Peripheral Output Assignment Procedure



(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

■ Port Function (Input) Assignment

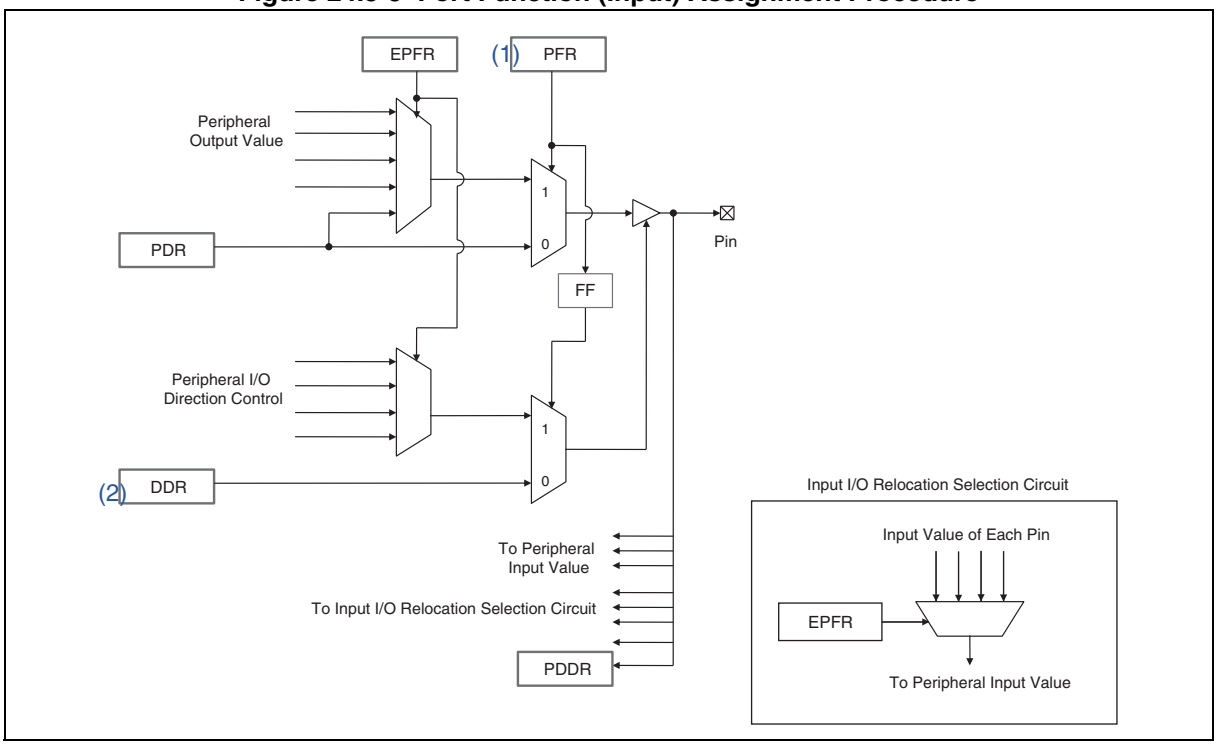
(Preparation)

- For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode". For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".
- For a pin with the D/A converter function, set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(1) Set the PFR to enable the port function.

(2) Set the DDR for input.

Figure 24.5-5 Port Function (Input) Assignment Procedure



(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

■ Port Function (Output) Assignment

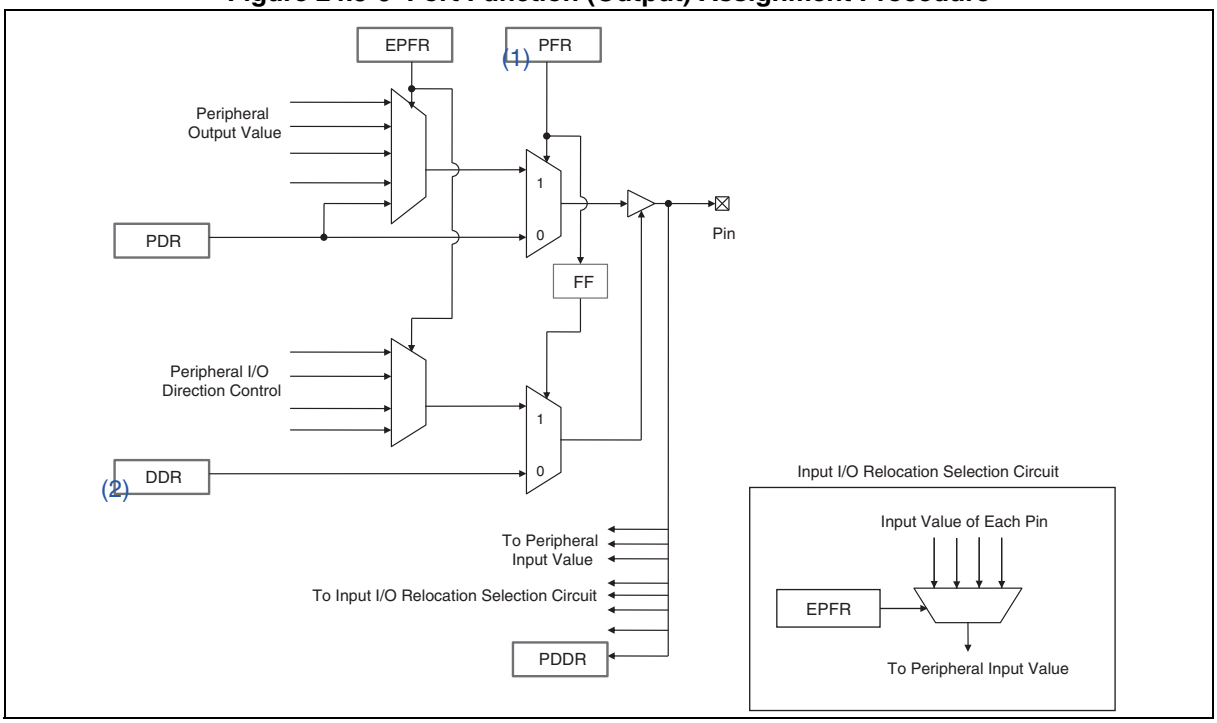
(Preparation)

- For a pin with the A/D converter function, set the bit in the A/D converter analog input enable register (ADER) to "Port I/O mode". For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER".
- For a pin with the D/A converter function, set the bit in the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER".

(1) Set the PFR to enable the port function.

(2) Set the DDR for output.

Figure 24.5-6 Port Function (Output) Assignment Procedure



(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

■ A/D Converter Input Assignment

(1) Set the A/D converter analog input enable register (ADER) to analog input mode. See "CHAPTER 45 12-BIT A/D CONVERTER".

Since A/D converter assignment is given the highest priority, no other configuration is required.

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers. It is necessary to set the key code in order to execute writing.

■ D/A Converter Output Assignment

(1) Set the D/A converter analog output enable register (DAER) to analog input mode. See "CHAPTER 48 D/A CONVERTER".

Since D/A converter assignment is given the highest priority, no other configuration is required.

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR and ADER are the target key code registers.It is necessary to set the key code in order to execute writing.

24.5.2 Notes on Port Changes

This product is structured to ensure that when peripheral output of pin and port input (Hi-Z state) are switched using the PFR, transmission to the output enable control signal is delayed by one clock cycle (1PCLK) in comparison to the data signal selection from the PFR as shown in Figure 24.5-1. Therefore, when the PFR is changed like 1 → 0 → 1 while DDR = 0, the operation is as follows:

Expected operation if there is no delay structure:
Peripheral output → Hi-Z (port input) → Peripheral output
Actual operation:
Peripheral output → PDR value output (1 clock cycle) → Hi-Z → Peripheral output

This means that Hi-Z state comes after a period of one clock cycle during which the PDR value is output. Also, note that the PDR value after a reset is indefinite.

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

24.5.3 EPFR Setting Priority

If the PFR is set for the peripheral and the EPFR setting for one pin can be used for multiple peripherals, the effective peripheral is determined based on the following priorities:

1. CAN
2. Multi-function serial interface
3. PPG
4. Base timer
5. Reload timer
6. FlexRay
7. Clock monitor
8. Waveform generator
9. (No EPFR setting) PDR value

24.5.4 Notes on Setting Input I/O Relocation

Take note that if levels between pins before and after switching differ when switching from an input pin to an external pin, I/O relocation switching may become a trigger input to a peripheral that triggers a pertinent pin.

24.5.5 Noise Filter

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
- A/D converter trigger input
- Base timer
- UP/DOWN COUNTER
- Dead timer interrupt request
- Input capture
- FlexRay STOPWT input

(Note) For information on the noise filter characteristics, see the pertinent data sheet.

24.5.6 Disabling of input by GPORTEN

Before a power-on reset on a product, inputs to most pins are disabled in order to avoid pass-through current fluctuations before the ports are configured by software. For information on pins whose inputs are disabled, see the pin status table in "APPENDIX D. Pin Statuses in CPU Status". On how to enable inputs, see Section "24.4.9 PORTEN (PORT ENable register) Port Input Enable Register".

24.5.7 Notes on Pins with the A/D Converter Function

If you want to use a pin with the A/D converter function to perform another function, set the pertinent bit of the A/D converter analog input enable register (ADER) to "Port I/O mode".For information on the setting method, see "CHAPTER 45 12-BIT A/D CONVERTER". In analog input mode, inputs from ports and from peripheral functions are fixed at "0" and outputs are fixed at Hi-Z regardless of the port function register (PFR) and extended port function register (EPFR) settings, inputs from ports and from peripheral functions are fixed at "0" and outputs are fixed at Hi-Z regardless of the port function register (PFR) and extended port function register (EPFR) settings.

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers. It is necessary to set the key code in order to execute writing.

24.5.8 Notes on Pins with the D/A Converter Function

If you want to use a pin with the D/A converter function to perform another function, set the pertinent bit of the D/A converter analog output enable register (DAER) to "Port I/O mode". For information on the setting method, see "CHAPTER 48 D/A CONVERTER". If analog output 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 (PFR) and extended port function register (EPFR) settings.

(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers. It is necessary to set the key code in order to execute writing.

24.5.9 Setting when Using the Base Timer TIOA1 Pin

If the base timer TIOA1 pin is to be used, it must be set for input if base timer I/O mode 1 prevails or for output otherwise. If the base timer TIOA1 pin is to be used, it must be set for peripheral input if base timer I/O mode 1 prevails (see "■ Peripheral Input Assignment") or for peripheral output otherwise (see "■ Peripheral Output Assignment").

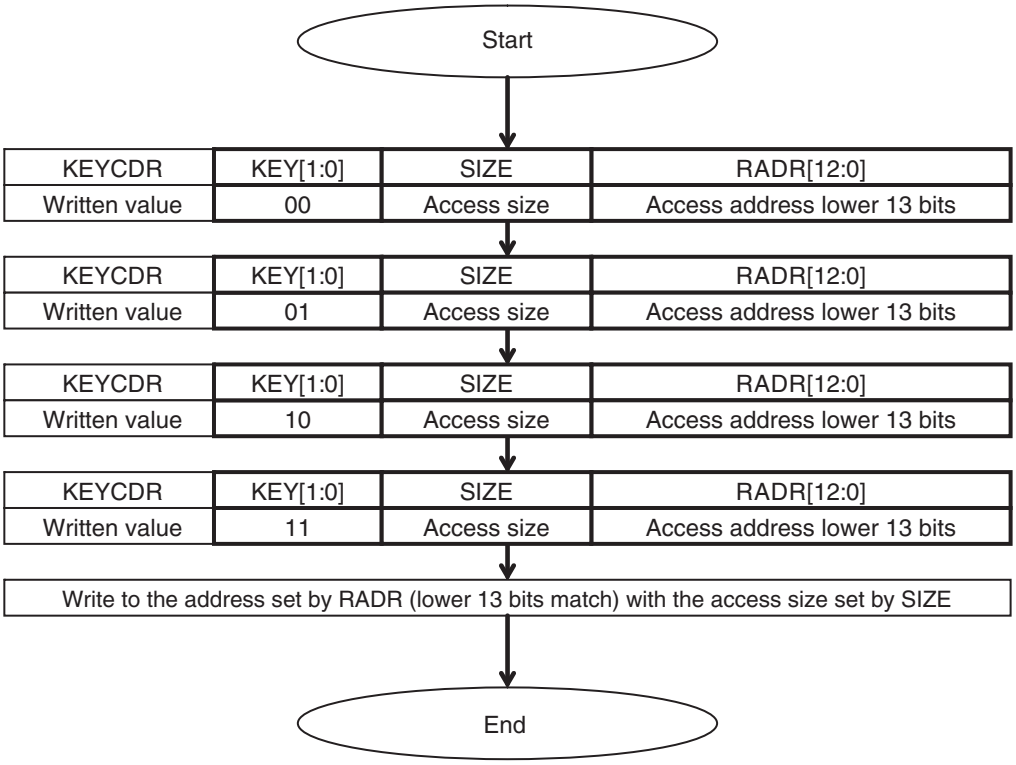
(Note) The DDR, PFR, EPFR, PPER, PILR, PODR, ADER, and DAER are the target key code registers.It is necessary to set the key code in order to execute writing.

24.5.10 Key Code Register Function Settings

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.



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

(Note) 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.

CHAPTER 25 CLOCK MONITOR

25.1 Overview

The clock monitor is a macro that outputs internal clock signals to external pins. The clock monitor has a function for dividing the frequency of a clock signal before output to the pin, allowing clock signals to be used for synchronization of external circuits with MCU functions.

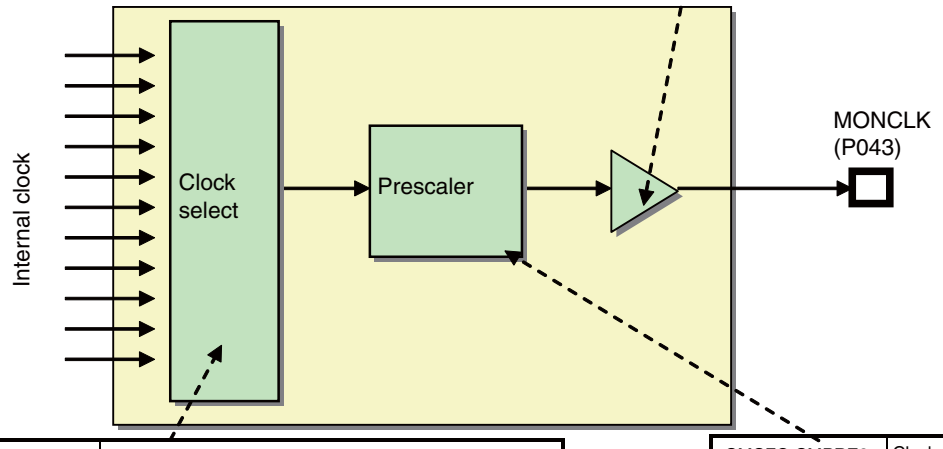
25.2 Features

- 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

Code: FJ43-1v0-91580MS-1-E

25.3 Configuration

Figure 25.3-1 Configuration Diagram of Clock Monitor



CMCFG.CMSEL3 to CMSEL0	Selected clock
0000B	Clock monitor output disabled
Other	Clock monitor output enabled

CMCFG.CMSEL3 to CMSEL0	Selected clock
0000B	Clock monitor output disabled
0001B	Main oscillation before CSV
0010B	CR oscillation
0011B	Main oscillation after CSV
0100B	HCLK for FlexRay
0101B	SCLK for FlexRay
0110B	PLL output for FlexRay
0111B	PLL automatic gear output for FlexRay
1000B	PLL output
1001B	SSCG output
1010B	PLL output after CAN prescaler (CAN system clock)
1011B	CCLK
1100B	HCLK
1101B	PCLK1
1110B	PCLK2
1111B	MTRCLK

CMCFG.CMPRE3 to CMPRE0	Clock output frequency
0000B	Division by 1
0001B	Division by 2
0010B	Division by 3
0011B	Division by 4
0100B	Division by 5
0101B	Division by 6
0110B	Division by 7
0111B	Division by 8
1000B	Division by 9
1001B	Division by 10
1010B	Division by 11
1011B	Division by 12
1100B	Division by 13
1101B	Division by 14
1110B	Division by 15
1111B	Division by 16

PCLK1:spread peripheral clock
PCLK2:peripheral clock after spread/ non spread selection

25.4 Registers

Table 25.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x0004E0	Reserved	Reserved	CSCFG	CMCFG	Clock Monitor Configuration Registers

25.4.1 Clock Monitor Configuration Registers

- **CMCFG: Address 0004E3$_H$ (Access: Byte, Half-word, Word)**

bit 7	6	5	4	3	2	1	0	
CMPRE3	CMPRE2	CMPRE1	CMPRE0	CMSEL3	CMSEL2	CMSEL1	CMSEL0	
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

See "Meanings of Bit Attribute Symbols" for details on the attributes.
The following limitations apply to when rewriting this register. See Section "25.8 Notes".

[bit7 to bit4] CMPRE3 to CMPRE0 (Output Frequency Prescaler Bits)

CMPRE3	CMPRE2	CMPRE1	CMPRE0	Clock frequency output to the MONCLK pin
0	0	0	0	Source clock divided by 1 (selected by CMSEL) (Initial value)
0	0	0	1	Source clock divided by 2 (selected by CMSEL)
0	0	1	0	Source clock divided by 3 (selected by CMSEL)
0	0	1	1	Source clock divided by 4 (selected by CMSEL)
0	1	0	0	Source clock divided by 5 (selected by CMSEL)
0	1	0	1	Source clock divided by 6 (selected by CMSEL)
0	1	1	0	Source clock divided by 7 (selected by CMSEL)
0	1	1	1	Source clock divided by 8 (selected by CMSEL)
1	0	0	0	Source clock divided by 9 (selected by CMSEL)
1	0	0	1	Source clock divided by 10 (selected by CMSEL)
1	0	1	0	Source clock divided by 11 (selected by CMSEL)
1	0	1	1	Source clock divided by 12 (selected by CMSEL)
1	1	0	0	Source clock divided by 13 (selected by CMSEL)
1	1	0	1	Source clock divided by 14 (selected by CMSEL)
1	1	1	0	Source clock divided by 15 (selected by CMSEL)
1	1	1	1	Source clock divided by 16 (selected by CMSEL)

Specifies the output frequency of the clock signal to the clock monitor pin.

[bit3 to bit0] CMSEL3 to CMSEL0 (Output Source Clock Selection Bits)

CMSEL3	CMSEL2	CMSEL1	CMSEL0	Clock source output to MONCLK pin
0	0	0	0	MONCLK output disabled (high impedance state) (initial value)
0	0	0	1	Main oscillation before CSV
0	0	1	0	CR oscillation
0	0	1	1	Main oscillation after CSV
0	1	0	0	HCLK for FlexRay
0	1	0	1	SCLK for FlexRay
0	1	1	0	PLL output for FlexRay
0	1	1	1	PLL automatic gear output for FlexRay
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	MTRCLK

- **CSCFG: Address 0004E2$_H$ (Access: Byte, Half-word, Word)**

bit 7	6	5	4	3	2	1	0	
Reserved	Reserved	Reserved	MONCKI	Reserved				
0	0	0	0	0	0	0	0	Initial value
R/W0	R/WX	R/WX	R/W	R/W0	R/W0	R/W0	R/W0	Attribute

See "Meanings of Bit Attribute Symbols" for details on the attributes.
The following limitations apply to when rewriting this register. See Section "25.8 Notes".

[bit7] Reserved

This bit is reserved. Always set this bit to "0" when writing.

[bit6] Reserved

This bit is reserved. Always set this bit to "0" when writing.

[bit5] Reserved

This bit is reserved. Always set this bit to "0" when writing.

[bit4] MONCKI (Clock Monitor MONCLK Inverter)

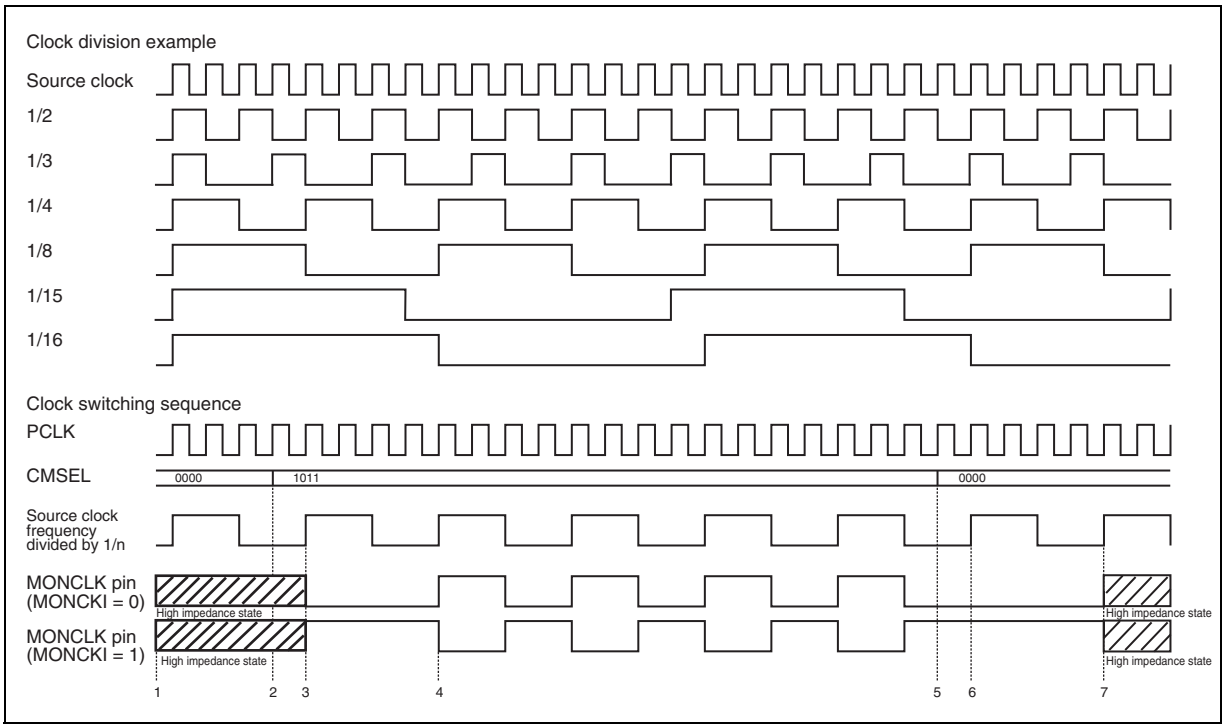
MONCKI	Function
0	MONCLK mark level is low level (initial value)

MONCKI	Function
1	MONCLK mark level is high level

[bit3 to bit0] Reserved

These bits are reserved. Always set these bits to "0" when writing.

25.5 Operation Description



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 "H" output 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 "H" output if MONCKI is set to "1") for the duration of one internal (prescaled) clock.
7. The MONCLK pin switches to the high impedance state.

25.6 Setup

Setting	Configured register	Configuration procedure
Setting the prescaler value	Clock monitor prescaler (CMCFG: CMPRE3 to CMPRE0)	Section 25.7.2
Setting the source clock	Clock monitor selection (CMCFG: CMSEL3 to CMSEL0)	Section 25.7.1
Setting the mark level	Clock monitor inverter (CSCFG: MONCKI)	Section 25.7.4
Enabling clock monitor output	Clock monitor selection (CMCFG: CMSEL3 to CMSEL0)	Section 25.7.3

25.7 Q&A

25.7.1 How Do I Configure the Output Pin (MONCLK)?

Use the clock monitor selection bits (CMCFG: CMSEL[3:0]).

25.7.2 How Do I Select the Output Frequency?

Use the output frequency selection bits (CMCFG: CMPRE[3:0]).

Clock division ratio	When output frequency (Example) HCLK is selected		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

25.7.3 How Do I Enable or Disable Clock Monitor Output?

Use the output enable bits (CMCFG: CMSEL[3:0]).

Operation Description	Output enable 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 between 0001_B and 1111_B

25.7.4 How Do I Set the Clock Output Mark Level?

Use the clock monitor MONCLK inverter bit (CSCFG: MONCKI).

25.8 Notes

In order to achieve glitch-free switching, use the following procedure when changing the clock source (CMSEL3 to CMSEL0) or prescaler ratio (CMPRE3 to CMPRE0).

- The CMPRE3 to CMPRE0 registers can be written only when the CMSEL3 to CMSEL0 registers are "0_H".
- The CMPRE3 to CMPRE0 registers can be written only when "0_H" is written to the CMSEL3 to CMSEL0 registers during the same write access.
- At least two cycles of the monitor clock division are required during the two write accesses to CMPRE and CMCFG.
- When selecting another effective clock while something 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

CMCFG_CMSEL = 0

CMCFG_CMPRE = Prescaler

2.Access

CMCFG_CMSEL = Clock

The CSCFG.MONCKI flag can also be written the same as above only when CMSEL3 to CMSEL0 are "0_H".

CHAPTER 26 WDT1 CALIBRATION

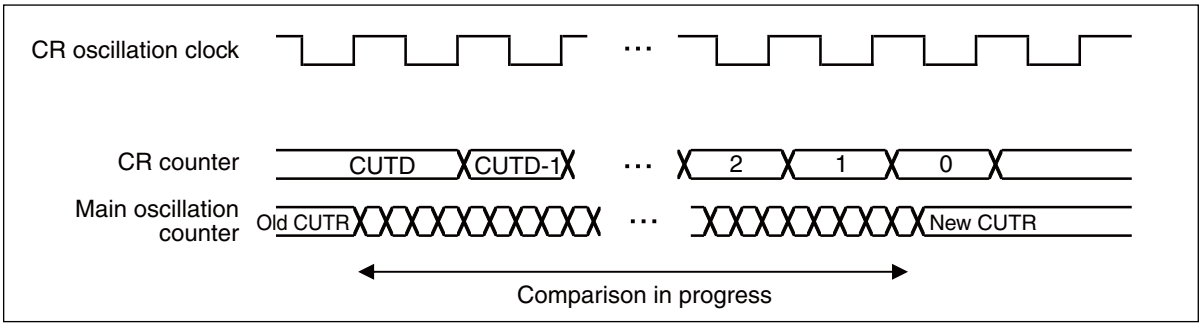
26.1 Overview

This module calculates the values for frequency calibrations of on-board CR oscillation circuit for WDT1.

26.2 Features

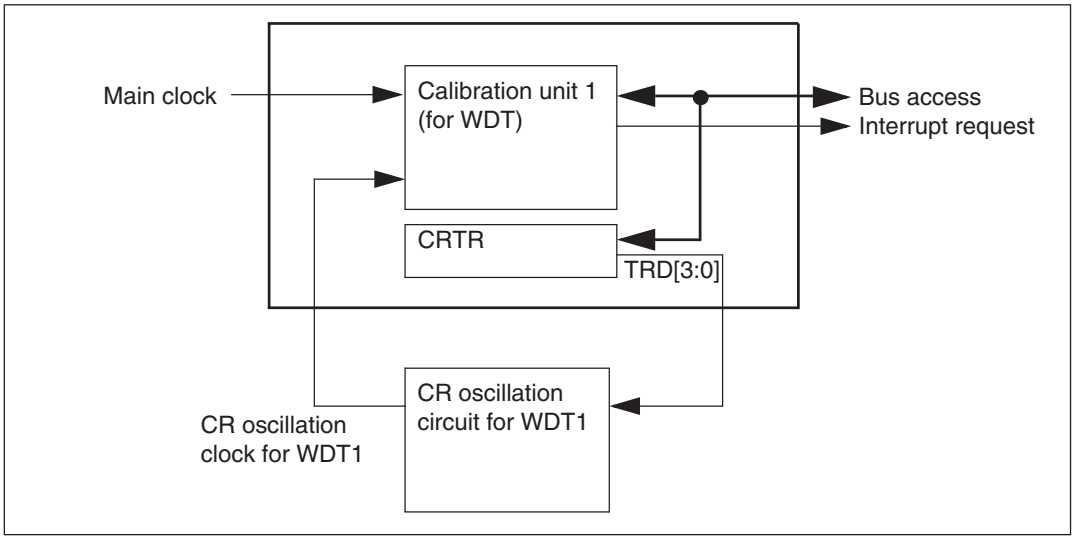
- WDT1 (CR clock) calibration
  - Operate the main clock driven counter and the CR clock driven counter concurrently (Figure 26.2-1), calculate the CR clock frequency from the main clock frequency, and set the CR clock trimming value.

Figure 26.2-1 Comparison for Counters Driven by Different Clocks



26.3 Configuration

Figure 26.3-1 Block Diagram of WDT1 Calibration



Code: BJ42RC-0v1-91580MS-1-E

26.4 Registers

Table 26.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x04C4	CUCR1		CUTD1		Calibration unit control register 1 CR oscillation timer data register
0x04C8	CUTR1				Main oscillation timer data register 1
0x0570	CRTR	Reserved	Reserved	Reserved	CR oscillation trimming setting register

26.4.1 CUCR1: (Calibration Unit Control Register 1) Calibration Unit Control Register 1

This register configures calibration start and interrupts for calibration unit for WDT.

- **CUCR1:Address 04C4$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
-								
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	Attribute

7	6	5	4	3	2	1	0	bit
-	-		STRT	-		INT	INTEN	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,WX	R0,WX	R,W	R0,WX	R0,WX	R,W0	R/W	Attribute

[bit15 to bit8] Undefined

These bits always read "1". Writing has no effect on operation.

[bit7] Reserved

"0" should be written to this bit.

[bit6, bit5] Undefined

These bits always read "0". Writing has no effect on operation.

[bit4] STRT (calibration STaRT) Calibration start

This bit starts counters driven by main clock and CR clock. INT bit will be set at the end of comparison. Setting "0" to this bit stops comparison. While comparing, writing "1" to this bit will not take effect.

STRT	Function
"0" write	Stops comparison.
"1" write	Starts comparison.

This bit will be cleared to "0" at the end of comparison.

[bit3, bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[bit1] INT (calibration INTerrupt) Interrupt

INT bit will be set to "1" at the end of comparison. If INTEN bit is set, an interrupt occurs. This bit is cleared when "0" is written.

[bit0] INTEN (calibration INTerrupt Enable) Interrupt enable

This bit configures whether to generate an interrupt or not when INT bit is set.

INTEN	interrupt
0	Disabled
1	Enabled

26.4.2 CUTD1: (Calibration Unit Timer Data register 1) CR Clock Timer Data Register

This register configures the time interval for driving CR clock driven counter.

- **CUTD1:Address 04C6$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
TDD[15:8]								
1	1	0	0	0	0	1	1	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
TDD[7:0]								
0	1	0	1	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] TDD15 to TDD0 (Timer Data Data field) Timer data

These bits configure the comparison time interval in number of CR clocks.

26.4.3 CUTR1: (Calibration Unit Timer Result register 1) Main Oscillation Timer Data Register 1

This register displays the value of the main clock driven counter in the interval set with CUTD1.

- **CUTR1:Address 04C8$_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
-								
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

23	22	21	20	19	18	17	16	bit
TDR[23:16]								
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

15	14	13	12	11	10	9	8	bit
TDR[15:8]								
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
TDR[7:0]								
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

[bit31 to bit24] Undefined

These bits always read "0". Writing has no effect on operation.

[bit23 to bit0] TDR23 to TDR0 (Timer Data Register) Timer data

These bits display the number of the count in the comparison interval. Read out after comparing. The value read would be indefinite while comparing. Data writing is ineffective.

26.4.4 CRTR: (CR oscillator calibration Trimming Register) CR Oscillation Trimming Setting Register

This register configures trimming values of CR oscillation circuit.

- **CRTR: Address 0570_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
TRD[7:0]								
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 to bit0] TRD (TRimming Data) Trimming value

TRD7	TRD6	TRD5	TRD4	TRD3	TRD2	TRD1	TRD0	Trimming value *	n value
0	0	0	0	0	0	0	0	-48.01%	0
0	0	0	0	0	0	0	1	-47.61%	1
0	0	0	0	0	0	1	0	-47.23%	2
• •	• •	• •	• •	• •	• •	• •	• •	• •	• •
0	1	1	1	1	1	1	1	0% [Initial value]	127
• •	• •	• •	• •	• •	• •	• •	• •	• •	• •
1	1	1	1	1	1	0	1	+45.62%	253
1	1	1	1	1	1	1	0	+45.98%	254
1	1	1	1	1	1	1	1	+46.37%	255

* You should follow the steps explained in 5.2 because the values may be changed for conditions such as temperature.

26.5 Operation

26.5.1 WDT1 Calibration (CR Clock Calibration)

Calculate the trimming values as follows.

(1) Set TDR[7:0] to "00000000", then execute calibration unit, and get the CUTR value. Make the CR oscillation frequency calculated from this CUTR value to be "Fmin".

(2) Set TDR[7:0] to "11111111", then execute calibration unit, and get the CUTR value. Make the CR oscillation frequency calculated from this CUTR value to be "Fmax".

(3) Substitute 0 to 255 for n in the formula below. Find n which makes Fer the minimum, which is the trimming value.

$Fstep = (Fmax - Fmin) \div 255$

$Fer = | (100kHz) - (Fmin + Fstep \times n) |$

26.5.2 Notes

A counter value which inputs the factor during the standby mode transition period will be invalid. Write "0" to STRT bit and stop, then write "1" once again.

CHAPTER 27 DMA CONTROLLER (DMAC)

27.1 Overview

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 CPU, thus increases the system performance.

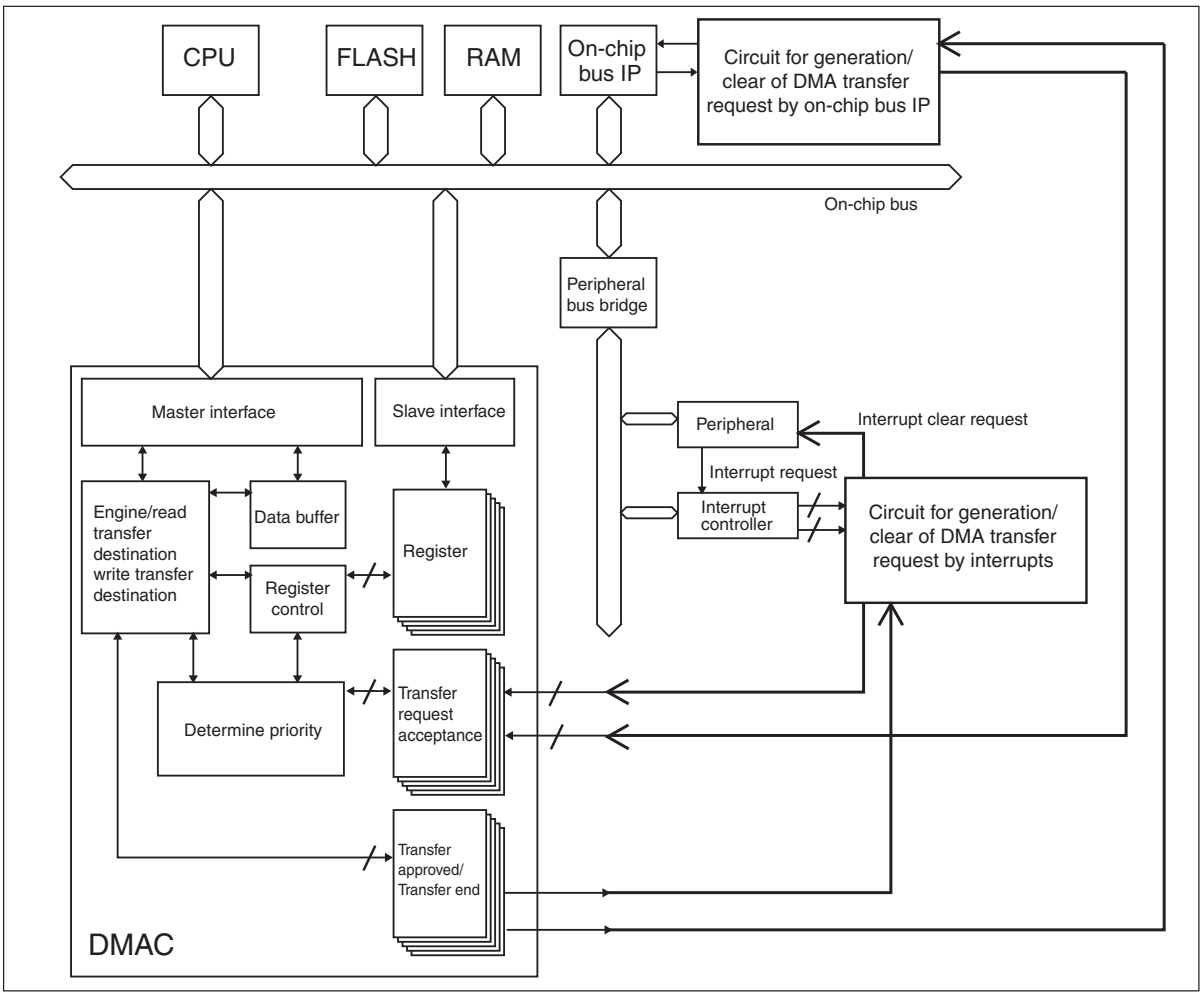
27.2 Features

- Channels : 8 channels
- Address space : 32-bit address space (4 GB)
- Transfer mode : Block/burst transfer
- Address update : Increment/Decrement/Fixed (Address increment/decrement range = 1, 2, 4 fix)
- 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 28 GENERATION AND CLEARING OF DMA TRANSFER REQUESTS".),
  Transfer request by on-chip bus IP (DMAC channel numbers corresponding to each on-chip bus IP cannot be selected. See Section "27.5.2 Table for On-chip Bus IP and DMAC Channels".)
- Transfer stop request : Transfer stop request by interrupts
- Reload function : All channels can be specified for reload.
  - Transfer source address reload
  - Transfer destination address reload
  - Transfer count reload
- Priority : Fixed (ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7), or round robin
- Interrupt request : Normal completion interrupt requests, abnormal completion interrupt requests, and transfer suspend interrupt requests by transfer stop requests can be generated.

Code: FR81S09_DMA-1v0-91580MS-1-E

27.3 Configuration

Figure 27.3-1 Block Diagram of DMA Controller



27.4 Registers

Table 27.4-1 Base Address (Base_addr)

Channel number	Base_addr	Remark
0	0x0C00	DMA ch.0 control register
1	0x0C10	DMA ch.1 control register
2	0x0C20	DMA ch.2 control register
3	0x0C30	DMA ch.3 control register
4	0x0C40	DMA ch.4 control register
5	0x0C50	DMA ch.5 control register
6	0x0C60	DMA ch.6 control register
7	0x0C70	DMA ch.7 control register

Table 27.4-2 Register Map (1 / 2)

Address	Register				Register function
	+0	+1	+2	+3	
0x0C00	DCCR0				DMA channel control register 0
0x0C04	DCSR0		DTCR0		DMA channel status register 0 DMA transfer count register 0
0x0C08	DSAR0				DMA transfer source address register 0
0x0C0C	DDAR0				DMA transfer destination address register 0
0x0C10	DCCR1				DMA channel control register 1
0x0C14	DCSR1		DTCR1		DMA channel status register 1 DMA transfer count register 1
0x0C18	DSAR1				DMA transfer source address register 1
0x0C1C	DDAR1				DMA transfer destination address register 1
0x0C20	DCCR2				DMA channel control register 2
0x0C24	DCSR2		DTCR2		DMA channel status register 2 DMA transfer count register 2
0x0C28	DSAR2				DMA transfer source address register 2
0x0C2C	DDAR2				DMA transfer destination address register 2
0x0C30	DCCR3				DMA channel control register 3
0x0C34	DCSR3		DTCR3		DMA channel status register 3 DMA transfer count register 3
0x0C38	DSAR3				DMA transfer source address register 3
0x0C3C	DDAR3				DMA transfer destination address register 3
0x0C40	DCCR4				DMA channel control register 4
0x0C44	DCSR4		DTCR4		DMA channel status register 4 DMA transfer count register 4
0x0C48	DSAR4				DMA transfer source address register 4
0x0C4C	DDAR4				DMA transfer destination address register 4
0x0C50	DCCR5				DMA channel control register 5

Table 27.4-2 Register Map (2 / 2)

Address	Register				Register function
	+0	+1	+2	+3	
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
0x0DF4	Reserved	Reserved	DNMIR	DILVR	DMA transfer suppression NMI flag register DMA transfer suppression interrupt level register
0x0DF8	DMACR				DMA control register

27.4.1 DMACR: (DMA Control Register) DMA Control Register

DMA control register is a 32-bit register to control entire DMAC (all channels). This register must be accessed in 32-bit mode.

- **DMACR: Address $0DF8_H$ (Access: Word)**

31	30	29	28	27	26	25	24	bit
DME	-							
0	0	0	0	0	0	0	0	Initial value
R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

23	22	21	20	19	18	17	16	bit
-								
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

15	14	13	12	11	10	9	8	bit
AT	-							
0	0	0	0	0	0	0	0	Initial value
R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

7	6	5	4	3	2	1	0	bit
-								
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

[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", it operates according to the settings for each channel.

If "0" is written while a DMA transfer is in progress, A DMA transfer stops operation in blocks specified in DCCRx:BLK.

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

[bit30 to bit16] Undefined

These bits always read "0". Writing has no effect on operation.

[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), the priority of the channel which started the transfer is the lowest and the priority of following channels increases one by one. Decision on priority is done by blocks specified in DCCRx:BLK regardless of the priority setting.

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

[bit14 to bit0] Undefined

These bits always read "0". Writing has no effect on operation.

27.4.2 DCCR0 to DCCR7: (DMA Channel Control Register 0 to 7) DMA Channel Control Registers 0 to 7

DMA channel control registers are 32-bit registers to control operation of DMAC channels, which exists independently for each channel. These registers must be accessed in 32-bit mode.

- **DCCR0 to DCCR7:Address BASE + 0000_H (Access: Word)**

31	30	29	28	27	26	25	24	bit
CE	-				AIE	SIE	NIE	
0	0	0	0	0	0	0	0	Initial value
R,W	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

23	22	21	20	19	18	17	16	bit
-		RS[1:0]		-		TM[1:0]		
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
ST	SAR	SAC[1:0]		DT	DAR	DAC[1:0]		
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
TCR	-	TS[1:0]		BLK[3:0]				
0	0	0	0	0	0	0	0	Initial value
R/W	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[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, 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, CE bit will not be automatically cleared if transfer count reload (DCCRx:TCR) is specified. When transfer count reload is disabled, CE bit will be cleared when all transfers are finished. If "0" is written while the operation is going on regardless of the transfer source, stops transfer in blocks specified in DCCRx:BLK. When writing "1" again and detecting a new transfer request, the operation restarts.

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

[bit30 to bit27] Undefined

These bits always read "0". Writing has no effect on operation.

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

This bit controls the generation of interrupts when setting the prohibited values to DMA channel control register (DCCR). 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

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

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

[bit25] SIE (Stop Interrupt Enable) Transfer suspend interrupt enable 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. For the interrupt factor, refer to the status register (DCSRx).

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

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

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

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

[bit23, bit22] Undefined

These bits always read "0". Writing has no effect on operation.

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

These bits select the transfer request source for the channel.

Settings of RS[1:0]=2’b11 are prohibited as there are no transfers between channel 2 and 7 that specify the on-chip bus IP as transfer request source.

RS[1:0]	DMA transfer request source
00	Software (initial value)
01	Interrupts
10	Reserved (setting disabled)
11	On-chip bus IP

[bit19, bit18] Undefined

These bits always read "0". Writing has no effect on operation.

[bit17, bit16] TM[1:0] (Transfer Mode): Transfer mode

These bits specify the DMA transfer mode.

TM[1:0]	Transfer mode
00	Block transfer (initial value)
01	Burst transfer
10	Reserved (setting disabled)
11	Demand transfer (setting disabled/no support for this product type)

[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). 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, returns the transfer source address register value to the initial value at the end of the transfer. When disabling a reload, the transfer source address register value will point to the next value to the last address at the end of the transfer.

SAR	Transfer Source address reload specified
0	Reload disabled (initial value)
1	Reload

[bit13, bit12] SAC[1:0] (Source Address Count): Transfer Source address count

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

SAC[1:0]	Transfer Source address count
00	Address increment (initial value)
01	Address decrement
10	Reserved (setting disabled)
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).

DT	Transfer destination type
0	See "■ Setting the ST bit (Transfer source type) and DT bit (Transfer destination type)".
1	

[bit10] DAR (Transfer 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 value will point to the next value 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 disabled)
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] Undefined

This bit always reads "0". Writing has no effect on operation.

[bit5, bit4] TS (Transfer Size): Transfer size

These bits specify transfer size. DMA transfers will be performed once with the bit width specified here.

TS[1:0]	Transfer size
00	8 bits: byte (initial value)
01	16 bits: halfword
10	32 bits: word
11	Reserved (setting disabled)

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[3:0] (Block Size): Block size

These bits specify block size. 1 block transfer will be repeated for the number of blocks of transfer size specified with DCCRx:TS bit.

BLK[3:0]	Block size
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

27.4.3 DCSR0 to DCSR7: (DMA Channel Status Register 0 to 7) DMA Channel Status Registers 0 to 7

These registers are 16-bit registers to indicate the status for each DMAC channel, which exist independently for each channel. These registers must be accessed in 16-bit mode.

- **DCSR0 to DCSR7: Address BASE + 0004_H (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
CA	-							
0	0	0	0	0	0	0	0	Initial value
R,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
-					AC	SP	NC	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	R,W	Attribute

[bit15] CA (Channel Active) Channel active

This bit indicates the operation of the channels. Writing "1" to the corresponding DCCRx:CE bit for the channel makes it in 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 ineffective.

CA	Channel operating state
0	Stop state (initial value)
1	Channel operating

[bit14 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2] AC (Abnormal Completion): Abnormal completion state

This bit indicates that a prohibited value has been set to DMA channel control register (DCCR). 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

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

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 transfer suspend interrupt (DCCRx:SIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is ineffective.

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

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

27.4.4 DTCR0 to DTCR7:(DMA Transfer Count Register 0 to 7) DMA Transfer Count Registers 0 to 7

These registers are 16-bit registers to indicate the transfer count for each DMAC channel, which exists independently for each channel. These registers must be accessed in 16-bit mode.

- **DTCR0 to DTCR7: Address BASE + 0006_H (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
DTC[15:8]								
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
DTC[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 bit0] DTC[15:0] (DMA Transfer Count) DMA transfer count

These registers indicate the number of transfer times. They decrement at the end of each block transfer and stop the transfer when "0" is reached. 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.

27.4.5 DSAR0 to DSAR7: (DMA Source Address Registers 0 to 7) DMA Transfer Source Registers 0 to 7

These registers are 32-bit registers to indicate the transfer source address of each DMAC channel, which exist independently for each channel. These registers must be accessed in 32-bit mode.

- **DSAR0 to DSAR7: Address BASE + 0008_H (Access: Word)**

31	30	29	28	27	26	25	24	bit
DSA[31:24]								
X	X	X	X	X	X	X	X	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
DSA[23:16]								
X	X	X	X	X	X	X	X	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
DSA[15:8]								
X	X	X	X	X	X	X	X	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
DSA[7:0]								
X	X	X	X	X	X	X	X	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[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 (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)".

27.4.6 DDAR0 to DDAR7: (DMA Destination Address Registers 0 to 7) DMA Transfer Destination Registers 0 to 7

These registers are 32-bit registers to indicate the transfer destination address of each DMAC channel, which exist independently for each channel. These registers must be accessed in 32-bit mode.

- **DDAR0 to DDAR7: Address BASE + 000C$_H$ (Access: Word)**

31	30	29	28	27	26	25	24	bit
DDA[31:24]								
X	X	X	X	X	X	X	X	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
DDA[23:16]								
X	X	X	X	X	X	X	X	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
DDA[15:8]								
X	X	X	X	X	X	X	X	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
DDA[7:0]								
X	X	X	X	X	X	X	X	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

- **[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 (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)".

27.4.7 DNMIR: (DMA-halt by NMI Register) DMA Transfer Suppression NMI Flag Register

This register is 8-bit register to suppress DMA transfer by the user NMI. This register must be accessed in 8-bit mode.

- **DNMIR: Address $0DF6_H$ (Access: Byte)**

7	6	5	4	3	2	1	0	bit
NMIH	-						NMIHD	
0	0	0	0	0	0	0	0	Initial value
R,W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	Attribute

[bit7] NMIH (NMI Halt): DMA suppression flag (by NMI cause)

If NMIHD is "0", this flag shows an occurrence of 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 ineffective.

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] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] NMIHD (NMI Halt Disable) DMA suppress 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 suppress control
0	Stops the DMA transfer by the user NMI. (initial value)
1	Does not stop the DMA transfer by the user NMI.

27.4.8 DILVR: (DMA-halt by Interrupt Level Register) DMA Transfer Suppress Level Register

This register is 8-bit register to control the DMA transfer suppression by peripheral interrupts. This register must be accessed in 8-bit mode.

- **DILVR: Address $0DF7_H$ (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-			LVL[4]	LVL[3:0]				
0	0	0	1	1	1	1	1	Initial value
R0,WX	R0,WX	R0,WX	R1,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit5] Undefined

These bits always read "0". Writing has no effect on operation.

[bit4 to bit0] LVL (Level) DMA suppress 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 suppress control
11111	Suppresses the DMA transfer when all peripheral interrupt requests are 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.

27.5 Operation

27.5.1 Configuration

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

27.5.1.1 Common Items for All Channels

The following explains the register settings for entire DMAC control.

■ DMA operation enable

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

- Disable DMA operations. (DMACR:DME=0)
- Enable DMA operations. (DMACR:DME=1)

■ Channel priority

A channel priority can be set by the DMACR:AT.

- Fixed (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 that precedes the DMA transfer can be set by DILVR:LVL when an interrupt is generated. Levels 0x1F to 0x10 are allowed.

27.5.1.2 Separate Items for Each Channel

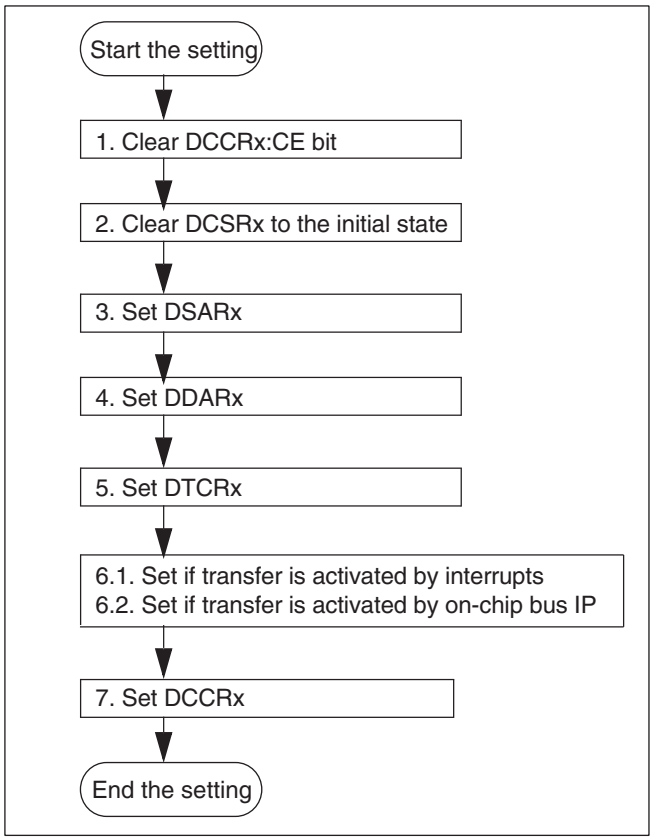
The following explains both the items to be set separately for each channel and the register setup procedure.

■ Register setup procedure

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

(1) Clear the DCCRx:CE bit to disable the channel operation.
(2) Clear each bit of DCSRx register to initialize the channel status flag.
(3) Set the transfer source address (to be used when the transfer starts) in the DSARx register.
(4) Set the transfer destination address (to be used when the transfer starts) in the DDARx register.
(5) Set the transfer count in the DTCRx register. This count must be 1 or a larger value.
(6.1) If transfer is started by a peripheral interrupt, the occurrence of each peripheral interrupt must be enabled and the ICSEL and IORR registers must be set. (For the ICSEL and IORR registers, see "CHAPTER 28 GENERATION AND CLEARING OF DMA TRANSFER REQUESTS".)
(6.2) If transfer is started by an on-chip bus IP, enable DMA transfer request by each on-chip bus IP.
(7) Set the DCCRx register. During this time, the channel operation is enabled when the DCCRx:CE bit is set.

Figure 27.5-1 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.

■ 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 permission 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)
- Request by on-chip bus peripheral (DCCRxRS=11)(* x is either 0 or 1 only)

■ Transfer mode setting

Set the DMA transfer mode using the DCCRx:TM.

- Block transfer (DCCRx:TM=00)
- Burst transfer (DCCRx:TM=01)

■ Setting the ST bit (Transfer source type) and DT bit (Transfer destination type)

Set them by following the table definition below. The DMA transfer is not supported in combinations (5) and (9).

Table 27.5-1 ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type) Setting

	Combination of transfer request source, transfer source, and transfer destination			DMA support	ST and DT bit setting
	Transfer request source (DCCR.RS[1:0])	Transfer source address (DSAR)	Transfer destination address (DDAR)		
①	Request by software (DCCR.RS=00)	Any combination		Supported	ST=0, DT=0
②	Peripheral bus and peripheral interrupt <Interrupt from peripheral bus area *1> (DCCR.RS=01)	Peripheral bus area *1	All areas except peripheral bus area	Supported	ST=1, DT=0
③		All areas except peripheral bus area	Peripheral bus area *1	Supported	ST=0, DT=1
④		Peripheral bus area *1	Peripheral bus area *1	Supported	ST=0, DT=1
⑤		All areas except peripheral bus area	All areas except peripheral bus area	Not supported	-
⑥	On-chip bus and peripheral interrupt <Interrupt from on-chip bus area *2> (DCCRxRS=11)	On-chip bus area *2	All areas except on-chip bus area	Supported	ST=1, DT=0
⑦		All areas except on-chip bus area	On-chip bus area *2	Supported	ST=0, DT=1
⑧		On-chip bus area *2	On-chip bus area *2	Supported	ST=0, DT=1
⑨		All areas except on-chip bus area	All areas except on-chip bus area	Not supported	-

If the ST and DT bits are set in a combination other than above, the interrupt may not be cleared automatically after occurrence of DMA transfer request.

For the address areas indicated with *1 and *2 and their functions, see Table 27.5-2.

Table 27.5-2 Support for Address Area

	*1 Peripheral bus area	*2 On-chip bus area
Address area	000000_H to $0002FF_H$ 000400_H to $0005FF_H$ $000E00_H$ to $0001FFF_H$	000900_H to $000AFF_H$ 002000_H to `$00EFFF_H$ 030000_H to $03FFFF_H$

■ Transfer address reload setting

Using the DCCRx:SAR, set the reload control of transfer source address at the end of transfer.

- The transfer source address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.SAR=0)
- The transfer source address is returned to the initial value at the end of transfer. (DCCRx:SAR=1)
- Using the DCCRx:DAR, set the reload control of transfer destination address at the end of transfer.
- The transfer destination address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.DAR=0)
- The transfer destination address is returned to the initial value at the end of transfer. (DCCRx:DAR=1)

■ Transfer address update setting

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

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

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

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

■ Transfer count reload setting

Using the DCCRx:TCR, set the reload control of transfer count at the end of transfer.

- The transfer count is not reloaded after the transfer. (After the normal completion of transfer, the transfer count is set to "0".) (DCCRx.TCR=0)
- The transfer count is returned to the initial value at the end of transfer. (DCCRx:TCR=1)

■ Transfer size and block size setting

To set a data 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 bits, 16 bits, or 32 bits).

- 8 bits (DCCRx:TS=00)
- 16 bits (DCCRx:TS=01)
- 32 bits (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.

27.5.1.3 Operations

This section explains the DMAC operations as follows.

(1) Channel status check

(2) Data transfer

■ (1) 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 ineffective.

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.

■ (2) 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 bits, 16 bits, or 32 bits) 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.

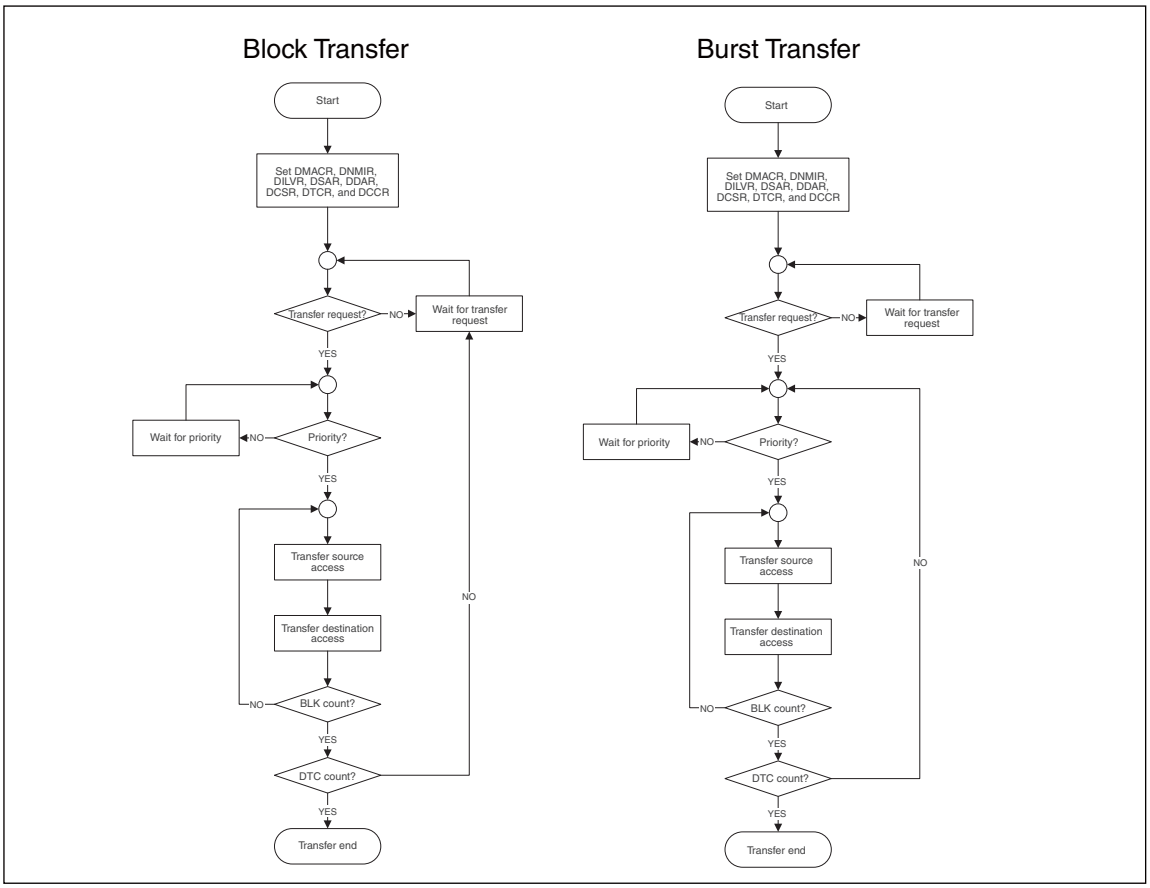
• Block transfer mode

1-time transfer request causes the 1-time 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 is transferred for the number of times being set by the block size.

• 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 is transferred continuously for the block size × number of times.)

Figure 27.5-2 Each Transfer Mode



Transfer request

The transfer request has a request by software or a request by interrupt. The following explains the relationship between the transfer request detect conditions and the transfer mode.

• Request by software

If 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. If the DMA operation is enabled (DMACR:DME=1), the priority is determined and the data transfer is started immediately.

When a transfer stop request is asserted from the peripheral, a transfer request is not detected.

Also, an interrupt vector to be used for transfer request must be set for each channel. See "CHAPTER 28 GENERATION AND CLEARING OF DMA TRANSFER REQUESTS".

Table 27.5-3 Relationship between Transfer Request Detect 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
Request by on-chip bus IP	Edge detection	Edge detection

Also, the relationship between the detected transfer request and the DMACR:DME and DCCRx:CE bits is given on Table 27.5-4. If the DME bit or CE bit is cleared during transfer, the block transfer is stopped.

Table 27.5-4 Relationship between Transfer Requests and DME/CE Bits

		DME bit	CE bit
DME/CE clear		The already detected transfer request is not cleared.	The already detected transfer request is cleared.
DME/CE setting after the transfer interrupt	Block transfer	When a new transfer request is detected, the data transfer is restarted based on the priority.	When a new transfer request is detected, the data transfer is restarted based on the priority.
	Burst transfer	When the DME bit is set, the data transfer is restarted immediately based on the priority.	

● Standby recovery request by DMA transfer request

If the MCU receives a transfer request in the standby mode, the DMAC requests the MCU to recover from the standby mode. If data transfer is enabled and if a transfer request is asserted by the transfer request source, a standby recovery is requested.

● Channel priority

If multiple transfer requests are issued, the DMAC starts data transfer on the channel having the highest priority. The channel priority can be fixed or can be set by round robin. The priority is determined for each block transfer or when data transfer ends.

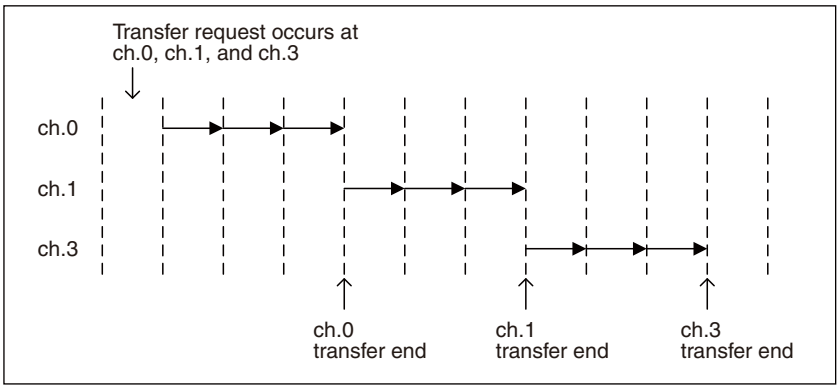
• Fixed priority (DMACR:AT=0)

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

Example 1: If transfer requests are issued on channels 0, 1 and 3 simultaneously, data transfer starts from channel 0. When data transfer ends on channel 0, the next data transfer starts on channel 1. After data transfer on channel 1, the next data transfer starts on channel 3. The following gives transfer examples. Dotted lines in the figure show the block delimiters.

Transfer request : Requests are issued for channels 0, 1 and 3 simultaneously.
Setting : Channels 0, 1 and 3 are set to the burst transfer mode, and data transfer occurs 3 times.

Figure 27.5-3 Data Transfer Example 1 If Channel Priority Is Fixed

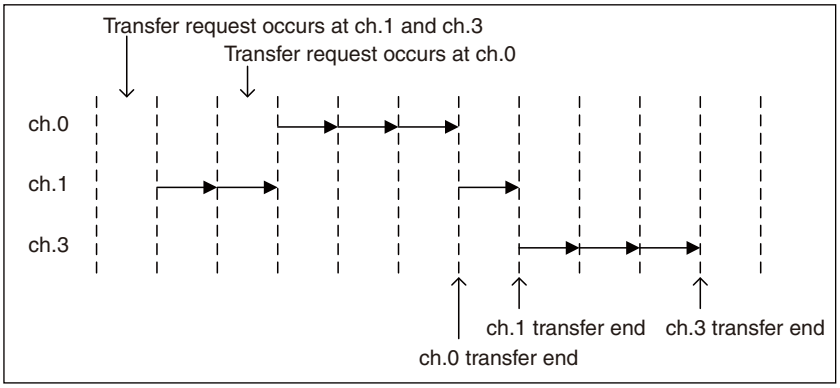


Example 2: If transfer requests are issued simultaneously for channel 1 and channel 3 and if a transfer request on channel 0 is issued during data transfer on channel 1, the data transfer on channel 1 is temporarily stopped and data transfer on channel 0 is started. During this time, the channel transition occurs in units of blocks. When the requested data transfer ends on channel 0, the data transfer is started on channel 1. Dotted lines in the figure show the block delimiters.

Transfer request : Requests are issued for channels 1 and 3 simultaneously. When data is transferred on channel 1, another request for transfer on channel 0 is issued.

Setting : Channels 0, 1 and 3 are set to the burst transfer mode, and data transfer occurs 3 times.

Figure 27.5-4 Data Transfer Example 2 If Channel Priority Is Fixed



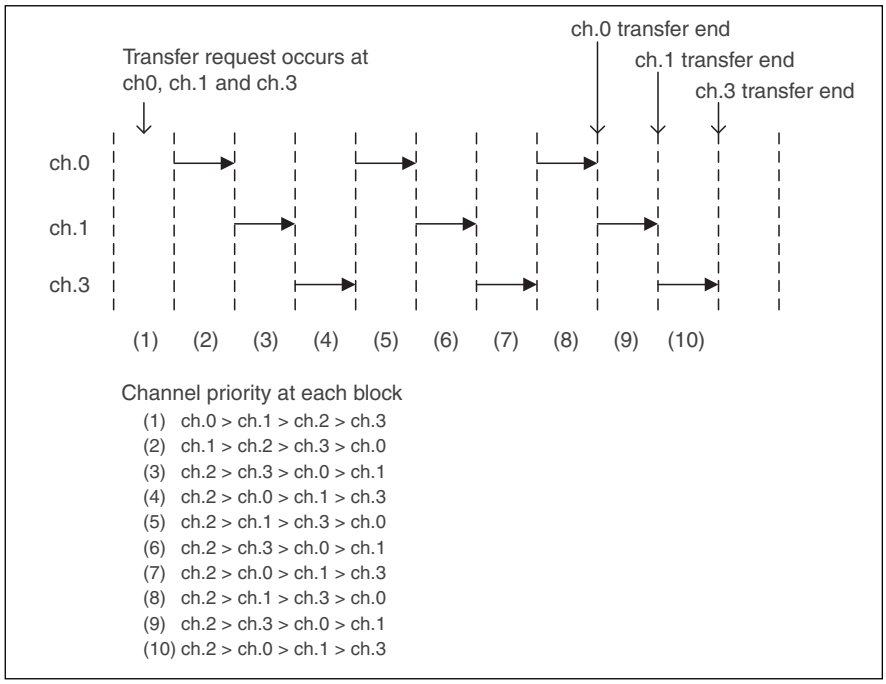
- **Round robin (DMACR:AT=1)**

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 channels 0, 1 and 3 simultaneously.

Channels 0, 1 and 3 are set to the burst transfer mode; and data transfer occurs 3 times.

Figure 27.5-5 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 27.5-5 shows the address increasing or decreasing width during address updating. If an overflow occurs due to address updating, the relevant bit is discarded.

Table 27.5-5 Updating of Transfer Source Address And Transfer Destination Address (1 / 2)

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 bits ("00")	Increments by 1	Increments by 1
		16 bits ("01")	Increments by 2	Increments by 2
		32 bits ("10")	Increments by 4	Increments by 4
	Decrements ("01")	8 bits ("00")	Increments by 1	Decrements by 1
		16 bits ("01")	Increments by 2	Decrements by 2
		32 bits ("10")	Increments by 4	Decrements by 4
	Fixed ("11")	8 bits ("00")	Increments by 1	Not updated
		16 bits ("01")	Increments by 2	
		32 bits ("10")	Increments by 4	

Table 27.5-5 Updating of Transfer Source Address And Transfer Destination Address (2 / 2)

Address setting		Transfer size (TS)	Address updating for each data transfer	
Transfer source (SAC)	Transfer destination (DAC)		Transfer source (DSA)	Transfer destination (DDA)
Decrements ("01")	Increments ("00")	8 bits ("00")	Decrements by 1	Increments by 1
		16 bits ("01")	Decrements by 2	Increments by 2
		32 bits ("10")	Decrements by 4	Increments by 4
	Decrements ("01")	8 bits ("00")	Decrements by 1	Decrements by 1
		16 bits ("01")	Decrements by 2	Decrements by 2
		32 bits ("10")	Decrements by 4	Decrements by 4
	Fixed ("11")	8 bits ("00")	Decrements by 1	Not updated
		16 bits ("01")	Decrements by 2	
		32 bits ("10")	Decrements by 4	
Fixed ("11")	Increments ("00")	8 bits ("00")	Not updated	Increments by 1
		16 bits ("01")		Increments by 2
		32 bits ("10")		Increments by 4
	Decrements ("01")	8 bits ("00")		Decrements by 1
		16 bits ("01")		Decrements by 2
		32 bits ("10")		Decrements by 4
	Fixed ("11")	8 bits ("00")		Not updated
		16 bits ("01")		
		32 bits ("10")		

● Reloading of transfer address

The DMAC can reload the transfer address after the specified number of data transfer has completed.

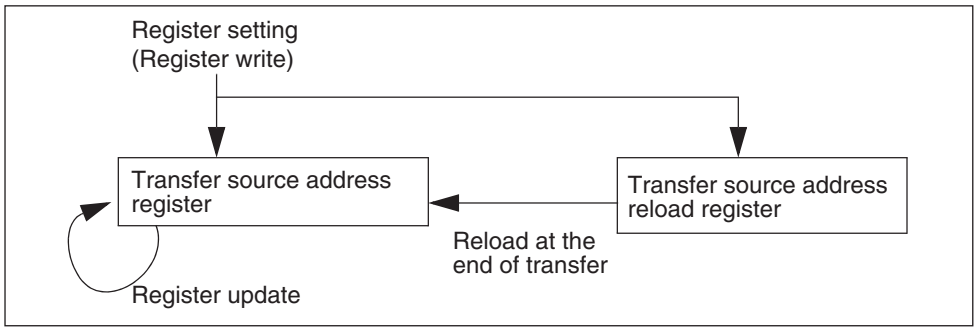
• Reloading of transfer source address

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

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

If the specified number of times of transfer is suspended or abnormally terminated, the DSARx:DSA indicates the next access address (after the terminated address) regardless of the reload setting of transfer source address.

Figure 27.5-6 Reloading of Transfer Source Address Register



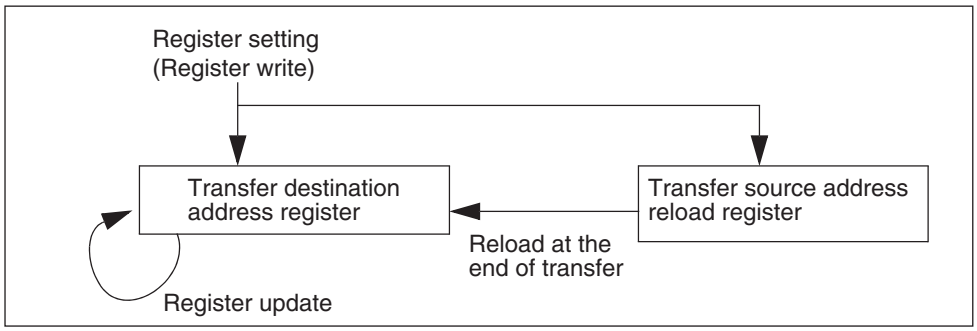
- **Reloading of transfer destination address register**

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

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

If the specified number of times of transfer is suspended or abnormally terminated, the DDARx:DDA indicates the next access address (after the terminated address) regardless of the reload setting of transfer destination address.

Figure 27.5-7 Reloading of Transfer Destination Address Register



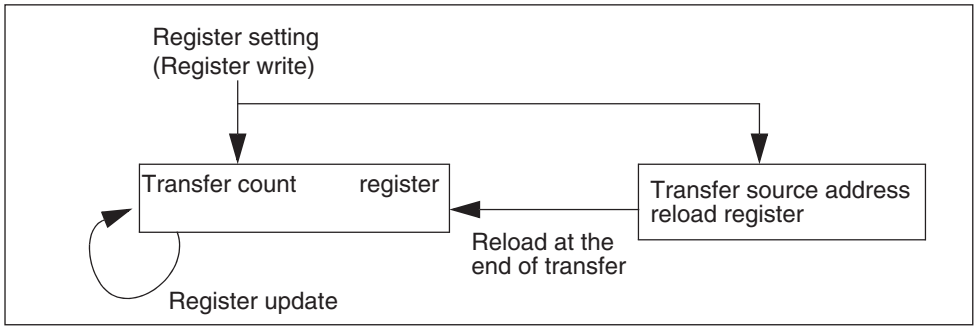
● Reloading of transfer count

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

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

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

Figure 27.5-8 Reloading of Transfer Count Address Register



The DCCRx:CE bit status varies after the data transfer, depending on the reload setting of transfer count. The following explains the relation between the transfer count reload setting and the transfer request source.

Table 27.5-6 DCCRx:CE Bit at the End of Transfer

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

● Transfer suspend

The DMAC suspends DMA transfer due to the following causes.

- An suspend as the DMACR:DME bit is cleared
- An suspend as the DCCRx:CE bit is cleared
- A transfer stop request from 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 suspend cause.

• An suspend 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.

• An suspend as the DCCRx:CE bit is cleared

If the DCCRx:CE bit is cleared, its 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) Serial interface (LIN2.1)
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 followings occur.

- The SP bit of DMA channel status registers (DCSR0 to DCSR7) is set to "1".
- The CE bit of DMA channel control registers (DCCR0 to DCCR7) is set to "0".
- The already detected transfer request is cleared.

While a transfer stop request being issued, a new transfer request is rejected. Restart the DMA transfer in the following procedure.

(1) Clear the 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 DCSR7) of the corresponding channel to "0".
(3) Set the CE bit of DMA channel control registers (DCCR0 to DCCR7) to "1".
(4) Issue a new transfer request.

Table 27.5-7 Settings to Restart the Interrupted 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) A transfer request is issued

● Transfer end

Data transfer can terminate normally or abnormally.

• Normal termination

After data transfer has completed for the specified number of times, set by the transfer count (DTCRx:DTC bit), the data transfer terminates normally. 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 something other than software from the transfer request source, the DCCRx:CE bit of this channel is not cleared.

Also, if the transfer count (DTCRx:DTC bit) is "0" and if the DCCRx:CE bit of the corresponding channel is set to "1", 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 during transfer suspend 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 cause or to clear the interrupt request.

• Interrupt request during normal termination

If the normal termination interrupt of a channel is enabled (DCCRx:NIE=1), the DCSRx:NC bit of the corresponding channel is set at normal termination. Also, an interrupt request is issued.

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

• Interrupt request during abnormal termination

If the abnormal termination interrupt of a channel is enabled (DCCRx:AIE=1), the DCSRx:AC bit of the corresponding channel is set at the abnormal termination. Also, an interrupt request is issued.

Clear the interrupt request by clearing the DCSRx:AC bit of the corresponding channel.

• A transfer suspend interrupt request by a transfer stop request

If a transfer suspend interrupt of a channel is enabled (DCCRx:SIE=1), the DCSRx:SP bit of the corresponding channel is set if data transfer is suspended by a transfer stop request. Also, an interrupt request is issued.

Clear the interrupt request by clearing the DCSRx:SP bit of the corresponding channel.

• Notes on the interrupt request clear

When an interrupt request is issued, check the status register (DCSRx) of the corresponding channel. If some status flags (DCSRx.AC/SP/NC) are "1", be careful when clearing the interrupt request.

If any flag (DCSR.AC/SP/NC) of status register is "1", the interrupt request cannot be cleared. When some status flags are "1", clear the interrupt request by clearing all flags, regardless of interrupt enable/disable bit (DCCR.AIE/SIE/NIE) settings.

DMA transfer suppressing

DMA transfer is suppressed due to the following causes.

- 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 cause.

- **DMA transfer suppress request from DSU/OCD (for debugging)**

If a DMA transfer suppress request from the DSU/OCD is asserted, new data transfer does not start. The current DMA transfer is stopped in units of blocks. The DMA transfer suppress request from the DSU/OCD is not acknowledged.

- **DMA transfer suppressing by NMI**

If the NMIHD bit is set to "0" and if a user NMI occurs, the NMIH flag is set. After the current block has been transferred, the DMA transfer is suppressed.

To restart the DMA transfer, set the NMIH flag to "0".

- **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 level drops to LVL[4:0] or lower level, the DMA transfer restarts.

Table 27.5-8 LVL[4:0] Settings to Suppress DMA Transfer

LVL[4:0]	DMA suppress control
11111	Suppresses the DMA transfer when all peripheral interrupt requests are 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.

27.5.2 Table for On-chip Bus IP and DMAC Channels

Each DMAC channel has been assigned the following on-chip bus IP.

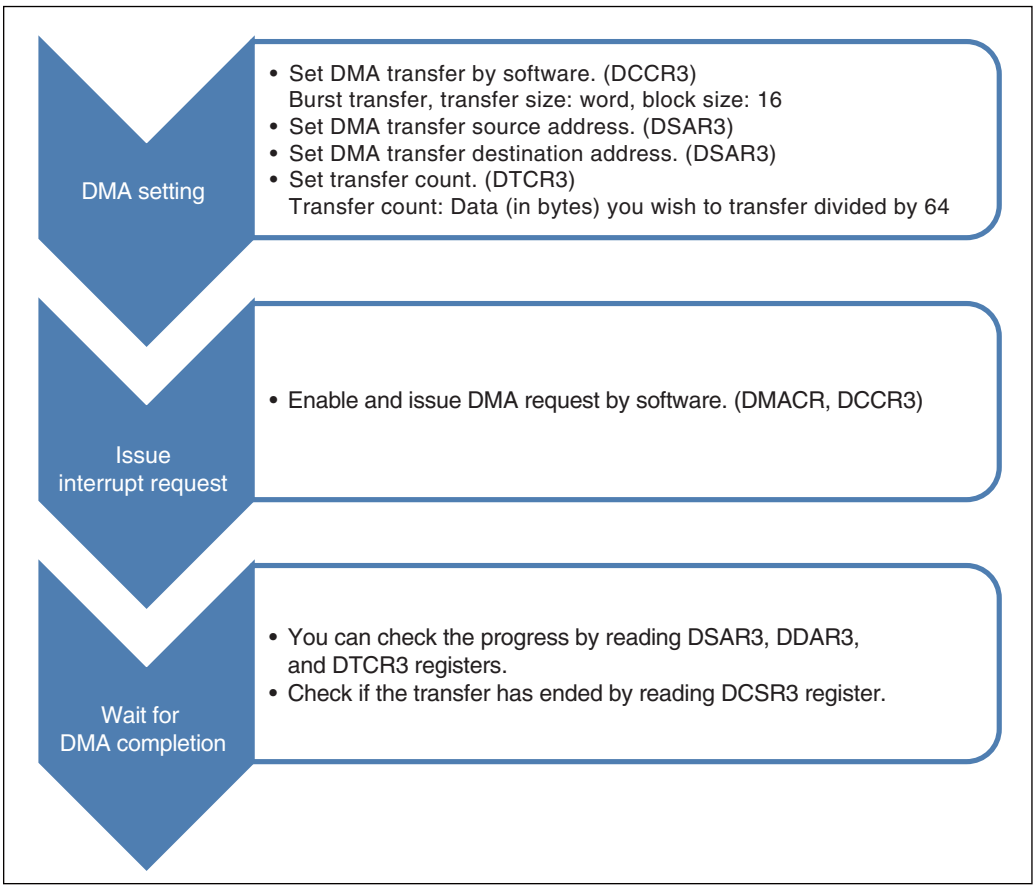
Table 27.5-9 Table for On-chip Bus IP and DMAC Channels

ch	On-chip bus IP
0	Transfer request by FlexRay output buffer busy (CIF1.DREQO)
1	Transfer request by FlexRay input buffer host busy (CIF1.DREQI)
2	There is no corresponding on-chip bus IP.
3	There is no corresponding on-chip bus IP.
4	There is no corresponding on-chip bus IP.
5	There is no corresponding on-chip bus IP.
6	There is no corresponding on-chip bus IP.
7	There is no corresponding on-chip bus IP.

27.6 DMA Example

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

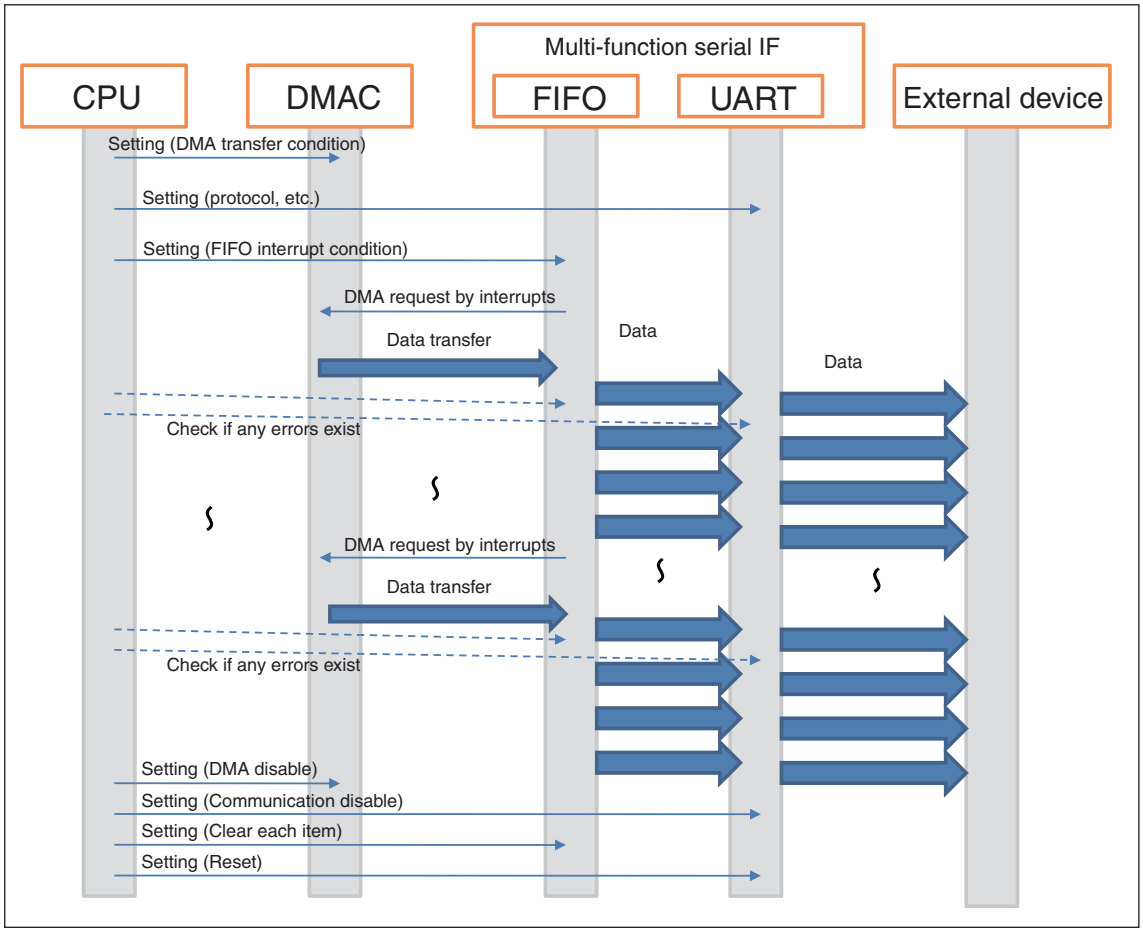
Figure 27.6-1 memcpy Example Using the DMA (Ch.3 Is used)



This example uses the external interrupt pin to control DMA transfer from the external companion chip.

This is a communication example via the multi-function serial interface that uses the DMA transfer. 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 27.6-2 Communication Example via the Multi-function Serial Interface That Uses DMA



CHAPTER 28 GENERATION AND CLEARING OF DMA TRANSFER REQUESTS

28.1 Overview

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.

28.2 Features

■ Transfer Request Generation Specification

For each 8-channel DMA transfer request, you need to specify what interrupt from interrupt vector numbers 0x10 to 0x3F is used to generate the DMA transfer request.

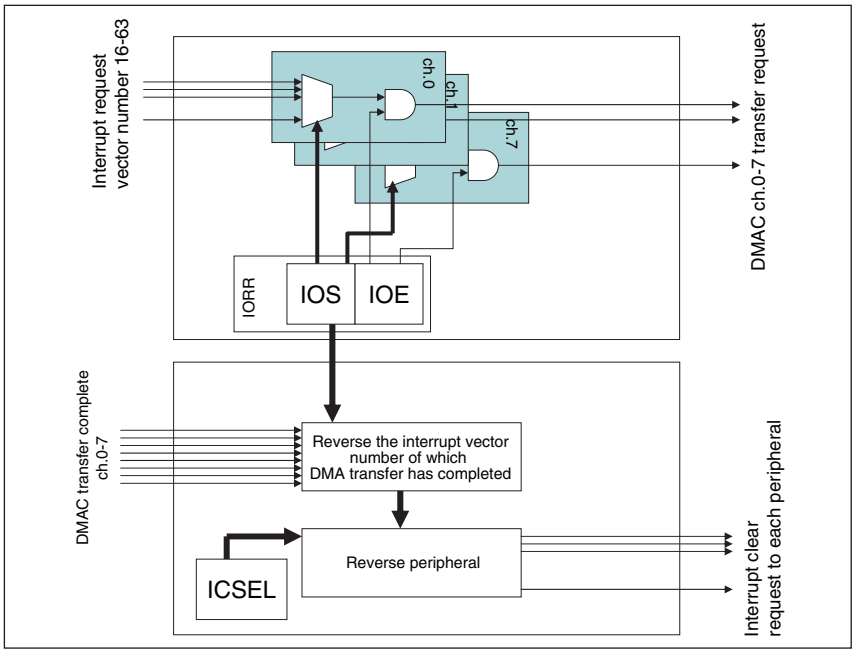
■ Interrupt Clearing Setting

After the DMA transfer ends, the interrupt source peripheral that has issued the interrupt request to be cleared is identified if the transfer request source is a vector number to which multiple interrupt source peripherals belong.

Code: BICSEL-0v3-91580MS-1-E

28.3 Configuration

Figure 28.3-1 Block Diagram of Generation and Clearing of DMA Transfer Request



28.4 Registers

Table 28.4-1 Register Map (1 / 2)

Address	Register				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 #36)
0x0404	ICSEL4	ICSEL5	ICSEL6	ICSEL7	DMA clear request register 4(for vector number #38) DMA clear request register 5(for vector number #39) DMA clear request register 6(for vector number #40) DMA clear request register 7(for vector number #41)
0x0408	ICSEL8	ICSEL9	ICSEL10	ICSEL11	DMA clear request register 8(for vector number #42) DMA clear request register 9(for vector number #43) DMA clear request register 10(for vector number #44) DMA clear request register 11(for vector number #45)
0x040C	ICSEL12	ICSEL13	ICSEL14	ICSEL15	DMA clear request register 12(for vector number #46) DMA clear request register 13(for vector number #47) DMA clear request register 14(for vector number #48) DMA clear request register 15(for vector number #49)
0x0410	ICSEL16	ICSEL17	ICSEL18	ICSEL19	DMA clear request register 16(for vector number #50) DMA clear request register 17(for vector number #51) DMA clear request register 18(for vector number #52) DMA clear request register 19(for vector number #53)
0x0414	ICSEL20	ICSEL21	ICSEL22	ICSEL23	DMA clear request register 20(for vector number #54) DMA clear request register 21(for vector number #55) DMA clear request register 22(for vector number #56) DMA clear request register 23(for vector number #57)

Table 28.4-1 Register Map (2 / 2)

Address	Register				Register function
	+0	+1	+2	+3	
0x0418	ICSEL24	ICSEL25	ICSEL26	ICSEL27	DMA clear request register 24(for vector number #58) DMA clear request register 25(for vector number #59) DMA clear request register 26(for vector number #60) DMA clear request register 27(for vector number #61)
0x0490	IORR0	IORR1	IORR2	IORR3	IO transfer request register 0 IO transfer request register 1 IO transfer request register 2 IO transfer request register 3
0x0494	IORR4	IORR5	IORR6	IORR7	IO transfer request register 4 IO transfer request register 5 IO transfer request register 6 IO transfer request register 7

28.4.1 ICSEL0 (Interrupt Clear SELect register 0) DMA request clear register 0

- **ICSEL0: Address 0400_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL00[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL00[2:0] Clear selection bits

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

ICSEL00[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

(Note) On the MB91F583AS/F584AS/F585AS, setting of CSEL00[2:0]= "111" is prohibited.

28.4.2 ICSEL1 (Interrupt Clear SELect register 1) DMA request clear register 1

- **ICSEL1: Address 0401_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL01	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL01 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #17).

ICSEL01	Clear target
0	Reload timer 0
1	Reload timer 1

28.4.3 ICSEL2 (Interrupt Clear SELect register 2) DMA request clear register 2

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

7	6	5	4	3	2	1	0	bit
-							ICSEL02	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL02 Clear selection bit

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

ICSEL02	Clear target
0	Reload timer 2
1	Reload timer 3

28.4.4 ICSEL3 (Interrupt Clear SELect register 3) DMA request clear register 3

- **ICSEL3: Address 0403_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL03	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL03 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #36).

ICSEL03	Clear target
0	Main timer
1	PLL timer

28.4.5 ICSEL4 (Interrupt Clear SELect register 4) DMA request clear register 4

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

7	6	5	4	3	2	1	0	bit
-							ICSEL04	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL04 Clear selection bit

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

ICSEL04	Clear target
0	Up/ down counter 0
1	Up/ down counter 1

28.4.6 ICSEL5 (Interrupt Clear SELect register 5) DMA request clear register 5

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

7	6	5	4	3	2	1	0	bit
-							ICSEL05	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL05 Clear selection bit

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

ICSEL05	Clear target
0	Free-run timer 0 (0 detection)
1	Free-run timer 0 (compare clear)

28.4.7 ICSEL6 (Interrupt Clear SELect register 6) DMA request clear register 6

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

7	6	5	4	3	2	1	0	bit
-							ICSEL06	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL06 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #40).

ICSEL06	Clear target
0	Free-run timer 1 (0 detection)
1	Free-run timer 1 (compare clear)

28.4.8 ICSEL7 (Interrupt Clear SELect register 7) DMA request clear register 7

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

7	6	5	4	3	2	1	0	bit
-					ICSEL07[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL07[2:0] Clear selection bits

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

ICSEL07[2:0]	Clear target
000	Free-run timer 2 (0 detection)
001	Free-run timer 2 (compare clear)
010	PPG0
011	PPG1
100	PPG2
101	PPG3
110	Setting disabled (Nothing cleared)
111	

28.4.9 ICSEL8 (Interrupt Clear SELect register 8) DMA request clear register 8

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

7	6	5	4	3	2	1	0	bit
-							ICSEL08	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL08 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #42).

ICSEL08	Clear target
0	Free-run timer 3 (0 detection)
1	Free-run timer 3 (compare clear)

28.4.10 ICSEL9 (Interrupt Clear SELect register 9) DMA request clear register 9

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

7	6	5	4	3	2	1	0	bit
-							ICSEL09	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL09 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #43).

ICSEL09	Clear target
0	Free-run timer 4 (0 detection)
1	Free-run timer 4 (compare clear)

28.4.11 ICSEL10 (Interrupt Clear SELect register 10) DMA request clear register 10

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

7	6	5	4	3	2	1	0	bit
-						ICSEL10[1:0]		
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	Attribute

[bit7 to bit2] Undefined

These bits always read "0". Writing has no effect on operation.

[bit1 to bit0] ICSEL10[1:0] Clear selection bits

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

ICSEL10[1:0]	Clear target
00	Free-run timer 5 (0 detection)
01	Free-run timer 5 (compare clear)
10	PPG4
11	PPG5

28.4.12 ICSEL11 (Interrupt Clear SELect register 11) DMA request clear register 11

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

7	6	5	4	3	2	1	0	bit
-							ICSEL11	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL11 Clear selection bits

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #45).

ICSEL11	Clear target
0	ICU0
1	ICU1

28.4.13 ICSEL12 (Interrupt Clear SELect register 12) DMA request clear register 12

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

7	6	5	4	3	2	1	0	bit
-							ICSEL12	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL12 Clear selection bits

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #46).

ICSEL12	Clear target
0	ICU2
1	ICU3

28.4.14 ICSEL13 (Interrupt Clear SELect register 13) DMA request clear register 13

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

7	6	5	4	3	2	1	0	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	R/W0	Attribute

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] Reserved

Always write "0" to this bit.

28.4.15 ICSEL14 (Interrupt Clear SELect register 14) DMA request clear register 14

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

7	6	5	4	3	2	1	0	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	R/W0	Attribute

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] Reserved

Always write "0" to this bit.

28.4.16 ICSEL15 (Interrupt Clear SELect register 15) DMA request clear register 15

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

7	6	5	4	3	2	1	0	bit
-							ICSEL15	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL15 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #49).

ICSEL15	Clear target
0	OCU0
1	OCU1

28.4.17 ICSEL16 (Interrupt Clear SELect register 16) DMA request clear register 16

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

7	6	5	4	3	2	1	0	bit
-							ICSEL16	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL16 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #50).

ICSEL16	Clear target
0	OCU2
1	OCU3

28.4.18 ICSEL17 (Interrupt Clear SELect register 17) DMA request clear register 17

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

7	6	5	4	3	2	1	0	bit
-							ICSEL17	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL17 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #51).

ICSEL17	Clear target
0	OCU4
1	OCU5

28.4.19 ICSEL18 (Interrupt Clear SELect register 18) DMA request clear register 18

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

7	6	5	4	3	2	1	0	bit
-							ICSEL18	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL18 Clear selection bit

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

ICSEL18	Clear target
0	OCU6
1	OCU7

28.4.20 ICSEL19 (Interrupt Clear SELect register 19) DMA request clear register 19

- **ICSEL19: Address 0413_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL19	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL19 Clear selection bit

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

ICSEL19	Clear target
0	OCU8
1	OCU9

28.4.21 ICSEL20 (Interrupt Clear SELect register 20) DMA request clear register 20

- **ICSEL20: Address 0414_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL20	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL20 Clear selection bit

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

ICSEL20	Clear target
0	OCU10
1	OCU11

28.4.22 ICSEL21 (Interrupt Clear SELect register 21) DMA request clear register 21

- **ICSEL21: Address 0415_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL21[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

- **[bit7 to bit3] Undefined**

These bits always read "0". Writing has no effect on operation.

- **[bit2 to bit0] ICSEL21[2:0] Clear selection bits**

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #55).

ICSEL21[2:0]	Clear target
000	WG dead timer underflow 0
001	WG dead timer underflow 1
010	WG dead timer underflow 2
011	WG dead timer reload 0
100	WG dead timer reload 1
101	WG dead timer reload 2
110	WG DTTI0
111	Setting disabled (Nothing cleared)

28.4.23 ICSEL22 (Interrupt Clear SELect register 22) DMA request clear register 22

- **ICSEL22: Address 0416_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL22[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL22[2:0] Clear selection bits

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #56).

ICSEL22[2:0]	Clear target
000	WG dead timer underflow 3
001	WG dead timer underflow 4
010	WG dead timer underflow 5
011	WG dead timer reload 3
100	WG dead timer reload 4
101	WG dead timer reload 5
110	WG DTTI1
111	Setting disabled (Nothing cleared)

28.4.24 ICSEL23 (Interrupt Clear SELect register 23) DMA request clear register 23

- **ICSEL23: Address 0417_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL23[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL23[2:0] Clear selection bits

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #57).

ICSEL23[2:0]	Clear target
000	AD converter 0
001	AD converter 1
010	AD converter 2
011	AD converter 3
100	AD converter 4
101	AD converter 5
110	AD converter 6
111	AD converter 7

28.4.25 ICSEL24 (Interrupt Clear SELect register 24) DMA request clear register 24

- **ICSEL24: Address 0418_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL24[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL24[2:0] Clear selection bits

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #58).

ICSEL24[2:0]	Clear target
000	AD converter 8
001	AD converter 9
010	AD converter 10
011	AD converter 11
100	AD converter 12
101	AD converter 13
110	AD converter 14
111	Setting disabled (Nothing cleared)

28.4.26 ICSEL25 (Interrupt Clear SELect register 25) DMA request clear register 25

- **ICSEL25: Address 0419_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-					ICSEL25[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	Attribute

[bit7 to bit3] Undefined

These bits always read "0". Writing has no effect on operation.

[bit2 to bit0] ICSEL25[2:0] Clear selection bits

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #59).

ICSEL25[2:0]	Clear target
000	AD converter 16
001	AD converter 17
010	AD converter 18
011	AD converter 19
100	AD converter 20
101	AD converter 21
110	AD converter 22
111	AD converter 23

(Note) On the MB91F583AS/F584AS/F585AS, settings of ICSEL25[2:0]= "000" - "010" and "101" - "111" are prohibited.

28.4.27 ICSEL26 (Interrupt Clear SELect register 26) DMA request clear register 26

- **ICSEL26: Address $041A_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL26	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL26 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #60).

ICSEL26	Clear target
0	Base timer 0 IRQ0
1	Base timer 0 IRQ1

28.4.28 ICSEL27 (Interrupt Clear SELect register 27) DMA request clear register 27

- **ICSEL27: Address 041B$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-							ICSEL27	
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

[bit7 to bit1] Undefined

These bits always read "0". Writing has no effect on operation.

[bit0] ICSEL27 Clear selection bit

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #61).

ICSEL27	Clear target
0	Base timer 1 IRQ0
1	Base timer 1 IRQ1

28.4.29 IORR0-7 (IO triggered DMA Request Register for ch. 0-7) IO transfer request setting registers 0 to 7

If the DMA transfer request generation factor is specified as a peripheral interrupt request, these registers are used to identify the interrupt request of the vector number that has generated the DMA transfer request.

An instance of these registers is provided for each DMA controller (DMAC) channel.

- **IORR0-IORR7: Address 0490_H -- 0497_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	IOE	IOS[5:0]						
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

[bit7] Reserved

Make sure to write "0".

[bit6] IOE (IO Enabled) Transfer request enable bit

When an interrupt request specified by the IOS5 to IOS0 bits has been generated, this bit is used to notify the DMA controller (DMAC) for the pertinent channel whether to output the DMA transfer request.

IOE	Function
0	No DMA transfer request output -- The interrupt request generated by the peripheral is not used as a DMA transfer request (Initial value).
1	DMA transfer request output

[bit5 to bit0] IOS5 to IOS0 (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.

IOS5 to IOS0	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).

28.5 Operation

■ Configuration

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 set to the vector number selected in step (1).

(3) Set the interrupt-related registers for the peripheral.

(4) Configure the DMAC.

■ Notes

- Do not change the IORR and ICSEL when the DMAC enables DMA transfer requests issued by peripherals.
- Peripherals to which resource numbers are not assigned (see "Table C-1 Interrupt Vectors (MB91F583AM/F584AM/F585AM)" in "APPENDIX C. Interrupt Vector") 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 29 BUS PERFORMANCE COUNTERS

29.1 Overview

This model has a built-in bus performance counters (BPC) for measuring the performance of the on-chip bus. BPC measures the breakdown of traffic on the on-chip bus, and provides information for strategies to improve bus performance. Because the counters do not count while the on-chip bus is idle, use the timers in the system at the same time to measure the time.

29.2 Features

■ Counter configuration

- Count clocks: Clock for the on-chip bus
- Counter bit length:32 bits × 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

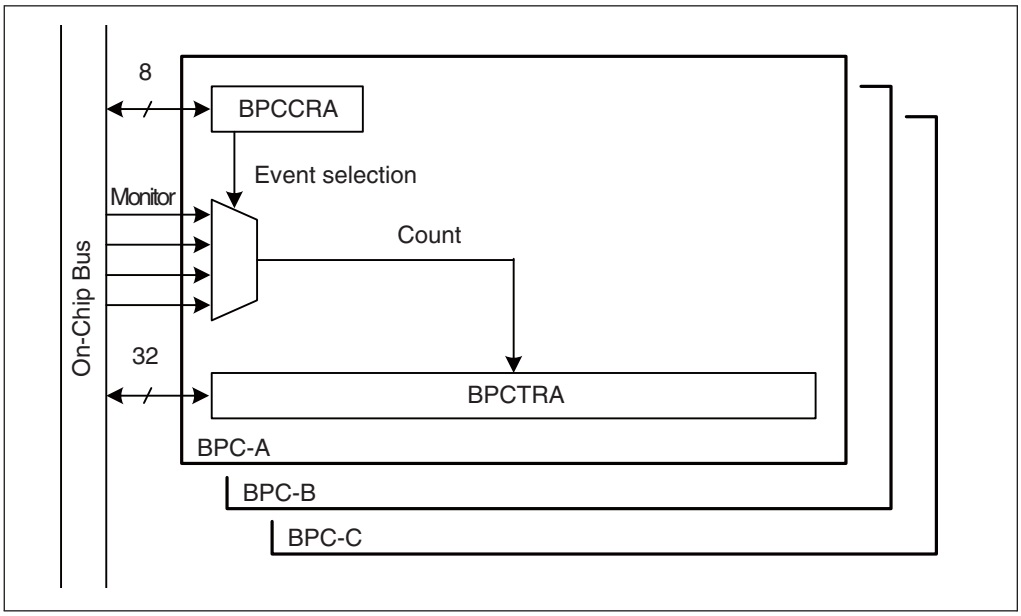
- Number of read accesses in the on-chip bus
- Number of write accesses in the on-chip bus
- Number of wait cycles in the on-chip bus

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)

29.3 Configuration

Figure 29.3-1 Block diagram of Bus Performance Counters



Code: FR81S09_BPC-1v0-91580MS-1-E

29.4 Registers

Table 29.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0710	BPCCRA	BPCCRB	BPCCRC	Reserved	BPC-A control register BPC-B control register BPC-C control register
0x0714	BPCTRA				BPC-A count register
0x0718	BPCTRB				BPC-B count register
0x071C	BPCTRC				BPC-C count register

29.4.1 BPCCRA (Bus Performance Counter Control Register A) BPC-A Control Register

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)**

7	6	5	4	3	2	1	0	bit
FUNC[1:0]		MST[3:0]				SLV[1:0]		
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

[bit7, bit6] FUNC1, FUNC0 (Function Selection) Measurement event selection

These bits select the event measured by BPC.

FUNC1, FUNC0	Event
00	BPC-A operation stopped (initial value)
01	Number of read accesses
10	Number of write accesses
11	Number of wait cycles

[bit5 to bit2] MST3 to MST0 (bus MaSTer select) Bus master selection

These bits select the bus master of which events are measured by BPC.

MST3 to MST0	Bus master
0000	All bus masters (initial value)
0001	CPU (XBS)
0010	DMAC
0011	Reserved
0100	Reserved
0101 to 1111	Reserved

[bit1, bit0] SLV1, SLV0 (SLaVe select) Slave selection

These bits select the slave of which events are measured by BPC.

SLV1, SLV0	Slave
00	All slaves (initial value)
01	MCH (registers)
10	ICH (peripherals)
11	Anything other than MCH/ICH

29.4.2 BPCCRB (Bus Performance Counter Control Register B) BPC-B control register

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)**

7	6	5	4	3	2	1	0	bit
FUNC[1:0]		MST[3:0]				SLV[1:0]		
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

29.4.3 BPCCRC (Bus Performance Counter Control Register C) BPC-C control register

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)**

7	6	5	4	3	2	1	0	bit
FUNC[1:0]		MST[3:0]				SLV[1:0]		
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

29.4.4 BPCTRA (Bus Performance CounTer Register A) BPC-A Count Register

This register is a 32-bit long count register that counts the events configured by BPCCRA register.

- **BPCTRA: Address 0714_H (Access: Word)**

31	30	• • •	3	2	1	0	bit
BPCTRA[31:0]							
0	0	• • •	0	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	R/W	Attribute

[bit31 to bit0] BPCTRA31 to BPCTRA0 (Bus Performance CounTer Register A) BPC-A count

If bits 7, 6: FUNC of the BPCCRA register are set to a value other than "00", the count of the target events begins. This register is readable and writable, and can only be accessed using 32-bit access. Because the counter is not initialized when the count is started, set the initial value when starting a new count. Furthermore, because there is no overflow control, if the counter overflows it returns to zero and continues counting.

29.4.5 BPCTRB (Bus Performance CounTer Register B) BPC-B Count Register

This register is a 32-bit long count register that counts the events configured by BPCCRB register. The usage is the same as BPCTRA.

- **BPCTRB: Address 0718_H (Access: Word)**

31	30	• • •	3	2	1	0	bit
BPCTRB[31:0]							
0	0	• • •	0	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	R/W	Attribute

29.4.6 BPCTRC (Bus Performance CounTer Register C) BPC-C Count Register

This register is a 32-bit long count register that counts the events configured by BPCCRC register. The usage is the same as BPCTRA.

- **BPCTRC: Address $071C_H$ (Access: Word)**

31	30	• • •	3	2	1	0	bit
BPCTRC[31:0]							
0	0	• • •	0	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	R/W	Attribute

29.5 Operation Description

29.5.1 Setup

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. For combinations that do not exist in the table below, events are not counted. Furthermore, counting is not performed in emulator mode.

Table 29.5-1 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
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

29.5.2 Starting and Stopping

The target event count is started by setting the FUNC[1:0] field of the bus performance counter A control register (BPCCRA) to a value other than "00". However, at this time the count starts from the current value without initializing the bus performance counter A register (BPCTRA). The operation of the bus performance counter stops when BPCCRA.FUNC[1:0] is set to "00".

29.5.3 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 after a reset occurs.

29.5.4 Measurement and Result Processing

The use of BPC is anticipated for when ICE is connected or when using a monitor debugger. Configuring of measurements and reading of results are performed in debug mode while user program execution is halted.

Example 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: Execution starting point of the user program

  - 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 each reference time, and read out the measurement results and initialize the counters.

  The following two methods are available for switching to debug mode each reference time.

  - Assert a tool break from the ICE 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 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 the debugger host program. Visualize the analysis results by displaying them in a graph so that they can be understood intuitively (pie graph, bar graph, line graph, etc.), and provide information that is beneficial for user program tuning (bus performance analysis function). The following is an analysis example.

Analysis example:

1. Bus master access proportion

   Ex. Proportion of DMAC access vs. CPU access, specific bus master access that occupies the total access, etc.
2. Occurred event proportion

   Ex. Proportion of write access vs. read access, proportion of total cycles made up of wait cycles, etc.
3. Target accessed proportion

   Ex. Proportion of MCH vs. ICH, proportion of total accesses made up of accesses to a specific target, etc.
4. Proportion of specific accesses from a specific bus master to a specific target

   Ex. Proportion of total access made up of read accesses from CPU to MCH, etc.
5. Proportion of wait cycles occurring in specific target

   Ex. Proportion of total cycles made up of wait cycles during MCH access, etc.
6. Analyze operation of each bus between two specific points in a program

   Ex. Proportion of total cycles between two specific points in the program consisting of read, write, wait cycles, etc.
7. Analyze operation of each bus during progress of each specific time

   Ex. Time course of proportion of all accesses consisting of accesses to specific bus masters and specific targets, etc.

CHAPTER 30 FLASH MEMORY

30.1 Overview

The capacity of the flash memory built into this product is 576 KBytes (512K + 64KBytes). Error correction codes (ECC) are attached.

30.2 Features

- Usable capacity
  MB91F583: 320 KBytes (large sector of 128KB each x 2 + small sector of 16KB each x 4)
  MB91F584: 448 KBytes (large sector of 128KB each x 3 + small sector of 16KB each x 4)
  MB91F585: 576 KBytes (large sector of 128KB each x 4 + small sector of 16KB each x 4)

Because this model has ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes adding to above.

(Note) A large sector is 64 KB and a small sector is 8 KB. If a contiguous area is used, it consists of paired small sectors and/or paired large sectors. This means that you can increase an area consisting of large sectors in 128-KB increments and an area consisting of small sectors in 16-KB increments.

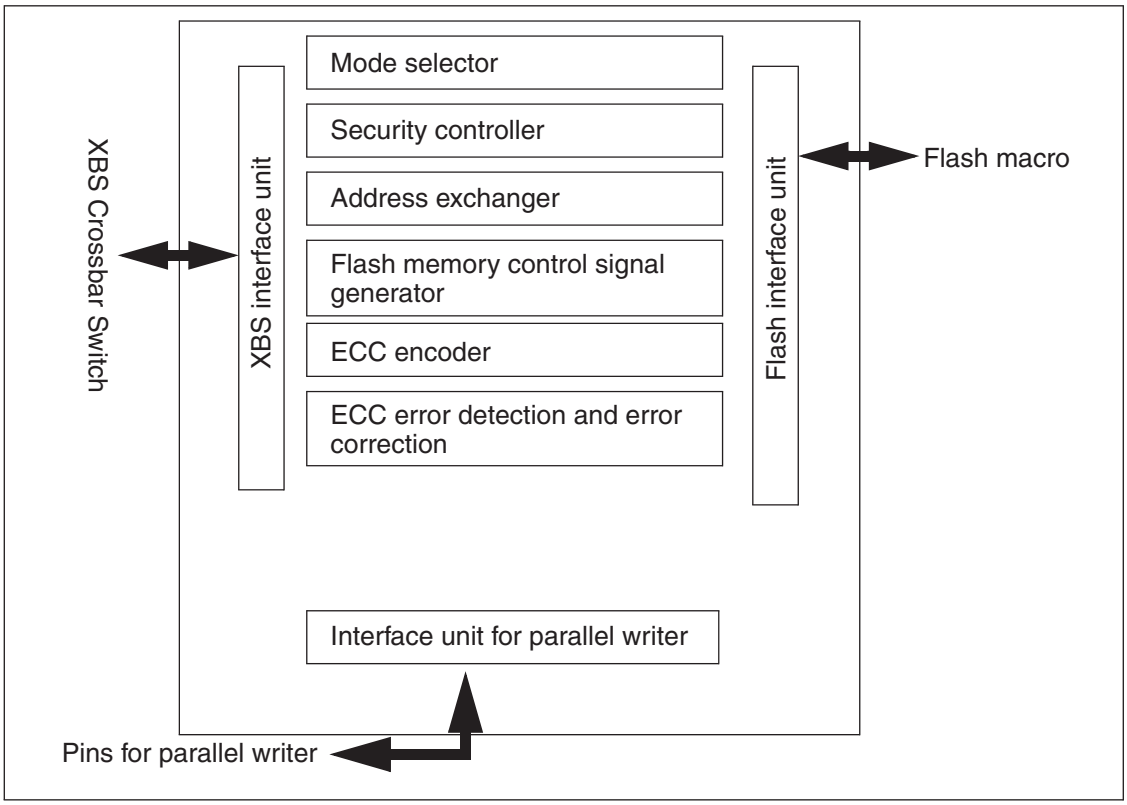
- High speed operation: Reads at the word (32-bit) unit can be performed in 1 cycle at 80MHz.
  Reads at the word (32-bit) unit can be performed in two cycles at 128 MHz.
- Write from external: Possible from ROM writer
- Operating mode:
  (1) CPU-ROM mode
  (CPU/DMA accesses the flash memory. Read-only)
  (2) CPU programming mode
  (CPU accesses the flash memory. Read, write, and erase)
  (3) Flash memory mode (flash memory accessible from external)
- Can be read, written, or erased (automatic program algorithm) by CPU
- Can be read, written, or erased (automatic program algorithm) by ROM writer
- Security function
  - In order to prevent the content of flash memory being read by a third party, when security is on, operation from external source after instruction fetch and writing/erasing other than chip erase are suppressed
  - After password authentication when using the on-chip debugger (OCD), this can be read from the external using the OCD even when security is on.
- Error Correction Code (ECC) security function
  - There is an error correction code (ECC) function that corrects errors of up to one bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read out. 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 if reading data in the state of chip erase/sector erase. To read the data correctly in the state of erase ($FFFF_H$), always write "$FFFF_H$" to read the data.

Code: FZ5C4-1v9-91580MS-1-E

30.3 Configuration

30.3.1 Block Diagram

Figure 30.3-1 Block diagram



30.3.2 Sector Configuration Diagram

Figure 30.3-2 Sector configuration diagram (MB91F585)

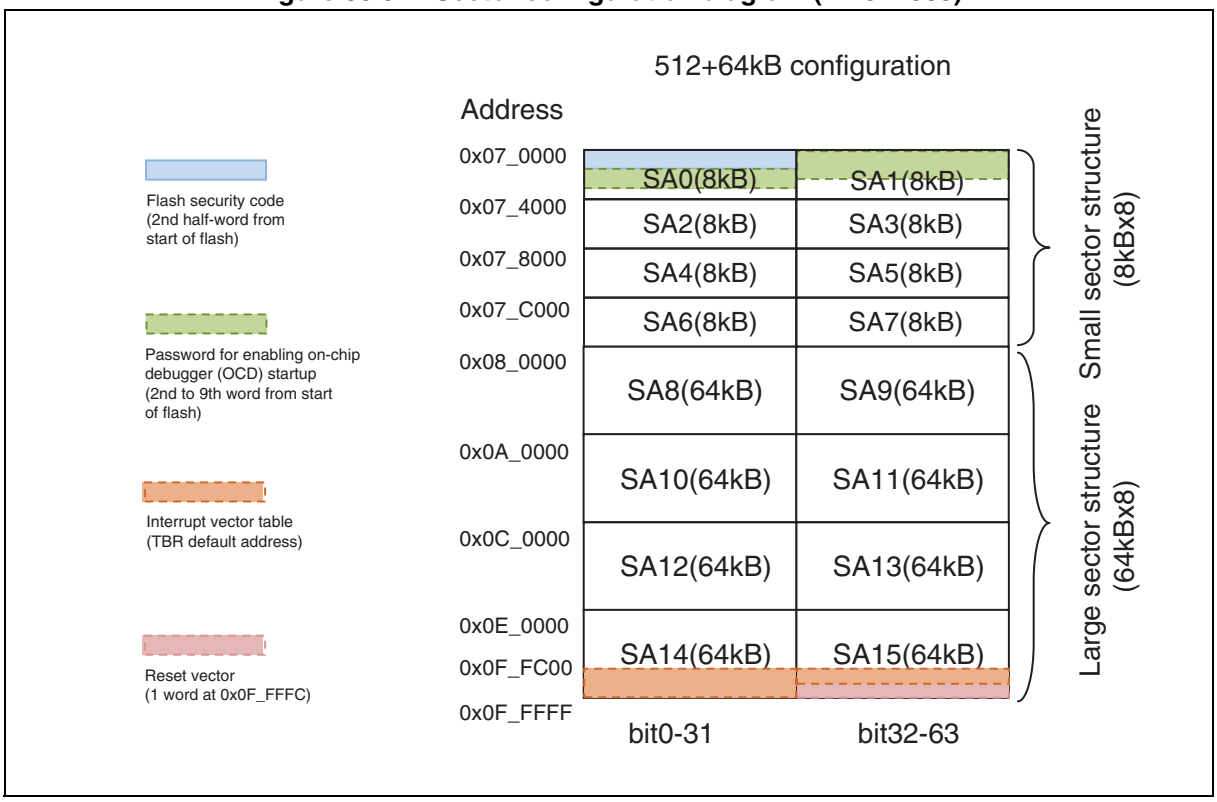


Figure 30.3-3 Sector configuration diagram (MB91F584)

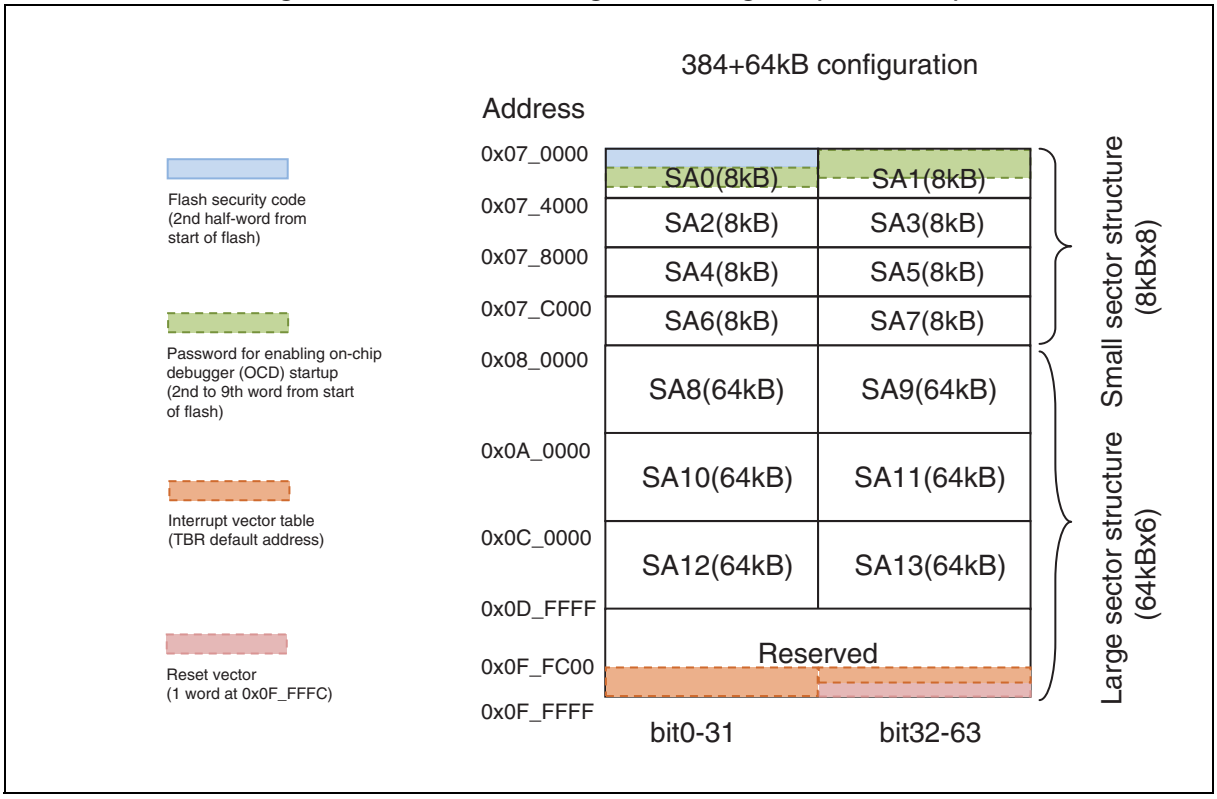
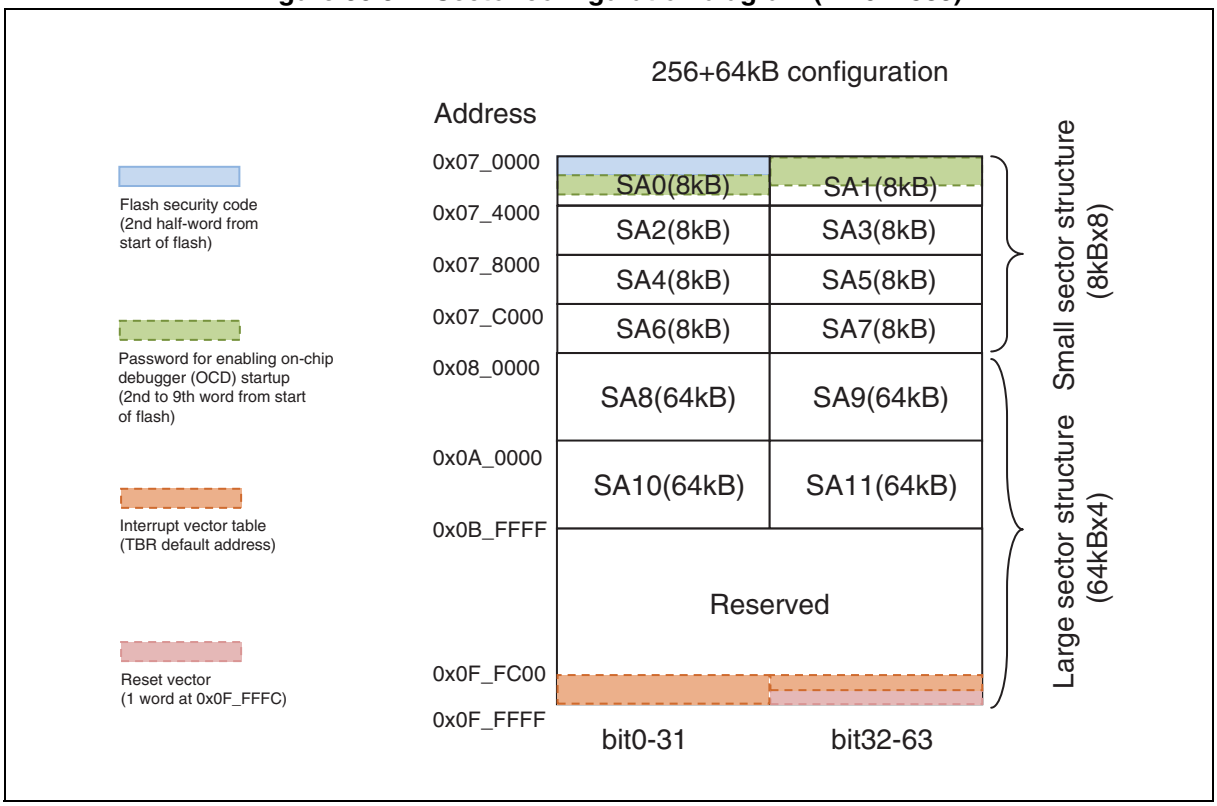


Figure 30.3-4 Sector configuration diagram (MB91F583)



(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. See "CHAPTER 32 FIXEDVECTOR FUNCTION" for details.
- Please refer to "CHAPTER 49 ON-CHIP DEBUGGER (OCD)" for the password setting for on-chip debugger (OCD) start enable. Please do not write anything in this area, and hold the initial state (all bit=1) immediately after the flash erase when the security function of on-chip debugger (OCD) is not used.

30.3.3 Sector Number and Flash Macro Number Correspondence Chart

Table 30.3-1 Sector number chart MB91F585 (512+64KB model)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (low-order 32 bits)	8KB	Flash security code area (0x07_0002 to 0x07_0003) On-chip debugger (OCD) startup permission password area (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 (high-order 32 bits)	8KB	On-chip debugger (OCD) startup permission password area (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 (low-order 32 bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (high-order 32 bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (low-order 32 bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (high-order 32 bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (low-order 32 bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (high-order 32 bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (low-order 32 bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (high-order 32 bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (low-order 32 bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (high-order 32 bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (low-order 32 bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (high-order 32 bits)	64KB	
SA14	0x0E_0000 to 0x0F_FFFB (low-order 32 bits)	64KB	
SA15	0x0E_0004 to 0x0F_FFFF (high-order 32 bits)	64KB	Interrupt vector table position (TBR default) (0x0F_FC00 to 0x0F_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Table 30.3-2 Sector number chart MB91F584 (384+64KB model)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (low-order 32 bits)	8KB	Flash security code area (0x07_0002 to 0x07_0003) On-chip debugger (OCD) startup permission password area (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 (high-order 32 bits)	8KB	On-chip debugger (OCD) startup permission password area (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 (low-order 32 bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (high-order 32 bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (low-order 32 bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (high-order 32 bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (low-order 32 bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (high-order 32 bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (low-order 32 bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (high-order 32 bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (low-order 32 bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (high-order 32 bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (low-order 32 bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (high-order 32 bits)	64KB	
-	0x0E_0000 to 0x0F_FFFF	-	Reserved Interrupt vector table position (TBR default) (0x0F_FC00 to 0x0F_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

Table 30.3-3 Sector number chart MB91F583 (256+64KB model)

Sector number	Address	Sector size	Remark
SA0	0x07_0000 to 0x07_3FFB (low-order 32 bits)	8KB	Flash security code area (0x07_0002 to 0x07_0003) On-chip debugger (OCD) startup permission password area (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 (high-order 32 bits)	8KB	On-chip debugger (OCD) startup permission password area (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 (low-order 32 bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (high-order 32 bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (low-order 32 bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (high-order 32 bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (low-order 32 bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (high-order 32 bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (low-order 32 bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (high-order 32 bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (low-order 32 bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (high-order 32 bits)	64KB	
-	0x0C_0000 to 0x0F_FFFF	-	Reserved Interrupt vector table position (TBR default) (0x0F_FC00 to 0x0F_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

30.4 Registers

Table 30.4-1 Register map

Address	Register				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

30.4.1 FCTLR (Flash ConTroL Register) Flash Control Register

This register configures the access control to flash memory.

- **FCTLR : Address 0840_H (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
Reserved	FWE	Reserved		FSZ[1:0]		FAW[1:0]		
1	0	0	0	1	0	0	0	Initial value
R1,WX	R/W	RX,W0	RX,W0	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
FDSBL	Reserved		RDYF	Reserved				
0	0	0	0	0	0	0	0	Initial value
R/W	RX,W0	RX,W0	R/W	RX,W0	RX,W0	RX,W0	RX,W0	Attribute

[bit15] Reserved

This bit is reserved. This bit always reads "1". Writing has no effect on operation.

[bit14] FWE (Flash Write Enable) Flash Write enable

Write enable bit to flash memory. Setting this bit configures CPU programming mode. Use the FSTR.FRDY bit to check whether or not writing is enabled.

Setting this bit disables the ECC error detection and data correction functions for data fetch to flash memory.

FWE	Description
0	Flash memory write not enabled (Initial value)
1	Flash memory write enabled

(Note) When writing to flash memory, any instruction fetch from flash memory is disabled.

[bit13, bit12] Reserved

These bits are reserved. Always write "0" to these bits. The read value is undefined.

[bit11, bit10] FSZ[1:0] (Flash access SiZe) Flash access size setting

These bits specify a flash write access size used in CPU mode. Any accesses to data with a bit width other than that specified with these bits are disabled. The read access size is not influenced by these bits.

In read mode, the flash macro is always read in 32-bit units. When a wait cycle is inserted based on the FAW bits, 64-bit read access is performed.

FSZ[1:0]	Description
00	8 bits
01/10/11	16 bits

[bit9, bit8] FAW1 and FAW0 (Flash Access Wait) Flash access wait setting

These bits specify the presence of a wait cycle for flash access performed in CPU mode. Since the flash memory read time is 12.5 ns, access to flash memory at a speed of greater than 80 MHz would be impossible without a wait. A wait must be inserted using these bits.

If the access speed is greater than 80 MHz, specify FAW = 1 (one wait cycle).

If you want to insert a wait cycle using the FAW feature, set these bits before setting the clock to high-speed.

Also, if you want to delete the wait cycle, set the clock to low-speed, and then set these bits.

FAW1, FAW0	Description
00	0 cycles (initial value)
01	1 cycle
10/11	Disabled

(Note) If one wait cycle is set using these bits, the wild register function is disabled.
If you want to use the wild register function, reduce the core operation speed to 80 MHz or lower and set the FAW bits to 2’b00 (0 cycle).

[bit7] FDSBL (Flash DiSaBLe) Flash Disable directive

This bit configures the flash memory access disabled state (both reads and writes).

FDSBL	Description
0	Flash enable (initial value)
1	Flash disable

[bit6, bit5] Reserved

These bits are reserved. Always write "0" to these bits. The read value is undefined.

[bit4] RDYF (ReaDY Flag) RDY negation instruction at branch access

This bit specifies the insertion of a wait cycle used at branch access. If this bit is set to "1", a wait cycle will be inserted at the time of branch access. The purpose of this insertion is processing cycle adjustment. If this bit is "0", wait cycles will be controlled based on the internal status of the flash interface when branch access occurs. No wait cycle will be inserted if cycle time compensation is unnecessary when a branch access request is accepted. A wait cycle will be inserted if cycle time compensation is necessary.

RDYF	Flash write cycle
0	Depends on the flash interface status (initial value)
1	Wait cycle insertion

[bit3 to bit0] Reserved

These bits are reserved. Always write "0" to these bits. The read value is undefined.

30.4.2 FSTR (Flash STatus Register) Flash Status Register

This register indicates the flash memory state.

- **FSTR : Address 0843_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
Reserved					FECCERR	FHANG	FRDY	
0	0	0	0	0	0	0	1	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,WX	R,WX	Attribute

[bit7 to bit3] Undefined

The read value is always "0". Writing has no effect.

[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	Programming
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.

Writing 0 is prioritized when ECC error and writing 0 occur concurrently.

[bit1] FHANG (Flash HANG) Flash Hang State

This bit indicates the flash memory HANG state. If there is a timing overrun (see "[bit5] TLOV (Timing Limit Elapsed Flag Bit)") and the flash memory hangs and this bit is set to "1", execute the reset command (see Section "30.5.3.1 Command Sequence").

FHANG	Description
0	Normal state
1	HANG UP state

The correct value might not be read out immediately after a command has been issued by the automatic algorithm. Therefore, ignore the first value read from 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. Data cannot be written to or erased from the flash memory 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 has been issued by the automatic algorithm. Therefore, ignore the first value read from this bit after the command issuance.

30.4.3 Flash interface control register: FLIFCTLR(Flash I/F Control Register)

This register controls the flash interface. It is used for both program flash and WorkFlash.

- **FLIFCTLR: Address 2308_H (Access: Byte, half word, or word)**

7	6	5	4	3	2	1	0	bit
Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0	
0	0	0	0	0	0	0	0	Initial value
RX,WX	RX,WX	RX,WX	R/W	RX,WX	RX,WX	R/W	R/W	Attribute

[bit7 to bit5] Reserved

These bits are reserved. The read value is undefined. Writing to these bits has no effect.

[bit4] DFWDSBL (Data Fetch Wait cycle DiSaBLe) Data fetch wait cycle disable

If this bit is set to "1", a wait cycle that is inserted at data fetch if waiting is specified is disabled. However, you cannot disable wait cycles to compensate for cycle time.

DFWDSBL	Description
0	Wait cycle enabled (initial value)
1	Wait cycle disabled

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing to this bit has no effect.

[bit2] Reserved

This bit is reserved. Always write "0" to this bit.

[bit1] ECCDSBL1 (ECC DiSaBLe 1) ECC function disable 1

This bit enables or disables the ECC function for write access or data fetch access to WorkFlash memory in CPU mode.

ECCDSBL1	Description
0	ECC function enabled (initial value)
1	ECC function disabled

[bit0] ECCDSBL0 (ECC DiSaBLe 0) ECC function disable 0

This bit enables or disables the ECC function for write access or data fetch access to program flash memory in CPU mode.

ECCDSBL0	Description
0	ECC function enabled (initial value)
1	ECC function disabled

30.4.4 Flash interface feature extension register 1 : FLIFFER1 (Flash I/F Feature Extension Register 1)

Reserved register. When writing to this register, write 0xFF.

- **FLIFFER1: Address 230A$_H$ (Access: Byte, half word, or word)**

7	6	5	4	3	2	1	0	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	Attribute

[bit7 to bit0] Reserved

These bits are reserved. When writing, always write "1" to these bits. The read value is "1".

30.4.5 Flash interface feature extension register 2 : FLIFFER2 (Flash I/F Feature Extension Register 2)

Reserved register. When writing to this register, write 0xFF.

- **FLIFFER2: Address 230B$_H$ (Access: Byte, half word, or word)**

7	6	5	4	3	2	1	0	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	Attribute

[bit7 to bit0] Reserved

These bits are reserved. When writing, always write "1" to these bits. The read value is "1".

30.5 Operation Description

This section explains the method for accessing the flash memory area.

30.5.1 Access Mode Setting

The flash memory in this model 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 the flash memory. Read-only, byte/half-word/word access)
(2) CPU programming mode
(CPU accesses the flash memory. Read/write, half-word access only)
(3) Flash memory mode (flash memory accessible from external)

■ Configuring CPU-ROM Mode

CPU ROM mode is in effect when the FWE bit of the flash control register (FCTLR) is "0". In CPU ROM mode, reading flash memory is enabled when the FRDY bit of the flash status register (FSTR) is "1". In CPU ROM mode, writing to flash memory is disabled. After releasing of reset, the flash memory enters this state.

■ Configuring CPU Programming Mode

CPU programming mode is in effect when the FWE bit of the flash control register (FCTLR) is "1". In CPU programming mode, reading and writing flash memory are enabled when the FRDY bit of the flash status register (FSTR) is "1".

30.5.2 Programming Flash Memory by CPU

After configuring CPU programming mode, perform erasing and programming using the automatic algorithm. In this model, because error correction codes (ECC) are added to each single word, programming needs to be performed for each single word. In the following procedure, each word is programmed by two operations to write one half-word. If this procedure is not followed, the written values will not be read correctly because the values will be written to flash memory without calculating the ECC.

(1) Set the flash access size to 16 bits. (FCTLR.FSZ[1:0]=01)

(2) Issue the write command. Write address = PA, write data = PD[31:16]
See Section "30.5.5 Write Command" for details on the write command.

(3) Read the hardware sequence flag until the write has finished.
See Section "30.5.3.2 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 that was written and check that the correct value is read. Even if the correct value could be read, check the FSTR.FECCERR bit to check that there was no ECC correction. If ECC correction occurred, write again starting from erasing the flash memory.

PA: Write address (word aligned)
PD[31:0]: Write data
PD[31:16]: Write data upper 16 bits
PD[15:0]: Write data lower 16 bits

30.5.3 Automatic Algorithm

When using CPU programming mode, writes and erasures of flash memory are performed by starting the automatic algorithm.

30.5.3.1 Command Sequence

The automatic algorithm starts when half-word (16-bit) data is written to the flash memory one to six times in a row. This is called a command sequences. The command sequences are shown below.

Table 30.5-1 Command sequence

Command	Number of writes	First		Second		Third		Fourth		Fifth		Sixth	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	Any	$F0_H$	RA	RD								
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 suspended	1	Any	$B0_H$										
Resumption of sector erasure	1	Any	30_H										

(Notes)
- In the table above, only the low-order 8 bits of data are shown. The high-order 8 bits are arbitrary. Write commands on the assumption of half-word or byte access.
- In the table above, only the low-order 16 bits of addresses are shown. For the high-order 16 bits, specify any address included in the range of applicable flash macro addresses.
  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 16 bits.)
SA: Sector address (specify any address included in the range of the addresses of erasure target sectors)
RA: Read address
RD: Read data (the read width is arbitrary.)

(Notes)
- When the command address and the sector erase command are issued, the lowest 2 bits of sector address (SA) to be input must be as follows.
  Half-word access: 2’b00
  Byte access: 2’b01 or 2’b11
  Example 1
  At byte access with command address = (lowest 2 bits of the standard command address changed to 2'b01)
  $yAA8_H \rightarrow yAA9_H$, $x554_H \rightarrow x555_H$, SA → {SA[31:2], 2’b01}
  (SA: An arbitrary address in the erase target sector to be input when the sector erase command is issued)
  Example 2
  At byte access with command address = (the lowest 2 bits of the standard command address changed to 2'b11)
  $yAA8_H \rightarrow yAAB_H$, $x554_H \rightarrow x557_H$, SA → {SA[31:2], 2’b11
  (SA: An arbitrary address in the erase target sector to be input when the sector erase command is issued)
- If an incorrect address or data value is written or if an address or data is written in an incorrect sequence, the command written so far will be cleared.

■ Reset command

By sending the reset command to the target flash memory, you can cancel the commands (shown in Table 30.5-1) you have entered so far and re-enter commands from the first one.

However, if a command was completely entered so that the automatic algorithm has started, you cannot use this reset command to stop the automatic algorithm.

Only if the automatic algorithm execution has exceeded the time limit, you can enter the reset command to restore the flash memory to its reset state.

■ Read command

The flash memory can be read by sending a read command to the target sector. Once a read command has been issued, the flash memory maintains the read state until another command is issued.

■ Program (write) command

If a write command is sent to the target sector four times in a row, the automatic algorithm starts and writes data to the flash memory. Programming (writing) of data can be performed in any order of addresses or across a sector boundary. In CPU programming mode, data is written in half-words or bytes. Once the fourth write has finished, the automatic algorithm starts and the automatic write to flash memory is started. After executing the automatic write algorithm command sequence, there is no need to control the flash memory externally.

See Section "30.5.5 Write Command" for details on the actual operation.

(Notes)
- If the fourth write command (write data cycle) is written to an odd address when writing in half-word, the write is not performed correctly. Always write to an even address.
- In the first write command sequence, only a single half-word data can be written. If you want to write multiple data, issue one write command sequence for each data.
- When security is on, there are restrictions on writing to the flash. See "■ Flash Access Restrictions When Security is On" for details.

■ Chip erase command

If the chip erase command is sent to the target sector six times in a row, all sectors of the flash memory can be erased in one step. Once the sixth write has finished, the automatic algorithm starts and the chip erase operation is started. When the automatic erase algorithm is started, "0" is written to all of the cells in the flash memory chip before erasing the entire chip, and there is no need to write to flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

See Section "30.5.6 Chip Erase Command" for details on the actual operation.

■ Sector erase command

If the sector erase command is sent to the target sector six times in a row, the sector of the flash memory is erased. When 40 μs elapses (timeout period) after the sixth write has finished, the automatic algorithm starts and the sector erase operation is started. If you want to erase multiple sectors, write the erase code (30_H) to the address of the sector to erase within the 40 μs (timeout period). If the next sector is not input within the timeout period, the sector erase command may become invalid. When the automatic erase algorithm is started, "0" is written to the cells in the sector to erase in flash memory before erasing the sector, and there is no need to write to flash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

See Section "30.5.7 Sector Erase Command" for details on the actual operation.

(Note) When security is on, there are restrictions on writing to the flash. See "■ Flash Access Restrictions When Security is On" for details.

■ Sector erase suspend command

If the sector erase suspend command is sent while executing sector erase or during command timeout, it can transit to the sector erase suspended state.

In the sector erase suspended state, the memory cell of the sector not to be erased will be available for reading operation.

In this case, a new write/erase command cannot be accepted.

To resume the paused erase operation from the sector erase suspended state, issue the erase resume command.

Once the erase resume command has been accepted, it returns to the sector erase state and resumes the erase operation.

Even in the case where transiting from the command timeout state to this state, if the erase resume command has been written normally, it transits to the sector erase state and resumes the sector erase operation without transiting to the command timeout state.

For the actual operation, see Section "30.5.8 Sector erase suspend command".

(Note) It takes up to 16.7 μs from the time of issuing the sector erase suspend command until the time when reading the sector not to be erased becomes available after suspending the sector erase operation.

To verify if it has become available for reading, check the FRDY bit of the flash status register (FSTR) or TOGG1 of the hardware sequence flag.

30.5.3.2 Automatic Algorithm Execution State

Because writing and erasing flash memory is performed by an automatic algorithm, the operating state can be checked by the hardware sequence flag using the FRDY bit of the flash status register (FSTR) to determine whether or not the automatic algorithm is executing.

■ Hardware sequence flag

This flag indicates the state of the automatic algorithm. When the FRDY bit of the flash status register (FSTR) is "0", the operating state can be checked by reading from an arbitrary address in flash memory. The bit configuration of the hardware sequence flag is shown below.

Figure 30.5-1 Bit configuration of hardware sequence flag

For half-word access

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
bit	7	6	5	4	3	2	1	0
	DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

For byte access

bit	7	6	5	4	3	2	1	0
	DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

(Notes)
- Cannot be 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 state

The correspondence between the state of each bit of the hardware sequence flag and the flash memory state is shown in Table 30.5-2.

Table 30.5-2 Correspondence between flags and flash memory states

State		DPOLL	TOGG1	TLOV	SETI	TOGG2
Executing	Writing	Inverted data*	Toggle	0	0	-
	During sector/chip erase	0	Toggle	0	1	-
Time limit exceeded	Write command	Inverted data*	Toggle	1	0	-
	Sector/chip erase command	0	Toggle	1	1	-
Sector erase suspend	Target sector to erase	-	-	-	-	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 a write/erase target address, this bit indicates whether or not the automatic algorithm is running using a data polling function.

The value being read differs depending on the operating state.

(1) During write

Writing:

Reads out the opposite value (inverse 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

Sector erase execution in progress: Reads "0" from the sector being erased.

After erasing the sector: Always reads out "1".

(3) During chip erase

Chip erase execution in progress: Always reads out "0".

After erasing the chip: Always reads out "1".

(4) During sector erase suspension

Suspended state (incomplete): Reads "0" from the sector erase suspended sector.

Sector erase operation complete: Reads "1" from the sector erase suspended 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 state.

During write / sector erase / chip erase

During write / sector erase / chip erase:

If this bit is read sequentially, "1" and "0" are returned 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 Elapsed 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 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 rewrite an address written with "0" with "1", the flash memory is locked and the automatic algorithm does not end. In this case, the value of the DPOLL bit remains invalid and the value read from the TOGG1 bit continues to alternate between "1" and "0". If the rated time is exceeded in this state, this bit changes to "1". If this bit becomes "1", issue a reset command.

(Note) When this bit is "1", this indicates that the flash memory was not used correctly. The flash memory is not faulty. Perform the appropriate processing after issuing the reset command.

[bit4] Undefined bit

[bit3] SETI (Sector Erase Timer Flag Bit)

During sector erase, a timeout period of 40 µs is required from when the sector erase command is issued until the sector erase actually starts. When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the sector erase command is within the timeout period is running. The value that is read differs depending on the state.

During sector erase:

When erasing sectors, you can check whether the next sector erase code is ready to be accepted by checking this bit before inputting the next sector erase code. Reads out the next value without accessing the address specified for reading the hardware sequence flag.

0: Within sector erase wait period (the next sector erase code (0x30) can be accepted.)

1: The sector erase wait period has been exceeded (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 suspended state, the sector not to be erased can be read but the sector to be erased cannot be read.

This flag indicates that output data toggles and it is the sector to be erased if the read address is the erase target sector while sector erase is being suspended.

Read out to the sector to be erased:	If this bit is read sequentially, "1" and "0" are returned alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
Read out to the sector not to be erased:	Reads out the data of the specified address.

[bit1, bit0] Undefined bits

30.5.4 Reset command

The flash memory can be reset by sending a reset command to the target flash memory. Since this state is the initial state of the flash memory, the flash memory always returns to the reset state when the power gets turned on or a command ends normally. When powering on, it is not necessary to issue a reset command. In addition, data can be read using normal read access while program can be accessed by the CPU in the reset state; thus it is not necessary to issue a reset command when reading data.

30.5.5 Write Command

Writes are performed in the following order.

(1) Send write commands sequentially to the target sector

The automatic algorithm is started and data is written to the flash memory.
After issuing a write command, there is no need to control the flash memory externally.

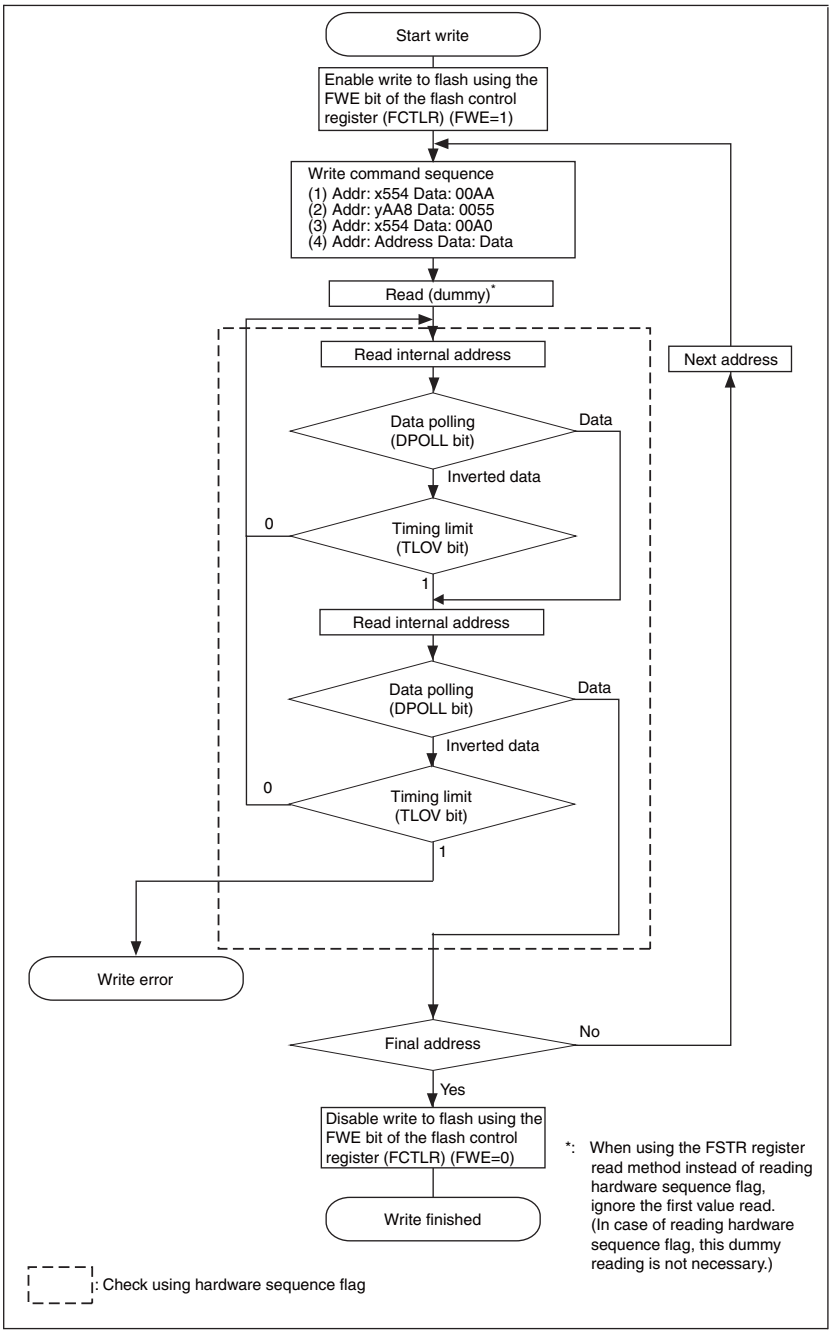
(2) Perform read access to the written address

The read data is the hardware sequence flag. Therefore, if bit7 (DPOLL bit) of the read data matches the written data, the write to the flash memory has finished.

If the write has not finished, the opposite value (inverse data) of the value of bit7 of the last written data is read out.

An example of a write operation to the flash memory is shown in Figure 30.5-2.

Figure 30.5-2 Write procedure example



(Notes)

- Once the write has finished, because the flash memory returns to read mode, write addresses are no longer accepted.
- See Section "30.5.3 Automatic Algorithm" for details on write commands.
- The value of the DPOLL bit of the hardware sequence flag changes at the same time as the TLOV bit, and needs to be re-checked even if the TLOV bit is "1".
- The toggle operation stops at the same time as the TOGG1 bit and TLOV bit of the hardware sequence flag change to "1". Therefore, even if the TLOV bit is "1", the TOGG1 bit needs to be re-checked.
- Although flash memory can be written to in any order of addresses, even if it crosses a sector boundary, only a single half-word data can be written in each write command sequence. If you want to write multiple data, issue one write command sequence for

each data.

- Data that has been written to "0" once cannot be returned to "1". If "0" is rewritten with "1", one of the following occurs.
  -The element is judged as faulty by the data polling algorithm.
  -The write rated time is exceeded, and the TLOV bit of the hardware sequence flag changes to "1".
  -It appears to have been written as "1".
  However, even if it appears to have been written as "1", the actual data remains "0" and "0" will be read out when the data is read in read/reset mode. If you want to return data to "1", perform a chip erase or sector erase.
- During write operations, all commands written to flash memory are ignored.
- If this device is reset during a write, the data that was written cannot be guaranteed.
- Because this model has the ECC bit added, data always needs to be written as 32 bits by two 16-bit writes. See Section "30.5.2 Programming Flash Memory by CPU" for the procedure.

30.5.6 Chip Erase Command

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 Section "30.5.3 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 (pre-program)]. 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 need to write to flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.
- When security is on, there are restrictions in the procedure for erasing the flash.
  See "■ Unlocking Flash Security" for details.

30.5.7 Sector Erase Command

A sector in the flash memory can be selected and only data in the selected sector can be erased. 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

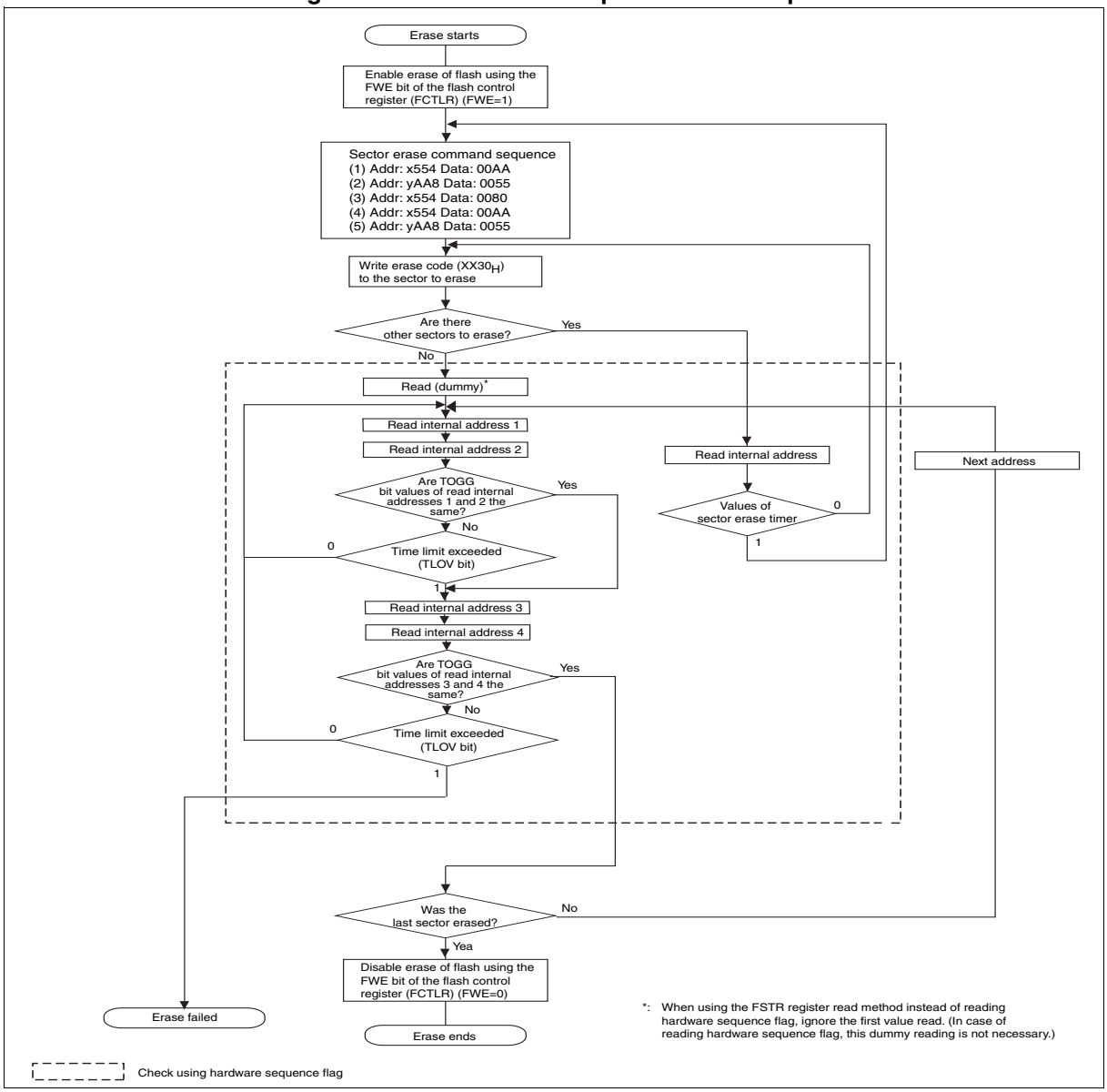
Once 40µs has elapsed (timeout period), the automatic algorithm starts and the sector erase operation is started. If you want to erase multiple sectors, write the erase code (30_H) to the address of the sector to erase within the 40 µs (timeout period). If the write is performed after the timeout period has elapsed, the sector erase command may be invalid.

(2) Perform read access to an arbitrary address

The read data is the hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", the sector erase has finished. Furthermore, you can use the TOGG1 bit to check whether or not the sector erase has finished.

An example of the sector erase procedure is shown in Figure 30.5-3 taking the example of using the TOGG1 bit in the check operation.

Figure 30.5-3 Sector erase procedure example



(Notes)

- The time required to erase the sector is [(sector erasure time + sector write time (pre-program)) × no. of sectors].
- When the sector erase operation finishes, the flash memory returns to the read/reset state.
- See Section "30.5.3 Automatic Algorithm" for details on the sector erase command.
- The value of the DPOLL bit of the hardware sequence flag changes at the same time as the TLOV bit, and needs to be re-checked even if the TLOV bit is "1".
- The toggle operation stops at the same time as the TOGG1 bit and TLOV bit of the hardware sequence flag change to "1". Therefore, even if the TLOV bit is "1", the TOGG1 bit needs to be re-checked.
- The only write commands that can be accepted in a command time-out state are the add erased sector and erase suspend commands. The only command that can be accepted in a sector erase state is the erase suspend command.
- When the automatic erase algorithm is started, "0" is written to the cells to erase in flash memory before erasing the sector, and there is no need to write to flash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

30.5.8 Sector erase suspend command

Sector erase can be suspended while executing sector erase or during command timeout.

In the sector erase suspended state, the memory cell of the sector not to be erased will be available for reading operation. In this case, a new write/erase command cannot be accepted.

To suspend the sector erase, send a sector erase suspend command to an arbitrary address of the target flash macro.

After sector erase has been suspended, reading from the target flash macro is enabled.

At this time, the hardware sequence flag is read from the sector of which sector erase is suspended.

Once it has entered in the sector erase suspended state, the following conditions apply.

- TOGG1 bit that toggles during sector erase will not toggle in the sector erase suspended state.
- FRDY of the flash status register becomes "1".

You can verify if it has entered in the sector erase suspended state judging from the above conditions.

(Note) It takes up to 16.7 μs from the time of issuing the sector erase suspend command until the time when reading the sector not to be erased becomes available after suspending the sector erase operation.

Even while sector erase is suspended, bit2:TOGG2 of the hardware sequence flag toggles. You can verify the sector under suspension using this bit.

To resume the paused erase operation from the sector erase suspended state, issue the sector erase resume command shown in "Table 30.5-1 Command sequence".

The sector erase resume command can be accepted in the sector erase suspended state only.

Send the command after making sure that it has entered in the sector erase suspended state.

Once the erase resume command has been accepted, it returns to the sector erase state and resumes the erase operation.

30.5.9 Security Function

This flash memory is equipped with a security function. When the security function is off, the flash memory can be used without limits. However, when the security function is on, writes and erases other than chip erase are suppressed. See "■ Flash Access Restrictions When Security is On" for details on the restrictions.

■ Flash Security On/Off Determination When Reset Released

The flash interface of this model 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.

■ Flash Security Setting Method

When reset is input and released after writing 0x0001 to the flash security code area (see "Figure 30.3-2 Sector configuration diagram (MB91F585)" in "CHAPTER 30 FLASH MEMORY"), security is turned on. Once security has been turned on once, the security does not turn off unless the entire flash memory area is erased.

■ Unlocking Flash Security

The chip erase command can be performed on all flash macros using the following procedure.

(1) Erase WorkFlash.

(2) Erase program flash that 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 user mode (internal FLASH start), issuing the erase command to an arbitrary flash macro and erasing data in the flash macro are available. For protecting data stored in the flash macro, following the above procedure in the described order is recommended for erasing the chip.

■ Flash Access Restrictions When Security is On

When security is on, the restrictions shown below are created by the start mode.

Table 30.5-3 Access restrictions when security is on

Operating mode	Access restriction
User	In normal mode (the state where there are no access restrictions due to the following flash security violations), writing in the security information area (first nine words of the FLASH memory) is canceled. Moreover, sector erase command to sector 0 and sector 1 is ignored. If an instruction fetch is performed to the on-chip bus area, a reset request is issued by the flash security violation reset source. Accesses to the flash memory are not accepted thereafter. The flash memory returns to normal state by reset.
Other than above (writer, etc.)	Access to flash memory is restricted. The data from reads is masked and 0xFFFF_FFFF is returned. Write commands and sector erase commands are ignored. Chip erase commands are accepted. See "■ Unlocking Flash Security".

Furthermore, when security is ON, if a data read is performed to the security information storage area (9 words at the start of the flash memory), it operates as follows.

- It will be a data access error where an illegal instruction exception or a data access error interrupt occurs. (See the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL" for details.)
- 0xFFFFFFFF is returned as the return value.
  However, when the OCD tool is connected, this restriction does not apply to access from OCDU or read during debug state.

30.5.10 Warnings When Using Flash Memory

- 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 Section "30.5.2 Programming Flash Memory by CPU" for the procedure.
- Do not issue commands to multiple macros simultaneously (i.e. in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- Once authentication is completed using the on-chip debugger (OCD) password, OCD can be used to read the content of flash memory externally even if security is on. If you want to prevent a third party from reading, always set a password for enabling on-chip debugger (OCD) activation.
- Transiting to the standby state during flash program/erase is prohibited.
- Since this flash memory is equipped with ECC, data cannot be overwritten to the address that already has some kind of values written.

CHAPTER 31 WorkFlash MEMORY

31.1 Overview

The flash memory capacity built in this product is 64 KBytes. ECC(Error Correction Code) are attached.

31.2 Features

- Usable capacity
  MB91F583 : 64 Kbytes (8 Kbytes x 8 sectors)
  MB91F584 : 64 Kbytes (8 Kbytes x 8 sectors)
  MB91F585 : 64 Kbytes (8 Kbytes x 8 sectors)
  For ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes adding to above.
- High speed operation: Reads at the word (32-bit) unit can be performed in 2 cycles at 80MHz.
  Reads at the word (32-bit) unit can be performed in 4 cycles at 128MHz.
- Write from external: Possible from ROM writer
- Operating mode:
  (1) CPU-ROM mode
  (CPU/DMA accesses the flash memory. Read-only)
  Only data access is enabled. Instruction fetch is not enabled.
  (2) CPU programming mode
  (CPU accesses the flash memory. Read, write, and erase)
  (3) Flash memory mode (flash memory accessible from external)
- Security function
  - In order to prevent the content of flash memory being read by a third party, when security is on, operation from external source after instruction fetch and writing/erasing other than chip erase are suppressed
  - After password authentication when using the on-chip debugger (OCD), this can be read from the external using the OCD even when security is on.
- Error Correction Code (ECC) security function
  - There is an error correction code (ECC) function that corrects errors of up to one bit in each word.
    (A function for detecting 2-bit errors is not provided.)
    Errors are automatically corrected during read.
    Furthermore, ECC codes are automatically added when writing to flash memory.
    Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development.
  - An error is detected if reading data in the state of chip erase/sector erase. To read the data correctly in the state of erase ($FFFF_H$), always write "$FFFF_H$" to read the data.

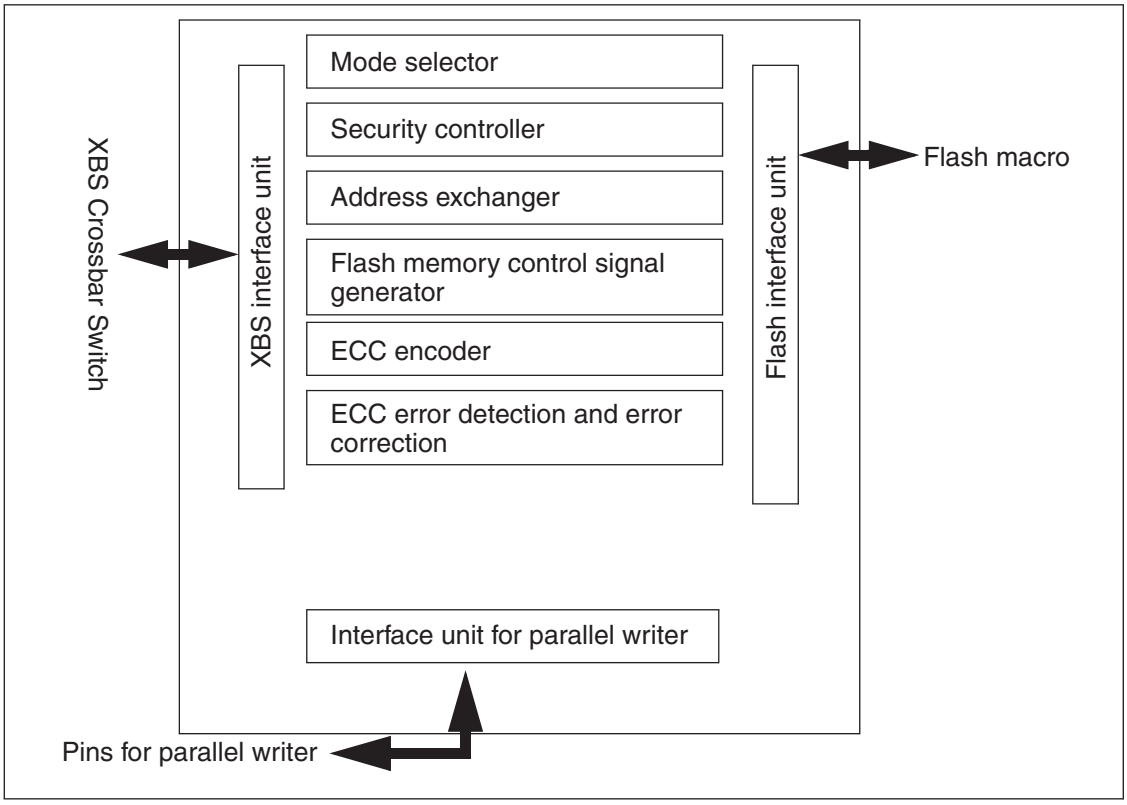
*: Embedded Algorithm

Code: FZ5C0-1v9-91580MS-1-E

31.3 Configuration

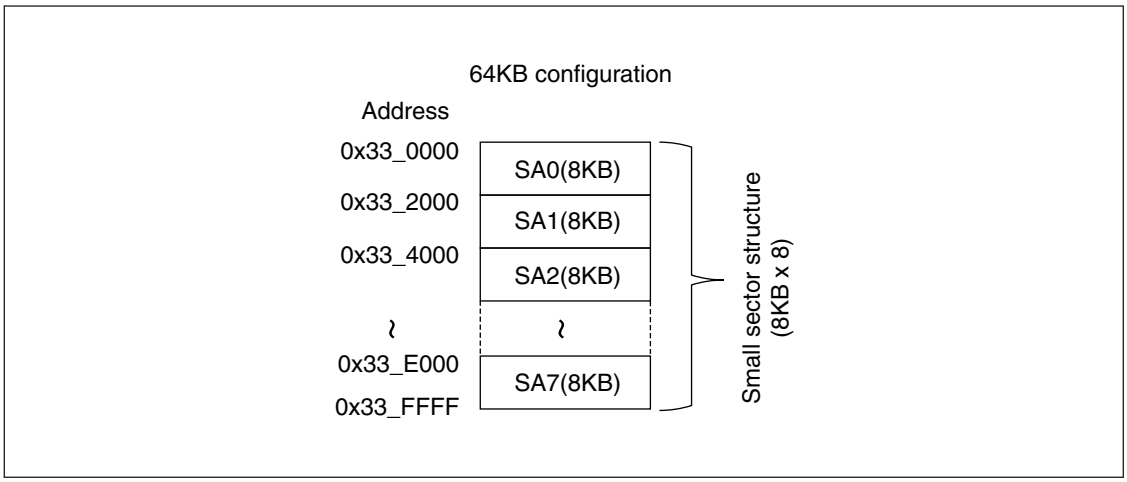
31.3.1 Block Diagram

Figure 31.3-1 Block Diagram (64KB model)



31.3.2 Sector configuration diagram

Figure 31.3-2 Sector configuration diagram



31.4 Registers

Table 31.4-1 Register map

Address	Register				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

31.4.1 DFCTLR (Data Flash ConTroL Register) WorkFlash Control Register

This register configures the access control to data flash.

- **DFCTLR : Address 2300_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	FWE	Reserved						
0	0	0	0	0	0	0	0	Initial value
RX,WX	R/W	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	Attribute

7	6	5	4	3	2	1	0	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

[bit15] Reserved

This bit is reserved. The read value is undefined. Writing has no effect.

[bit14] FWE (Flash Write Enable): Flash Write enable

This bit is a control bit to enable write to data flash in the CPU mode.

Setting this bit disables the ECC error detection and data correction functions for data fetch to WorkFlash memory.

FWE	Description
0	Flash write not enabled (Initial value)
1	Flash write enabled

[bit13 to bit0] Reserved

These bits are reserved. The read value is undefined. Writing has no effect.

31.4.2 DFSTR (WorkFlash STatus Register) WorkFlash Status Register

This register indicates the WorkFlash state.

- **DFSTR : Address 2303_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved					DFECCERR	DFHANG	DFRDY	
0	0	0	0	0	0	0	1	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R/W	R,WX	R,WX	Attribute

[bit7 to bit3] Reserved

These bits are reserved. The read value is undefined. Writing has no effect.

[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 bits or more in a single word, the read value of this bit is undefined.

[bit1] DFHANG (WorkFlash HANG): WorkFlash Hang State

This bit indicates the WorkFlash memory HANG 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 "■ Command Sequence").

The correct value might not be read out immediately after a command of automatic program algorithm has been issued. Therefore, ignore the first read value of this bit after the command issuance.

DFHANG	Description
0	Normal state
1	HANG UP state

[bit0] DFRDY (Flash ReaDY): Data 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.

DFRDY	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 program algorithm has been issued. Therefore, ignore the first read value of this bit after the command issuance.

31.4.3 FLIFCTLR(Flash I/F Control Register) Flash Interface Control Register

Flash I/F is controlled. This register is shared by program flash and WorkFlash.

- **FLIFCTLR : Address 2308_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0	
0	0	0	0	0	0	0	0	Initial value
RX,WX	RX,WX	RX,WX	R/W	RX,WX	RX,WX	R/W	R/W	Attribute

[bit7 to bit5] Reserved

These bits are reserved. The read value is undefined. Writing has no effect.

[bit4] DFWDSBL (Data Fetch Wait cycle Disable) Data Fetch Wait cycle Disable

If this bit is set to "1", wait cycle to be inserted while data fetching during wait configuration is disabled. However, the wait cycle for cycle time guarantee cannot be disabled.

DFWDSBL	Description
0	Wait cycle Enabled (Initial value)
1	Wait cycle Disabled

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect.

[bit 2]

This bit is reserved. Always write "0" to this bit.

[bit1] ECCDSBL1(ECC Disable1) ECC function Disable 1

Under CPU mode, enabling/disabling of ECC function is configured against write access / data fetching to WorkFlash memory.

ECCDSBL1	Description
0	ECC function Enabled (Initial value)
1	ECC function Disabled

[bit0] ECCDSBL0(ECC Disable0) ECC function Disable 0

Under CPU mode, enabling/disabling of ECC function is configured against write access / data fetching to program flash memory.

ECCDSBL0	Description
0	ECC function Enabled (Initial value)
1	ECC function Disabled

31.5 Operation

This section explains the method for accessing the flash area.

31.5.1 Access Mode Setting

The flash memory in this model 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
(Flash memory accessible from external)

■ Configuring CPU-ROM Mode

CPU ROM mode is in effect when the FWE bit of the WorkFlash control register (DFCTLR) is "0". In CPU ROM mode, reading flash memory is enabled when the DFRDY bit of the WorkFlash status register (FSTR) is "1". In CPU ROM mode, writing to flash memory is disabled. After releasing of reset, the flash memory enters this state.

■ Configuring CPU Programming Mode

CPU programming mode is in effect when the FWE bit of the WorkFlash control register (DFCTLR) is "1". In CPU programming mode, reading and writing flash memory are enabled when the DFRDY bit of the WorkFlash status register (FSTR) is "1".

31.5.2 Programming Flash Memory by CPU

After configuring CPU programming mode, perform erasing and programming using the automatic program algorithm. In this model, because error correction codes (ECC) are added to each single word, programming needs to be performed for each single word. In the following procedure, each word is programmed by two operations to write one half-word. If this procedure is not followed, the written values will not be read correctly because the values will be written to flash memory without calculating the ECC.

(1) Set the flash access size to 16 bits. (FCTLR:FSZ[1:0]=01)
(Note) See "CHAPTER 30 FLASH MEMORY" for FCTLR.
(2) Issue the write command. Write address = PA, write data = PD[31:16]
See "31.5.5 Write Command" for details on the write command.
(3) Read the hardware sequence flag until the write has finished.
See "■ 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 that was written and check that the correct value is read. Even if the correct value could be read, check the DFSTR:FECCERR bit to confirm that there was no ECC correction.
If ECC correction occurred, write again starting from erasing the flash memory.

PA: Write address (word alignment)
PD[31:0]: Write data
PD[31:16]: Write data upper 16 bits
PD[15:0]: Write data lower 16 bits

31.5.3 Automatic Algorithm

When using CPU programming mode, writes and erasures of flash memory are performed by starting the automatic algorithm. This section explains the automatic algorithm.

■ Command Sequence

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 31.5-1 Command Sequence

Command	Number of writes	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	Optional	$F0_H$										
Read	1	RA	RD										
Write	4	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	$A0_H$	PA	PD				
Chip erasure	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 erasure	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 suspended	1	Optional	$B0_H$										
Resumption of sector erasure	1	Optional	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 bytes or half-words.

* The addresses written in the table only show the lower 12 bits. Set the upper 20 bits to any address within the address range of the target flash macro.

PA: Write address (half-word alignment)

PD: Write data (Write as 16 bits.)

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 command address and the sector erase command are issued, the lowest 2 bits of sector address (SA) to be input must be as follows.
  Half-word access: 2’b00
  Byte access: 2’b01 or 2’b11
  Example 1
  At byte access with command address = (lowest 2 bits of the standard command address changed to 2'b01)
  $AA8_H \rightarrow AA9_H$, $x554_H \rightarrow 555_H$, SA → {SA[31:2], 2’b01}
  (SA: An arbitrary address in the erase target sector to be input when the sector erase command is issued)
  Example 2
  At byte access with command address = (lowest 2 bits of the standard command address changed to 2'b11)
  $AA8_H \rightarrow AAB_H$, $554_H \rightarrow 557_H$, SA → {SA[31:2], 2’b11}
  (SA: An arbitrary address in the erase target sector to be input when the sector erase command is issued)
- If an invalid address or data is written, or writing is performed in wrong sequence, the commands written till then will be cleared.

● Reset command

Sending reset command to a target FLASH memory, each command input entered as shown in Table 31.5-1 is cancelled, and can reenter command again from the 1st round.

However, if each command entry is complete, and automatic algorithm is started, this reset command cannot stop such automatic algorithm.

Only in the case if execution of automatic program algorithm exceeds the timing limit, reset command can be entered, and flash memory is restored to the reset status.

● Read command

Flash memory can be read by sending read command to a target sector. When read command is issued, flash memory retains its read status until other 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. 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 "31.5.5 Write Command" for details on the actual operation.

(Notes)
- When writing half word, if the 4th write command (write data cycle) is written to an odd address, writing cannot be completed. It must be written in an even address.
- In the 1st write command sequence, only one half word data can be written. In order to write more than one data, write once to 1 data, then issue command sequence.
- When security is on, there are restrictions on writing to the flash. See "■ Flash Access Restrictions When Security is On" for details.

● Chip erase command

If the chip erase command is sent to the target sector six times in a row, all sectors of the flash memory can be erased in one step. Once the sixth write has finished, the automatic algorithm starts and the chip erase operation is started. When the automatic erase algorithm is started, "0" is written to all of the cells in the flash memory chip before erasing the entire chip, and there is no need to write to flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

See Section "31.5.6 Chip Erase Command" for details on the actual operation.

● Sector erase command

If the sector erase command is sent to the target sector six times in a row, the sector of the flash memory is erased. When 40 μs elapses (timeout period) after the sixth write has finished, the automatic algorithm starts and the sector erase operation is started. If you want to erase multiple sectors, write the erase code (30_H) to the address of the sector to erase within the 40 μs (timeout period). If the next sector is not input within the timeout period, the sector erase command may become invalid. When the automatic erase algorithm is started, "0" is written to the cells in the sector to erase in flash memory before erasing the sector, and there is no need to write to flash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

See Section "31.5.7 Sector Erase Command" for details on the actual operation.

(Note) When security is on, there are restrictions on writing to the flash. See "■ Flash Access Restrictions When Security is On" for details.

● Sector erase suspend command

If sector erase suspend command is issued while sector erase is in effect, or during command timeout, shifting to the sector erase suspend state is initiated.

Under sector erase suspend state, read operation of memory sector cell that is not subjected to erase can be initiated. However, new write/erase command cannot be accepted.

To resume erase operation from sector erase suspend state, you must issue erase resume command.

Upon receiving of erase resume command, sector erase state is restored, and erase operation is resumed.

If shifting to this state from command timeout state, it shifts to sector erase state and immediately resumes sector erase operation, as soon as erase resume command is properly written, without shifting to command timeout state.

See Section "31.5.8 Sector Erase Suspend Command" for details on the actual operation.

(Notes)
- From issuing of sector erase suspend command, it requires up to 16.7μs until sector erase operation is halted, and read-out of sectors (that are not targeted for erase) can be initiated.
- You can check the DFRDY bit of WorkFlash status register (DFSTR) or TOGG1 of hardware sequence flag to see if those sectors are read-enabled.

■ Automatic Algorithm Execution State

Because writing and erasing flash memory is performed by an automatic algorithm, the operating state can be checked by the hardware sequence flag using the DFRDY bit of the WorkFlash status register (DFSTR) to determine whether or not the automatic program 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 31.5-1 Bit configuration of hardware sequence flag

For half-word access

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

For byte access

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

(Notes)
- Cannot be 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 state

Table 31.5-2 shows the correspondence between the state of each bit of the hardware sequence flag and the flash memory state.

Table 31.5-2 Correspondence between flags and flash memory states

Table		DPOLL	TOGG	TLOV	SETI	TOGG2
Executing	During writing	Inverted data*	Toggle	0	0	-
	During sector/chip erase	0	Toggle	0	1	-
Time limit exceeded	Write operation	Inverted data*	Toggle	1	0	-
	Sector/chip erase operation	0	Toggle	1	1	-
During sector erase suspend	Erase target sector	-	-	-	-	Toggle

*: See "● Bit descriptions" for the values that are read out.

● Bit descriptions

[bit15 to bit8] Undefined bit

[bit7] DPOLL: (Data polling flag bit)

When the hardware sequence flag is read by specifying 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. During write

During 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

Sector erase execution in progress	Reads "0" from the sector being erased.
After erasing the sector	Always reads out "1".

3. During chip erase

Chip erase execution in progress	Always reads out "0".
After erasing the chip	Always reads out "1".

4. During sector erase suspend

Pause state (incomplete)	"0" is read from the sector erase suspend sector.
Sector erase operation Completed	"1" is read from the sector erase suspend sector.

(Notes) 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 Bit 1)

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the automatic program algorithm is running. The value that is read differs depending on the state.

During write / sector erase / chip erase

During write / sector erase / chip erase	If this bit is read sequentially, "1" and "0" are returned 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 Elapsed 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 rewrite an address written with "0" with "1", 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.

(Notes) 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 is running. The value that is read differs depending on the state.

During sector erase:

When erasing sectors, you can check whether the next sector erase code is ready to be accepted by checking this bit before inputting the next sector erase code. Reads out the next value without accessing the address specified for reading the hardware sequence flag.

0	Within sector erase wait period (the next sector erase code (0x30) can be accepted.)
1	The sector erase wait period has been exceeded (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 Bit 2)

Under sector erase suspend state, read-out can be initiated to the sector not targeted for erase, but for sectors targeted for erase, reading cannot be done. While sector erase suspend is in effect, and read address sector is targeted for erase, output data is toggled and indicates with this flag that this sector is targeted for erase.

Reading from erase-targeted sector	Reading this bit sequentially results to alternation of "1" and "0" (toggle operation). Address assigned to read hardware sequence flag will not be accessed.
Reading to non-erase-targeted sector	Specified address data is read.

[bit1, bit0] Undefined bits

31.5.4 Reset command

The flash memory can be reset by sending reset commands sequentially to the target FLASH memory. Because this state is the flash memory initial state, the flash memory always returns to the reset state when the power is turned on or a command finishes successfully. When the power is turned on, there is no need to issue a 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.

31.5.5 Write Command

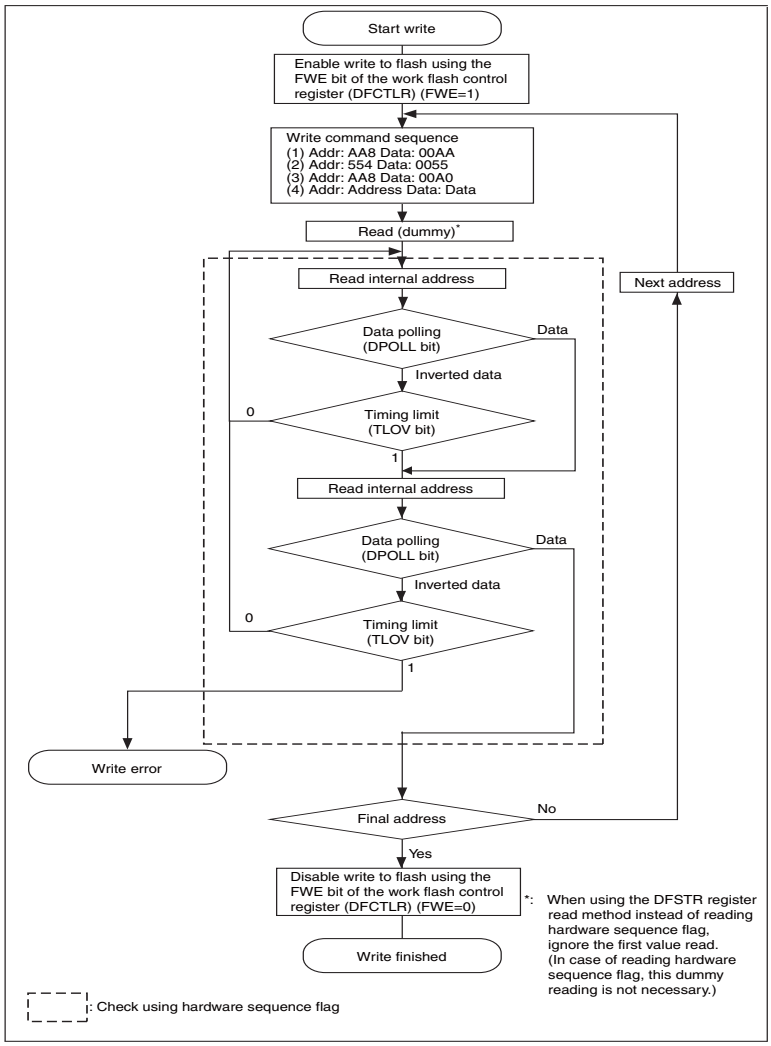
Writes are performed in the following order.

(1) Send write commands sequentially to the target sector
The automatic algorithm is started and data is written to the flash memory.
After issuing a write command, there is no need to control the flash memory externally.

(2) Perform read access to the written address
The read data is the hardware sequence flag. Therefore, if bit7 (DPOLL bit) of the read data matches the written data, the write to the flash memory has finished.
If the write has not finished, the opposite value (inverted data) of the value of bit7 of the last written data is read out.

Examples of writing operation to flash memory are shown below.

Figure 31.5-2 Write procedure example



(Notes)

- Once the write has finished, because the flash memory returns to read mode, write addresses are no longer accepted.
- See "31.5.3 Automatic Algorithm" for details on write commands.
- The value of the DPOLL bit of the hardware sequence flag changes at the same time as the TLOV bit, and needs to be re-checked even if the TLOV bit is "1".
- The toggle operation stops at the same time as the TOGG1 bit and TLOV bit of the hardware sequence flag change to "1". Therefore, even if the TLOV bit is "1", the TOGG1 bit needs to be re-checked.
- Although flash memory can be written to in any order of addresses, even if it crosses a sector boundary, only a single half-word or byte data can be written in each write command sequence. If you want to write multiple data, issue one write command sequence for each data.
- Data that has been written to "0" once cannot be returned to "1". If "0" is rewritten with "1", one of the following occurs.

  -The element is judged as faulty by the data polling algorithm.

  -The write rated time is exceeded, and the TLOV bit of the hardware sequence flag changes to "1".

  -It appears to have been written as "1".

  However, even if it appears to have been written as "1", the actual data remains "0" and "0" will be read out when the data is read in read/reset mode. If you want to return data to "1", perform a chip erase or sector erase.
- During write operations, all commands written to flash memory are ignored.
- If this device is reset during a write, the data that was written cannot be guaranteed.
- Because this model has the ECC bit added, data always needs to be written as 32 bits by two 16-bit writes. See Section "31.5.2 Programming Flash Memory by CPU" for the procedure.

31.5.6 Chip Erase Command

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 Section "31.5.3 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 need to write to flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.
- When security is on, there are restrictions in the procedure for erasing the flash. See "■Unlocking Flash Security" for details.

31.5.7 Sector Erase Command

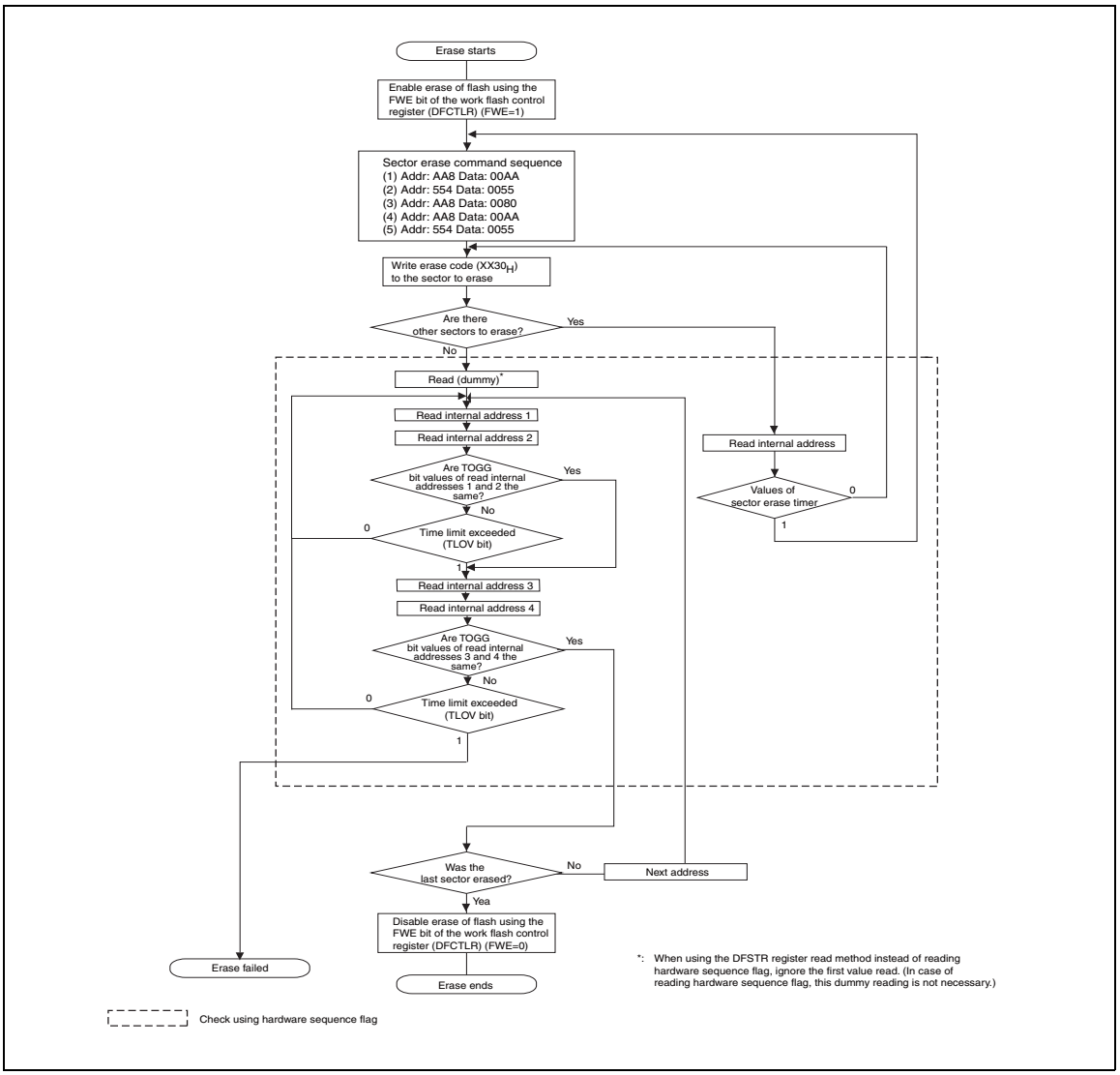
A sector in the flash memory can be selected and only data in the selected sector can be deleted. Multiple sectors can also be specified at the same time. Sector erase is performed in the following order.

(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 or not the sector erase has finished.

Figure 31.5-3 shows an example of the sector erase procedure taking the example of using the TOGG1 bit in the check operation.

Figure 31.5-3 Sector erase procedure example



(Notes)

- The time required to erase the sector is [(sector erasure time + sector write time (preprogram)) × no. of sectors].
- When the sector erase operation finishes, the flash memory returns to the read/reset state.
- See Section "31.5.3 Automatic Algorithm" for details on the sector erase command.
- The value of the DPOLL bit of the hardware sequence flag changes at the same time as the TLOV bit, and needs to be re-checked even if the TLOV bit is "1".
- The toggle operation stops at the same time as the TOGG1 bit and TLOV bit of the hardware sequence flag change to "1". Therefore, even if the TLOV bit is "1", the TOGG1 bit needs to be re-checked.
- The only write commands that can be accepted in a command time-out state are the add erased sector and erase suspend commands. The only command that can be accepted in a sector erase state is the erase suspend command.
- When the automatic erase algorithm is started, "0" is written to the cells to erase in flash memory before erasing the sector, and there is no need to write to flash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

31.5.8 Sector Erase Suspend Command

While sector erase is in progress, or during command timeout, sector erase can temporarily be suspended.

Under sector erase suspend state, memory cell of non-erase-targeted sector can be read. However, new write/erase command cannot be accepted.

In order to temporarily suspend sector erase, issue sector erase suspend command to an arbitrary address of target FLASH macro.

After sector erase is halted, read operation from target FLASH macro is enabled.

Under this circumstance, hardware sequence flag is read from erase-suspended sector.

Under sector erase suspend state, the following conditions are true:

- TOGG1 bit that toggles while erasing a sector does not toggle under this state.
- DFRDY of WorkFlash status register becomes "1".

You can confirm the status of sector erase suspend mode by observing these conditions.

(Notes) From issuing of sector erase suspend command, it requires up to 16.7μs until sector erase operation is halted, and read-out of non-erase-targeted sectors can be initiated.

Even under sector erase suspend state, hardware sequence flag (bit2:TOGG2) toggles. You can utilize this bit to confirm the stopped sectors.

To resume erase operation from under sector erase suspend state, you can send sector erase resume command indicated in Table 31.5-1.

Sector erase resume command can only be received under sector erase suspend state.

You must issue the command after making sure that sector erase suspend state is in effect.

After erase resume command is received, it goes back to sector erase state, and resumes erase operation.

31.5.9 Security Function

This flash memory is equipped with a security function. When the security function is off, the flash memory can be used without limits. However, when the security function is on, writes and erases other than chip erase are suppressed. See "■ Flash Access Restrictions When Security is On" for details on the restrictions.

■ Flash Security On/Off Determination When Reset Released

The flash interface of this model 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.

■ Flash Security Setting Method

When reset is input and released after writing 0x0001 to the flash security code area (see "Figure 30.3-2 Sector configuration diagram (MB91F585)" in "CHAPTER 30 FLASH MEMORY"), security is turned on. Once security has been turned on once, the security does not turn off unless the entire flash memory area is erased.

■ Unlocking Flash Security

The chip erase command can be performed on all flash macros using the following procedure.

1. WorkFlash is erased.
2. Program flash that contains flash security code is erased.

As indicated above, erasing of program flash must be done at last. Otherwise, erase command issued to program flash will be ignored. Also, if reset is issued between each erase, you must start over from step 1.

(Note) In user mode (internal FLASH start), issuing the erase command to an arbitrary flash macro and erasing data in the flash macro are available. For protecting data stored in the flash macro, following the above procedure in the described order is recommended for erasing the chip.

■ Flash Access Restrictions When Security is On

When security is on, the restrictions shown below are created by the start mode.

Table 31.5-3 Access restrictions when security is on

Operating mode	Access restriction
User	In normal mode (the state where there are no access restrictions due to the following flash security violations), 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 above (writer, etc.)	Access to flash memory is restricted. The data from reads is masked and 0xFFFF_FFFF is returned. Write commands and sector erase commands are ignored. Chip erase commands are accepted. See "■ Unlocking Flash Security".

- Data access error is raised, as invalid instruction exception or data access error interrupt is generated. (See the "FR Family FR81 32-BIT MICROCONTROLLER PROGRAMMING MANUAL" for details.)
- "0xFFFFFFFF" is returned as a reading value.

However, this limitation does not exist for the access by OCDU when the OCD tool is connected or reading at the debug state.

31.5.10 Warnings When Using Flash Memory

- 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 (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 flash 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 bits by two 16-bit writes. See Section "31.5.2 Programming Flash Memory by CPU" for the procedure.
- Do not issue commands to multiple macros simultaneously (i.e. in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- Once authentication is completed using the on-chip debugger (OCD) password, OCD can be used to read the content of flash memory externally even if security is on. If you want to prevent a third party from reading, always set a password for enabling on-chip debugger (OCD) startup.
- While FLASH program/erase is in effect, shifting to standby state is prohibited.
- This flash memory has built-in ECC, and overwriting of data to addresses with values already assigned cannot be done.

CHAPTER 32 FIXEDVECTOR FUNCTION

32.1 Overview

The FixedVector function is a function for returning the starting address of flash memory + 0x0024 instead of the content of flash memory at the address (0xF_FFFC) corresponding to the interrupt vector on reset.

32.2 Features

Interrupt vector on reset returned by the FixedVector function: 0x0007_0024

32.3 Configuration

See "Figure 30.3-2 Sector configuration diagram (MB91F585)" in "CHAPTER 30 FLASH MEMORY" for the configuration diagram.

32.4 Registers

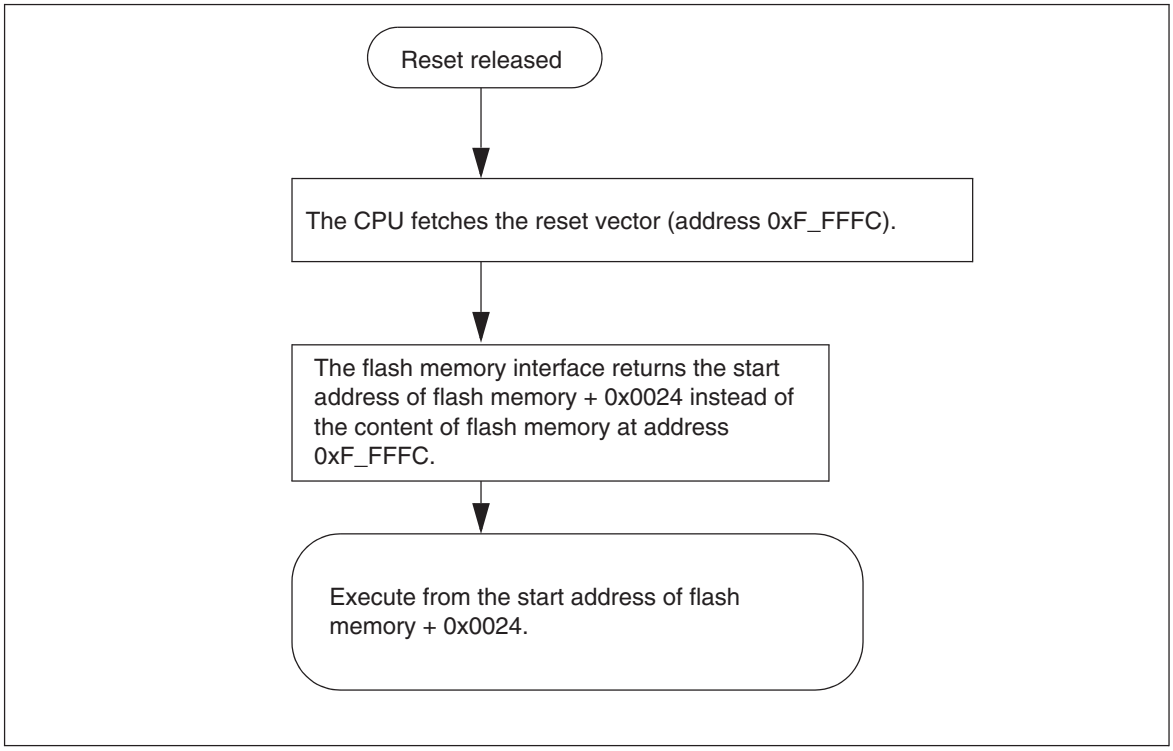
None.

32.5 Operation Description

32.5.1 Operation After Reset Released

In the following sequence, the starting address of flash memory + 0x0024 is returned instead of the content of 0xF_FFFC in flash memory when reset is released.

Figure 32.5-1 Operation sequence after reset



Code: ZBXFLASH-0v4-91580MS-1-E

32.5.2 Usage

After the reset is released, this model executes from the starting address of flash memory + 0x0024 instead of the value written at address 0x000F_FFFC.

32.5.3 Notes

During reads from addresses 0x000F_FFFC to 0x000F_FFFF other than reset vector fetch (Example: the call destination when INT #00H is executed while TBR is its initial value (=0x000F_FC00)), the content of flash memory at the addresses 0x000F_FFFC to 0x000F_FFFF is returned.

CHAPTER 33 WILD REGISTER

33.1 Overview

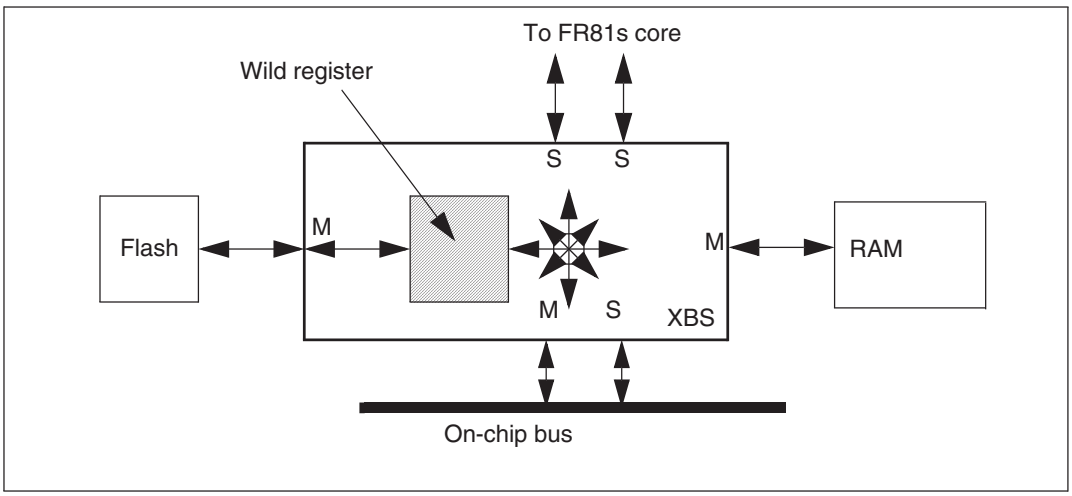
The function of the wild register is to switch the patch target address data that has been set to the address register with the data that has been set to the data register.

33.2 Features

- 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

33.3 Configuration

Figure 33.3-1 Configuration diagram



(Note) This function cannot be used if access wait to the FLASH memory is set to 1 cycle.

Code: WR-1v0-91580MS-1-E

33.4 Registers

Table 33.4-1 Register map

Address	Register				Register function
	+0	+1	+2	+3	
0x0858	Reserved		WREN		Wild register data enabled register
0x0880	WRAR00				Wild register address register 00
0x0884	WRDR00				Wild register data register 00
0x0888	WRAR01				Wild register address register 01
0x088C	WRDR01				Wild register data register 01
0x0890	WRAR02				Wild register address register 02
0x0894	WRDR02				Wild register data register 02
0x0898	WRAR03				Wild register address register 03
0x089C	WRDR03				Wild register data register 03
0x08A0	WRAR04				Wild register address register 04
0x08A4	WRDR04				Wild register data register 04
0x08A8	WRAR05				Wild register address register 05
0x08AC	WRDR05				Wild register data register 05
0x08B0	WRAR06				Wild register address register 06
0x08B4	WRDR06				Wild register data register 06
0x08B8	WRAR07				Wild register address register 07
0x08BC	WRDR07				Wild register data register 07
0x08C0	WRAR08				Wild register address register 08
0x08C4	WRDR08				Wild register data register 08
0x08C8	WRAR09				Wild register address register 09
0x08CC	WRDR09				Wild register data register 09
0x08D0	WRAR10				Wild register address register 10
0x08D4	WRDR10				Wild register data register 10
0x08D8	WRAR11				Wild register address register 11
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
0x08B0	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

33.4.1 WREN (Wild Register ENable register) Wile Register Data Enable Register

This register sets whether the wild register function is enabled or disabled on each channel.

- **WREN :Address 085A$_H$ (Access: Half-word)**

15	14	• • •	2	1	0	bit
WREN[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] WREN15 to WREN0 (Wild Register ENable) Enable Bits

These bits set whether the wild register function is enabled or disabled on each channel.

WREN0 to WREN15	Function
0	Disables the wild register function of ch.n
1	Enables the wild register function of ch.n

33.4.2 WRAR00 to 15 (Wild Register Address Register 00 to 15)Wild Register Address Registers 00 to 15

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)**

31	30	29	28	27	26	25	24	bit
-								
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

23	22	21	20	19	18	17	16	bit
-		WRAR[21:16]						
0	0	X	X	X	X	X	X	Initial value
R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
WRAR[15:8]								
X	X	X	X	X	X	X	X	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
WRAR[7:2]						-		
X	X	X	X	X	X	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R0,WX	Attribute

[bit31 to bit22] Undefined

These bits always read "0". Writing has no effect on operation.

[bit21 to bit2] WRAR21 to WRAR2 (Wild Register Address Register) Address Register

Address to be patched is configured. Target address is (WRAR&0x003FFFFC).

While wild register operation is enabled, read value is undefined.

[bit1, bit0] Undefined

These bits always read "0". Writing has no effect on operation.

33.4.3 WRDR00 to 15 (Wild Register Data Register 00 to 15) Wild Register Data Registers 00 to 15

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)**

31	30	• • •	2	1	0	bit
WRDR[31:0]						
X	X	• • •	X	X	X	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit31 to bit0] WRDR31 to WRDR0 (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.

33.5 Operation Description

This function is used to patch the flash area. Because the enable register is initialized by reset, this register needs to be set on each reset when being used.

The address to be configured must not overlap. Read value of overlapped address is undefined.

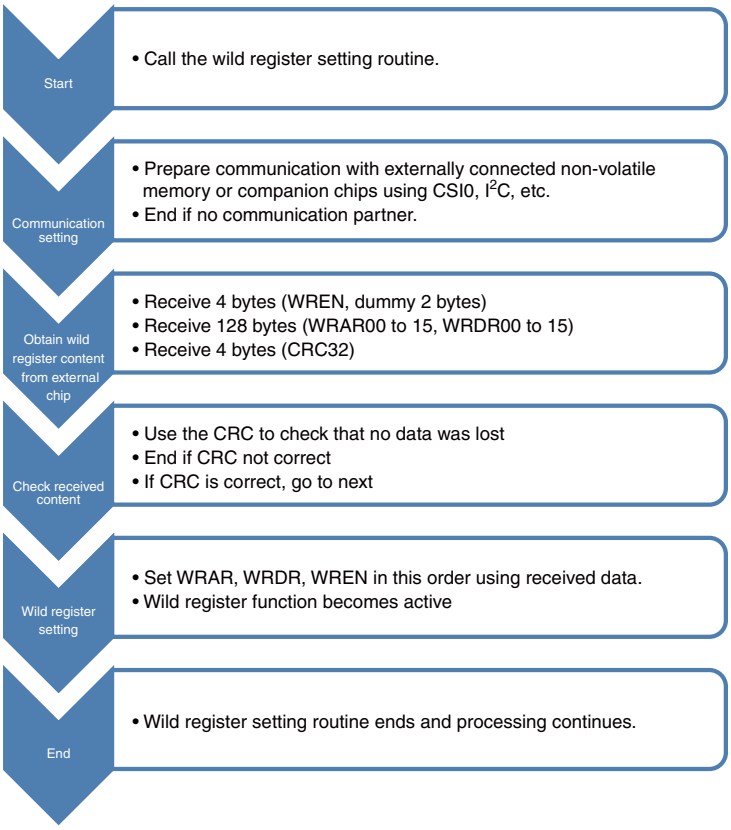
The byte order of data is big-endian.

The target area to be replaced is in the FLASH area only.

33.6 Usage Example

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 33.6-1 Usage example



CHAPTER 34 CRC

34.1 Overview

This product includes 2 CRC channels.

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.

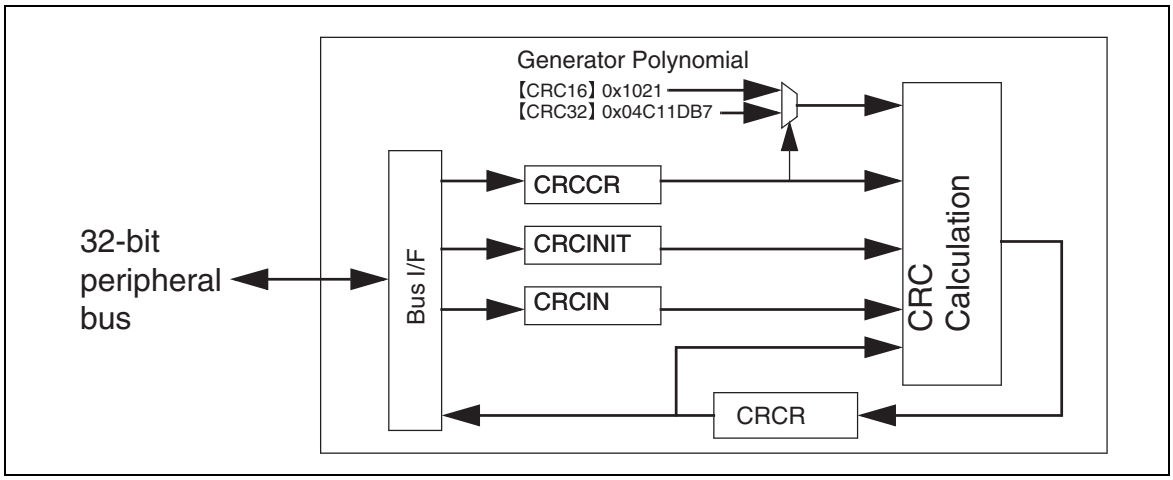
34.2 Features

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

34.3 Configuration

Figure 34.3-1 Block Diagram



Code: FS15-2v1-91580MS-1-E

34.4 Registers

Table 34.4-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x10C0	Reserved			CRCCR	CRC control register 0
0x10C4	CRCINIT				Initial value register 0
0x10C8	CRCIN				Input Data register 0
0x10CC	CRCR				CRC register 0
0x10D0	Reserved			CRCCR1	CRC control register 1
0x10D4	CRCINIT1				Initial value register 1
0x10D8	CRCIN1				Input Data register 1
0x10DC	CRCR1				CRC register 1

34.4.1 CRCCR (CRC Control Register)

This register controls CRC calculation.

- **CRCCR: Address $10C3_H$ (Access: Byte, Half-word, Word)**
- **CRCCR1: Address $10D3_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W	Attribute

[bit7] Reserved

This bit must always be written to "0".

[bit6] FXOR (Final XOR) Final XOR Control Bit

Computes XOR of CRC results and XOR values and outputs the results. 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

Bit to set 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

Bit to set 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

Bit to set 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

Bit to set 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

Bit to select 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 CRC calculation.

34.4.2 CRCINIT (CRC Initial value register)

This register sets the initial value for CRC calculation.

- **CRCINIT: Address $10C4_H$ (Access: Byte, Half-word, Word)**
- **CRCINIT1: Address $10D4_H$ (Access: Byte, Half-word, Word)**

31	30	• • •	2	1	0	bit
D[31:0]						
1	1	• • •	1	1	1	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit31 to bit0] D (Data) Initialization bit

Stores the initial value for CRC calculation. Software writes the initial value for CRC calculation. (0xFFFF_FFFF is applied after reset.) For CRC16, D15 to D0 are used and D31 to D16 are ignored.

34.4.3 CRCIN (CRC INput data register)

This register sets the input data for CRC calculation.

- **CRCIN: Address $10C8_H$ (Access: Byte, Half-word, Word)**
- **CRCR1: Address $10D8_H$ (Access: Byte, Half-word, Word)**

31	30	• • •	2	1	0	bit
D[31:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit31 to bit0] D (Data) Input Data bit

Sets the input data for CRC calculation. Software writes the input data for CRC calculation. The bit width of 8, 16 or 32 is used. These bit 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.

34.4.4 CRCR (CRC Register)

This register outputs the result for CRC calculation.

- **CRCR: Address $10CC_H$ (Access: Byte, Half-word, Word)**
- **CRCR1: Address $10DC_H$ (Access: Byte, Half-word, Word)**

31	30	• • •	2	1	0	bit
D[31:0]						
1	1	• • •	1	1	1	Initial value
R,WX	R,WX	• • •	R,WX	R,WX	R,WX	Attribute

[bit31 to bit0] D (Data) CRCbit

Outputs the result for CRC calculation. When software writes "1" to the initialization bit (CRCCR.INIT), the initial value register (CRCINIT) value is loaded to this register. When software writes the input data for 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.

34.5 Operation

34.5.1 CRC Definition

■ CCITT CRC16 Standard

Generator polynomials	0x1021	(CRCCR:CRC32=0)
Initial value	0xFFFF	
Final XOR value	0x0000	(CRCCR:FXOR=0)
Bit order	MSB First	(CRCCR:LSBFST=0)
Output bit order	MSB First	(CRCCR:CRCLSF=0)

(Any byte order can be set for input and output.)

■ IEEE-802.3 CRC32 Ethernet Standard

Generator polynomials	0x04C11DB7	(CRCCR:CRC32=1)
Initial value	0xFFFF_FFFF	
Final XOR value	0xFFFF_FFFF	(CRCCR:FXOR=1)
Bit order	LSB First	(CRCCR:LSBFST=1)
Output bit order	LSB First	(CRCCR:CRCLSF=1)

(Any byte order can be set for input and output.)

34.5.2 Reset Operation

To reset, set 0xFFFF_FFFF to initial value register (CRCINIT) and CRC register (CRCR). The other registers are cleared to "0".

34.5.3 Initialization

In initialization by CRCCR.INIT, the value of the initial value register is loaded to the CRC register (CRCR).

34.5.4 Byte and Bit Orders

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:

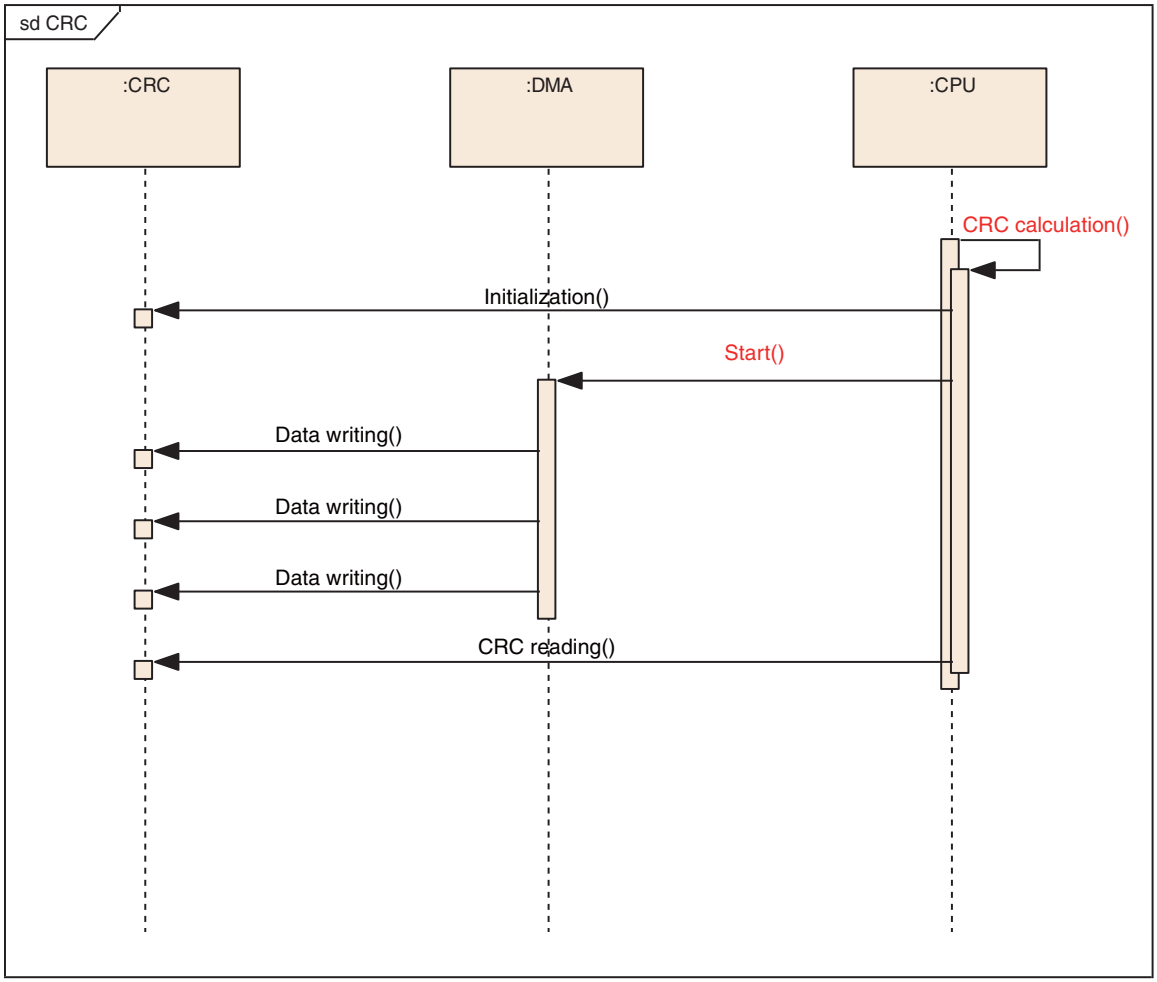
10100001	01001010	11010101	10000000
(first)			(last)

(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.

34.5.5 CRC Calculation Sequence

The sequence for 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 34.5-1 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 CRC calculation. Input data can be written continuously. In addition, there can be different bit widths of writing in a sequence.
- CRC code is obtained with the readout of the CRC register (CRCR).

34.6 Example

34.6.1 Example 1 CRC16, Fixed Byte Input

Figure 34.6-1 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);
```

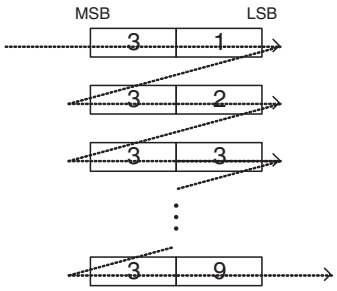
(The following is assumed.)

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

Image of input order into CRC calculator



- Bytes and half words can be written into any position. In this example, data is written into +0 position continuously.
- When CRC16 is used, the CRC result is output in bit15 to bit0 for big-endian byte order and thus the address for H_READ (Half-word reading) is +2 in the example.

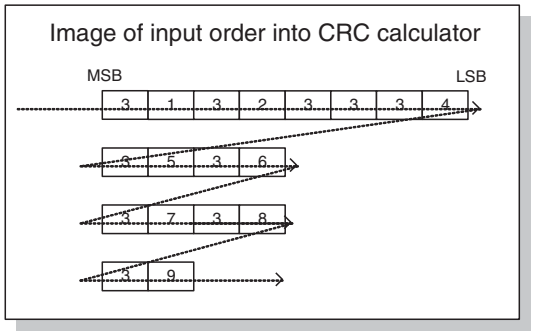
34.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

Figure 34.6-2 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

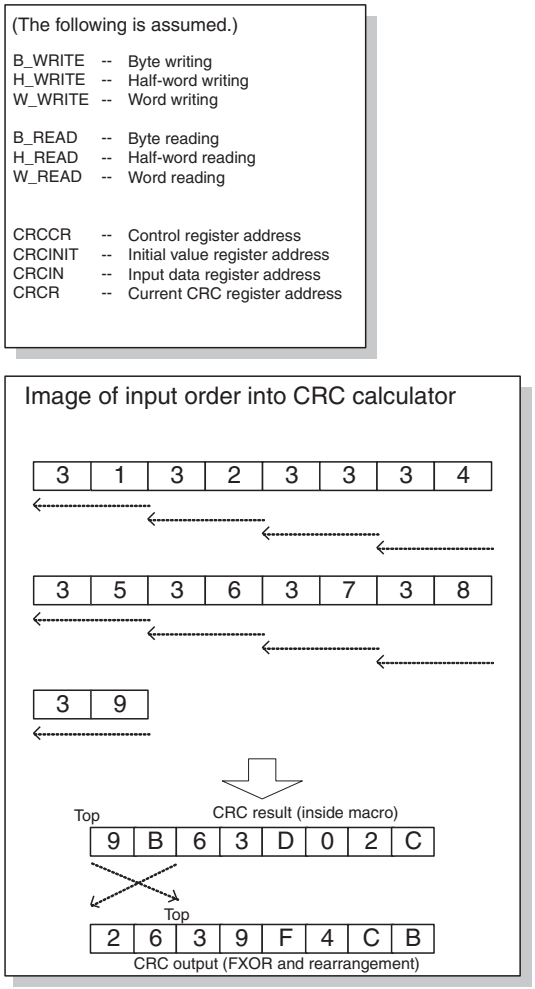


- When the byte and bit orders are set correctly and the orders to input bits to CRC calculator are the same, any writing width can be used.
- For example, there is a case that words are written basically and bytes or a half word is written if there is a fraction of 1, 2, or 3 bytes at the end.

34.6.3 Example 3 CRC32, Byte Order, Big Endian

Figure 34.6-3 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
```



- When CRC32 (IEEE-802.3) is used, the bit order is LSB First. This CRC calculator supports both byte orders and the above figure shows the case for big endian.

34.6.4 Example 4 CRC32, Byte Order, Little Endian

Figure 34.6-4 Example 4

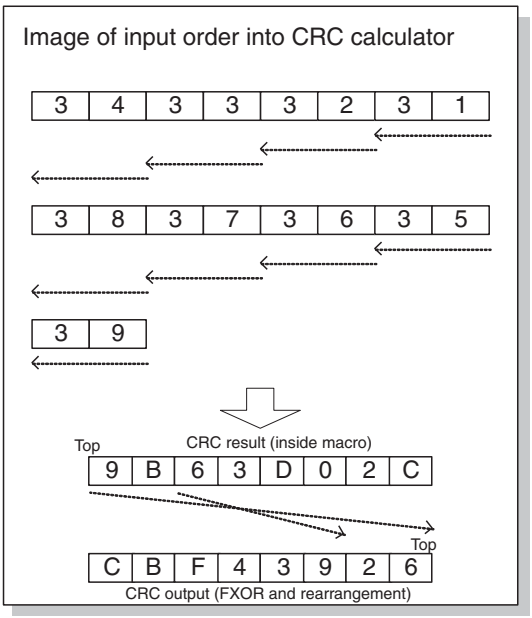
```
//*****************************************************************
//  CRC32 (IEEE-802.3)
//    polynomial:          0x04C11DB7
//    initial value:       0xFFFF_FFFF
//    CRCCR.CRC32     1          // CRC32
//    CRCCR.LTLEND:   1          // little endian
//    CRCCR.LSBFST:   1          // LSB 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 above figure shows the case for little endian.
- When the bit inversion for CRC results is not necessary, the bit inversion for the current result can be canceled with the arithmetic operation after the initialization using 0x3F or setting of CRCCR.FXOR bit to 0 (for example, CRCCR=0x3E) after data input.

CHAPTER 35 MULTI-FUNCTION SERIAL INTERFACE

35.1 Overview

This module provides UART (Asynchronous Serial Interface), CSIO (SPI-supported Clock Synchronous Serial Interface), a LIN interface (v2.1) (LIN Communication Control Interface (v2.1)), and the I^2C serial communication function.

35.2 Features

This product is equipped with a 4ch multi-function serial interface communication module. To use this device, you select one of the UART, CSIO, LIN interface (v2.1), or I^2C by using the serial mode register (SMR) for each channel.

The MB91F583AS/F584AS/F585AS multi-function serial interface communication module contains 2 channels.

(Note) The MB91F583AM/F584AM/F585AM I^2C feature support is only for ch.0, ch.2, and ch.3.
The MB91F583AS/F584AS/F585AS I^2C feature support is only for ch.0.

■ 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	• 5-9 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 the transmission FIFO is empty) • Both transmission and reception employ DMA function • Status interrupt (Serial timer interrupt)

Code: FIP002-1v2-91580MS-1-E

Item	Function
Master/slave mode communication function (multi-processor mode)	1 (Master)-to-n (slave) communication is supported (both master and slave systems are supported)
FIFO option	• Transmission/reception FIFO equipped (transmission FIFO: 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 the transmission FIFO is empty) • Both transmission and reception employ DMA function • Status interrupt (Serial timer interrupt)
Serial chip select	• ch.0: No serial chip select function • ch.1: 1ch control (single control) • ch.2: 1ch control (single control) • ch.3: 1ch control (single control) • Variable setup/hold/deselect times can be set • Active level can be select 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

■ LIN interface (v2.1) (LIN Communication Control Interface (v2.1))

The LIN interface (v2.1) (LIN Communication Control Interface (v2.1)) provides functions to support the 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 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 the 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 sync field connected to input capture by input capture (See Section "43.4.1.3 LIN SYNC FIELD Switching Register (LSYNS)".)
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

■ I²C

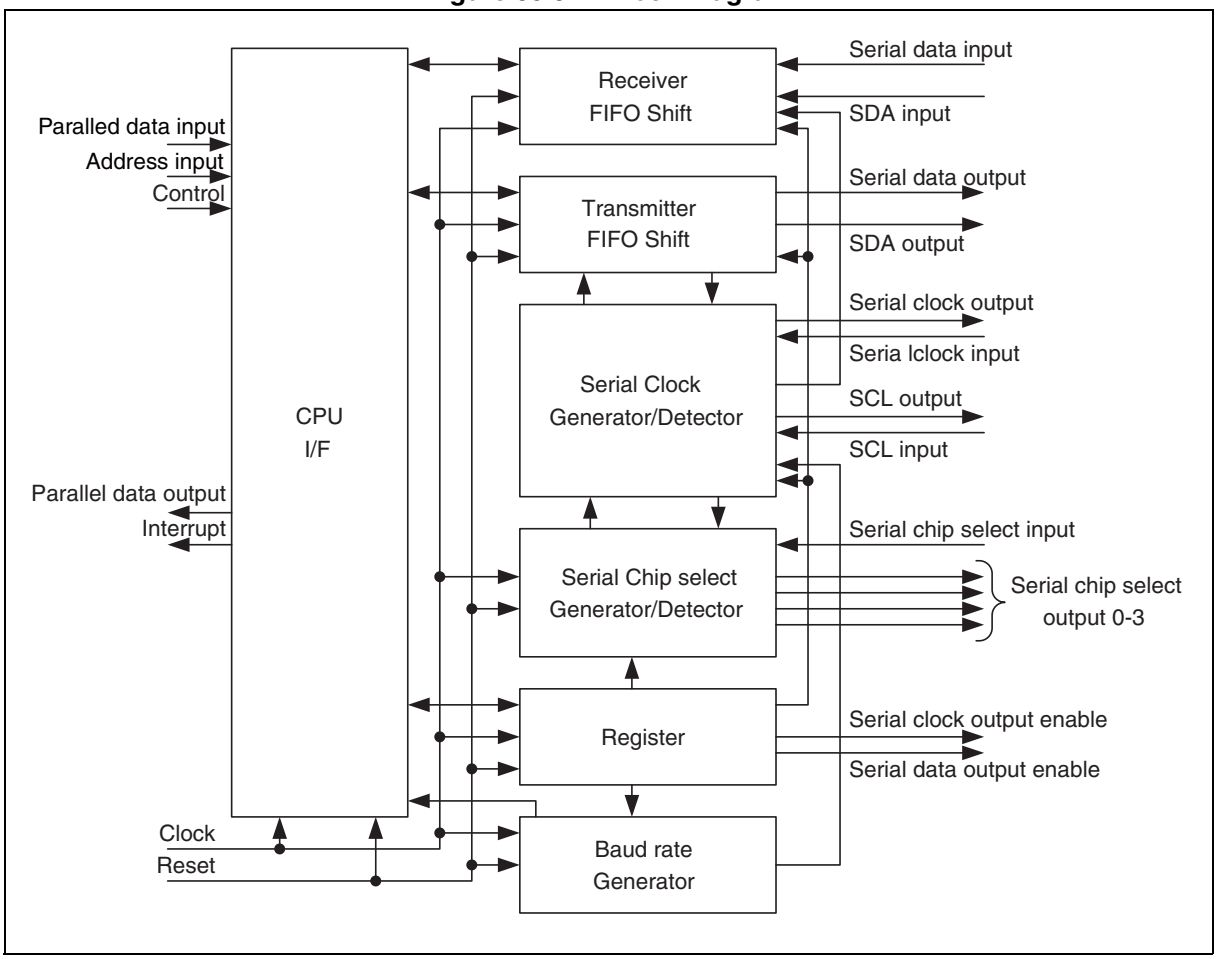
The I²C interface (I²C communication control interface) supports the I²C bus, and runs as a master/slave device on the I²C bus. It is also equipped with the FIFO for transmission/reception (64 bytes each).

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 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) • Transmission FIFO interrupt (when the transmission FIFO is empty) • Both transmission and reception employ DMA function
I²C	• Master/slave transmission/reception function • Adjustment function • Clock synchronous function • Transmission direction detection function • Generation of iterative start condition and detection function • Bus error detection function • General call addressing function • 7-bit addressing as master or slave • Interrupt can be generated at transmission or bus error • 10-bit addressing function is supported by a program
Timer feature	• Employs 16-bit serial timer • Dividing ratio of operating clock is selectable (1/1 to 1/256)
FIFO	• 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) The MB91F583AM/F584AM/F585AM I²C feature support is only for ch.0, ch.2, and ch.3.
The MB91F583AS/F584AS/F585AS I²C feature support is only for ch.0.

35.3 Configuration

Figure 35.3-1 Block Diagram



35.4 Registers

■ Table of Base Addresses (Base_addr) and External Pins

Table 35.4-1 Table of Base Addresses (Base_addr) and External Pins

Channels	Base_addr	External pin			
		SCK SCL*1	SOT SDA*1	SIN	SCS
0	0x1500	SCK0_0/ SCK0_1 *2	SOT0_0/ SOT0_1 *2	SIN0_0/ SIN0_1 *2	-
1	0x1524	SCK1	SOT1	SIN1	SCS1
2	0x1548	SCK2	SOT2	SIN2	SCS2
3	0x156C	SCK3	SOT3	SIN3	SCS3

(Notes) *1: For I^{2}C
*2: I/O relocation is supported only by MB91F583AM/F584AM/F585AM.

■ Registers Map

Table 35.4-2 Registers Map (1 / 5)

Address	Registers				Registers function
	+0	+1	+2	+3	
0x1500	[UART] SCR0 [CSIO] SCR0 [LIN] SCR0 [I^{2}C] IBCR0	[Common] SMR0	[UART] SSR0 [CSIO] SSR0 [LIN] SSR0 [I^{2}C] SSR0	[UART] ESCR0 [CSIO] ESCR0 [LIN] ESCR0 [I^{2}C] IBSR0	––– ch.0 ––– [UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [I^{2}C] I^{2}C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [I^{2}C] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register [I^{2}C] I^{2}C Bus status register
0x1504	[UART] Reserved [CSIO] RDR10/TDR10 [LIN] Reserved [I^{2}C] Reserved		[UART] RDR00/TDR00 [CSIO] RDR00/TDR00 [LIN] RDR00/TDR00 [I^{2}C] RDR00/TDR00		––– ch.0 ––– [CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register [I^{2}C] Transmission/receive data register
0x1508	[UART] SACSR0 [CSIO] SACSR0 [LIN] SACSR0 [I^{2}C] SACSR0		[UART] STMR0 [CSIO] STMR0 [LIN] STMR0 [I^{2}C] STMR0		––– ch.0 ––– [UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [I^{2}C] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register [I^{2}C] Serial timer register
0x150C	[UART] STMCR0 [CSIO] STMCR0 [LIN] STMCR0 [I^{2}C] STMCR0		[UART] Reserved [CSIO] Reserved [LIN] SFUR0 [I^{2}C] Reserved		––– ch.0 ––– [UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [I^{2}C] Serial timer compare register [LIN] Sync field upper limit register

Table 35.4-2 Registers Map (2 / 5)

Address	Registers				Registers function
	+0	+1	+2	+3	
0x1510	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] SFLR10 [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] SFLR00 [I²C] Reserved	––– ch.0 ––– [LIN] Sync field lower limit register
0x1514	Reserved				
0x1518	Reserved				
0x151C	[UART] BGR0 [CSIO] BGR0 [LIN] BGR0 [I²C] BGR0		[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] ISMK0	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] ISBA0	––– ch.0 ––– [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register [I²C] Baud rate generator register [I²C] 7-bit slave address mask register [I²C] Bit slave address register
0x1520	[Common] FCR10	[Common] FCR00	[Common] FBYTE0		––– ch.0 ––– [Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register
0x1524	[UART] SCR1 [CSIO] SCR1 [LIN] SCR1	[Common] SMR1	[UART] SSR1 [CSIO] SSR1 [LIN] SSR1	[UART] ESCR1 [CSIO] ESCR1 [LIN] ESCR1	––– ch.1 ––– [UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register
0x1528	[UART] Reserved [CSIO] RDR10/TDR11 [LIN] Reserved [I²C] Reserved		[UART] RDR01/TDR01 [CSIO] RDR01/TDR01 [LIN] RDR01/TDR01		––– ch.1 ––– [CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register
0x152C	[UART] SACSR1 [CSIO] SACSR1 [LIN] SACSR1		[UART] STMR1 [CSIO] STMR1 [LIN] STMR1		––– ch.1 ––– [UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register
0x1530	[UART] STMCR1 [CSIO] STMCR1 [LIN] STMCR1		[UART] Reserved [CSIO] SCSCR1 [LIN] SFUR1 [I²C] Reserved		––– ch.1 ––– [UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [CSIO] Serial chip select control register [LIN] Sync field upper limit register
0x1534	[UART] Reserved [CSIO] SCSTR31 [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] SCSTR21 [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] SCSTR11 [LIN] SFLR11 [I²C] Reserved	[UART] Reserved [CSIO] SCSTR01 [LIN] SFLR01 [I²C] Reserved	––– ch.1 ––– [CSIO] Serial chip select timing register [LIN] Sync field lower limit register
0x1538	Reserved				

Table 35.4-2 Registers Map (3 / 5)

Address	Registers				Registers function
	+0	+1	+2	+3	
0x153C	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] TBYTE01 [LIN] Reserved [I²C] Reserved	––– ch.1 ––– [CSIO] Transfer byte register
0x1540	[UART] BGR1 [CSIO] BGR1 [LIN] BGR1		[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	––– ch.1 ––– [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register
0x1544	[Common] FCR11	[Common] FCR01	[Common] FBYTE1		––– ch.1 ––– [Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register
0x1548	[UART] SCR2 [CSIO] SCR2 [LIN] SCR2 [I²C] IBCR2	[Common] SMR2	[UART] SSR2 [CSIO] SSR2 [LIN] SSR2 [I²C] SSR2	[UART] ESCR2 [CSIO] ESCR2 [LIN] ESCR2 [I²C] IBSR2	––– ch.2 ––– [UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [I²C] I²C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [I²C] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register [I²C] I²C Bus status register
0x154C	[UART] Reserved [CSIO] RDR12/TDR12 [LIN] Reserved [I²C] Reserved		[UART] RDR02/TDR02 [CSIO] RDR02/TDR02 [LIN] RDR02/TDR02 [I²C] RDR02/TDR02		––– ch.2 ––– [CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register [I²C] Transmission/receive data register
0x1550	[UART] SACSR2 [CSIO] SACSR2 [LIN] SACSR2 [I²C] SACSR2		[UART] STMR2 [CSIO] STMR2 [LIN] STMR2 [I²C] STMR2		––– ch.2 ––– [UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [I²C] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register [I²C] Serial timer register
0x1554	[UART] STMCR2 [CSIO] STMCR2 [LIN] STMCR2 [I²C] STMCR2		[UART] Reserved [CSIO] SCSCR2 [LIN] SFUR2 [I²C] Reserved		––– ch.2 ––– [UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [I²C] Serial timer compare register [CSIO] Serial chip select control register [LIN] Sync field upper limit register
0x1558	[UART] Reserved [CSIO] SCSTR32 [LIN] Reserved	[UART] Reserved [CSIO] SCSTR22 [LIN] Reserved	[UART] Reserved [CSIO] SCSTR12 [LIN] SFLR12	[UART] Reserved [CSIO] SCSTR02 [LIN] SFLR02	––– ch.2 ––– [CSIO] Serial chip select timing register [LIN] Sync field lower limit register
0x155C	Reserved				

Table 35.4-2 Registers Map (4 / 5)

Address	Registers				Registers function
	+0	+1	+2	+3	
0x1560	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] TBYTE02 [LIN] Reserved	––– ch.2 ––– [CSIO] Transfer byte register
0x1564	[UART] BGR2 [CSIO] BGR2 [LIN] BGR2 [I^{2}C] BGR2		[UART] Reserved [CSIO] Reserved [LIN] Reserved [I^{2}C] ISMK2	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I^{2}C] ISBA2	––– ch.2 ––– [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register [I^{2}C] Baud rate generator register [I^{2}C] 7-bit slave address mask register [I^{2}C] Bit slave address register
0x1568	[Common] FCR12	[Common] FCR02	[Common] FBYTE2		––– ch.2 ––– [Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register
0x156C	[UART] SCR3 [CSIO] SCR3 [LIN] SCR3 [I^{2}C] IBCR3	[Common] SMR3	[UART] SSR3 [CSIO] SSR3 [LIN] SSR3 [I^{2}C] SSR3	[UART] ESCR3 [CSIO] ESCR3 [LIN] ESCR3 [I^{2}C] IBSR3	––– ch.3 ––– [UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [I^{2}C] I^{2}C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [I^{2}C] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register [I^{2}C] I^{2}C Bus status register
0x1570	[UART] Reserved [CSIO] RDR13/TDR13 [LIN] Reserved [I^{2}C] Reserved		[UART] RDR03/TDR03 [CSIO] RDR03/TDR03 [LIN] RDR03/TDR03 [I^{2}C] RDR03/TDR03		––– ch.3 ––– [CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register [I^{2}C] Transmission/receive data register
0x1574	[UART] SACSR3 [CSIO] SACSR3 [LIN] SACSR3 [I^{2}C] SACSR3		[UART] STMR3 [CSIO] STMR3 [LIN] STMR3 [I^{2}C] STMR3		––– ch.3 ––– [UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [I^{2}C] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register [I^{2}C] Serial timer register
0x1578	[UART] STMCR3 [CSIO] STMCR3 [LIN] STMCR3 [I^{2}C] STMCR3		[UART] Reserved [CSIO] SCSCR3 [LIN] SFUR3 [I^{2}C] Reserved		––– ch.3 ––– [UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [I^{2}C] Serial timer compare register [CSIO] Serial chip select control register [LIN] Sync field upper limit register
0x157C	[UART] Reserved [CSIO] SCSTR33 [LIN] Reserved [I^{2}C] Reserved	[UART] Reserved [CSIO] SCSTR23 [LIN] Reserved [I^{2}C] Reserved	[UART] Reserved [CSIO] SCSTR13 [LIN] SFLR13 [I^{2}C] Reserved	[UART] Reserved [CSIO] SCSTR03 [LIN] SFLR03 [I^{2}C] Reserved	––– ch.3 ––– [CSIO] Serial chip select timing register [LIN] Sync field lower limit register
0x1580	Reserved				

Table 35.4-2 Registers Map (5 / 5)

Address	Registers				Registers function
	+0	+1	+2	+3	
0x1584	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] Reserved	[UART] Reserved [CSIO] TBYTE03 [LIN] Reserved [I²C] Reserved	––– ch.3 ––– [CSIO] Transfer byte register
0x1588	[UART] BGR3 [CSIO] BGR3 [LIN] BGR3 [I²C] BGR3		[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] ISMK3	[UART] Reserved [CSIO] Reserved [LIN] Reserved [I²C] ISBA3	––– ch.3 ––– [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register [I²C] Baud rate generator register [I²C] 7-bit slave address mask register [I²C] Bit slave address register
0x158C	[Common] FCR13	[Common] FCR03	[Common] FBYTE3		––– ch.3 ––– [Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register

35.4.1 Common Registers

35.4.1.1 Serial Mode Register: SMR

This register selects an operation mode. The functions of bits 3 to 0 vary depending on the mode, which is UART, CSIO, LIN, or I^{2}C.

- **SMR: Address 1501_H, 1525_H, 1549_H, $156D_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
MD[2:0]			Reserved	SBL/ SCINV/ RIE	BDS/TIE (Reserved)	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 blt5] MD[2:0] (MoDe): Operation mode

These bits are used to set the communication method.

000: Operating mode 0 (asynchronous normal mode) is set.
001: Operating mode 1 (asynchronous multi-processor mode) is set.
010: Operating mode 2 (CSIO mode) is set.
011: Operating mode 3 (LIN communication mode) is set.
100: Operating mode 4 (I^{2}C 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^{2}C] Before changing the operation mode, disable I^{2}C (ISMK:EN="0").
- I^{2}C support is only for ch.0, ch.2, and ch.3.
  For MB91F583AS/F584AS/F585AS, only ch.0 is supported.

[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 terminal 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 SCINV is set, set reception enable (SCR: RXE="1").
- Set this bit when transmission and reception are disabled (TXE=RXE="0").
- This bit is used in one of cases below.
  - When chip select terminal 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 terminal 0 is active
- Set this bit to "0" when serial test mode is enabled (SACSR:STST="1").

[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.

(Notes) 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]

This bit is reserved. Always set this bit to "0".

[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").

(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 terminal 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 terminal 0 is active

[I^{2}C]

- 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.

(Notes) Set this bit to "0" when transmitting data with the use of INT bit of I^{2}C 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 general-purpose I/O port or serial clock input pin.

When this bit is set to "1": It becomes a serial clock output terminal allowing clock output while transmitting.

(Notes)
- When you use a SCK terminal 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 terminal as a serial clock output, set the SCK terminal as a peripheral output terminal (set with EPFR). See "CHAPTER 24 I/O PORTS" for how to make setups.

[UART] [LIN] [I^{2}C]

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": The SOT pin stops serial data output.

When this bit is set to "1": The SOT pin functions as a serial data output pin (SOT).

(Note) Set a SOT terminal as a peripheral output terminal (set with EPFR). See "CHAPTER 24 I/O PORTS" for how to make setups.

[I^{2}C]

This bit is reserved. Always set this bit to "0".

35.4.1.2 FIFO Control Register 1: FCR1

FIFO control register (FCR1) is used for the test settings of FIFO, selection of transmission/reception FIFO, settings of transmission FIFO interrupt enable, and control of interrupt flag.

- **FCR1: Address 1520_H, 1544_H, 1568_H, $158C_H$ (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	R0,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) To use the transmission FIFO, 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

- FBYTE (for transmission) = 0 (transmission FIFO is empty)
- Resetting the transmission FIFO

FDRQ reset condition

- Writing "0" to this bit
- If the transmission FIFO becomes full

(Notes)
- When FBYTE (for transmission) is "0", writing "0" to this bit is prohibited.
- When this bit is "0", changing the FSEL bit is prohibited.
- When you set this bit to "1", it does not affect the operation.
- If a read-modify-write instruction is executed, "1" will be read.
- When the transmission FIFO is enabled, writing "0" is valid.
- After writing the necessary data in the transmission FIFO after a transmission interrupt occurs, write "0" to the FIFO transmission data request bit (FDRQ) to clear the interrupt

request.

[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.

35.4.1.3 FIFO Control Register 0: FCR0

FIFO control register 0 (FCR0) is used to enable/disable FIFO operation, reset FIFO, save read pointer, and configure retransmission.

- **FCR0: Address 1521_H, 1545_H, 1569_H, $158D_H$ (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

Always set this bit 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

- Writing to (overwriting) the FIFO when the FLSTE bit in FIFO control register 1 (FCR1) is "1" and the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit

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 is completed, this bit is set 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

In a case where a communication error occurs when the read pointer was saved before transmission, if the FLST bit is "0", retransmission is possible.

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)
- [UART] [CSIO] [LIN] After disabling transmission/reception, reset FIFO2.
- [I^2C] After disabling FIFO2, reset it.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- The valid data count of the FBYTE2 register is set to "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)
- [UART] [CSIO] [LIN] After disabling transmission/reception, reset FIFO1.
- [I^2C] After disabling FIFO1, reset it.
- 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".
- With this bit selected as the reception FIFO by the FCR1:FSEL bit, if a reception error occurs, this bit is cleared to "0". Then, until the reception error is cleared, this bit cannot be set to "1".
- When the transmission FIFO is used, if the transmission buffer is empty (SSR:TDRE="1"), or when the reception FIFO is used, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO2 disabled, the state of FIFO2 will be retained.
- [UART] [CSIO] [LIN] When FIFO2 is set for the transmission FIFO (FCR1:FSEL="1"), FIFO2 contains data, and UART, CSIO, or LIN is enabled for transmission (SCR:TXE="1"), writing "1" to this bit starts transmission immediately. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, SCR:TIE bit and SCR:TBIE bit.
- [UART] [CSIO] [LIN] When you use 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").
- [UART] [CSIO] [LIN] 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").

(Notes) [I^{2}C]

- 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 using 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 using 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".

[bit0] FE1 (FIFO Enable 1) FIFO1 operation enable bit

This bit enables/disables operation of FIFO1.

- To use FIFO1, set this bit to "1".
- With this bit selected as the reception FIFO by the FCR1:FSEL bit, if a reception error occurs, this bit is cleared to "0". Then, until the reception error is cleared, this bit cannot be set to "1".
- When the transmission FIFO is used, if the transmission buffer is empty (SSR:TDRE="1"), or when the reception FIFO is used, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO1 disabled, the state of FIFO1 will be retained.
- [UART] [CSIO] [LIN] When FIFO1 is set for the transmission FIFO (FCR1:FSEL="0"), FIFO1 contains data, and UART, CSIO, or LIN is enabled for transmission (SCR:TXE="1"), writing "1" to this bit starts transmission immediately. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, SCR:TIE bit and SCR:TBIE bit.
- [UART] [CSIO] [LIN] When you use 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 (FBYTE1="0").
- [UART] [CSIO] [LIN] 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").

(Notes) [I^{2}C]

- 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 using 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 using 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".

35.4.1.4 FIFO BYTE Register: FBYTE

- **FBYTE: Address 1522_H, 1546_H, $156A_H$, $158E_H$ (Access: Byte, Half-word, Word)**

Function of this register changes 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 number of data sets.

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 following table lists 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, it outputs serial clocks for the volume of data configured 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. It outputs the SCK clocks for the data volume configured. 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] After disabling reception, change the FBYTE register.
- [UART] [LIN] Set "00"$_H$ in the FBYTE register of the transmission FIFO.
- [CSIO] [I^{2}C] Other than the case of receiving data in the master operation mode, set FBYTE register of the transmission FIFO to "00"$_H$.
- [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.
- [I^{2}C] 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^{2}C] Before you disable the I^{2}C 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^{2}C] If any of the following conditions is met, change the FBYTE register.
  - I^{2}C interface disabled (ISMK:EN="0")
  - SSR:DMA="0" in master reception, and IBCR:INT="1"
  - SSR:DMA="1" in master reception, and SSR:TBI="1"
- [I^{2}C] To receive data in the master operation mode (master reception), clear the SMR:TIE bit to "0", set the reception data count in the FBYTE register of the transmission FIFO, and do not write dummy data to the transmission data register (TDR).
- [Common] Set data that is "1" or greater in FBYTE of the reception FIFO.
- [Common] You cannot use read-modify-write instructions for this register.
- [Common] Settings that exceed the FIFO capacity are prohibited.
- [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.

35.4.2 Registers for UART

35.4.2.1 Serial Control Register: SCR

The serial control register (SCR) allows you to disable/enable transmission and reception, disable/enable transmission/reception interrupts, disable/enable transmission bus idle interrupts, and reset UART.

- **SCR: Address 1500_H, 1524_H, 1548_H, $156C_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
UPCL	Reserved		RIE	TIE	TBIE	RXE	TXE	
0	-	-	0	0	0	0	0	Initial value
R0,W	RX,W0	RX,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] UPCL: Programmable clear bit

This bit initializes the internal state of UART.

When this bit is set to "1":

• UART is directly reset (software reset). In this case, the register settings are maintained.

Note that any active transmission or reception is cut off immediately.

• The baud rate generator restarts by reloading the setting value of the BGR register.

• All the transmission/reception and status interrupt factors (SSR:PE, FRE, ORE, RDRF, TDRE, TBI, TINT) are initialized ("0000110"b).

When this bit is set to "0": There is no effect.

A read always results in "0".

UPCL	Programmable Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Programmable clear	

(Notes)
- Execute a programmable clear after disabling interrupts.
- When using a FIFO, disable the FIFO (FCR0:FE2,FE1="0") before you execute a programmable clear (SSR:UPCL="1").
- Even after you execute a programmable clear (SSR:UPCL="1"), the value of the serial timer register (STMR) is not cleared.

[bit6, bit5] Reserved

Always set these bits to "0".

[bit4] RIE: Reception interrupt enable bit

This bit enables/disables output of reception interrupt requests to the CPU.

When the RIE bit and reception data flag bit (SSR:RDRF) are "1" or any of the error flag bits (SSR:PE, ORE, FRE) is "1", a reception interrupt request is output.

RIE	Reception Interrupt Enable Bit
0	Disables reception interrupt
1	Enables reception interrupt

[bit3] TIE: Transmission interrupt enable bit

This bit enables/disables output of transmission interrupt requests to the CPU.

When the TIE bit and SSR:TDRE bit are "1", a transmission interrupt request is output.

TIE	Transmission Interrupt Enable Bit
0	Disables transmission interrupt
1	Enables transmission interrupt

[bit2] TBIE: Transmission bus idle interrupt enable bit

This bit enables/disables output of transmission bus idle interrupt requests to the CPU.

When the TBIE bit and TBI bit are "1", a transmission bus idle interrupt request is output.

TBIE	Transmission Bus Idle Interrupt Enable Bit
0	Disables transmission bus idle interrupt
1	Enables transmission bus idle interrupt

[bit1] RXE: Reception enable bit

This bit enables/disables UART reception.

RXE	Reception Enable Bit
0	Disables reception
1	Enables reception

(Notes)
- Even when you enable reception (RXE="1"), reception does not start until the falling edge of the start bit is input (in the case of the NRZ format (ESCR:INV="0")). (In the case of the inverted NRZ format (ESCR:INV="1"), reception does not start until the rising edge is input.)
- If you disable reception (RXE="0") while a reception operation is in progress, the reception operation stops immediately.

[bit0] TXE: Transmission enable bit

This bit enables/disables UART transmission.

TXE	Transmission Enable Bit
0	Disables transmission
1	Enables transmission

(Note) If you disable transmission (TXE="0") while a transmission operation is in progress, the transmission operation stops immediately.

35.4.2.2 Serial Status Register: SSR

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag as well as to clear the reception error flag.

- **SSR: Address 1502_H, 1526_H, $154A_H$, $156E_H$ (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	0	1	1	Initial value
R0,W	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit7] REC: Reception error flag clear bit

This bit clears the PE, FRE, and ORE flags of the serial status register (SSR).

• To clear the error flag, write "1" to this bit.

• Writing "0" does not affect anything.

A read always results in "0".

REC	Reception Error Flag Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Clears reception error flag (PE, FRE, or ORE)	

[bit6] Reserved

Always set this bit to "0".

[bit5] PE: Parity error flag bit (functions only in operation mode 0)

• If a parity error occurs while reception is in progress (ESCR:PEN="1"), this bit is set to "1". To clear this bit, write "1" to the REC bit.

• When the PE bit and SCR:RIE bit are "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

PE	Parity Error Flag
0	No parity error
1	Parity error exists

[bit4] FRE: Framing error flag bit

• If a framing error occurs while reception is in progress, this bit is set to "1". To clear this bit, write "1" to the REC bit.

• When the FRE bit and SCR:RIE bit are "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

FRE	Framing Error Flag Bit
0	No framing error
1	Framing error exists

[bit3] ORE: Overrun error flag bit

• If an overrun error occurs while reception is in progress, this bit is set to "1". To clear this bit, write "1" to the REC bit.

• When the ORE bit and SCR:RIE bit are "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

ORE	Overrun Error Flag Bit
0	No overrun error
1	Overrun error exists

[bit2] RDRF: Reception data full flag bit

• This flag indicates the state of the reception data register (RDR).

• When reception data is loaded in RDR, this flag is set to "1". When RDR is read out, the flag is cleared to "0".

• When the RDRF bit and SCR:RIE bit are "1", a reception interrupt request is output.

• While the reception FIFO is in use, RDRF is set to "1" once the reception FIFO has received the specified number of data sets.

• While the reception FIFO is in use, if both of the following conditions are met when reception idle continues for 8 baud rate clocks or more, RDRF is set to "1".

  • The reception FIFO idle detection enable bit (FCR1:FRIIE) is "1".

  • The reception FIFO contains data without receiving the specified number of data sets.

If you read RDR while the counter is counting 8 baud rate clocks, the counter is reset to 0 and starts counting 8 clocks again.

• While the reception FIFO is in use, the bit is cleared to "0" once the reception FIFO becomes empty.

RDRF	Reception Data Full Flag Bit
0	Reception data register (RDR) is empty
1	Reception data register (RDR) contains data

[bit1] TDRE: Transmission data empty flag bit

• This flag indicates the state of the transmission data register (TDR).

• When transmission data is written to TDR, this flag becomes "0", which indicates that valid data exists in TDR.

Once transmission starts after the data has been loaded to the transmit shift register, the bit is set to "1", which indicates that TDR does not contain any valid data.

• When the TDRE bit and SCR:TIE bit are "1", a transmission interrupt request is output.

• When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit is set to "1".

• For details on the timing of setting/resetting the TDRE bit while using the transmission FIFO, see Section "35.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

TDRE	Transmission Data Empty Flag Bit
0	Transmission data register (TDR) contains data
1	Transmission data register (TDR) is empty

[bit0] TBI: Transmission bus idle flag bit

• This bit indicates that UART has no transmission in progress.

• When transmission data is written to the transmission data register (TDR), this bit is set to "0".

• When the transmission data register is empty (TDRE="1") and no transmission is in progress, this bit is set to "1".

• When you set the UPCL bit of the serial control register (SCR) to "1", the TBI bit is set to "1".

• When this bit is "1" and transmission bus idle interrupt is enabled (SCR:TBIE="1"), a transmission interrupt request is output.

TBI	Transmission Bus Idle Flag Bit
0	Transmission in progress
1	No transmission operation

35.4.2.3 Extended Serial Control Register: ESCR

The extended communication control register (ESCR) allows you to set the data length of transmission/reception, enable/disable the parity bit, select a parity bit, invert the serial data format, as well as to select the length of stop bit.

- **ESCR: Address 1503_H, 1527_H, $154B_H$, $156F_H$ (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

[bit7] Reserved

Always set this bit to "0".

[bit6] ESBL: Extended stop bit length select bit

This bit configures the bit length of stop bits (frame end mark for transmission data).

ESBL	Extended Stop Bit Length Select Bit	
0	SMR.SBL="0"	1 bit
	SMR.SBL="1"	2 bits
1	SMR.SBL="0"	3 bits
	SMR.SBL="1"	4 bits

(Notes)
- When receiving, only the first bit of the stop bits is always detected.
- This bit should be set when transmission is disabled (SCR:TXE="0").

[bit5] INV: Inverted serial data format bit

This bit selects either NRZ format or inverted NRZ format as the serial data format.

INV	Inverted Serial Data Format Bit
0	NRZ format
1	Inverted NRZ format

[bit4] PEN: Parity enable bit (functions only in operation mode 0)

This bit sets whether to enable addition (transmission) and detection (reception) of the parity bit.

PEN	Parity Enable Bit
0	Disables parity
1	Enables parity

(Note) In operation mode 1, this bit will be fixed to "0" internally.

[bit3] P: Parity selection bit (functions only in operation mode 0)

When parity is enabled (PEN="1"), this bit selects "1" for odd parity or "0" for even parity.

P	Parity Selection Bit
0	Even parity
1	Odd parity

[bit2 to bit0] L2, L1, L0: Data length select bits

These bits specify the data length of transmission/reception data.

L2	L1	L0	Data Length Select Bits
0	0	0	8 bits long
0	0	1	5 bits long
0	1	0	6 bits long
0	1	1	7 bits long
1	0	0	9 bits long

(Notes)
- Settings other than those listed above are prohibited.
- In operation mode 1, set the data length to 7 or 8 bits. The other settings are prohibited.

35.4.2.4 Receive Data Register/Transmit Data Register: RDR/TDR

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register. When FIFO is enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

- **RDR/TDR: Address 1506_H, $152A_H$, $154E_H$, 1572_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,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	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

■ Read

The receive data register (RDR) is a 9-bit data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN 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 be 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 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: PE, ORE, or FRE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.

■ Write

The transmit data register (TDR) is the 9-bit data buffer register for sending serial data.

- When the transmission operation is enabled (SCR:TXE="1"), writing transmission data to the transmission data register (TDR) transfers the data to the transmit shift register. The transmission data is converted into serial data and 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 Section "35.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

35.4.2.5 Serial Aid Control Status Register: SACSR

Serial Aid Control Status Register (SACSR) has setups for serial test operation control, serial timer startup selection, timer interrupt enable/disable, operating clock division rate for serial timer, and serial timer enable/disable.

- **SACSR: Address 1508_H, $152C_H$, 1550_H, 1574_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(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	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, 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, terminal SOUT will be fixed to "H" and data input to terminal SIN will be ignored.

STST	Serial Test Bit
0	Disables serial test mode
1	Enables serial test mode

(Note) This bit can be changed only when transmission/reception is disabled (SCR:TXE="0", 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.

35.4.2.6 Serial Timer Register: STMR

Serial Timer Register (STMR) indicates the timer values of the serial timer.

- **STMR: Address $150A_H$, $152E_H$, 1552_H, 1576_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.

35.4.2.7 Serial timer Comparison Register: STMCR

Serial Timer Comparison Register (STMCR) sets the comparison value of a timer of the serial timer.

- **STMCR: Address $150C_H$, 1530_H, 1554_H, 1578_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 time, the timer interrupt flag (SACSR:INT) is set to "1".

The following operation is performed at the time interval of (STMCR:TC + 1) x timer operating clock (set by 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 in this register, the timer interrupt flag (SACSR:TINT) is fixed at "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").

35.4.2.8 Baud rate Generator Register: BGR

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

- **BGR: Address $151C_H$, 1540_H, 1564_H, 1588_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.

EXT	External Clock Select Bit
0	Uses internal clock
1	Uses external clock

[bit14 to bit0] BGR14 to BGR0 (Baud rate GeneratoR): Baud rate generator bits

- These bits set division rate 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 "0000"h. 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, the length of "H" part of reception serial clock will be longer than "L" part by one bus clock. When the reload value is an odd number, the lengths of "H" and "L" are 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 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 setup (EXT="1").

35.4.3 Registers for CSIO

35.4.3.1 Serial Control Register: SCR

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.

- **SCR: Address 1500_H, 1524_H, 1548_H, $156C_H$ (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

[bit7] UPCL: Programmable clear bit

This bit initializes the internal state of CSIO.

When this bit is set to "1":

• CSIO is directly reset (software reset). In this case, the register settings are maintained.

Note that any active transmission or reception is cut off immediately.

• The baud rate generator restarts by reloading the setting value of the BGR register.

• All the transmission/reception and status interrupt factors (SSR:TDRE, TBI, RDRF, ORE, TINT, CSE) are initialized.

When this bit is set to "0": There is no effect on operation.

A read always results in "0".

UPCL	Programmable Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Programmable clear	

(Notes)
- Execute a programmable clear after disabling interrupts.
- When using a FIFO, disable the FIFO (FCR0:FE2, FE1="0") before you execute a programmable clear (SSR:UPCL="1").
- Even after you execute a programmable clear (SSR:UPCL="1"), the value of the serial timer register (STMR) is not cleared.

[bit6] MS: Master/Slave function select bit

This bit selects master or slave mode.

MS	Master/Slave Function Select Bit
0	Master mode
1	Slave mode

(Notes)
- When SMR:SCKE="0" and slave mode is selected, an external clock is input directly.
- Set this bit when transmission and reception are disabled (TXE=RXE="0").
- After the MS bit is set, set reception enable (RXE="1").

[bit5] SPI: SPI support bit

This bit is used to execute an SPI communication. When chip select is used in master mode (MS="0"), the bit is used for communication through serial chip selection pin 0.

SPI	SPI Support Bit
0	Normal synchronous transfer
1	SPI communication supported

(Notes)
- Set this bit when transmission and reception are disabled (TXE=RXE="0").
- This bit is used in any of the following cases:
  - Chip selection pin 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 selection pin 0 is active

[bit4] RIE: Reception interrupt enable bit

This bit enables/disables output of reception interrupt requests to the CPU.

When the RIE bit and reception data flag bit (SSR:RDRF) are "1" or the error flag bit (SSR:ORE) is "1", a reception interrupt request is output.

RIE	Reception Interrupt Enable Bit
0	Disables reception interrupt
1	Enables reception interrupt

[bit3] TIE: Transmission interrupt enable bit

This bit enables/disables output of transmission interrupt requests to the CPU.

When the TIE bit and SSR:TDRE bit are "1", a transmission interrupt request is output.

TIE	Transmission Interrupt Enable Bit
0	Disables transmission interrupt
1	Enables transmission interrupt

[bit2] TBIE: Transmission bus idle interrupt enable bit

This bit enables/disables output of transmission bus idle interrupt requests to the CPU.

When the TBIE bit and TBI bit are "1", a transmission bus idle interrupt request is output.

TBIE	Transmission Bus Idle Interrupt Enable Bit
0	Disables transmission bus idle interrupt
1	Enables transmission bus idle interrupt

[bit1] RXE: Reception enable bit

This bit enables/disables CSIO reception.

RXE	Reception Enable Bit
0	Disables reception
1	Enables reception

(Notes)
- If you disable reception (RXE="0") while a reception operation is in progress, the reception operation stops immediately.
- After the MS and SMR:SCINV bits are set, enable reception (RXE="1").

[bit0] TXE: Transmission enable bit

This bit enables/disables CSIO transmission.

TXE	Transmission Enable Bit
0	Disables transmission
1	Enables transmission

(Note) If you disable transmission (TXE="0") while a transmission operation is in progress, the transmission operation stops immediately.

35.4.3.2 Serial Status Register: SSR

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag as well as to clear the reception error flag.

- **SSR: Address 1502_H, 1526_H, $154A_H$, $156E_H$ (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

[bit7] REC: Reception error flag clear bit

This bit clears the ORE flag of the serial status register (SSR).

• To clear the error flag, write "1" to this bit.

• Writing "0" does not affect anything.

A read always results in "0".

REC	Reception Error Flag Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Clears reception error flag (ORE)	

[bit6, bit5] Reserved

Always set these bits to "0".

[bit4] AWC: Access width control bit

This bit selects 16-bit or 32-bit access for accessing the transmission data register (TDR) or reception data register (RDR).

AWC	Access Width Control Bit
0	16-bit access
1	32-bit access

(Notes)
- Change this bit when transmission and reception are disabled (SCR:TXE=RXE="0") and TDR and RDR are empty (SSR:TDRE="1", SSR:RDRF="0").
- For DMA transfer when the transmission FIFO is not used, set this bit to "1".
- When the data length is 20, 24, or 32 bits, set this bit to "1"
- To use SPI mode with slave mode, set "1" to this bit.

[bit3] ORE: Overrun error flag bit

• If an overrun error occurs while reception is in progress, this bit is set to "1". To clear this bit, write "1" to the REC bit.

• When the ORE bit and SCR:RIE bit are "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

ORE	Overrun Error Flag Bit
0	No overrun error
1	Overrun error exists

[bit2] RDRF: Reception data full flag bit

• This flag indicates the state of the reception data register (RDR).

• When reception data is loaded in RDR, this flag is set to "1". When RDR is read out, the flag is cleared to "0".

• When the RDRF bit and SCR:RIE bit are "1", a reception interrupt request is output.

• While the reception FIFO is in use, RDRF is set to "1" once the reception FIFO has received the specified number of data sets.

• While the reception FIFO is in use, if both of the following conditions are met when reception idle continues for 8 baud rate clocks or more, RDRF is set to "1".

  • The 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 is reset to 0 and starts counting 8 clocks again.

• While the reception FIFO is in use, the bit is cleared to "0" once the reception FIFO becomes empty.

RDRF	Reception Data Full Flag Bit
0	Reception data register (RDR) is empty
1	Reception data register (RDR) contains data

[bit1] TDRE: Transmission data empty flag bit

• This flag indicates the state of the transmission data register (TDR).

• When transmit data is written to TDR, this flag turns to "0", which indicates that valid data exists in the TDR. Once transmission starts after data has been loaded to the transmit shift register, the bit is set to "1", which indicates that TDR does not contain valid data.

• When the TDRE bit and SCR:TIE bit are "1", a transmission interrupt request is output.

• When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit is set to "1".

• For details on the timing of setting/resetting the TDRE bit while using the transmission FIFO, see Section "35.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

TDRE	Transmission Data Empty Flag Bit
0	Transmission data register (TDR) contains data
1	Transmission data register (TDR) is empty

[bit0] TBI: Transmission bus idle flag bit

• This bit indicates that CSIO has no transmission in progress.

• When transmission data is written to the transmission data register (TDR), this bit is set to "0".

• When the transmission data register is empty (TDRE="1") and no transmission is in progress, this bit is set to "1".

• When you set the UPCL bit of the serial control register (SCR) to "1", the TBI bit is set to "1".

• When this bit is "1" and transmission bus idle interrupt is enabled (SCR:TBIE="1"), a transmission interrupt request is output.

TBI	Transmission Bus Idle Flag Bit
0	Transmission in progress
1	No transmission operation

(Note) If a serial chip select error (SACSR:CSE="1") occurs when the transmission data register (TDR) is empty (TDRE="1"), this bit becomes "1".

35.4.3.3 Extended Serial Control Register: ESCR

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.

- **ESCR: Address 1503_H, 1527_H, $154B_H$, $156F_H$ (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

[bit7] SOP: Serial output pin set bit

This bit sets the serial output pin (SOT) at the "H" level.

When you write "1" to this bit, the SOT pin is set to "H". After this, "0" does not need to be written to this bit.

A read always results in "0".

SOP	Serial Output Pin Set Bit	
	Write	Read
0	No effect	"0" always read
1	Sets SOT pin to "H"	

(Note) Do not set this bit while serial data transmission is in progress.

[bit5] CSFE: Serial chip select format enable bit

This bit enables or disables the format setting of each serial chip selection pin.

When this bit is set to "1", the following settings are made for each serial chip selection pin:

- Inactive level for serial chip select
- Mark level of the serial clock
- Selection of SPI transfer/normal transfer
- Direction of serial data transfer
- Serial data length

CSFE	Serial Chip Select Format Enable Bit
0	Same data format and clock format are set for all serial chip selection pins
1	Data format and clock format are set for each serial chip selection pin

(Notes) This bit setting is invalid in either of the following cases:

- Chip selection pin disabled (SCSCR:CSEN3-0="0000"b)
- While in slave mode (SCR:MS="1")
- When "1" is set to this bit, configure as follows.
  - Enable the reception FIFO.
  - Set the hold-delay value to "2" or more.
    (Set the CSHD7-0 bit of SCSTR1 register to "2" or more.)
  - Set the data length of a serial chip select to under 9 bits or over 10 bits, communicating to multi slave devices.
    - Set the data length of a serial chipSetting the data length of each serial chip selects to under 9 bits or over 10 bits are prohibited.
      (Example of prohibition)
      Serial chip select terminal 0 = 9 bits
      Serial chip select terminal 1 = 10 bits
      (Example of permission)
      Serial chip select terminal 0 = 16 bits
      Serial chip select terminal 1 = 10 bits

[bit4, bit3] WT1, WT0: Data transmission/reception wait select bits

In master mode, these bits specify the number of waits for successive data transmission or reception. Operation in slave mode is "00".

WT1	WT0	Data Transmission/Reception Wait Select Bits
0	0	0 bits
0	1	1 bit
1	0	2 bits
1	1	3 bits

(Notes) Set "00" to "WT1" and "WT0" bit, when all the conditions listed below are met.

- Using serial chip select function.
- Using SPI mode (SCR:SPI="1").
- Setting "01" to TBYTE register.
- Setting "1" to SCAM bit of SCSCR register

[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"), the bits are used for communication through serial chip selection pin 0.

L3	L2	L1	L0	Data Length Select Bits
0	0	0	0	8 bits long
0	0	0	1	5 bits long
0	0	1	0	6 bits long
0	0	1	1	7 bits long
0	1	0	0	9 bits long
0	1	0	1	10 bits long
0	1	1	0	11 bits long
0	1	1	1	12 bits long
1	0	0	0	13 bits long
1	0	0	1	14 bits long
1	0	1	0	15 bits long
1	0	1	1	16 bits long
1	1	0	0	20 bits long
1	1	0	1	24 bits long
1	1	1	0	32 bits long

(Notes)

- Settings other than those listed above are prohibited.
- These bits are used in any of the following cases:
  - Chip selection pin 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 selection pin 0 is active

35.4.3.4 Receive Data Register/Transmit Data Register: RDR/TDR

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

- **RDR/TDR: Address 1504_H, 1528_H, $154C_H$, 1570_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

■ Read

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 is filled from the lower bit, and the bits other than filled bits are set to "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="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="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 AWC="0", the read values of D31 to D16 are indefinite.

■ Write

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 be come 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.

(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 "35.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

- **Relationship between the Transmission Data Register (TDR) and Transmission Data Empty Flag**

For 16-bit access (SSR:AWC="0"), TDR is located on a 16-bit boundary, and each write operation stores 16-bit transmission data in the register. If TDR contains 32-bit transmission data, the transmission data empty flag (SSR:TDRE) becomes "0".

For 32-bit access (SSR:AWC="1"), TDR is located on a 32-bit boundary, and each write operation stores 32-bit transmission data in the register.

Table 35.4-3 Relationship between the Transmission Data Register (TDR) and Transmission Data Empty Flag

Data Access Width	Data Length	Number of Bits Stored in TDR	TBI Flag	TDRE Flag	Transmission
16-bit access (SSR:AWC="0")	5 to 16 bits	0 bits	1	1	Impossible
		16 bits	0		Possible
		32 bits		0	
	20, 24, or 32 bits	0 bits	1	1	Impossible
		16 bits	0		
		32 bits		0	Possible
32-bit access (SSR:AWC="1")	All data lengths	0 bits	1	1	Impossible
		32 bits	0	0	Possible

35.4.3.5 Serial Auxiliary Control Status Register: SACSR

The serial auxiliary control status register (SACSR) allows you to control serial test operations, select how to activate the serial timer, 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.

- **SACSR: Address 1508_H, $152C_H$, 1550_H, 1574_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

Always set this bit to "0".

[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 either of the cases listed below where transfer byte error is enabled (TBEEN="1") in operation in master mode (SCR:MS="0"), this bit is set to "1" under the following circumstances. The number of transmitted frames is less than the TBYTE setting value, 1 frame has been transmitted at this point in time, and the transmission data register (TDR) contains no valid transmission data (SSR:TDRE="1").

- Chip select is used
- Synchronous transmission 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 not used (SCSCR:CSEN0="0") in slave mode (SCR:MS="1"), this bit is not set to "1".
- If a chip select error occurs (CSE="1"), disable transmission (SCR:TXE="0"), and then write "0" to this bit. To restart transmission, write "0" to this bit, then enable transmission (SCR:TXE="1"), and write transmission data in the transmission data register (TDR).
- If noise that lasts 1 bus clock or longer occurs at serial chip select input during slave transmission, "1" may be set in this bit. In this case, restart transmission after the master transfer is completed.

[bit10, bit9] Reserved

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

Read: The read value is "0".

Write: Always write "0" to this bit.

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	ϕ= 8MHz	ϕ= 10MHz	ϕ= 16MHz	ϕ= 20MHz	ϕ= 24MHz	ϕ= 32MHz
0	0	0	0	ϕ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	ϕ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	ϕ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	ϕ/8	1μs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	ϕ/16	2μs	1.6μs	1μs	800ns	666.67ns	500ns
0	1	0	1	ϕ/32	4μs	3.2μs	2μs	1.6μs	1.33μs	1μs
0	1	1	0	ϕ/64	8μs	6.4μs	4μs	3.2μs	2.67μs	2μs
0	1	1	1	ϕ/128	16μs	12.8μs	8μs	6.4μs	5.33μs	4μs
1	0	0	0	ϕ/256	32μs	25.6μs	16μs	12.8μs	10.67μs	8μs

ϕ: Bus clock

(Notes)
- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped. During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

(Notes)
- 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")

35.4.3.6 Serial Timer Register: STMR

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

- **STMR: Address $150A_H$, $152E_H$, 1552_H, 1576_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".

35.4.3.7 Serial Timer Compare Register: STMCR

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

- **STMCR: Address $150C_H$, 1530_H, 1554_H, 1578_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 compared 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 following operations are performed at the time interval of (STMCR:TC - 1) x timer operating clock (set by SACSR:TDIV3-0).

- SACSR:TINIT is set to "1".
- Transmission begins for synchronous transmission with the serial timer.

(Notes)
- When "0000"h is set to this register, the serial timer register will remain set to "0".
- With synchronous transmission disabled (SACSR:TSYNE="0") and "0000"h set in this register, the timer interrupt flag (SACSR:TINT) is fixed at "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 the serial timer is disabled (SACSR:TMRE="0").

35.4.3.8 Serial Chip Select Control Status Register: SCSCR

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.

- **SCSCR1-3: Address 1532_H, 1556_H, $157A_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved						SCAM	CDIV2	
0	0	0	0	0	0	0	0	Initial value
RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R/W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
CDIV1	CDIV0	CSLVL	Reserved			CSEN	CSOE	
0	0	1	0	0	0	0	0	Initial value
R,W	R,W	R,W	RX,W0	RX,W0	RX,W0	R,W	R,W	Attribute

[bit15 to bit10] Reserved

Always set these bits to "0".

[bit9] SCAM: Serial chip select active retain bit

This bit is used to select whether to retain the active state of the serial chip selection pin or not.

When this bit is set to "1", the serial chip selection pin will not become inactive, even if the transmission operation has completed (SSR:TBI="1"), after the serial chip selection pin becomes active.

If this bit is set to "0" when the serial chip selection pin is active and this bit is "1", the serial chip selection pin becomes inactive after transmission has completed.

SCAM	Serial Chip Select Active Retain Bit
0	Active state of serial chip selection pin not retained
1	Active state of serial chip selection pin retained

(Notes)
- If transmission is disabled (SCR:TXE="0") and a software reset is triggered (SCR:UPCL="1"), the serial chip selection pin becomes inactive regardless of the value of this bit.
- If a serial chip error occurs (SACSR:CSE="1"), the serial chip selection pin becomes inactive regardless of the value of this bit.
- Set "0" to this bit, when all the conditions listed below are met.
  - Using Master mode (SCR:MS="0").
  - Using serial chip select function.
  - Using SPI mode (SCR:SPI="1").
  - Setting "01" to TBYTE register.
  - Setting "00" to WT1 and WT0 bit of ESCR register.

[bit8 to bit6] CDIV3-0: 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 Bits						
			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 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 the serial chip selection pin is inactive.

The bit is used for communication through the chip selection pin.

CSLVL	Serial Chip Select Level Setting Bit
0	Inactive level "L"
1	Inactive level "H"

(Notes)
- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- This bit setting is used in any of the following cases:
  - 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 the serial chip selection pin is active

[bit4 to bit2] Reserved

Always write "0" to these bits.

[bit1] CSEN: Serial chip select enable bit

This bit is used to enable or disable the serial chip selection pin.

CSEN	Serial Chip Select Enable Bit
0	Operation of serial chip selection pin disabled
1	Operation of serial chip selection pin enabled

(Notes)
- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In master mode (SCR:MS="0"), if CSEN is set to "0", the transmission/reception operation is performed regardless of the serial chip selection pin.
- In slave mode (SCR:MS="1"), if CSEN is set to "0", the transmission/reception operation is performed regardless of the serial chip selection pin.

[bit0] CSOE: Serial chip select output enable bit

This bit is used to enable or disable output of the serial chip selection pin.

CSOE	Serial Chip Select Output Enable Bit
0	Output of serial chip selection pin disabled
1	Output of serial chip selection pin enabled

(Notes)
- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In slave mode (SCR:MS="1"), set this bit to "0".

35.4.3.9 Serial Chip Select Timing Register: SCSTR3-0

The serial chip select timing register (SCSTR3-0) is used to set the setup delay time for serial chip select, the hold delay time for serial chip select, and the deselect time for serial chip select.

(Notes)
- Setting value of the setup delay time (CSSU7-0) or hold delay time (CSHD7-0) should be satisfied either of the following condition.
  Baud rate/2[ns] < Hold delay time[ns] + 3 x Bus clock[ns]
  Hold delay time + Setup delay time < Baud rate - 2 x Bus clock[ns]

- **SCSTR1-0: Address 1536_H, $155A_H$, $157E_H$ (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 time when a serial chip selection pin becomes active and the time when the serial clock is output.

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.
- Setting "00"h in these bits is prohibited.

[bit7 to bit0] CSHD7-0: Serial chip select hold delay bits

These bits are used to set a time interval between the time when serial clock output is completed and the time when the serial chip selection pin becomes inactive. If "00"h is set in these bits, the time when serial clock output is completed and the time when the serial chip selection pin becomes inactive are the same.

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.
- Setting "00"h in these bits is prohibited.

- **SCSTR3-2: Address 1534_H, 1558_H, $157C_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[bit15 to bit0] CSDS15-0: Serial chip deselect bits

These bits are used to set a minimum time interval between the timing when a serial chip select pin becomes inactive and the timing when the serial chip select pin becomes active next time.

CSDS15	CSDS14	CSDS13	•••	CSDS2	CSDS1	CSDS0	Minimum deselect time
0	0	0	•••	0	0	0	No minimum deselect time
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, a serial chip selection pin takes at least 5 bus clocks after it becomes inactive until it next becomes active.

35.4.3.10 Transfer Byte Register: TBYTE

The transfer byte register (TBYTE) is used to indicate the transfer data count when each serial chip select pin is active.

- **TBYTE01: Address 153F$_H$ (Access: Byte, Half-word, Word)**
- **TBYTE02: Address 1563$_H$ (Access: Byte, Half-word, Word)**
- **TBYTE03: Address 1587$_H$ (Access: Byte, Half-word, Word)**

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

[bit7 to bit0] TBYTE: Transfer data count display bits

The transfer byte register allows you to set the transfer data count when each serial chip selection pin is active. After the serial chip selection pin becomes active, the serial chip selection pin becomes inactive when the transfer of data of the data count set in these bits is completed.

If the following condition is met, the transfer byte register (TBYTE) is used for synchronous transmission. After the synchronous transmission operation starts, as much data as the data count set in TBYTE is transferred.

- Serial chip select disabled (SCSCR:CSEN="0")

If the value of these bits is changed during the transmission operation (SSR:TBI="0"), the changed transfer data count setting takes effect after the transmission operation of the transfer data count that was set before the change is completed.

TBYTE	Transfer Byte Register
Write	Writing to TBYTE
Read	TBYTE setting value

(Notes)
- If "00"h is set to these bits, the transfer count is 8.
- Do NOT set "01" to this bit, when all the conditions listed below are met.
  - Using Master mode (SCR:MS="0").
  - Using serial chip select function.
  - Using SPI mode (SCR:SPI="1").
  - Setting "1" to SCAM bit of SCSCR register.
  - Setting "00" to WT1 and WT0 bits of ESCR register.

35.4.3.11 Baud rate Generator Register: BGR

Baud rate generator register (BGR) sets the division ratio of serial clock.

- **BGR: Address $151C_H$, 1540_H, 1564_H, 1588_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

Read: The read value is undefined.

Write: No effect.

[bit14 to bit0] BGR (Baud rate GeneratoR): Baud rate generator bits

- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

(Notes)

- Write to the baud rate generator (BGR) in 16-bit access mode.
- If the reload value is an even number, the "H" and "L" widths of the serial clock depend on the SCINV bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  If SMR:SCINV="0", the "H" width of the serial clock is longer by one cycle of the bus clock.
  If SMR:SCINV="1", the "L" width of the serial clock is longer by one cycle of the bus clock .
- Set the reload value to 3 or higher.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15h00". Thus, if you wish to validate a new setting value immediately, execute CSIO reset (SCR:UPCL) after you have changed the setting value of BGR.
- To operate in the slave mode by setting "1" to the reception FIFO idle detection enable bit (FCR1:FRIIE) when you use reception FIFO, set the baud rate at the BGR.

35.4.4 Registers for the LIN Interface (v2.1)

35.4.4.1 Serial Control Register: SCR

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).

- **SCR: Address 1500_H, 1524_H, 1548_H, $156C_H$ (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	R0,W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] UPCL: Programmable clear bit

This bit initializes the internal state of the LIN interface (v2.1).

When this bit is set to "1":

• The LIN interface (v2.1) is directly reset (software reset). In this case, the register settings are maintained. Note that any active transmission or reception is cut off immediately.

• The baud rate generator restarts by reloading the setting value of the BGR register.

• All transmission/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": There is no effect.

A read always results in "0".

UPCL	Programmable Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Programmable clear	

(Notes)
- Execute a programmable clear after disabling interrupts.
- When using a FIFO, disable the FIFO (FCR0:FE2, FE1="0") before you execute a programmable clear.
- Even after you execute a programmable clear (SSR:UPCL="1"), the value of the serial timer register is not cleared.
- Transmission/reception FIFO is not cleared by a programmable clear.

[bit6] MS: Master/Slave function select bit

This bit selects master or slave mode.

MS	Master/Slave Function Select Bit
0	Master mode
1	Slave mode

[bit5] LBR: LIN Break field setting bit (functions only in master operation)

When this bit is set to "1", the LIN Break field and LIN Break delimiter with the length set in the ESCR:LBL1/0 bits and ESCR:DEL1/0 are generated.

Write:

Writing "0": There is no effect.

Writing "1": The LIN Break field is generated.

Read: A read always results in "0".

LBR	LIN Break Field Setting Bit	
	Write	Read
0	No effect	"0" always read
1	Generates LIN Break field	

(Notes)
- This bit functions only in master operation (MS="0").
- Do not set this bit to "1" while the LIN Break field is being generated.

[bit4] RIE: Reception interrupt enable bit

This bit enables/disables output of reception interrupt requests to the CPU.

When the RIE bit and reception data flag bit (SSR:RDRF) are "1" or any of the error flag bits (SSR:FRE, ORE) is "1", a reception interrupt request is output.

RIE	Reception Interrupt Enable Bit
0	Disables reception interrupt
1	Enable reception interrupt

[bit3] TIE: Transmission interrupt enable bit

This bit enables/disables output of transmission interrupt requests to the CPU.

When the TIE bit and SSR:TDRE bit are "1", a transmission interrupt request is output.

TIE	Transmission Interrupt Enable Bit
0	Disables transmission interrupt
1	Enables transmission interrupt

[bit2] TBIE: Transmission bus idle interrupt enable bit

This bit enables/disables output of transmission bus idle interrupt requests to the CPU.

When the TBIE bit and TBI bit are "1", a transmission bus idle interrupt request is output.

TBIE	Transmission Bus Idle Interrupt Enable Bit
0	Disables transmission bus idle interrupt
1	Enables transmission bus idle interrupt

[bit1] RXE: Reception enable bit

This bit enables/disables LIN interface (v2.1) reception.

RXE	Reception Enable Bit
0	Disables reception
1	Enables reception

(Notes)
- Even when you enable reception (RXE="1"), reception does not start until the falling edge of the start bit is input.
- In master operation mode, data is not received even if reception is enabled (RXE="1") during LIN Break field transmission.
- If you disable reception (RXE="0") while a reception operation is in progress, the reception operation stops immediately.

[bit0] TXE: Transmission enable bit

This bit enables/disables LIN interface (v2.1) transmission.

TXE	Transmission Enable Bit
0	Disables transmission
1	Enables transmission

(Note) If you disable transmission (TXE="0") while a transmission operation is in progress, the transmission operation stops immediately.

35.4.4.2 Serial Status Register: SSR

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.

- **SSR: Address 1502_H, 1526_H, $154A_H$, $156E_H$ (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

[bit7] REC: Reception error flag clear bit

This bit clears the FRE and ORE flags of the serial status register (SSR).

- To clear the error flag, write "1" to this bit.
- Writing "0" does not affect anything.

A read always results in "0".

REC	Reception Error Flag Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Clears reception error flag (FRE or ORE)	

[bit6] Reserved

Always set this bit to "0".

[bit5] LBD: LIN Break field detection flag bit

This bit indicates that the LIN Break field has been detected.

When "L" data with 11 or more bits is input to the serial input (SIN), the LBD bit is set to "1". In this case, when "1" is set in the LIN Break field interrupt enable bit (LBIE), a status interrupt is generated.

(Read)

"1" is read: The LIN Break field has been detected.

"0" is read: The LIN Break field has not been detected.

(Write)

Writing "0": The LBD bit is cleared.

Writing "1": There is no effect.

LBD	LIN Break Field Detection Flag Bit	
	Write	Read
0	Clears LBD flag	LIN Break field not detected
1	No effect	LIN Break field detected

(Note) When a read-modify-write instruction is used, "1" is read.

[bit4] FRE: Framing error flag bit

• If a framing error occurs while reception is in progress, this bit is 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 "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

FRE	Framing Error Flag Bit
0	No framing error
1	Framing error exists

(Note) • 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.

[bit3] ORE: Overrun error flag bit

• If an overrun error occurs while reception is in progress, this bit is 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 "1", a reception interrupt request is output.

• If this flag is set, data contained in the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, the reception FIFO enable bit is cleared. As a result, no reception data is stored in the reception FIFO.

ORE	Overrun Error Flag Bit
0	No overrun error
1	Overrun error exists

[bit2] RDRF: Reception data full flag bit

• This flag indicates the state of the reception data register (RDR).

• When reception data is loaded in RDR, this flag is set to "1". When RDR is read out, the flag is cleared to "0".

• When the RDRF bit and RIE bit are "1", a reception interrupt request is output.

• While the reception FIFO is in use, RDRF is set to "1" once the reception FIFO has received the specified number of data sets.

• While the reception FIFO is in use, the bit is cleared to "0" once the reception FIFO becomes empty.

RDRF	Reception Data Full Flag Bit
0	Reception data register (RDR) is empty
1	Reception data register (RDR) contains data

[bit1] TDRE: Transmission data empty flag bit

• This flag indicates the state of the transmission data register (TDR).

• When transmission data is written to TDR, this flag becomes "0", which indicates that valid data exists in TDR.

Once transmission starts after the data has been loaded to the transmit shift register, the bit is set to "1", which indicates that TDR does not contain any valid data.

• When the TDRE bit and TIE bit are "1", a transmission interrupt request is output.

• When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit is set to "1".

• For details on the timing of setting/resetting the TDRE bit while using the transmission FIFO, see Section "35.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

TDRE	Transmission Data Empty Flag Bit
0	Transmission data register (TDR) contains data
1	Transmission data register (TDR) is empty

[bit0] TBI: Transmission bus idle flag bit

• This bit indicates that the LIN interface (v2.1) has no transmission in progress.

• When transmission data is written to the transmission data register (TDR), this bit is set to "0".

• When the LIN Break field is set (SMR:LBR="1"), this bit is set to "0".

• When the transmission data register (TDR) is empty (TDRE="1") and no transmission is in progress, this bit is set to "1".

• When LIN Break field transmission ends and the transmission data register is empty, this bit is set to "1".

• When this bit is "1" and transmission bus idle interrupt is enabled (SCR:TBIE="1"), a transmission interrupt request is output.

TBI	Transmission Bus Idle Flag Bit
0	Transmission in progress
1	No transmission operation

35.4.4.3 Extended Serial Control Register: ESCR

Extended communication control register (ESCR) is used to select LIN Break field interrupt enable/disable, LIN Break field detection, LIN Break field length, LIN Break delimiter length settings, and stop bit length.

- **ESCR: Address 1503_H, 1527_H, $154B_H$, $156F_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	ESBL	Reserved	LBIE	LBL[1:0]		DEL[1:0]		
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W	RX,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] Reserved

Always set this bit to "0".

[bit6] ESBL: Extended stop bit length select bit

This bit configures the bit length of stop bits (frame end mark for transmission data).

ESBL	Extended Stop Bit Length Select Bit	
0	SMR:SBL="0"	1 bit
	SMR:SBL="1"	2 bits
1	SMR:SBL="0"	3 bits
	SMR:SBL="1"	4 bits

(Notes)
- When receiving, only the first bit of the stop bits is always detected.
- This bit should be set when transmission is disabled (SCR:TXE="0").

[bit5] Reserved

Always set this bit to "0".

[bit4] LBIE: LIN Break field detection interrupt enable bit

This bit enables/disables LIN Break field detection interrupt.

When the LIN Break field detection flag (LBD) is "1" and interrupt is enabled (LBIE="1"), a reception interrupt is generated.

LBIE	LIN Break Field Detection Interrupt Enable Bit
0	Disables LIN Break field detection interrupt
1	Enables LIN Break field detection interrupt

[bit3, bit2] LBL1/0: LIN Break field length select bits (function only in master operation)

- These bits set the length of the LIN Break field generation time interval (in bits).
- Before setting the LBR bit of the serial control register (SCR) to "1" (LIN Break field transmission), set this bit.
- In slave operation, the timing of LIN Break field detection is always the 11th bit regardless of the set value of this bit.

LBL1	LBL0	LIN Break Field Length Select Bits
0	0	13-bit length
0	1	14-bit length
1	0	15-bit length
1	1	16-bit length

(Note) This function works only in master operation (SMR:MS="0").

[bit1, bit0] DEL1/0: LIN Break delimiter length select bits

• These bits set the length of the LIN Break delimiter (in bits).

• Before setting the LBR bit of the serial control register (SCR) to "1" (LIN Break field transmission), set this bit.

DEL1	DEL0	LIN Break Delimiter Length Select Bits
0	0	1-bit length
0	1	2-bit length
1	0	3-bit length
1	1	4-bit length

(Note) This function works only in master operation (SMR:MS="0").

35.4.4.4 Receive Data Register/Transmit Data Register: RDR/TDR

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

- **RDR/TDR: Address 1506_H, $152A_H$, $154E_H$, 1572_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
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

■ Read

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN 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 (SSR:either ORE or FRE is "1"), data in the receive data register (RDR) will become invalid.

(Notes)
- When you use reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to SSR:RDRF.
- When you are using reception FIFO, if the reception FIFO becomes empty, SSR:RDRF will be cleared to "0".
- If a reception error occurs (SSR: either ORE or FRE is set to "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.

■ Write

The transmit data register (TDR) is the data buffer register for sending serial data.

- When 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 "35.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

35.4.4.5 Serial Auxiliary Control Status Register: SACSR

The serial auxiliary control status register (SACSR) allows you to control serial test operations. You can also enable/disable automatic baud rate adjustment, enable/disable sync field interrupt, select how to activate the serial timer, enable/disable timer interrupt, set a serial timer operating clock division value, and enable/disable the serial timer.

- **SACSR: Address 1508_H, $152C_H$, 1550_H, 1574_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 the auto baud rate adjustment was executed by "Sync Filed" reception.

This bit is cleared when the fifth falling edge of LIN bus with Sync Field is detected.

BST	Baud rate setting flag	
	Write	Read
0	No effect	Disable auto baud rate adjustment function.
1		Enable auto baud rate adjustment function.

(Notes)
- This bit is fixed to "0" when the auto baud rate adjustment function is disabled (AUTE=0).
- This bit is cleared to "0" by executing programmable clear (SCR:UPCL="1").
- This bit is valid only when Sync Field detection flag is "1"(SACSR:SFD="1").
- Writing has no effect on operation.

[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
0	No Sync Field detected
1	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.
- "1" is read by the read operation of read-modify-write instructions.

[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 "3h" (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

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.

[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

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.

(Notes)
- 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")

35.4.4.6 Serial Timer Register: STMR

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

- **STMR: Address $150A_H$, $152E_H$, 1552_H, 1576_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".

35.4.4.7 Serial Timer Compare Register: STMCR

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

- **STMCR: Address $150C_H$, 1530_H, 1554_H, 1578_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 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". At this time, the time interrupt flag (SACSR:TINT) is set to "1".

The following operation is performed at the time interval of (STMCR:TC + 1) x timer operating clock (set by SACSR:TDIV3-0).

- SACSR:TINIT is set to "1".

(Notes)
- When "0000"h is set to this register, the serial timer register will remain set to "0".
- With "0000"h set in this register, the timer interrupt flag (SACSR:TINT) is fixed at "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 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)

35.4.4.8 Sync Field Upper Limit Register: SFUR

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.

- **SFUR: Address $150E_H$, 1532_H, 1556_H, $157A_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	TU14	TU13	TU12	TU11	TU0	TU9	TU8	
0	0	0	0	0	0	0	0	Initial value
R0,W0	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,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
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

[bit15] Reserved

Always set this bit to "0".

[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).

(Notes)
- These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".
- The read value is undefined.

35.4.4.9 Sync Field Lower Limit Register: SFLR

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.

- **SFLR: Address 1512_H, 1536_H, $155A_H$, $157E_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	TL14	TL13	TL12	TL11	TL0	TL9	TL8	
0	0	0	0	0	0	0	0	Initial value
R0,W0	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,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
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

[bit15] Reserved

Always set this bit to "0".

[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.

(Notes)
- These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".
- The read value is undefined.

35.4.4.10 Baud Rate Generator Register: BGR

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

- **BGR: Address $151C_H$, 1540_H, 1564_H, 1588_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 rate of the serial clock.
- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

(Notes)
- Write to the baud rate generator (BGR) in 16-bit access mode.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15h00". Thus, if you wish to validate a new setting value immediately, execute programmable clear (UPCL) after you have changed the setting value of BGR.
- When the reload value is an even number, the length of "L" part of reception serial clock is longer than "H" part by one bus clock. When the reload value is an odd number, the lengths of "H" and "L" for serial clock are 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 setup(EXT="1").

35.4.5 Registers for I²C

35.4.5.1 I²C Bus Control Register: IBCR

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.

- **IBCR: Address 1500_H, 1548_H, $156C_H$ (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

[bit7] MSS: Master/Slave select bit

When this bit is set to "1", master mode is selected when the I²C bus is in the idle state (ISMK:EN="1", IBSR:BB="0").

After this bit is set to "1" when the BB bit of the IBSR register is "1", this microcontroller waits for the start condition until the IBSR:BB bit becomes "0". While waiting, if it operates as a slave and the slave address matches, this bit is set to "0", and the AL bit of the IBSR register is set to "1".

If you write "0" to this bit when master operation is in progress (MSS="1", ACT="1") and the interrupt flag (INT) is "1", a stop condition occurs.

The MSS bit is cleared in any of the following conditions:

(1) I²C interface disabled (ISMK:EN="0")
(2) Arbitration lost occurred
(3) Bus error detected (BER="1")
(4) "0" written to the MSS bit when INT="1"
(5) DMA mode enabled (SSR:DMA="1") and "0" written to the MSS bit when SSR:TBI="1"

The relationship between the MSS bit and ACT bit is as follows.

MSS	ACT	State
0	0	Idle
0	1	Slave address matches or ACK response made to reserved address (*), and slave operation is in progress (slave mode)
1	0	Master operation wait
1	1	Master operation in progress (master mode)

*: ACK response: Indicates that SDA of the I²C bus is "L" in the acknowledge interval.

MSS	Master/Slave Select Bit
0	Selects slave mode
1	Select master mode

(Notes)
- To change the MSS bit to "0" when DMA mode is disabled (SSR:DMA="0") and the MSS bit is set to "1", do so when MSS bit="1" and INT bit="1". When the ACT bit is "1", writing "0" to the MSS bit also clears the INT bit to "0".
- To change the MSS bit to "0" when DMA mode is enabled (SSR:DMA="1") and the MSS bit is set to "1", do so when MSS bit="1", INT bit="1", or the SSR:TBI bit is "1". When the ACT bit is "1", writing "0" to the MSS bit also clears the INT bit to "0".
- When master operation is in progress, "1" is read out while the ACT bit stays at "1" even if you write "0" to the MSS bit.
- When slave operation is in progress, "1" is read out while the ACT bit stays at "1" even if you write "0" to the MSS bit. To use master mode after slave operation ends, perform the following steps.

1. Confirm the completion of slave operation by detecting the stop condition (IBCR:SPC="1")
2. Write a slave address and transmission data (only when the transmission FIFO is used).
3. Write "1" to this bit.

[bit6] ACT/SCC: Operation flag/repeated start condition generation bit

This bit differs in meaning between read and write.

Read: ACT bit

Write: SCC bit

The ACT bit indicates whether operation is in master mode or slave mode.

ACT bit set conditions:

(1) A start condition is output to the I^2C bus (master mode).

(2) The slave address matches the address sent from the master (slave mode).

(3) A reserved address is detected and an acknowledge response is made for it (slave mode entered when MSS="0").

ACT bit reset conditions:

<Master mode>

(1) A stop condition is detected.

(2) Arbitration lost is detected.

(3) A bus error is detected.

(4) The I^2C interface is disabled (ISMK:EN="0").

<Slave mode>

(1) A (repeated) start condition is detected.

(2) A stop condition is detected.

(3) No acknowledge response has been made for a detected reserved address (IBSR:RSA="1").

(4) The I^2C interface is disabled (ISMK:EN="0").

(5) A bus error occurs (BER="1").

In master mode, writing "1" to this bit executes a repeated start. Writing "0" is invalid.

ACT/SCC	Operation Flag/Repeated Start Condition Generation Bit	
	Write	Read
0	No effect	No operation
1	Generates repeated start condition	I^2C operation in progress

(Notes)

- Write "1" to the SCC bit during a master mode interrupt (MSS="1", ACT="1", INT="1"). When the ACT bit is "1", writing "1" to the SCC bit clears the INT bit to "0".
- In slave mode (MSS="0", ACT="1"), writing "1" to this bit is prohibited.
- When "1" is written to the SCC bit and "0" to the MSS bit, the MSS bit takes precedence.
- For read-modify-write instructions, the SCC bit is read.
- If both of the following conditions are met, "1" is set in the INT bit, and the I^2C bus waits (SCL="L"). To generate a repeated start condition, "1" must be written to the SCC bit again to clear the INT bit.
  - "1" is written to the SCC bit during a master mode interrupt (MSS="1", ACT="1", INT="1", WSEL="1") at the 8th bit.
  - NACK is received at the 9th bit.
- To generate a repeated start condition when DMA mode is enabled (SSR:DMA="1"), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following steps.
  1. Write "1" to the IBCR:INT bit.
  2. Confirm that "1" is set in the IBCR:INT bit.
  3. Write a slave address in TDR.
  4. Set "1" in this bit.

[bit5] ACKE: Data byte acknowledge enable bit

When this bit is set to "1", "L" is output at the acknowledge time.

Change this bit only when any of the following conditions is met.

- DMA mode is disabled (SSR:DMA="0"), ACT="1", and the INT bit is "1".
- DMA mode is enabled (SSR:DMA="1"), ACT="1", and the SSR:TBI bit is "1".
- DMA mode is enabled (SSR:DMA="1"), ACT="1", and SSR:RDRF is "1" during slave reception.
- ACT="0"

This bit is disabled in any of the following conditions.

(1) At acknowledgement for address fields except for reserved addresses (automatic generation)

(2) At data transmission (IBSR:RSA="0", IBSR:TRX="1", IBSR:FBT="0")

(3) In slave reception with the reception FIFO enabled (FCR0:FE="1", MSS="0", ACT="1"), the response is always ACK.

(4) The reception FIFO is enabled and WSEL is "0" in master reception (FCR0:FE="1", MSS="1", ACT="1", WSEL="0"). When the SSR:TDRE bit is "0", the response is ACK. Otherwise, when the SSR:TDRE bit is "1", the response is NACK.

(5) In slave transmission with the reception FIFO enabled, WSEL="0", and a reserved address detected (IBSR:RSA="1", IBSR:TRX="1", IBSR:FBT="1"), the response is always ACK. To respond with NACK at the interrupt time after the detection of a reserved address, disable the reception FIFO and set ACKE="0".

(6) The reception FIFO is enabled, WSEL is "1", and the transmission data register has data in master reception (FCR0:FE="1", MSS="1", ACT="1", WSEL="1", SSR:TDRE="0").

ACKE	Acknowledge Enable Bit
0	Disables acknowledge
1	Enables acknowledge

[bit4] WSEL: Wait select bit

When DMA mode is disabled (SSR:DMA="0"), this bit selects whether to generate an interrupt (INT="1") before or after acknowledgement, to cause the I^2C bus to wait.

When DMA mode is enabled (SSR:DMA="1"), this bit selects whether to generate an interrupt (INT="1", SSR:TBI="1" at transmission, SSR:RDRF="1" at reception) before or after acknowledgement, to cause the I^2C bus to wait.

The WSEL bit is invalid in any of the following conditions.

(1) An interrupt to the 1st byte *1 is generated (INT="1").

(2) A reserved address is detected (IBSR:FBT="1", IBSR:RSA="1").

(3) A NACK response *2 is detected (FCR0:FE="1", IBSR:RACK="1", ACT="1") while data transfer is in progress using a FIFO.

(4) The reception FIFO is being used and becomes full.

*1: 1st byte: Indicates the data after the (repeated) start condition.

*2: NACK response: Indicates that SDA of the I^2C bus is "H" in the acknowledge interval.

WSEL	Wait Select Bit
0	Wait after acknowledgement (9 bits)
1	Wait after data transmission/reception completion (8 bits)

[bit3] CNDE: Condition detection interrupt enable bit

This bit enables interrupt when a stop condition or a repeated start condition is detected in master mode or slave mode (ACT="1"). When the RSC or SPC bit in the IBSR register is "1" and this bit is "1", an interrupt is generated.

CNDE	Condition Detection Interrupt Enable Bit
0	Disables repeated start/stop condition interrupt
1	Enables repeated start/stop condition interrupt

[bit2] INTE: Interrupt enable bit

This bit enables interrupt for data transmission/reception and bus errors in master mode or slave mode (INT="1").

INTE	Interrupt Enable Bit
0	Disable interrupt
1	Enable interrupt

[bit1] BER: Bus error flag bit

This bit indicates that an error has been detected on the I^2C bus.

BER bit set conditions:

(1) A start condition or stop condition is detected during the transfer of the 1st byte (*).

(2) In the 2nd or subsequent byte, a (repeated) start condition or stop condition is detected between the 2nd and 9th (acknowledge) bits of data.

BER bit reset conditions:

(1) "0" is written to the INT bit with BER="1".

(2) The I^2C interface is disabled (ISMK:EN="0").

*: 1st byte: Indicates the data after the (repeated) start condition.

BER	Bus Error Detection Bit
0	No error
1	Error detected

(Note) Check this bit when the interrupt flag (INT bit) becomes "1". If this bit is "1", normal send/receive operations cannot be performed, so perform send/receive or other processing again.

[bit0] INT: Interrupt flag bit

Set this flag to "1" in master mode or slave mode, after 8 or 9 bits (ACK) of data transmission/reception or at the bus error time. Except for bus errors, the SCL state becomes "L" when the INT bit becomes "1", and the SCL state of "L" is canceled when the INT bit becomes "0".

INT bit set conditions:

<8th bit>

<When DMA mode is not a consideration>

(1) A reserved address is detected at the 1st byte.

(2) WSEL is "1", and arbitration lost is detected at the 2nd or subsequent byte.

<When DMA mode is disabled (SSR:DMA="0")>

(1) WSEL is "1", master operation is in progress, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.

(2) WSEL is "1", slave operation is in progress, the reception FIFO is disabled, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.

(3) WSEL is "1", slave transmission is in progress, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.

(4) WSEL is "1", slave reception is in progress, and the reception FIFO is disabled.

<When DMA mode is enabled (SSR:DMA="1")>

(1) WSEL is "1", master operation is in progress, the SSR:TBI bit is "1", and "1" is written to the INT bit at the 2nd or subsequent byte.

<9th bit>

<When DMA mode is not a consideration>

(1) Arbitration lost is detected at the 1st byte.

(2) NACK is received, except for the stop condition output setting (writing of "0" to the MSS bit in master operation).

(3) WSEL is "0", and arbitration lost is detected at the 2nd or subsequent byte.

(4) No reserved address has been detected at the 1st byte, the reception FIFO contains data in the reception direction in master mode or slave mode (IBSR:TRX="0"), when the reception FIFO is enabled.

<When DMA mode is disabled (SSR:DMA="0")>

(1) No reserved address has been detected at the 1st byte, and the SSR:TDRE bit is "1" in the transmission direction in master mode or slave mode (IBSR:TRX="1").

(2) No reserved address has been detected at the 1st byte, the SSR:TDRE bit is "1" in the reception direction in master mode or slave mode (IBSR:TRX="0"), and the reception FIFO is disabled.

(3) The SSR:TDRE bit is "1" at the 2nd or subsequent byte in master mode operation when WSEL="0".

(4) The SSR:TDRE bit is "1" at the 2nd or subsequent byte in transmission in slave mode when WSEL="0".

(5) Reception is performed in slave mode when the reception FIFO is disabled and WSEL="0".

However, in the 1st byte in cases where a reserved address is detected in reception in slave mode, no interrupt is generated at the 9th bit.

(6) The reception FIFO is enabled and becomes full at the reception time in slave mode.

<When DMA mode is enabled (SSR:DMA="1")>

(1) No reserved address has been detected at the 1st byte, and the SSR:TDRE bit is "1" in the transmission direction in slave mode (IBSR:TRX="1").

(2) No reserved address has been detected at the 1st byte, the SSR:TDRE bit is "1" in the reception direction in slave mode (IBSR:TRX="0"), and the reception FIFO is disabled.

(3) "1" is written to the INT bit when the SSR:TBI bit is "1" at the 2nd or subsequent byte in master mode operation when WSEL="0".

<Other>

A bus error is detected.

INT bit reset conditions:

(1) "0" is written to the INT bit.

(2) When the INT bit is "1" and the ACT bit is "1", "0" is written to the MSS bit.

(3) When the INT bit is "1" and the ACT bit is "1", "1" is written to the SCC bit.

When DMA mode is disabled (SSR:DMA="0"), writing "1" to the INT bit is invalid.

INT	Interrupt Flag Bit	
	Write	Read
0	Clears INT bit	No interrupt request
1	No effect	Interrupt request occurred

(Notes)

- With DMA mode enabled (SSR:DMA="1") and the SSR:TBI bit of "1" at the 2nd or subsequent byte in master mode, no status interrupt is generated when "1" is written to the INT bit.
- To issue a repeated start condition when the DMA mode is enabled (SSR:DMA="1"), the SSR:TBI bit is "1", and the IBCR:INT bit is "0", perform the following steps.
  1. Write "1" to the IBCR:INT bit.
  2. Confirm that "1" is set in the IBCR:INT bit.
  3. Write a slave address in TDR.
  4. Set "1" in the IBCR:SCC bit.
- If the INT flag is set to "1", writing "0" to the INT flag cancels the I^2C bus wait.
- When the ISMK:EN bit is "0", the SSR:RDRF bit and INT bit may become "1" depending on the reception timing. In this case, read the reception data to clear the INT bit.
- For read-modify-write instructions, "1" is read.
- When the reception FIFO is enabled, "1" is not set in the INT bit even if the reception FIFO becomes full in master reception operation.
- When issuing a start condition (IBCR:MSS="1"), write "1" to this bit.

35.4.5.2 Serial Status Register: SSR

Serial status register (SSR) checks for the transmission/reception states.

- **SSR: Address 1502_H, $154A_H$, $156E_H$ (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

[bit7] REC: Reception error flag clear bit

This bit clears the ORE flag of the serial status register (SSR).

• Writing "1" clears the ORE bit.

• Writing "0" does not affect anything.

A read always results in "0".

REC	Reception Error Flag Clear Bit	
	Write	Read
0	No effect	"0" always read
1	Clears reception error flag (ORE)	

[bit6] TSET: Transmission buffer empty flag set bit

This bit sets the TDRE bit of the serial status register (SSR).

• Writing "1" sets the TDRE bit. This also sets the TBI bit if DMA mode is enabled (DMA="1").

• Writing "0" does not affect anything.

A read always results in "0".

TSET	Transmission Data Empty Flag Set Bit	
	Write	Read
0	No effect	"0" always read
1	Sets TDRE bit	

(Note) Write "1" to this bit when the IBCR:INT bit is "1".

[bit5] DMA: DMA mode enable bit

This bit enables/disables DMA mode.

• When this bit is set to "1", interrupt conditions correspond to DMA transfer.

• When this bit is set to "0", interrupt conditions are those when DMA transfer is not performed.

For details, see "Table 35.8-1 I^2C Interface Interrupt Control Bits and Interrupt Factors".

DMA	DMA Mode Enable Bit
0	Disables DMA mode
1	Enables DMA mode

(Note) You can change this bit only when ISMK:EN="0".

[bit4] TBIE: Transmission bus idle interrupt enable bit (effective only when DMA mode is enabled)

• This bit enables/disables output of transmission bus idle interrupt requests to the CPU.

• When DMA mode is enabled (DMA="1") and both the TBIE bit and TBI bit are "1", a transmission bus idle interrupt request is output.

• When DMA mode is disabled (DMA="0"), this bit becomes "0" and is kept at "0", and any operating writing to the bit is ignored.

TBIE	Transmission Bus Idle Interrupt Enable Bit
0	Disables transmission bus idle interrupt
1	Enables transmission bus idle interrupt

[bit3] ORE: Overrun error flag bit

• If an overrun error occurs while reception is in progress, this bit is 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 "1", a reception interrupt request is output.

• If this flag is set, the reception data register (RDR) is invalid.

• If this flag is set while the reception FIFO is in use, no reception data is stored in the reception FIFO.

ORE	Overrun Error Flag Bit
0	No overrun error
1	Overrun error exists

[bit2] RDRF: Reception data full flag bit

• The flag indicates the state of the reception data register (RDR).

• When the SMR:RIE bit and reception data flag bit (RDRF) are "1", a reception interrupt request is output.

• When reception data is loaded in the reception data register (RDR), this flag is set to "1". When RDR is read out, the flag is cleared to "0".

• This flag is set at the falling time of SCL at the 8th bit of data.

• This flag is also set in a NACK response (*).

• While the reception FIFO is in use, RDRF is set to "1" once the reception FIFO has received the specified number of data sets.

• While the reception FIFO is in use, this bit is cleared to "0" once the reception FIFO becomes empty.

• If all of the following conditions are met when reception idle continues for 8 baud rate clocks or more, the interrupt flag (SSR:RDRF) is set to "1".

  • The reception FIFO idle detection enable bit (FCR:FRIIE) is "1".

  • The data count of data contained in the reception FIFO does not reach the transfer count.

  • The IBCR:BER bit is "0".

If you read RDR while the counter is counting 8 baud rate clocks, the counter is reset to 0 and starts counting 8 clocks again.

*: NACK response: Indicates that SDA of the I^2C bus is "H" in the acknowledge interval.

RDRF	Reception Data Full Flag Bit
0	Reception data register (RDR) is empty
1	Reception data register (RDR) contains data

(Notes) If all of the following conditions are met, SCL becomes "L" after ACK transmission, and the SCL state of "L" is canceled when the RDRF bit becomes "0":

• Reception FIFO not used
• DMA mode enabled (IBCR:DMA="1")
• RDRF bit is "1" when reception is in progress at the 2nd or subsequent byte (IBSR:TRX="0")
• IBCR:WSEL="0"

If all of the following conditions are met, SCL becomes "L" after 1-byte data reception, and the SCL state of "L" is canceled when the RDRF bit becomes "0":

• Reception FIFO not used
• DMA mode enabled (IBCR:DMA="1")
• RDRF bit is "1" when reception is in progress at the 2nd or subsequent byte (IBSR:TRX="0")
• IBCR:WSEL="1"

In reception with DMA mode enabled (DMA="1") and the reception FIFO used, SCL becomes "L" when the reception FIFO becomes full, and the SCL state of "L" is canceled when data is read out from RDR even once.

[bit1] TDRE: Transmission data empty flag bit

• This flag indicates the state of the transmission data register (TDR).

• When the SMR:TIE bit and TDRE bit are "1", a transmission interrupt request is output.

• When transmission data is written to TDR, this flag becomes "0", which indicates that valid data exists in TDR. Once transmission starts after the data has been loaded to the transmit shift register, the bit is set to "1", which indicates that TDR does not contain any valid data.

• This bit is set when "1" is written to the TSET bit of the serial status register (SSR). Use this flag to set "1" in the TDRE bit when any arbitration lost or bus error is detected.

TDRE	Transmission Data Empty Flag Bit
0	Transmission data register (TDR) contains data
1	Transmission data register (TDR) is empty

[bit0] TBI: Transmission bus idle flag bit

This bit indicates that I^2C has no transmission in progress when DMA mode is enabled (DMA="1").

With DMA mode enabled (DMA="1"), SCL becomes "L" when the TBI bit becomes "1" at the 2nd or subsequent byte, and the SCL state of "L" is canceled when the TBI bit becomes "0".

TBI bit set conditions:

<8th bit>

1. WSEL is "1", master operation is in progress, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.
2. WSEL is "1", transmission is in progress in slave mode, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.

<9th bit>

1. Master operation is in progress, no reserved address has been detected at the 1st byte, and the SSR:TDRE bit is "1".
2. WSEL is "0", master operation is in progress, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.
3. WSEL is "0", transmission is in progress in slave mode, and the SSR:TDRE bit is "1" at the 2nd or subsequent byte.

<Other>

The transmission buffer empty flag set bit (TSET) is set to "1".

TBI bit reset conditions:

Transmission data is written in the transmission data register (TDR).

When this bit is "1" and transmission bus idle interrupt is enabled (SCR:TBIE="1"), a transmission interrupt request is output.

• When DMA mode is disabled (DMA="0"), this bit becomes indefinite.

TBI	Transmission Bus Idle Flag Bit
0	Transmission in progress
1	No transmission operation

35.4.5.3 I²C Bus Status Register: IBSR

I²C bus status register (IBSR) indicates that repeat starts, acknowledges, data directions, arbitration lost, stop conditions, I²C bus states, and bus errors have been detected.

- **IBSR: Address 1503_H, $154B_H$, $156F_H$ (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

[bit7] FBT: First byte bit

This bit indicates the 1st byte.

FBT bit set condition:

(1) A (repeated) start condition is detected.

FBT bit clear conditions:

(1) The 2nd byte is transmitted/received.

(2) A stop condition is detected.

(3) The I²C interface is disabled (ISMK:EN="0").

(4) A bus error is detected (IBCR:BER="1")

FBT	First Byte Bit
0	Other than 1st byte
1	Transmission/Reception of 1st byte in progress

[bit6] RACK: Acknowledge flag bit

This bit indicates acknowledge received at the 1st byte in master mode or slave mode.

RACK bit update conditions:

(1) Acknowledge at the 1st byte

(2) Acknowledge of data in master mode or slave mode

RACK bit clear conditions (RACK bit="0"):

(1) A (repeated) start condition is detected.

(2) The I²C interface is disabled (ISMK:EN="0").

(3) A bus error is detected (IBCR:BER="1").

RACK	Acknowledge Flag Bit
0	"L" reception
1	"H" reception

[bit5] RSA: Reserved address detection bit

This bit indicates that a reserved address has been detected.

RSA bit set condition (RSA="1"):

(1) The 1st byte is "0000xxxx" or "1111xxxx". "x" represents "0" or "1".

RSA bit reset conditions (RSA="0"):

(1) A (repeated) start condition is detected.

(2) A stop condition is detected.

(3) The I^2C interface is disabled (ISMK:EN="0").

(4) A bus error is detected (IBCR:BER="1").

When the RSA bit is "1" at the 1st byte, the interrupt flag (IBCR:INT) becomes "1" and SCL becomes "L" at the fall of SCL at the 8th bit of the 1st byte regardless of the FIFO enable/disable state. To read reception data and operate as a slave at this time, set IBCR:ACKE to "1" and clear the interrupt flag (IBCR:INT) to "0". After this, if the TRX bit is "0", the device receives data as a slave. To disable data reception in process, set "0" in the IBCR:ACKE bit. Subsequently, data is not received.

RSA	Reserved Address Detection Bit
0	No reserved address detected
1	Reserved address detected

(Notes)
- When IBCR:ACKE becomes "0" while data transfer is in progress, setting IBCR:ACKE to "1" is prohibited until a stop condition or repeated start condition is detected.
- Any check of slave transmission at the reserved address detection interrupt time when the reception FIFO is enabled causes an ACK response, so disable the reception FIFO and set IBCR:ACKE to "0".

[bit4] TRX: Data direction bit

This bit indicates the direction of data.

TRX bit set conditions:

(1) A (repeated) start condition is transmitted in master mode.

(2) The 8th bit of the 1st byte is "1" in slave mode (transmission direction for a slave).

TRX bit reset conditions:

(1) Arbitration lost occurs (AL="1").

(2) The 8th bit of the 1st byte is "0" in slave mode (reception direction for a slave).

(3) The 8th bit of the 1st byte is "1" in master mode (reception direction for a master).

(4) A stop condition is detected.

(5) A (repeated) start condition is detected in a mode other than master mode.

(6) The I^2C interface is disabled (ISMK:EN="0").

(7) A bus error is detected (IBCR:BER="1").

TRX	Data Direction Bit
0	Reception direction
1	Transmission direction

[bit3] AL: Arbitration lost bit

This bit indicates arbitration lost.

AL bit set conditions:

(1) The data output in master mode is different from reception data.

(2) Although "1" is set in the IBCR:MSS bit, operation is still in slave mode.

(3) In master mode, a repeated start condition is detected at the 1st bit of the 2nd or subsequent byte of data .

(4) In master mode, a stop condition is detected at the 1st bit of the 2nd or subsequent byte of data.

(5) In master mode, an attempt to generate a repeated start condition fails.

(6) In master mode, an attempt to generate a stop condition fails.

AL bit reset conditions:

(1) "1" is written to the IBCR:MSS bit.

(2) "0" is written to the IBCR:INT bit.

(3) "0" is written to the SPC bit when AL="1" and SPC="1".

(4) The I^2C interface is disabled (ISMK:EN="0").

(5) A bus error is detected (IBCR:BER="1").

AL	Arbitration Lost Bit
0	No arbitration lost occurred
1	Arbitration lost occurred

[bit2] RSC: Repeated start condition check bit

This bit indicates that a repeated start condition has been detected in master mode or slave mode.

RSC bit set condition:

(1) A repeated start condition is detected after acknowledgement in master mode or slave mode.

RSC bit reset conditions:

(1) "0" is written to the RSC bit.

(2) "1" is written to the IBCR:MSS bit.

(3) The I^2C interface is disabled (ISMK:EN="0").

Writing "1" to this bit is invalid.

RSC	Repeated Start Condition Check Bit
0	No repeated start condition detected
1	Repeated start condition detected

(Notes)
- If no acknowledge response is made while reception is in progress in slave mode due to the detection of a reserved address, slave mode is terminated. As a result, the next time that a repeated start condition is detected, "1" is not set in this bit.
- For read-modify-write instructions, "1" is read.

[bit1] SPC: Stop condition check bit

This bit indicates that a stop condition has been detected in master mode or slave mode.

SPC bit set conditions:

(1) A stop condition is detected in master mode or slave mode.

(2) Arbitration lost occurs in stop condition generation in master mode.

SPC bit reset conditions:

(1) "0" is written to this bit.

(2) "1" is written to the IBCR:MSS bit.

(3) The I^2C interface is disabled (ISMK:EN="0").

Writing "1" to this bit is invalid.

SPC	Stop Condition Check Bit	
0	No stop condition detected	
1	Master	Stop condition detected or arbitration lost occurred during stop condition output
	Slave	Stop condition detected

(Notes)
- If no acknowledge response is made while reception is in progress in slave mode due to the detection of a reserved address, slave mode is terminated. As a result, the next time that a stop condition is detected, "1" is not set in this bit.
- For read-modify-write instructions, "1" is read.

[bit0] BB: Bus state bit

This bit indicates the bus state.

BB bit set condition:

(1) "L" is detected at SDA or SCL on the I^2C bus.

BB bit reset conditions:

(1) A stop condition is detected.

(2) The I^2C interface is disabled (ISMK:EN="0").

(3) A bus error is detected (IBCR:IBER="1").

BB	Bus State Bit
0	Bus idle state
1	Bus transmission/reception state

35.4.5.4 Receive Data Register/Transmit Data Register: RDR/TDR

Receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

- **RDR/TDR: Address 1506_H, $154E_H$, 1572_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,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	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

■ Read

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial data line (SOT 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

The transmit data register (TDR) is the data buffer register for sending serial data.

- Output to serial data line (SOT 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".

(Notes) 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.

35.4.5.5 Serial Auxiliary Control Status Register: SACSR

The serial auxiliary control status register (SACSR) allows you to select how to activate the serial timer, enable/disable timer interrupts, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

- **Address 1508_H, 1550_H, 1574_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

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) • 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

Read: The read value is "0".

Write: No effect.

[bit5] Reserved

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.

35.4.5.6 Serial Timer Register: STMR

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

- **STMR: Address $150A_H$, 1552_H, 1576_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".

35.4.5.7 Serial Timer Compare Register: STMCR

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

- **STMCR: Address 150C$_H$, 1554$_H$, 1578$_H$ (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 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".

The following operation is performed at the time interval of (STMCR:TC + 1) x timer operating clock (set by SACSR:TDIV3-0).

- SACSR:TINIT is set to "1".

(Notes)
- When "0000"h is set to this register, the serial timer register will remain set to "0".
- With "0000"h set in this register, the timer interrupt flag (SACSR:TINT) is fixed at "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 the serial timer is disabled (SACSR:TMRE="0").

35.4.5.8 7-bit Slave Address Mask Register: ISMK

7-bit slave address mask register (ISMK) compares and configures bits of slave address.

- **ISMK: Address $151E_H$, 1566_H, $158A_H$ (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 become 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 SOT/SCK 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".

35.4.5.9 7-bit Slave Address Register: ISBA

7-bit slave address register (ISBA) sets slave addresses.

- **ISBA: Address $151F_H$, 1567_H, $158B_H$ (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 ISMK:EN bit is "0".

35.4.5.10 Baud rate Generator Register: BGR

Baud rate generator register (BGR) sets the division ratio of serial clock.

- **BGR: Address $151C_H$, 1564_H, 1588_H (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	BGR[14:8]							
-	0	0	0	0	0	0	0	Initial value
RX,W0	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] Reserved

Always set this bit to "0".

[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.
- Set the baud rate generator register when the ISMK:EN bit is "0".
- Configure baud rate regardless of the master mode or slave mode.
- 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.

35.5 Operation of UART

35.5.1 Interrupt of UART

There are transmission/reception interrupts and status interrupts 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
- Matching values of the serial timer compare register (STMCR) and the serial timer register (STMR)

35.5.1.1 List of Interrupt of UART

Table 35.5-1 indicates how UART interrupt control bits relate to interrupt factors.

Table 35.5-1 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	❍	❍	Flaming error		
	PE	SSR	❍	×	Parity error		
Transmission	TDRE	SSR	❍	❍	Transmission register is empty	SCR:TIE	Writing to the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*1
	TBI	SSR	❍	❍	No transmission operation	SCR:TBIE	Writing the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*1
	FDRQ	FCR1	❍	❍	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or the transmission FIFO is full
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".

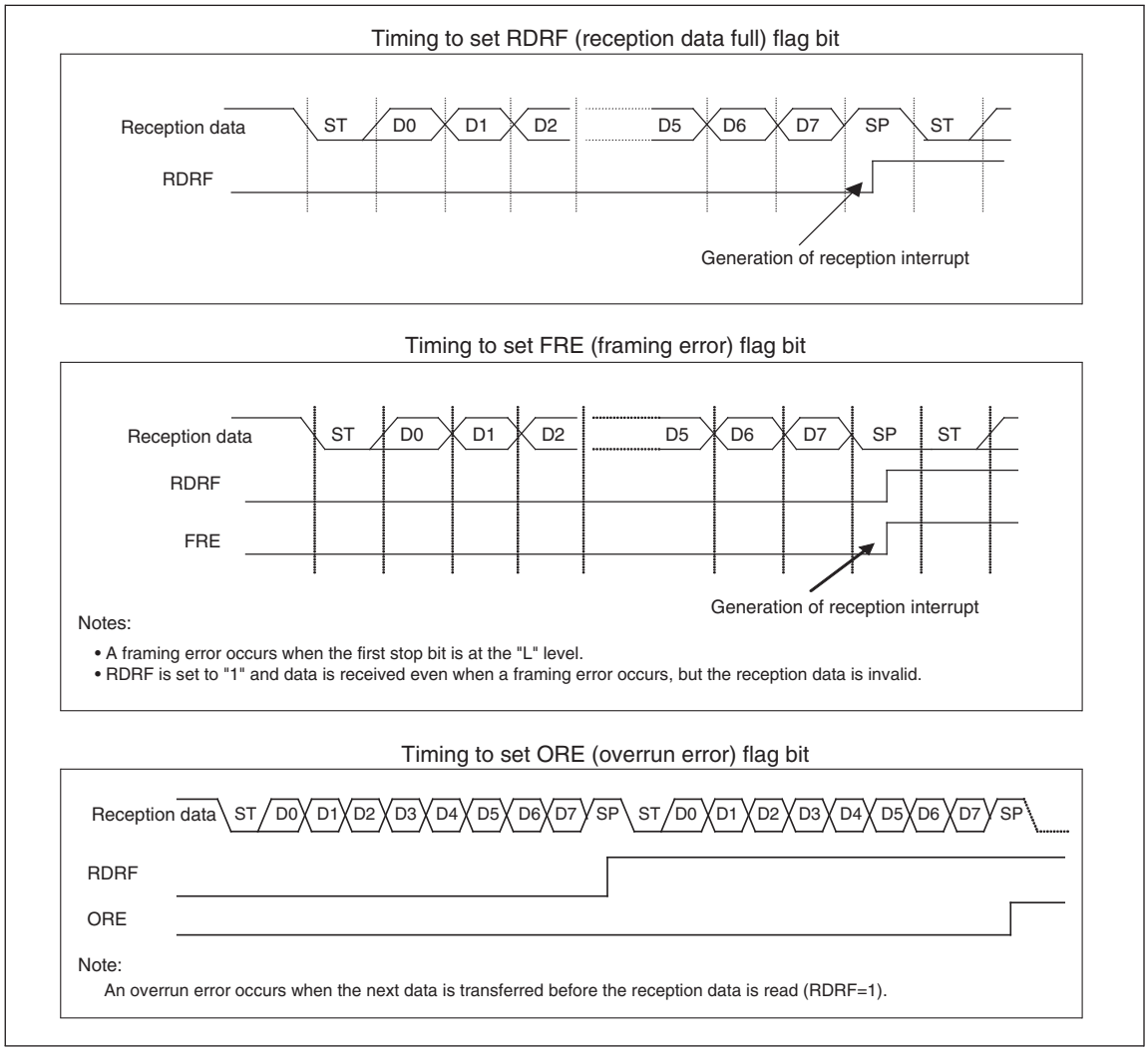
35.5.1.2 Reception Interrupts and Flag Setting Timing

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 35.5-1 Timing of Flag Bit Setting



(Notes) 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")

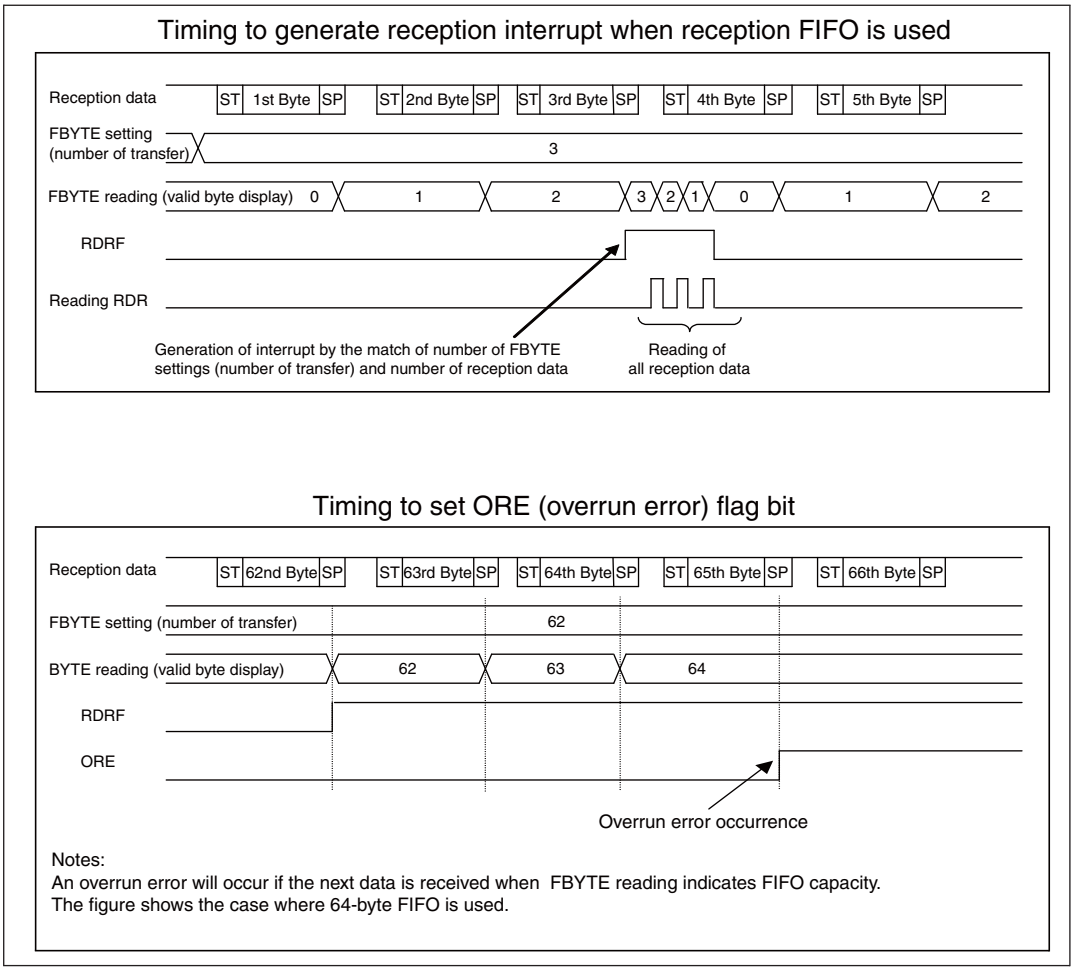
35.5.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

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.

- When 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 35.5-2 Timing of Using FIFO



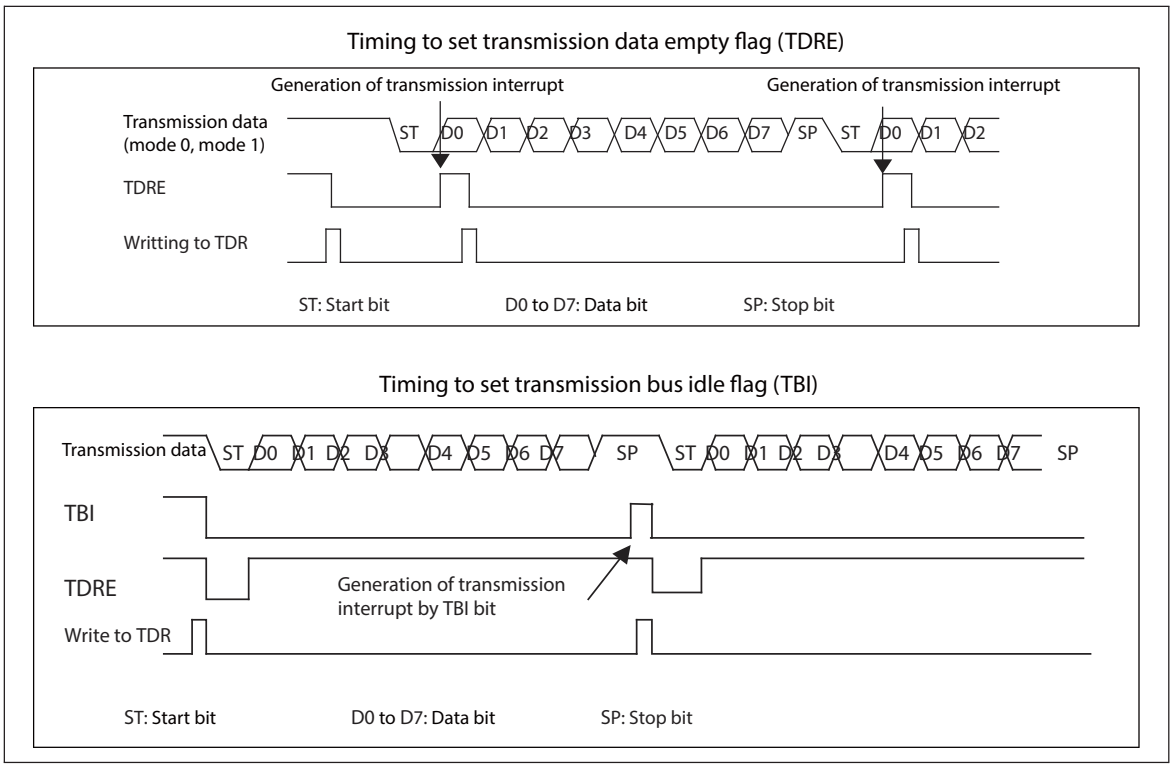
35.5.1.4 Transmission Interrupts and Flag Setting Timing

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 SSR:TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt will occur. When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 35.5-3 Timing of Transmission Interrupt Flag

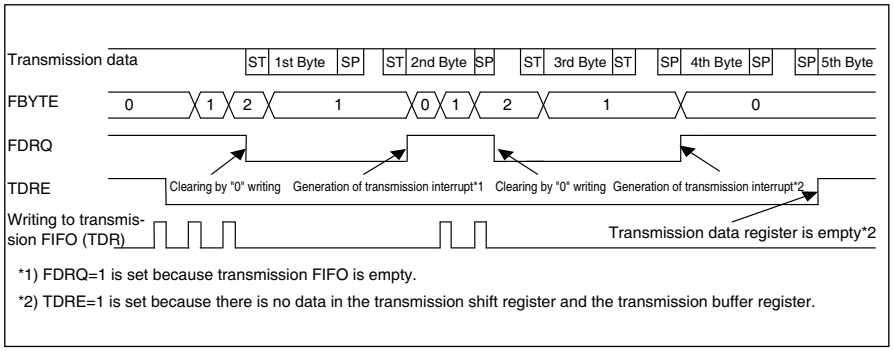


35.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

When the transmission FIFO is used, an interrupt occurs when there is no data in the transmission FIFO.

- When there is no data in the transmission FIFO, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1".
  If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE).
- When FBYTE=0x00, there is no data in the transmission FIFO.

Figure 35.5-4 Timing of Transmission Interrupts When Using Transmission FIFO

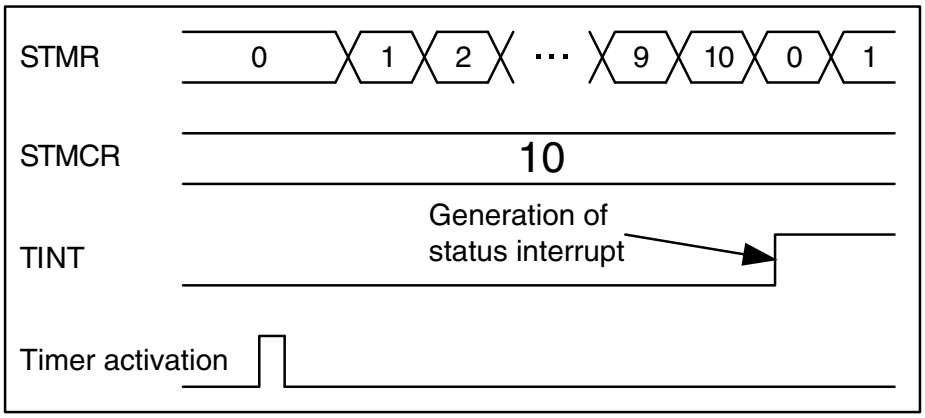


35.5.1.6 Timing of Timer Interrupt Generation and Flag Setting

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 35.5-5 Timer Interrupt Generation Timing



35.5.2 Operation of UART

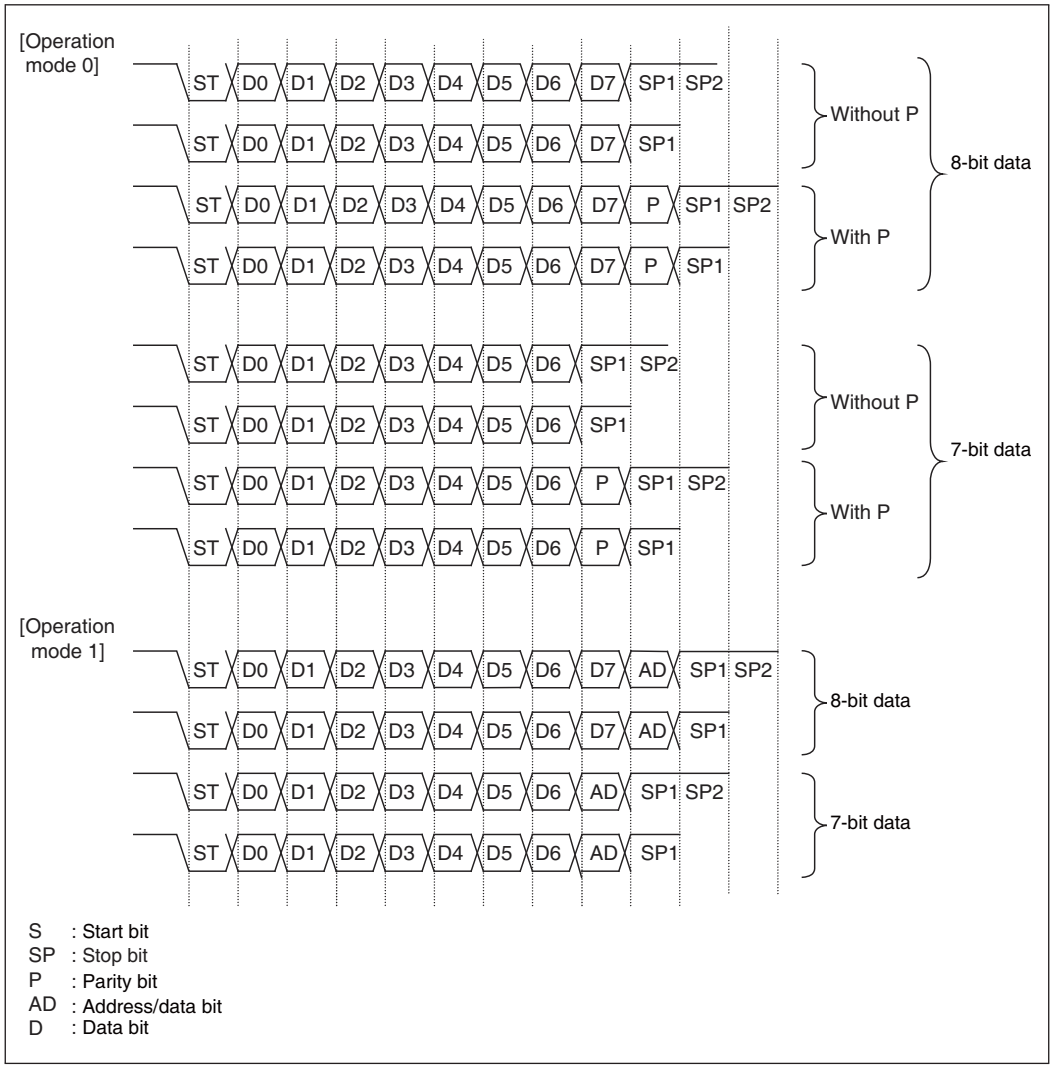
UART operates with the mode 0 bidirectional serial asynchronous communication and the mode 1 master/slave multiprocessor communication.

35.5.2.1 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 an operation with parity, the parity bit will always be placed between the last data bit and the first stop bit.
- In operation mode 0 (normal mode), you can select whether to use parity.
- In operation mode 1 (multiprocessor mode), the parity will not be added, instead AD bits will be added.

An example of transmission/reception data format (operation modes 0, 1) is shown in Figure 35.5-6:

Figure 35.5-6 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.

35.5.2.2 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) is set to "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. Following transmission data can be written to the transmit data register when processing interrupts.

(Notes)

- As soon as the transmission interrupt is enabled (SCR:TIE), a transmission interrupt occurs, because the transmission data empty flag bit (SSR:TDRE) has the initial value "1".
- As soon as the FIFO transmission interrupt is enabled (FCR1:FTIE=1), a transmission interrupt will occur, because the FIFO transmission data request bit (FCR1:FDRQ) has the initial value "1".

35.5.2.3 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.
- When you read a reception data, do it after the one frame data reception has completed, and check for the state of error flag of the serial status register (SSR). When a reception error has detected, correct the error.
- After a read of reception data, the reception data full flag bit (SSR:RDRF) will be cleared to "0".
- When reception FIFO is enabled, if as many frames as set in the reception FBYTE have been received, the reception data full flag bit (SSR:RDRF) will be set to "1".
- 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 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, when the error occurs due to adding the checksum of data at the end, send it again).
- 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")

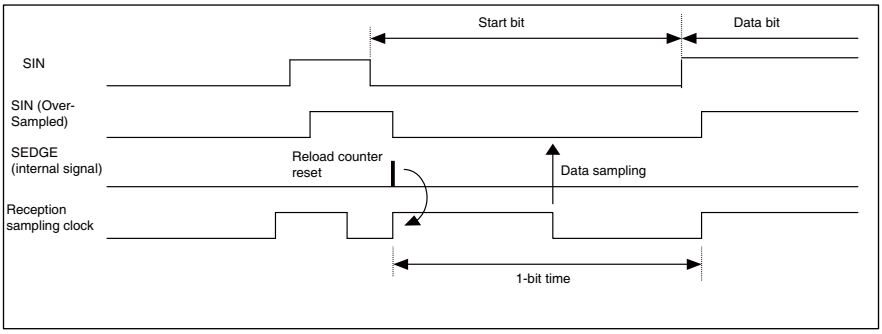
35.5.2.4 Clock Selection

- 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.

35.5.2.5 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 count down will start. This will always launch a data sampling aimed at the center of the data.

Figure 35.5-7 Start Bit Detection



35.5.2.6 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.

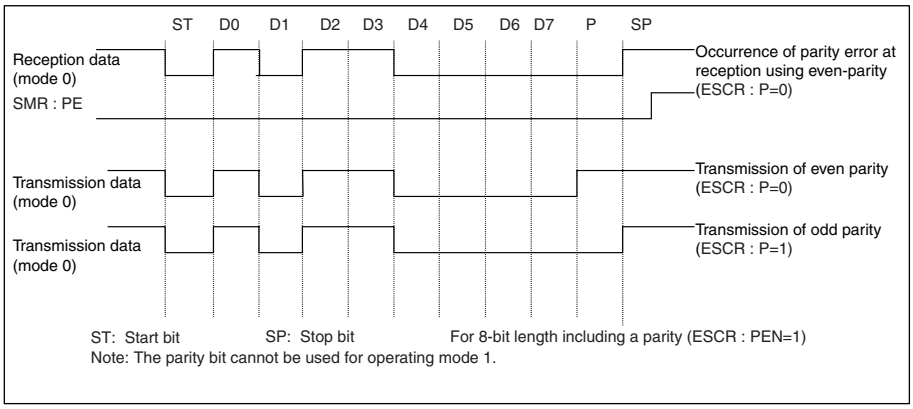
35.5.2.7 Error Detection

- 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.

35.5.2.8 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.

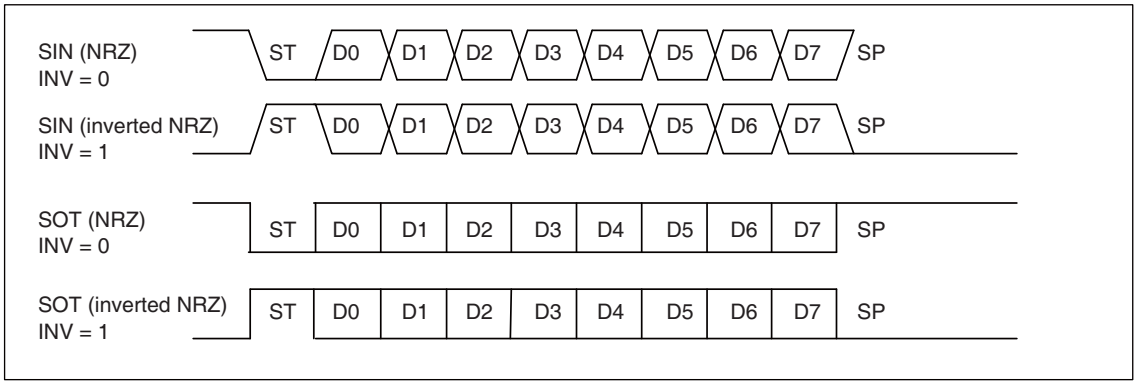
Figure 35.5-8 Operation with Parity Enabled



35.5.2.9 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 35.5-9 shows the NRZ signaling method and inverted NRZ signaling method.

Figure 35.5-9 NRZ (Non Return to Zero) Signaling Method and Inverted NRZ Signaling Method



35.5.2.10 Data Transfer Method

You can select LSB First or MSB First as the data bit transfer method.

35.5.2.11 Operation of Serial Timer

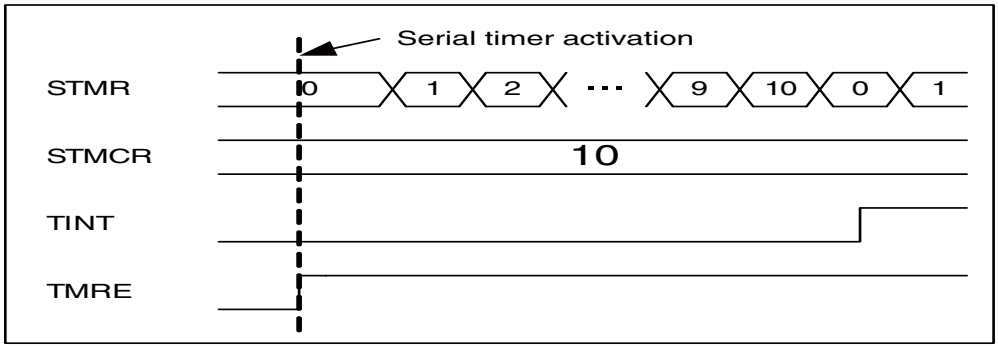
The serial timer can be used for the timer function.

● How to Start Serial Timer

To start the serial timer, set "1" in 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 35.5-10 Start by Using Serial Timer Enable Bit (STMCR="10", SACSR:TRGE="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

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 35.5-11 Timer Operation (STMCR="10")

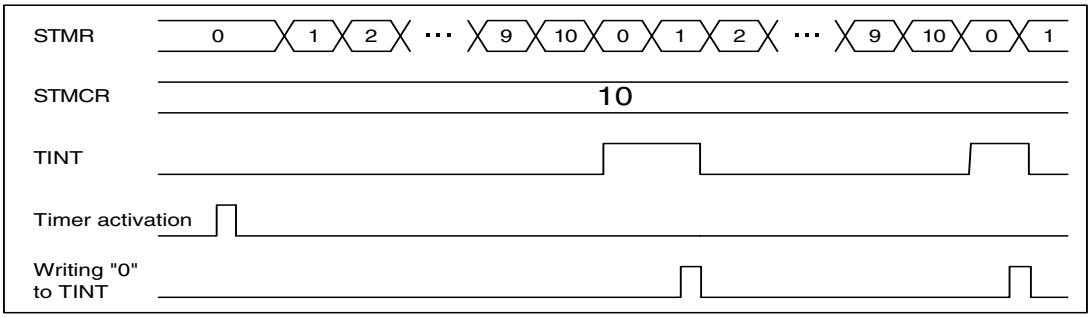


Figure 35.5-12 Flowchart of Serial Timer Initial Setting

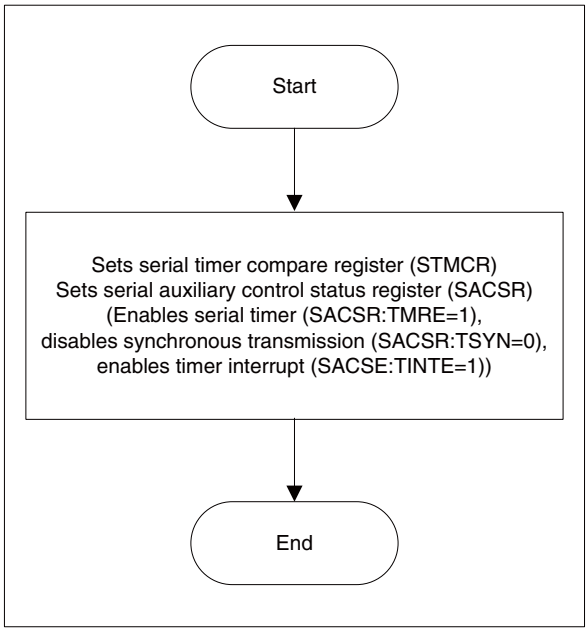
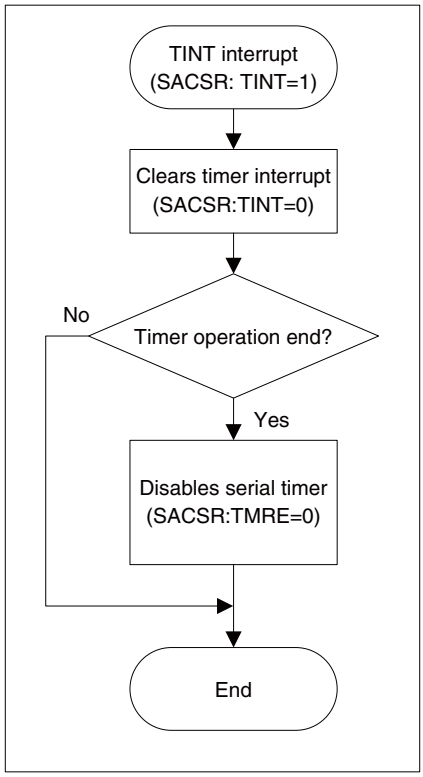


Figure 35.5-13 Flowchart of Serial Timer Interrupt Processing



(Note) With "0000"h set in the timer compare register (STMCR), the timer interrupt flag (SACSR:TINT) is fixed at "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000"b while the timer is running.

35.5.2.12 Test Mode

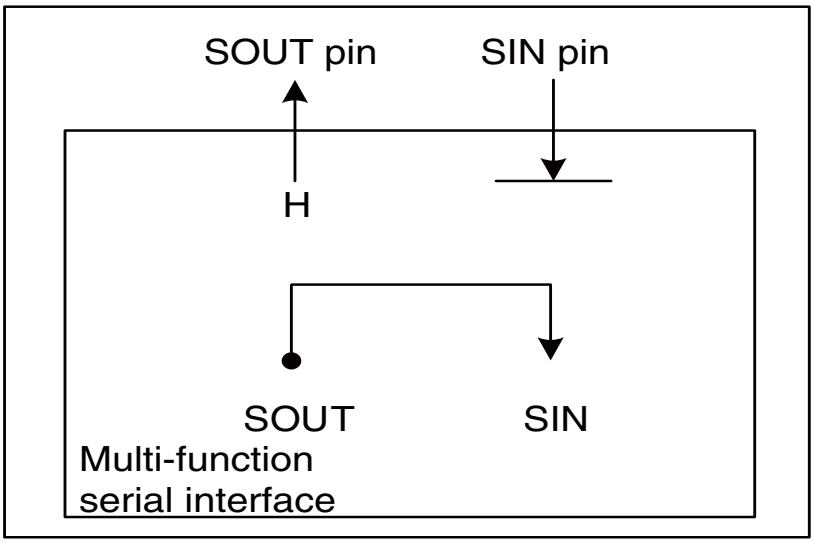
This section explains the operation of the test mode.

● Serial Test Mode

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 35.5-14 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").

35.5.2.13 UART Baud Rate Selection/Setting

You can select either of the following as the UART transmission/reception baud rate generator:

- Dedicated baud rate generator (reload counter)
- External clock input to the baud rate generator (reload counter)

● 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 (BGR:EXT="1") after stopping the reload counter (BGR="0000"h).
- When an external clock (BGR:EXT="1") has been set, the "H" width and "L" width of the external clock should be set to 2 bus clocks or more.

■ Baud Rate Calculation

Set two 15-bit reload counters in the baud rate generator register (BGR).

The baud rate calculation formulas are as follows:

(1) Reload value

$V = \phi / b - 1$

V: Reload value
b: Baud rate
ϕ: Bus clock frequency, external clock frequency

(2) Example of calculation

Reload values when setting the bus clock frequency at 16 MHz, usage of internal clock, and baud rate at 19200 bps are as follows:

Reload value:
$V = (16 \times 1{,}000{,}000) / 19200 - 1 = 832$
The baud rate is:
$b = (16 \times 1{,}000{,}000) / (832+1) = 19208$ bps

(3) Baud rate error

The baud rate error can be obtained using the following formula:

Error (%) = (calculated value - desired value) / desired value × 100

(Example) Bus clock 20MHz, Target baud rate value 153600 bps

Reload value = (20 × 1000000) / 153600 - 1 = 129
Baud rate (calculated value) = (20 × 1000000) / (129 + 1) = 153846 bps
Error (%) = (153846 - 153600)/ 153600 × 100 = 0.16(%)

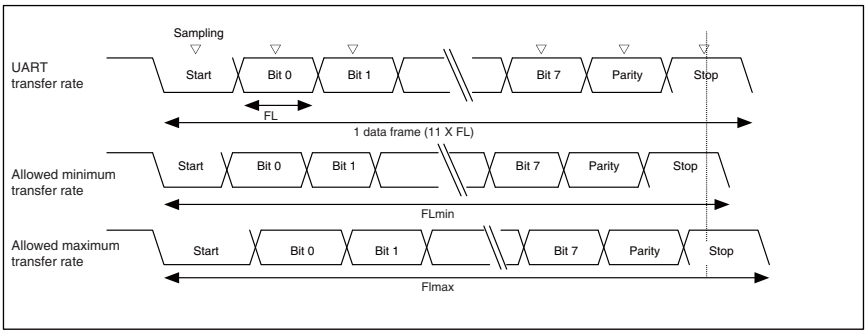
(Notes)
- Set the reload value to "0" to stop the reload counter.
- If the reload value is an even number, the "L" width of the reception serial clock is 1 bus clock longer than "H" width. If it is an odd number, the "H" and "L" widths of the serial clock are equal.
- Set the reload value to 4 or higher. A normal data reception operation, however, could not be achieved for some baud rate error and reload value settings.

■ Allowed Baud Rate Error Range at Reception

This section explains the amount of the destination baud rate error that can be allowed at reception.

The baud rate error at reception should be set within the allowed error range by using following formula.

Figure 35.5-15 Allowed Baud Rate Range at Reception



As shown in the figure, the counter set by the BGR register will determine the sampling timing of the reception data after having detected a start bit. A normal reception operation can be achieved if the last data (stop bit) have been completed on time at this sampling timing.

In theory, the following is expected when this is applied to 11-bit reception.

If the margin of sampling timing is 1 clock of 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 the allowance and maximum transfer rate (FLmax) receives the data, sampling is done in the starting point of receive data in the 11th bit.

Therefore, the allowance and maximum transfer rate (FLmax) is as follows.

$10/11 \times FLmax = (11bit \times (V+1) - (V+1) / 2) / \phi = (21/20 \times 11 \times (V+1) / 2) / \phi$ (s)
V: Reload value ϕ: Bus clock

When margin (ϕ) of the sampling timing is made two clocks, the allowance and maximum transfer rate (FLmax) is as follows:

$FLmax = (21/20 \times 11 \times (V+1) - 2) / \phi = (231V+191) / 20\phi$ (s)
V: Reload value ϕ: Bus clock

Therefore, the allowed minimum baud rate (BGmin) at the destination would be calculated as follows.

$BGmin = 11 / FLmax = 220\phi / (231V+191)$ (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 35.5-2 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 35.5-3 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. Therefore, if the external clock cannot be synchronized, operation is 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

35.5.3 Setup Procedure and Program Flow

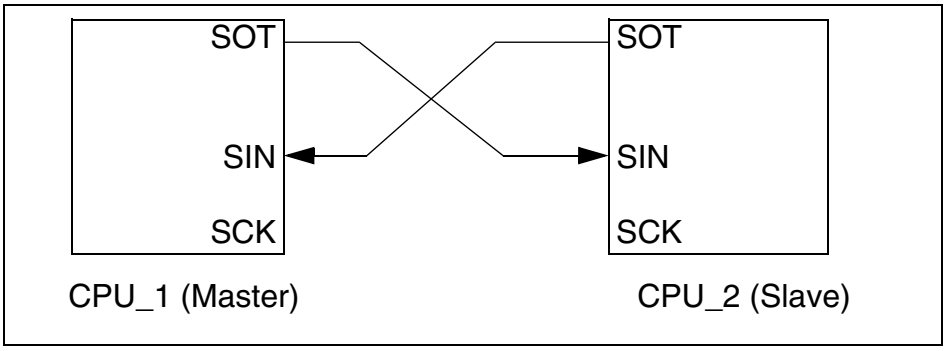
35.5.3.1 Operation Mode 0 (One-to-One Connection)

In operation mode 0, asynchronous serial bidirectional communications can be performed.

■ CPU Interconnection

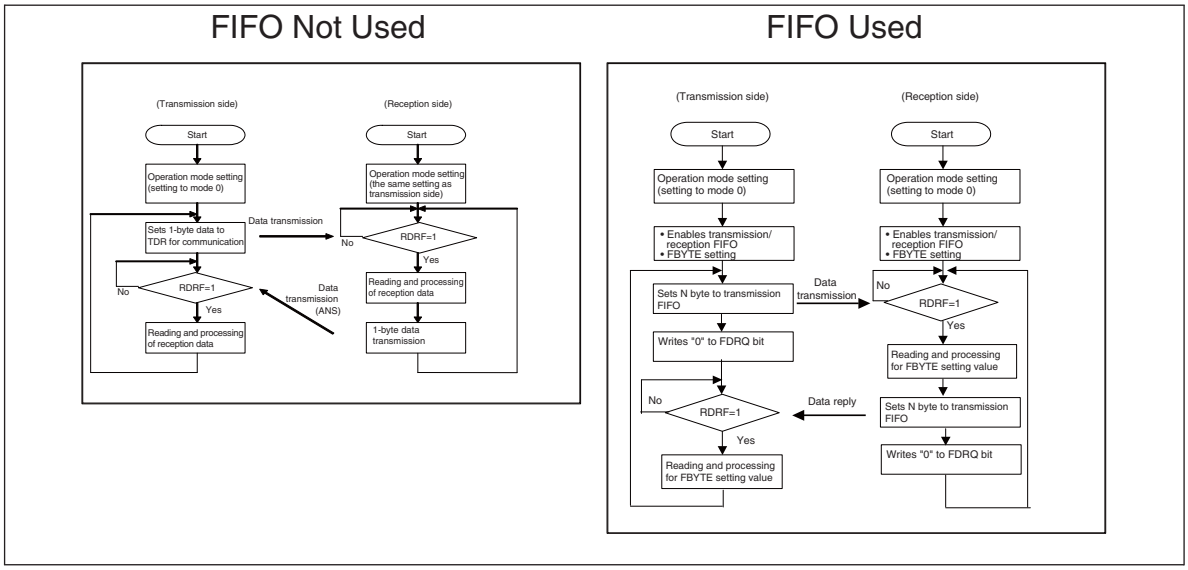
In operation mode 0 (normal mode), select bidirectional communications. Two CPUs are inter-connected as shown in Figure 35.5-16:

Figure 35.5-16 Example of Connection for Bidirectional Communications in UART Operation Mode 0



■ Flowchart

Figure 35.5-17 Example of Settings for Bidirectional Communications



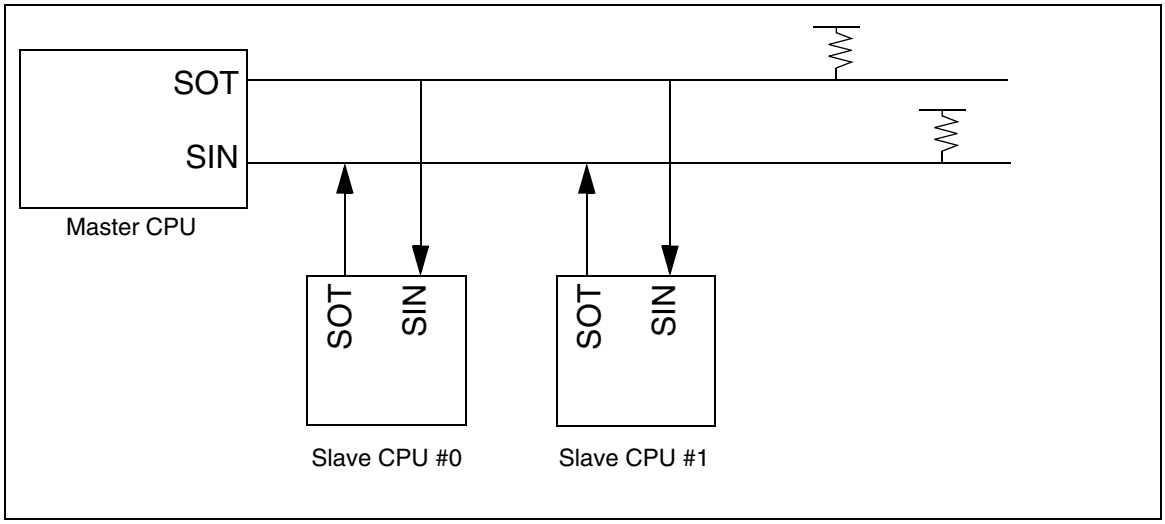
35.5.3.2 Operation Mode 1 (One-to-N Connection)

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 35.5-18 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 35.5-4:

Table 35.5-4 Selection of Master-Slave Communication Function

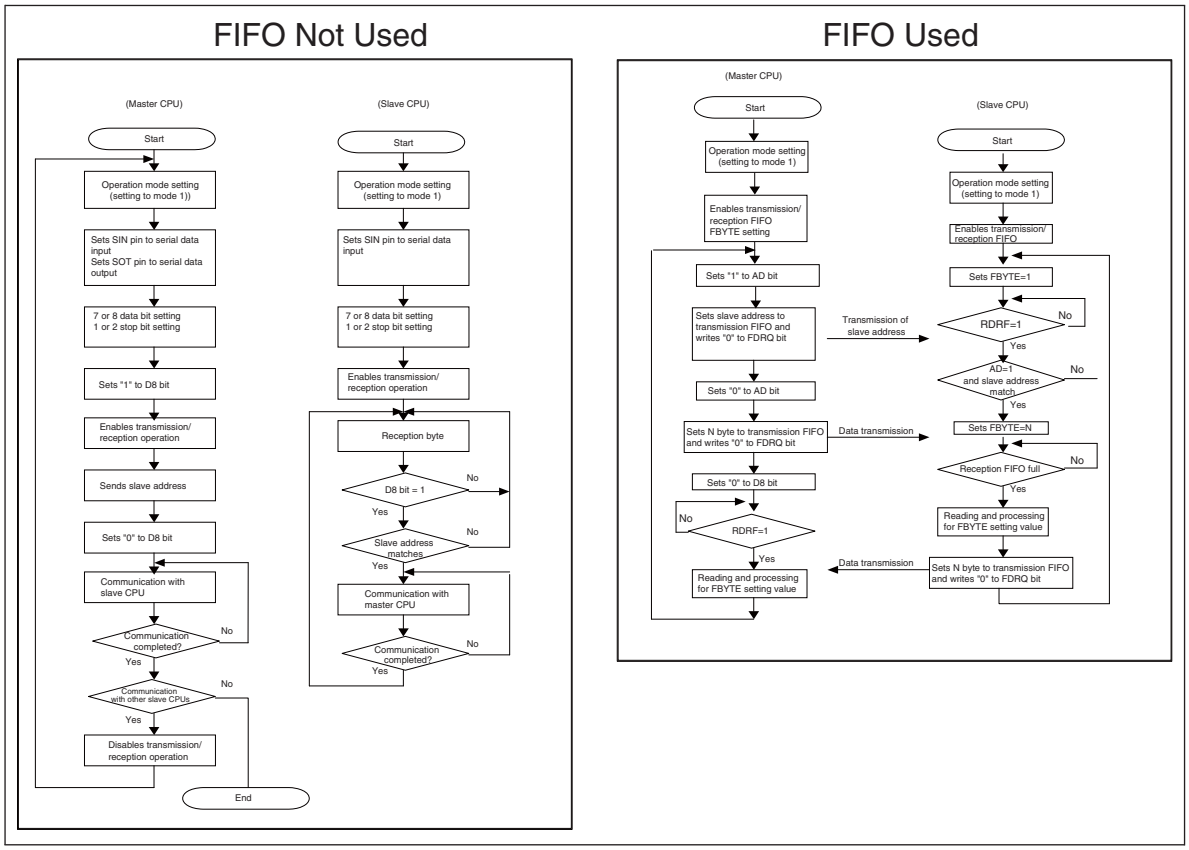
	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission and reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = "1" + 7 or 8-bit Address	None	1 bit or 2 bits	LSB or, MSB First
Data transmission and reception			AD = "0" + 7 or 8-bit Data			

(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 35.5-19 shows a flowchart of master-slave communications (multi-processor mode).

Figure 35.5-19 Example of Flowchart of Master-Slave Communications



35.6 Operation of CSIO

35.6.1 Interrupts of CSIO

The interrupts for the CSIO (clock synchronous serial interface) include reception, transmission, and status 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
- Matching values of the serial timer compare register (STMCR) and the serial timer register (STMR)
- Chip select error occurrence

35.6.1.1 List of Interrupts of CSIO

Table 35.6-1 Interrupt Control Bits and Interrupt Factors of CSIO

Interrupt Type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt source enable bit	Clearing of interrupt request flag
Reception	RDRF	SSR	1-byte reception	SCR: RIE	Reading of receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing of transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	TBI	SSR	No transmission operation	SCR: TBIE	Writing of transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	FDRQ	FCR1	Transmission FIFO is empty	FCR1: FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or transmission FIFO is full
	CSE	SACSR	Serial chip selection terminal is inactive while transmitting in slave mode (SCR:MS="1"). The transmission count in master mode (SCR:MS=0) is the TBYTE setting value or less, and the next transmission data has not been written in TDR (SSR:TDRE=1).	SACSR: SCEIE	Writing "0" to the serial chip selection flag bit (SACSR:CSE)
Status	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to the timer interrupt flag bit (SACSR:TINT)

*: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

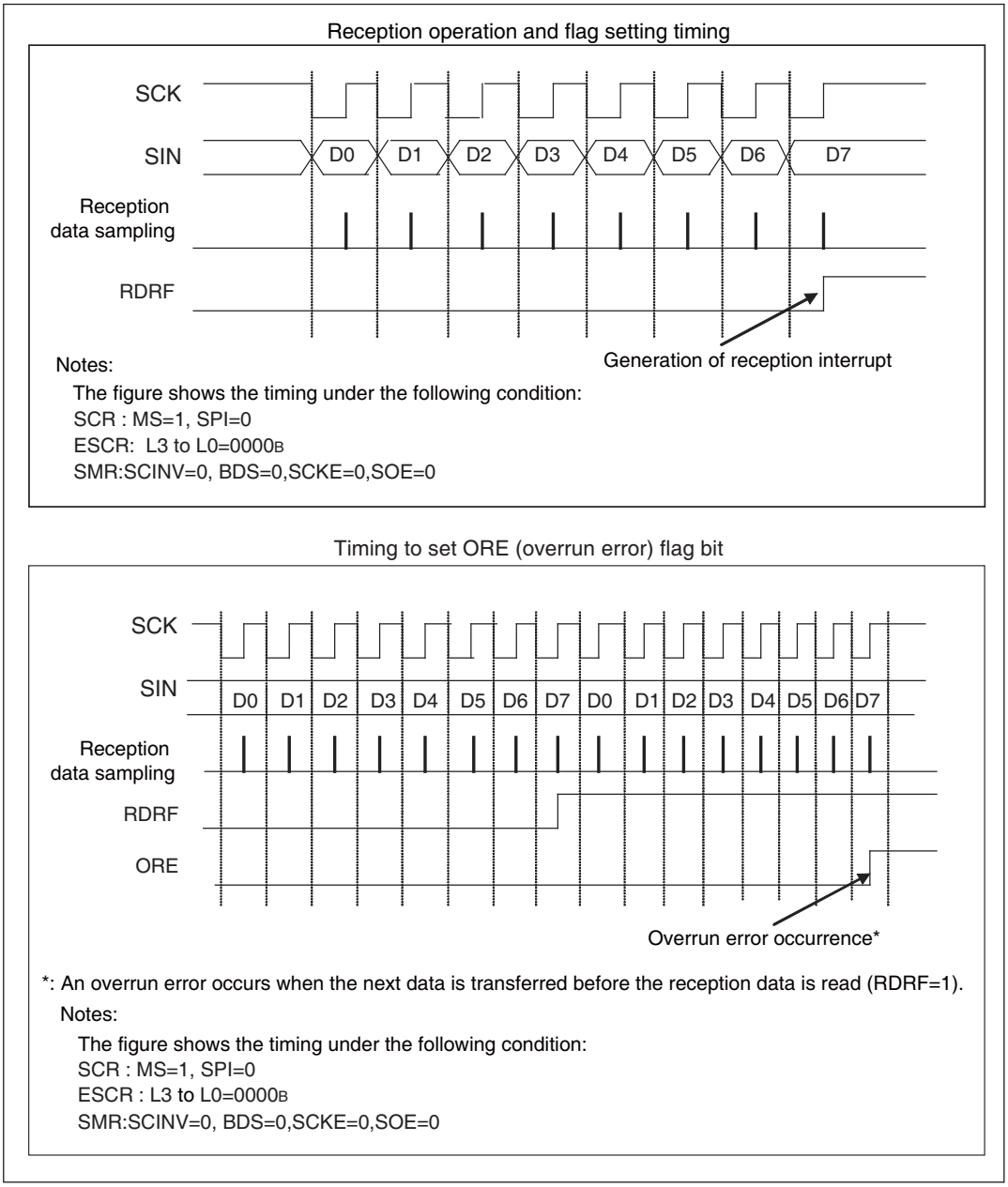
35.6.1.2 Reception Interrupts and Flag Setting Timing

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 35.6-1 Timing of Flag Setting



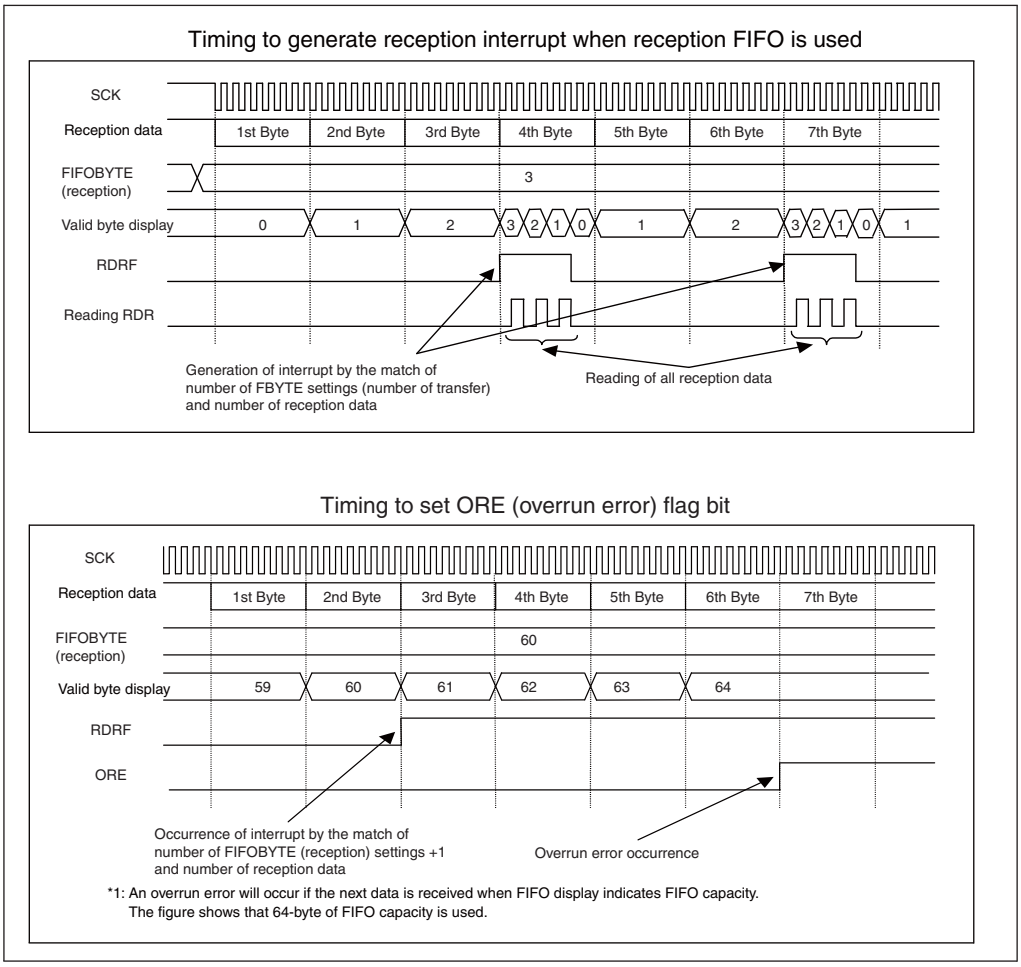
35.6.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

When the reception FIFO is used, an interrupt occurs in cases where 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.

- When 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 35.6-2 Timing of Interrupts and Flag Setting



35.6.1.4 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").

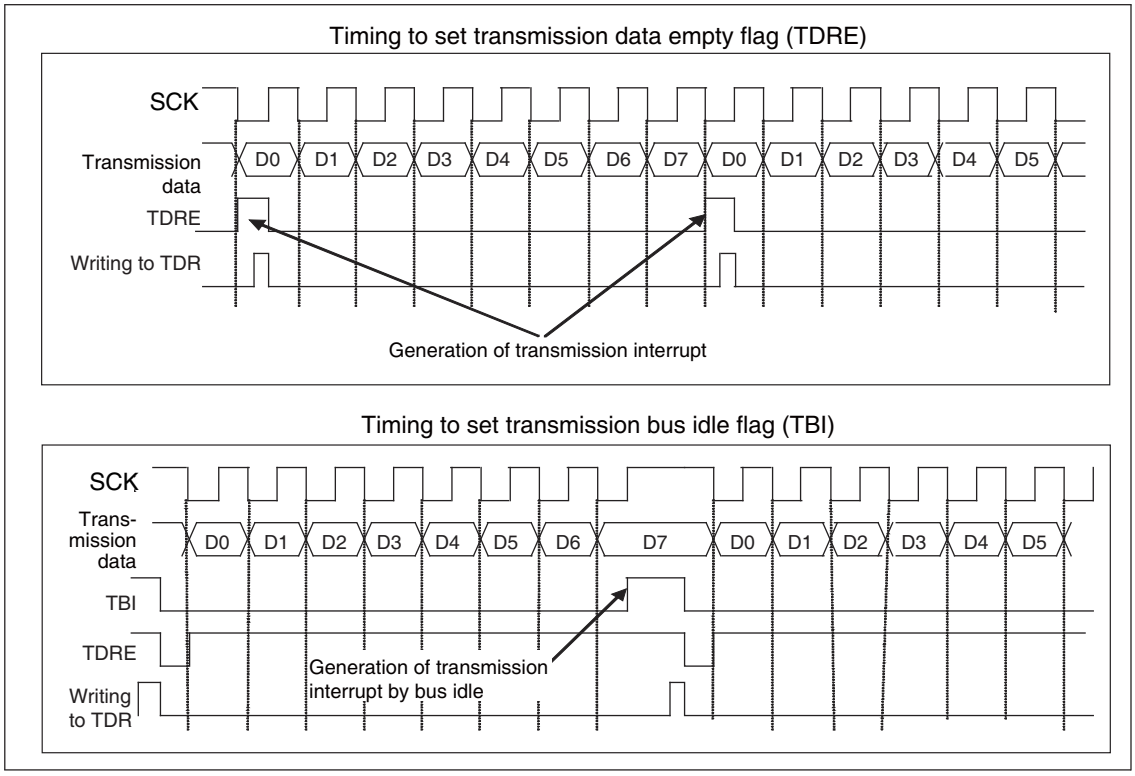
- Timing of transmission data empty flag (TDRE) setting

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 (SSR:TDRE="1") and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE="1") at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 35.6-3 Timing of Flag Setting

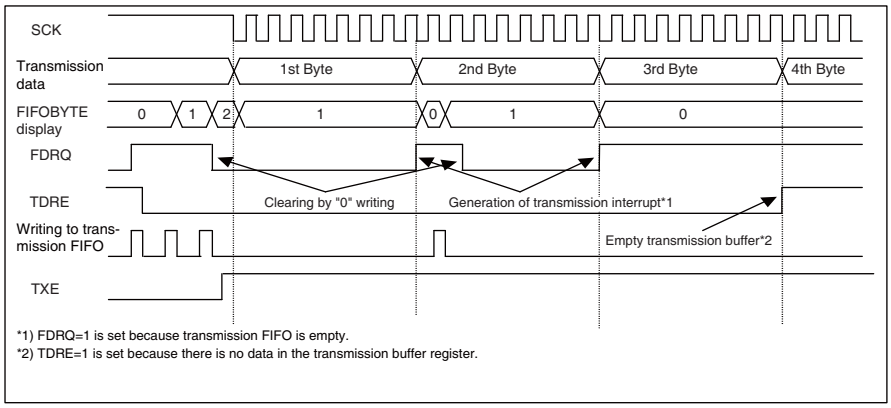


35.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

When the transmission FIFO is used, an interrupt occurs when there is no data in the transmission FIFO.

- When there is no data in the transmission FIFO, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE=1) at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE). When FBYTE=0x00, there is no data in the transmission FIFO.

Figure 35.6-4 Transmission Interrupt Generation Timing When Using the Transmission FIFO

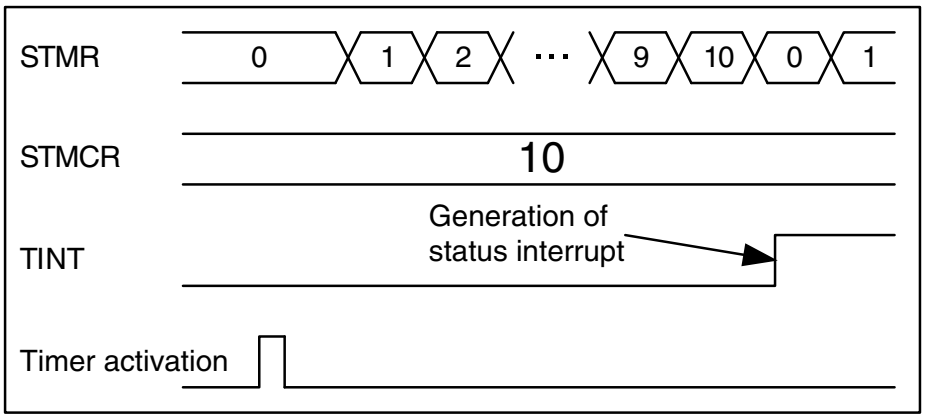


35.6.1.6 Timing of Timer Interrupt and Flag Setting

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 35.6-5 Timer Interrupt Generation Timing



35.6.1.7 Timing of Chip Selection Error Generation and Flag Setting

Chip selection 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 transmission 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 terminal becomes inactive while transmitting in slave mode (SCR:MS="1").

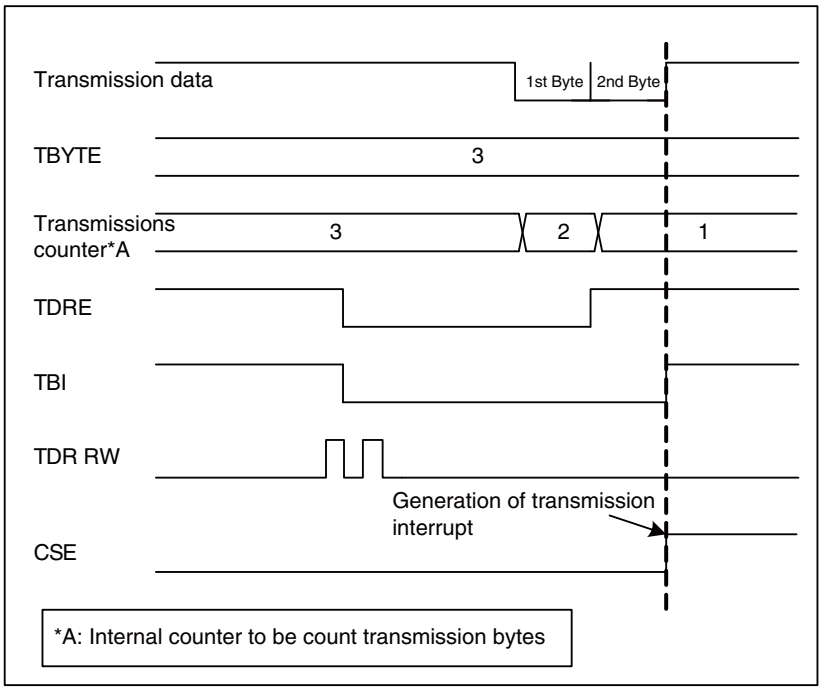
● Master Mode (SCR:MS="0")

Chip selection error will be generated with transfer byte error enabled (SACSR: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 selection error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 35.6-6 Chip Selection Error Generation Timing



(Notes)

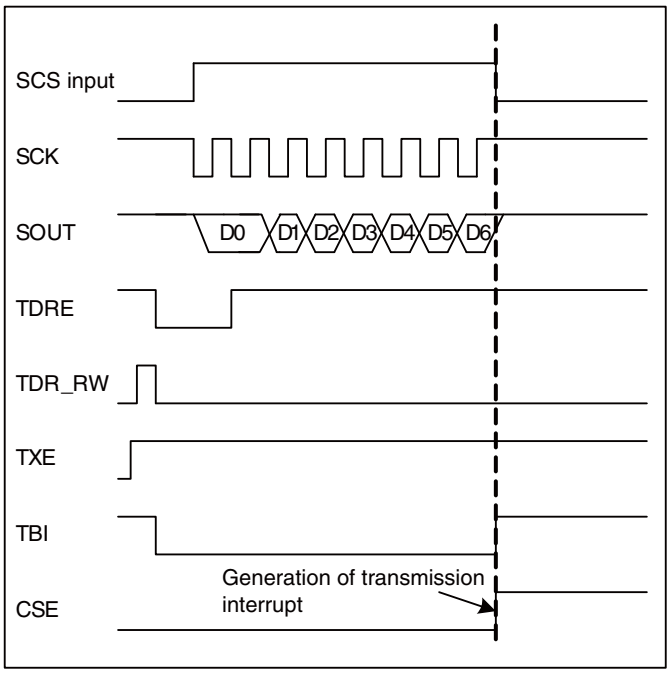
- When serial chip select is used, the chip select error flag (SACSR:CSE) is set to "1" after the deselect time has elapsed from the occurrence of a chip select error. Also, no transmission operation begins even if transmission data is written in the transmission data register (TDR) during the hold delay time, and the chip select error flag (SACSR:CSE) is set to "1" after the deselect time has elapsed.
- When "1" is set in the chip select error flag (SACSR:CSE), no transmission operation begins even if transmission data is written in the transmission data register (TDR).
- When the chip selection 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 selection error will be generated when chip select terminal becomes inactive while transmitting (SSR:TBI="0").

  In this case, when chip selection error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 35.6-7 Chip Selection Error Generation Timing



35.6.2 Operation of CSIO

35.6.2.1 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	5-16, 20, 24, 32 bits

■ Register Settings

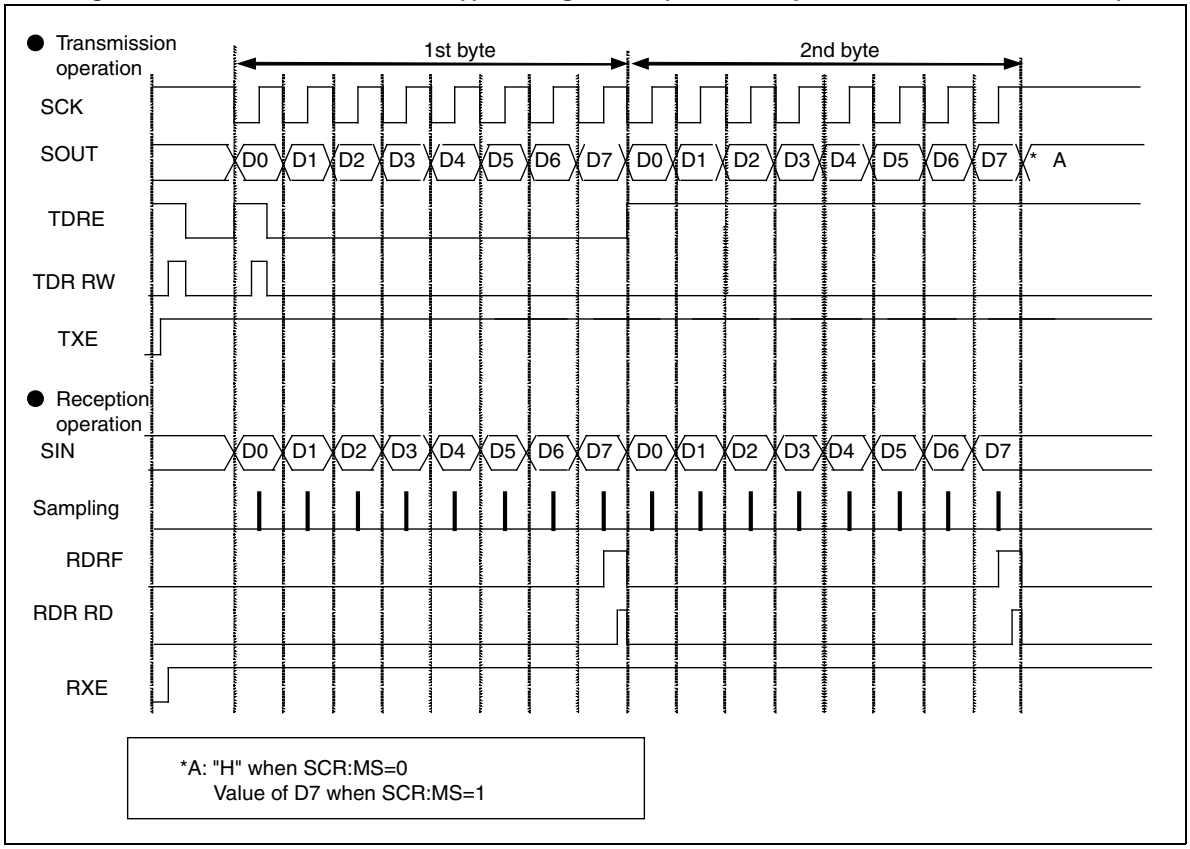
The following table lists the register settings required for normal transfer (I).

SCR:SPI=0*[1], SMR:MD2=0, MD1=1, MD0=0, SCINV=0*
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0
*: Bit settings depend on the condition. See "Table 35.6-3 Format Setting of Serial Chip Selection Pin" for details.

(Note) Use proper usage for setting the registers other than the bits above.

■ Normal Transfer (I) Timing Chart (Serial Chip Select Terminal Unused)

Figure 35.6-8 Normal Transfer (I) Timing Chart (Serial Chip Select Terminal Unused)



[1] Master operation (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 resulted 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").

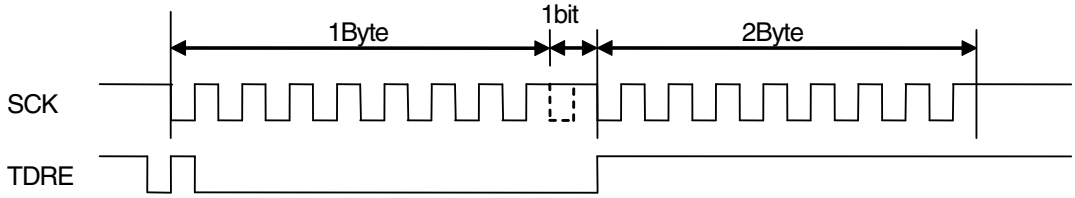
(2) When transmission data is written in TDR, SSR:TDRE="0" is set, and the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE="1" is set, and when transmission interrupt is enabled (SCR:TIE="1"), 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".

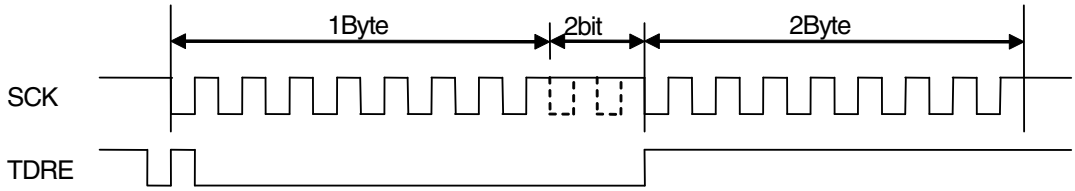
● Successive data transmission or reception wait operation

If setting other than (ESCR:WT1, ESCR:WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

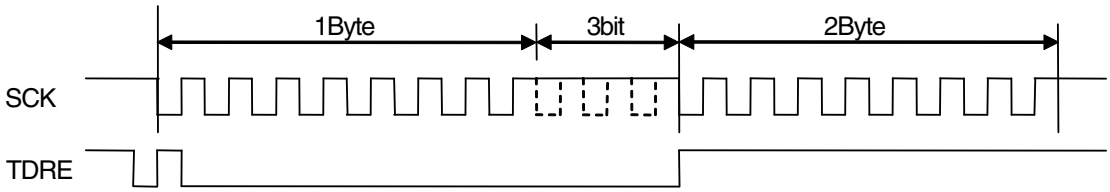
■ ESCR.WT1=0, ESCR.WT0=1 (for master)



■ ESCR.WT1=1, ESCR.WT0=0 (for master)



■ ESCR.WT1=1, ESCR.WT0=1 (for master)



[2] Slave operation (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 resulted in outputting 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

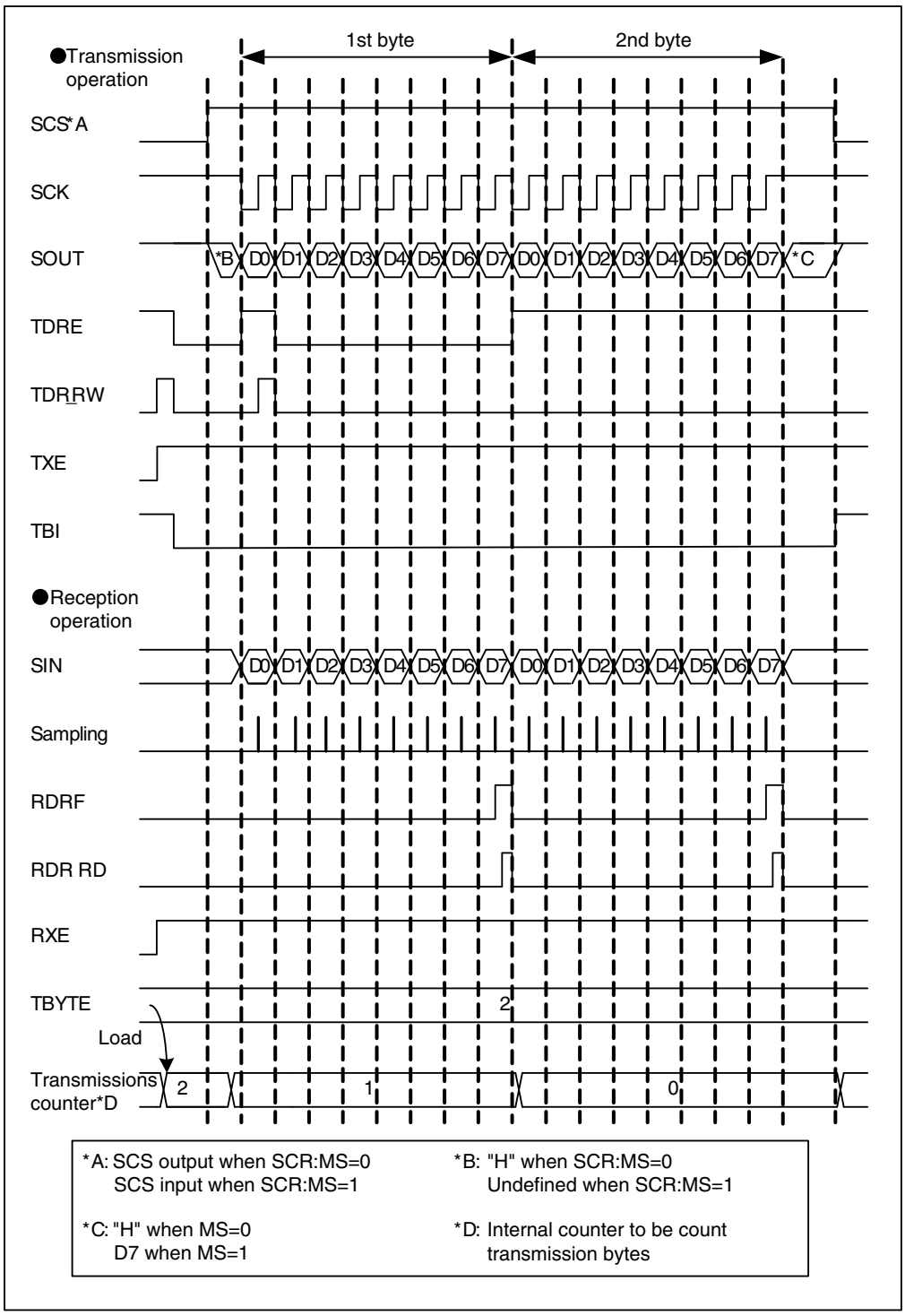
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE="1") and enable transmission/reception operation (SCR:TXE, RXE="1").

(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 Terminal Used)

Figure 35.6-9 Normal Transfer (I) Timing Chart (Serial Chip Select Terminal Used)



[1] Master operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: "n" shows the number of serial chip select terminal 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 terminal (SCS) will become active, transmission will be started after the setup time is lapsed for the serial chip select terminal. The start of the transmission resulted 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 terminal is lapsed after the transmission operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 terminal (SCS) active and reception operation will be started after the setup time is lapsed for the serial chip select terminal. The start of the reception resulted 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 received 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 terminal is lapsed after the reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 frame 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 terminal (SCS) will become active and transmission/reception will be started after the setup time is lapsed for the serial chip select terminal. 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.

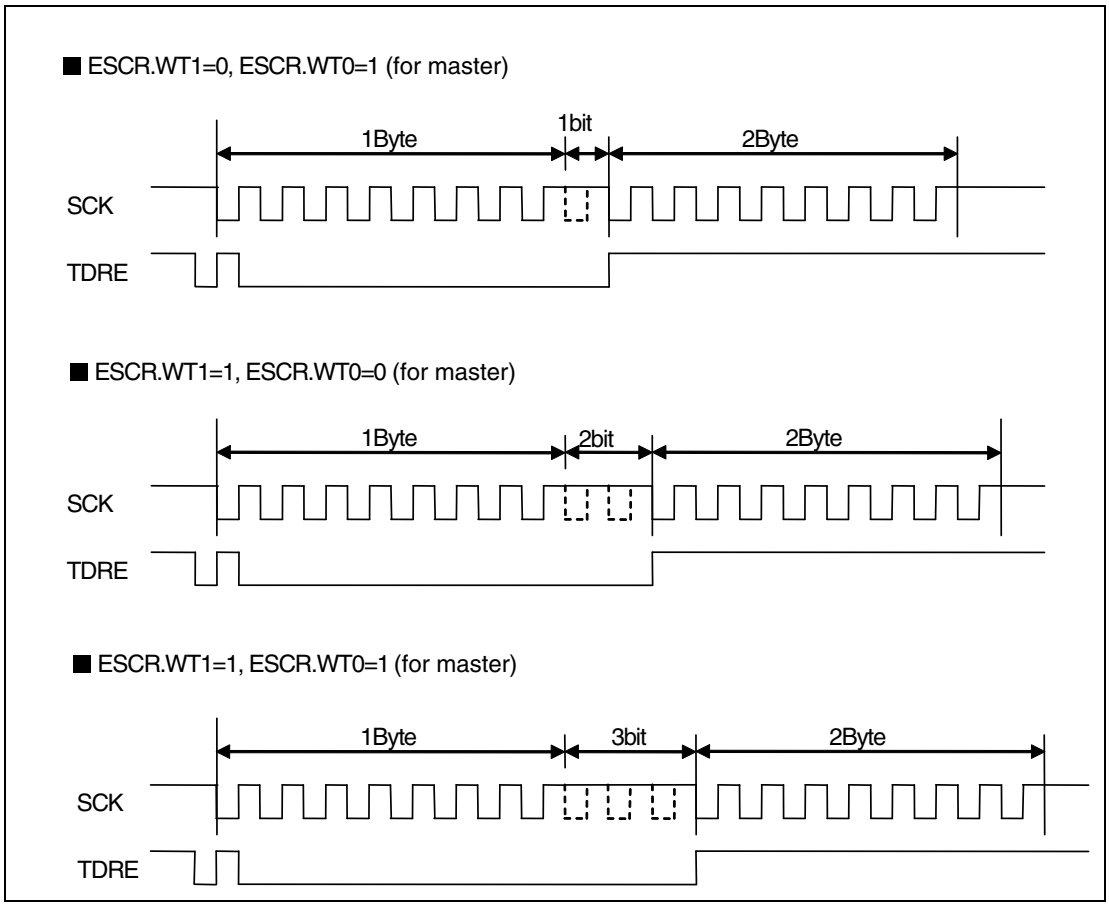
(3) While operating transmission/reception, the reception data will be sampled at a rising edge of the serial clock output (SCK). Receiving the last bit of receiving data sets SSR:RDRF="1" and when the reception interrupt is enabled (SCR:RIE="1"), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the received 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 terminal is lapsed after the transmission/reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (SCS) will remain active.

● Continuous Data Transmission or Reception Wait Operation

When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.



[2] Slave operation (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 terminal (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 terminal (SCS) becomes inactive, and serial output terminal (SOUT) 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 terminal (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.

(3) At this time, the receive data (RDR) can be read.

(4) Reading the received data (RDR) clears SSR:RDRF to "0".

(5) Reception operation will be terminated when serial chip select terminal (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 terminal (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 received data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when serial chip select terminal (SCS) becomes inactive, and serial output terminal (SOUT) becomes "H".

35.6.2.2 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	5-16, 20, 24, 32 bits

■ Register Settings

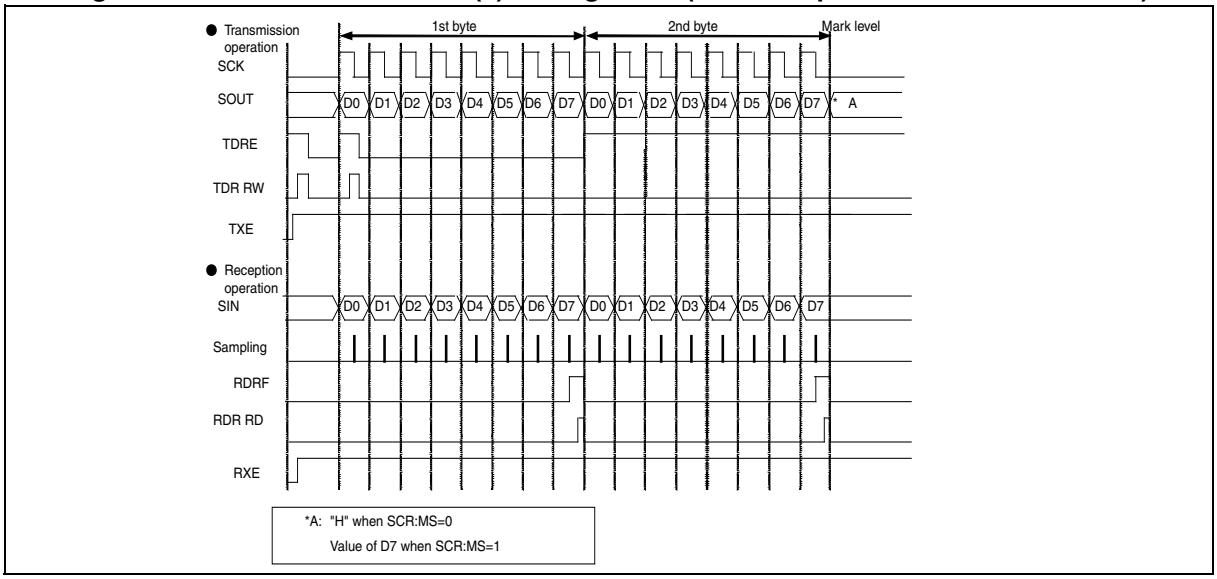
The following table lists the register settings required for normal transfer (II).

SCR:SPI*[1]=0, SMR:MD2=0, MD1=1, MD0=0, SCINV*=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0
*: Bit settings depend on the condition. See "Table 35.6-3 Format Setting of Serial Chip Selection Pin" for details.

(Note) Use proper usage for setting the registers other than the above.

■ Normal Transfer (II) Timing Chart (Serial Chip Select Terminal Unused)

Figure 35.6-10 Normal Transfer (II) Timing Chart (Serial Chip Select Terminal Unused)



[1] Master operation (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 resulted 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 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 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").

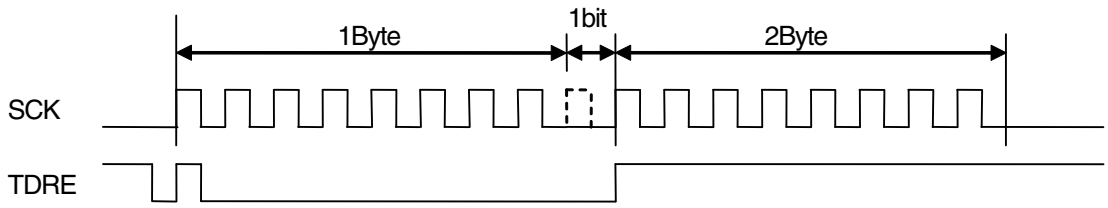
(2) When transmission data is written in TDR, SSR:TDRE="0" is set, and the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE="1" is set, and when transmission interrupt is enabled (SCR:TIE="1"), 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".

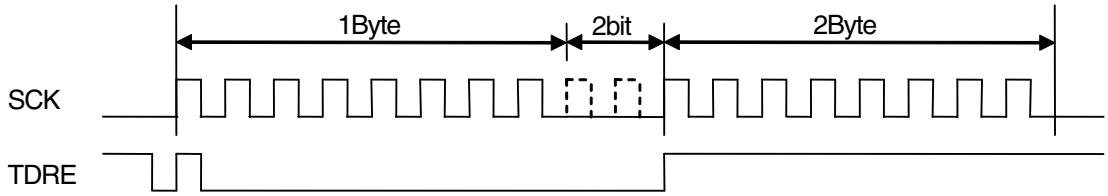
● Successive data transmission or reception wait operation

If setting other than (ESCR.WT1, ESCR.WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

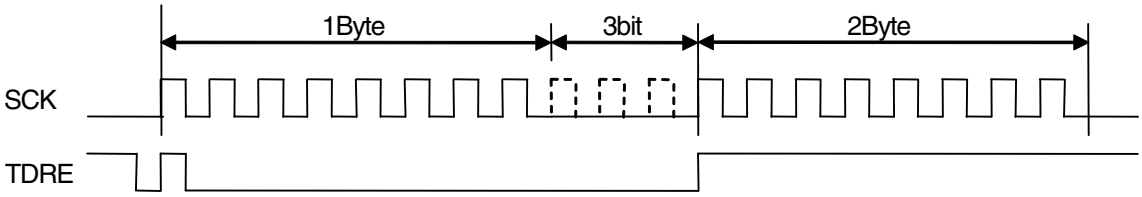
■ ESCR.WT1=0, ESCR.WT0=1 (for master)



■ ESCR.WT1=1, ESCR.WT0=0 (for master)



■ ESCR.WT1=1, ESCR.WT0=1 (for master)



[2] Slave operation (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 resulted 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"), 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 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

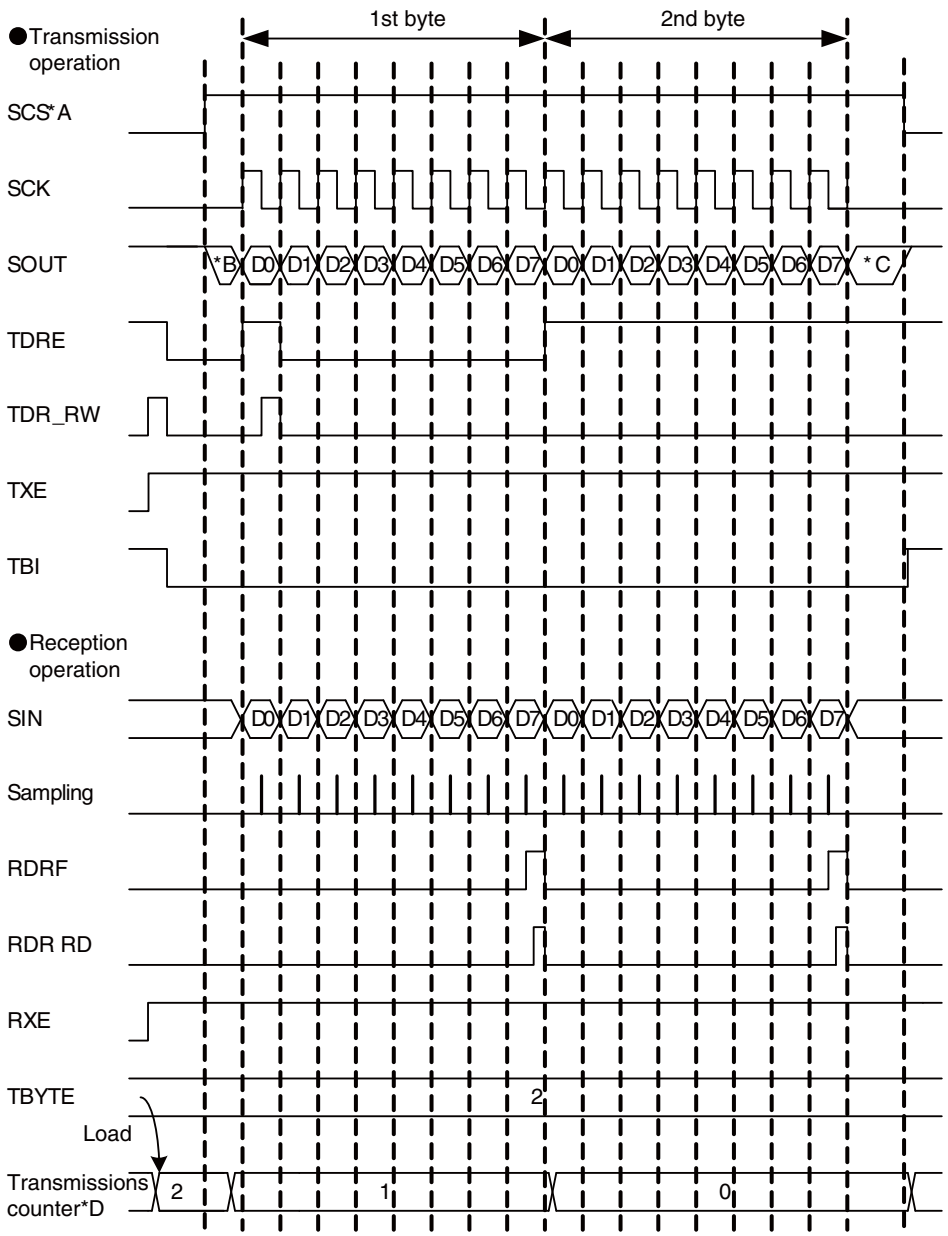
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE="1") and enable transmission/reception operation (SCR:TXE, RXE="1").

(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 Terminal Used)

Figure 35.6-11 Normal Transfer (I) Timing Chart (Serial Chip Select Terminal Used)



*A: SCS output when SCR:MS=0
SCS input when SCR:MS=1

*B: "H" when SCR:MS=0
Undefined when SCR:MS=1

*C: "H" when MS=0
D7 when MS=1

*D: Internal counter to be count transmission bytes

[1] Master operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: "n" shows the number of serial chip select terminal 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 terminal (SCS) will become active, transmission will be started after the setup time is lapsed for the serial chip select terminal. The start of the transmission resulted 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 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 terminal is lapsed after the transmission operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 terminal (SCS) active and reception operation will be started after the setup time is lapsed for the serial chip select terminal. The start of the reception resulted 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 received 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 terminal is lapsed after the reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 frame 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 terminal (SCS) will become active and transmission/reception will be started after the setup time is lapsed for the serial chip select terminal. 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.

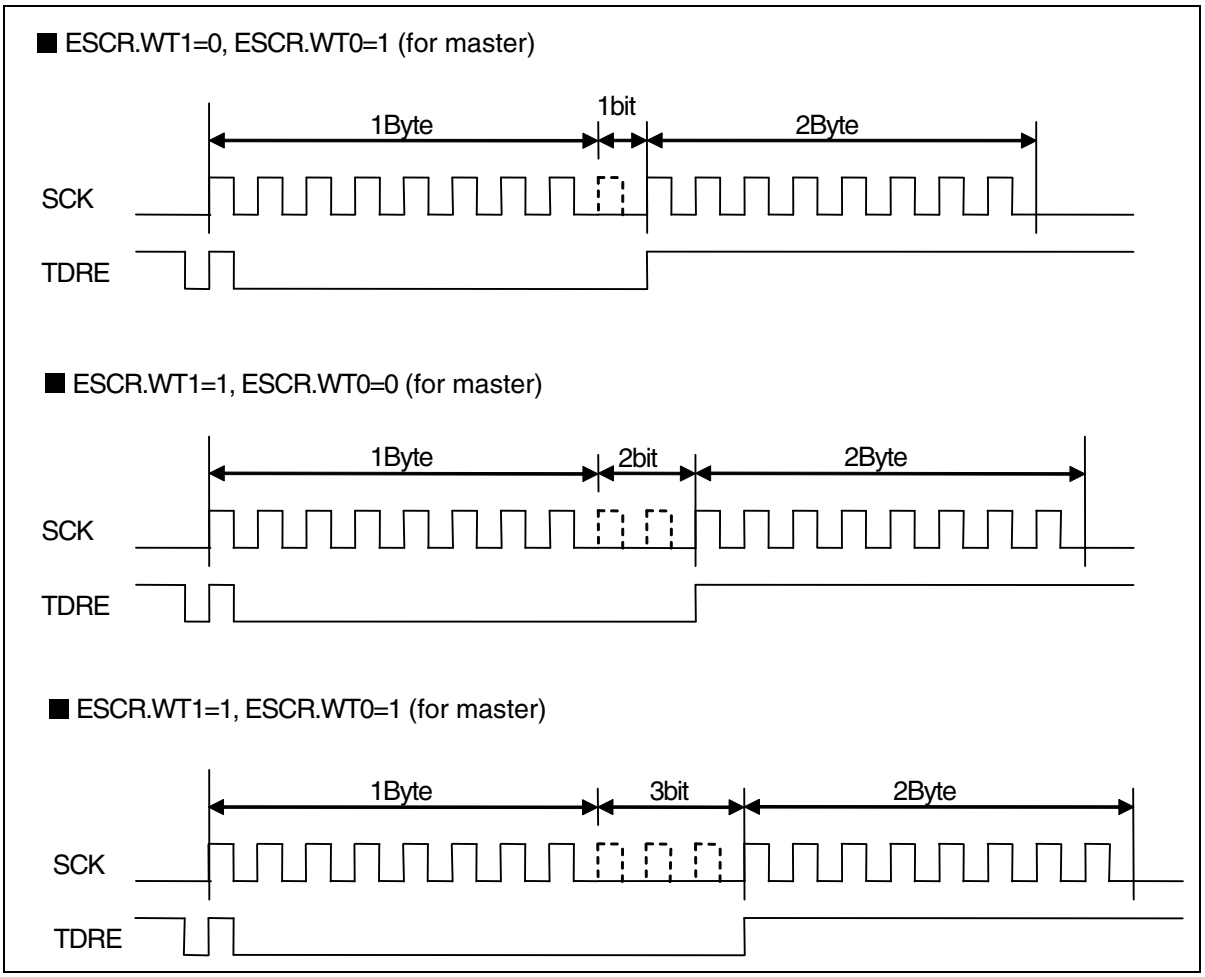
(3) While operating transmission/reception, the reception data will be sampled at a falling edge of the serial clock output (SCK). Receiving the last bit of receiving data sets SSR:RDRF="1" and when the reception interrupt is enabled (SCR:RIE="1"), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the received 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 terminal is lapsed after the transmission/reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (SCS) will remain active.

● Continuous Data Transmission or Reception Wait Operation

When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.



[2] Slave operation (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 terminal (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 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 terminal (SCS) becomes inactive, and serial output terminal (SOUT) 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 terminal (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.

(3) At this time, the receive data (RDR) can be read.

(4) Reading the received data (RDR) clears SSR:RDRF to "0".

(5) Reception operation will be terminated when serial chip select terminal (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 terminal (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 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 received data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when serial chip select terminal (SCS) becomes inactive, and serial output terminal (SOUT) becomes "H".

35.6.2.3 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 16, 20, 24, 32 bits

■ Register Settings

The following table lists the register settings required for SPI transfer (I).

SCR:SPI*=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*=0

Master operations: SCR:MS=0, SMR:SCKE=1

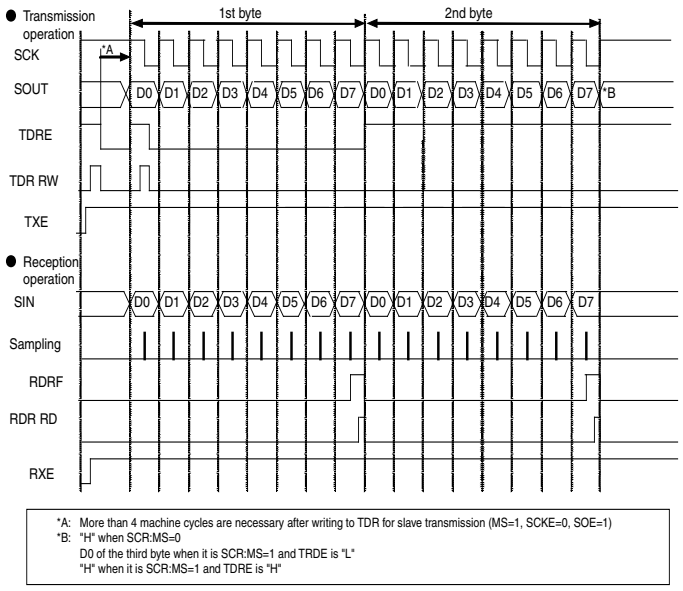
Slave operations: SCR:MS=1, SMR:SCKE=0

*: Bit settings depend on the condition. See "Table 35.6-3 Format Setting of Serial Chip Selection Pin" for details.

(Note) Use proper usage for setting the registers other than the above.

■ SPI Transfer (I) Timing Chart (Serial Chip Select Terminal Unused)

Figure 35.6-12 SPI Transfer (I) Timing Chart (Serial Chip Select Terminal Unused)



[1] Master operation (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 resulted 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").

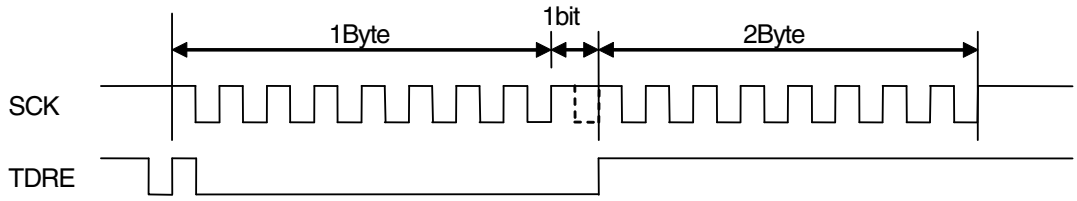
(2) When transmission data is written in TDR, SSR:TDRE="0" is set and the first bit is output. Then, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE="1" is set, and when 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".

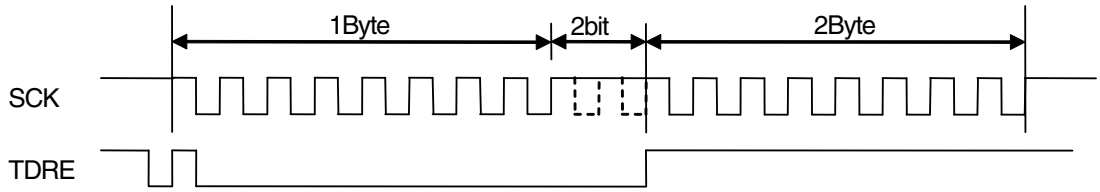
● Successive data transmission or reception wait operation

If setting other than (ESCR:WT1, ESCR:WT0)= (0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

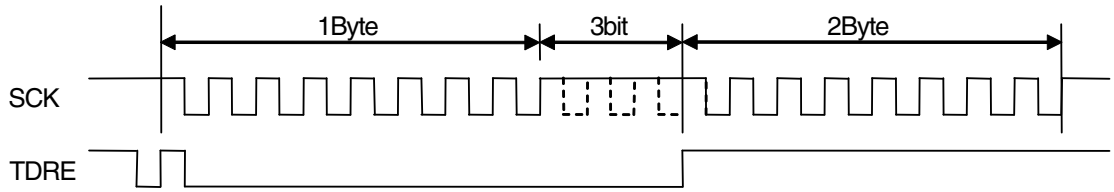
■ ESCR.WT1=0, ESCR.WT0=1 (for master)



■ ESCR.WT1=1, ESCR.WT0=0 (for master)



■ ESCR.WT1=1, ESCR.WT0=1 (for master)



[2] Slave operation(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 resulted 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.

(Notes) After transmission operation is enabled (SCR:TXE="1"), when you write first transmitting 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"), 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 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

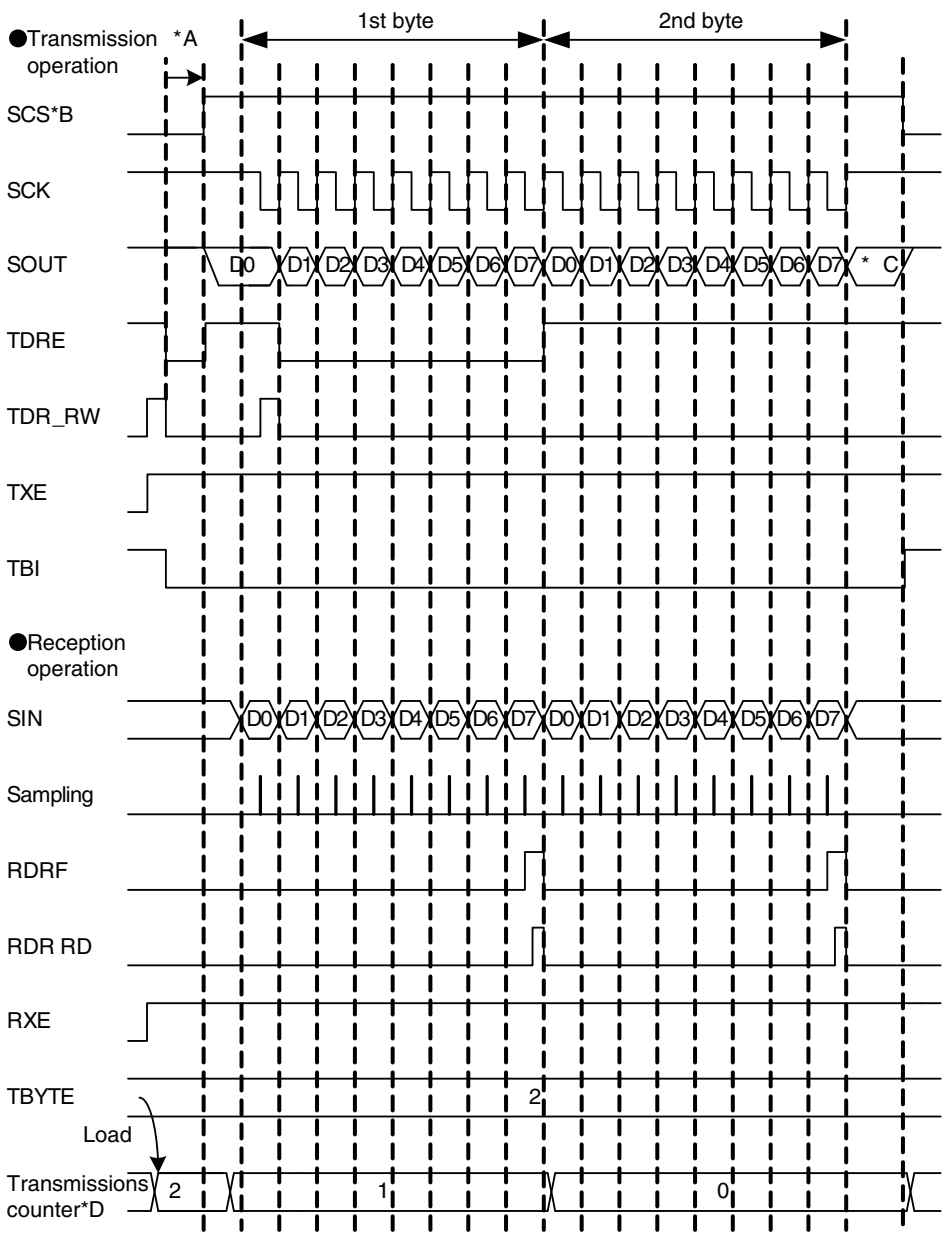
(1) Disable serial data output (SMR:SOE="0"), enable reception interrupt (SCR:RIE="1"), enable reception operation (SCR: RXE="1"), and enable transmission operation (SCR:TXE="1"). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the rising edge of 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, output transmission data in synchronization with the rising edge of the serial clock.

■ SPI Transfer (I) Timing Chart (Serial Chip Select Terminal Used)

Figure 35.6-13 SPI Transfer (I) Timing Chart (Serial Chip Select Terminal Used)



*A: More than 4 machine cycles are necessary after writing to TDR for slave transmission (MS=1, SCKE=0, SOE=1)
*B: SCS output when MS=0, SCS input when MS=1
*C: "D7" when SCR:MS=0
D0 of the third byte when it is SCR:MS=1 and TRDE is "L"
D7 when it is SCR:MS=1 and TDRE is "H"
*D: Internal counter to be count transmission bytes

[1] Master operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*:"n" shows the number of serial chip select terminal 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 terminal (SCS) will become active at the same time of outputting the first bit and transmission will be started after the setup time is lapsed for the serial chip select terminal. The start of the transmission resulted 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 terminal is lapsed after the transmission operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 terminal (SCS) active and reception operation will be started after the setup time is lapsed for the serial chip select terminal. The start of the reception resulted 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 received 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 terminal is lapsed after the reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (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 frame 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 terminal (SCS) will become active at the same time of outputting the first bit and transmission/reception will be started after the setup time is lapsed for the serial chip select terminal. 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.

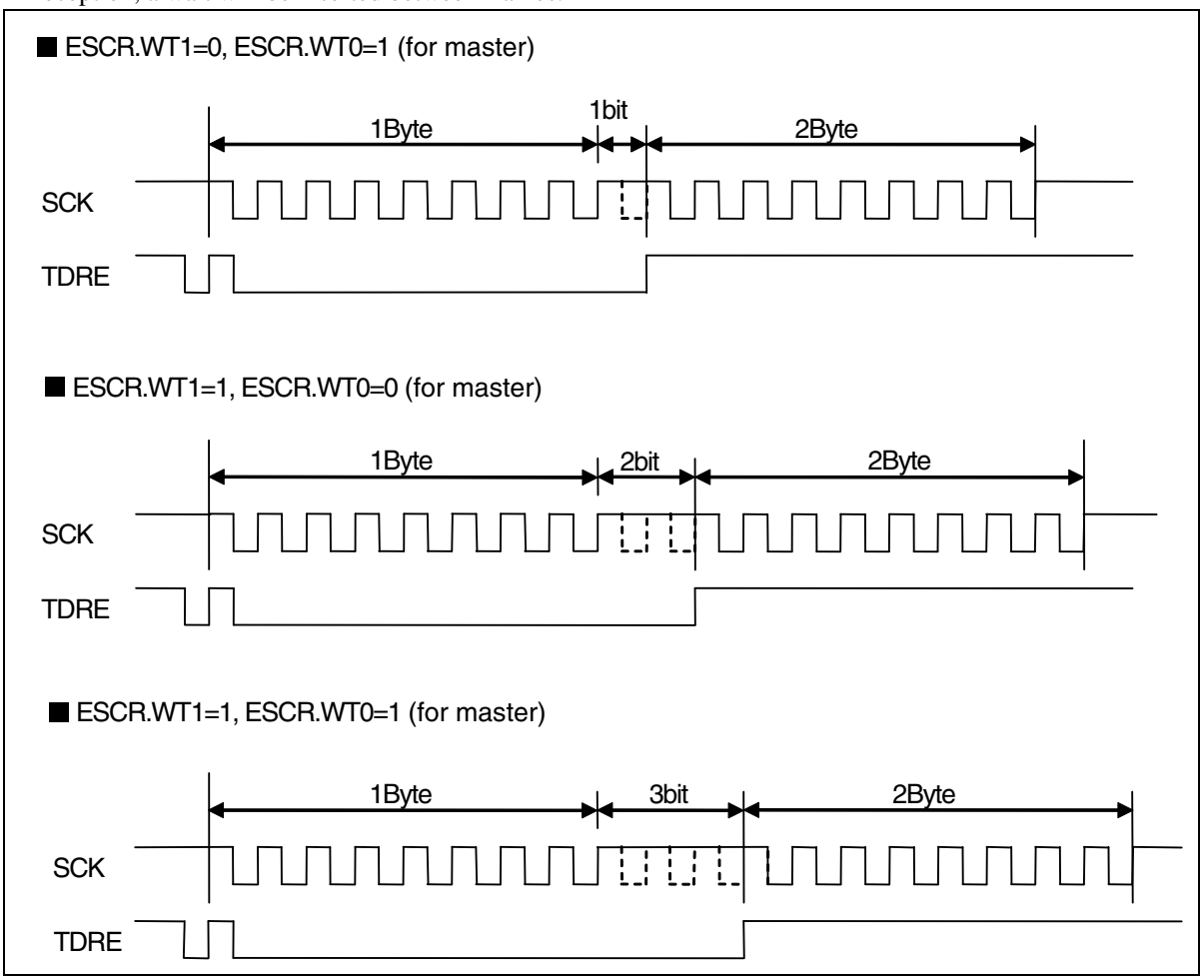
(3) The reception data will be sampled at a falling edge of the serial clock output (SCK). Receiving the last bit of receiving data sets SSR:RDRF="1" and when the reception interrupt is enabled (SCR:RIE="1"), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the received 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 terminal is lapsed after the transmission/reception operation, the serial chip select terminal (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select terminal (SCS) will remain active.

● Continuous Data Transmission or Reception Wait Operation

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.



[2] Slave operation (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 terminal (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 terminal (SCS) becomes inactive, and serial output terminal (SOUT) becomes "H".

(Notes) After transmission operation is enabled (SCR:TXE="1"), when you write first transmitting 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"), reception operation will be started when the serial chip select terminal (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.

(3) At this time, the receive data (RDR) can be read.

(4) Reading the received data (RDR) clears SSR:RDRF to "0".

(5) Reception operation will be terminated when serial chip select terminal (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 terminal (SCS) will become active, transmission/reception will be started and the first bit will be output. When transmission/reception is started, the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When the first bit of transmitting 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 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 received data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when serial chip select terminal (SCS) becomes inactive, and serial output terminal (SOUT) becomes "H".

35.6.2.4 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	5-16, 20, 24, 32 bits

■ Register Settings

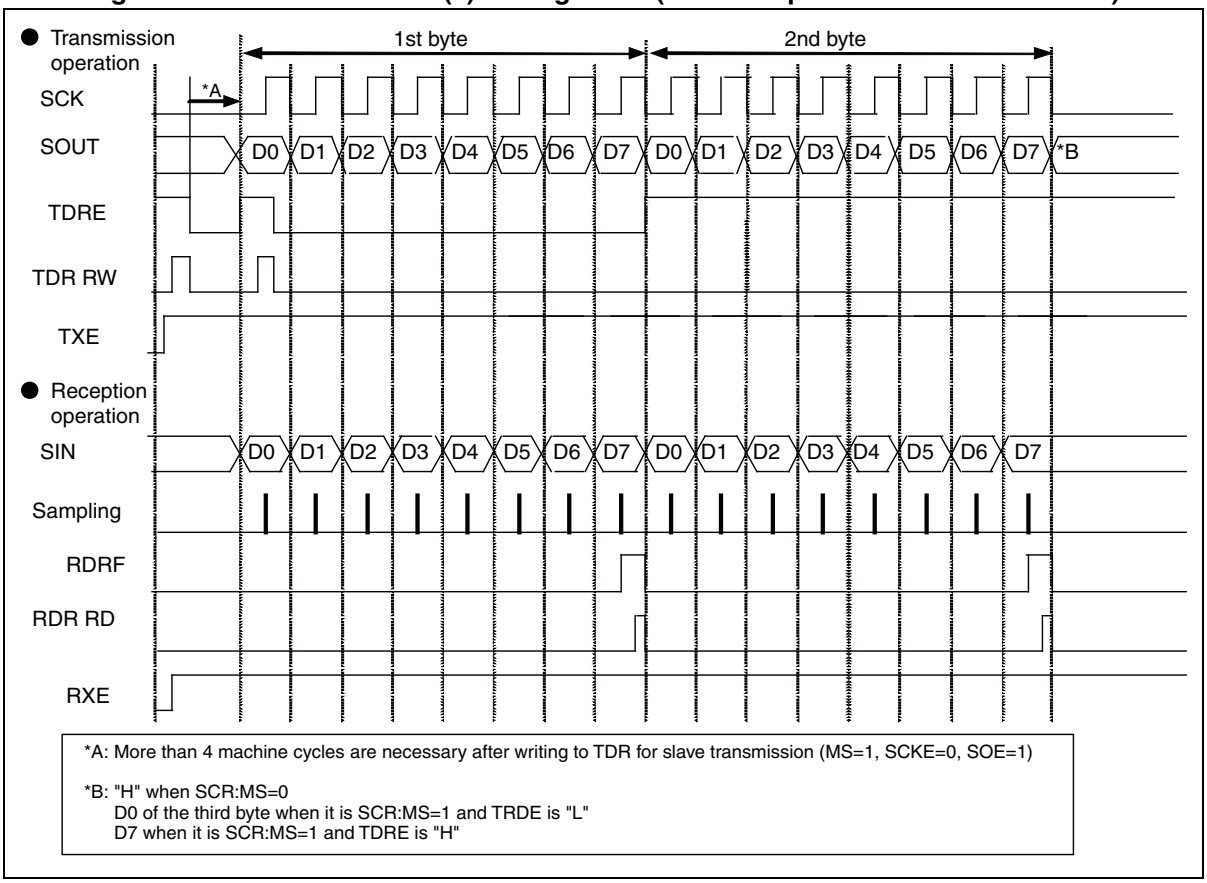
The following table lists the register settings required for SPI Transfer (II).

SCR:SPI*=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0
*: Bit settings depend on the condition. See "Table 35.6-3 Format Setting of Serial Chip Selection Pin" for details.

(Note) Use proper usage for setting the registers other than the above.

■ SPI Transfer (II) Timing Chart (Serial Chip Select Terminal Unused)

Figure 35.6-14 SPI Transfer (II) Timing Chart (Serial Chip Select Terminal Unused)



[1] Master operation (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 resulted in outputting transmitting 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 resulted 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 rising edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF="1" and, if the reception interrupt is enabled (SCR:RIE="1"), 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

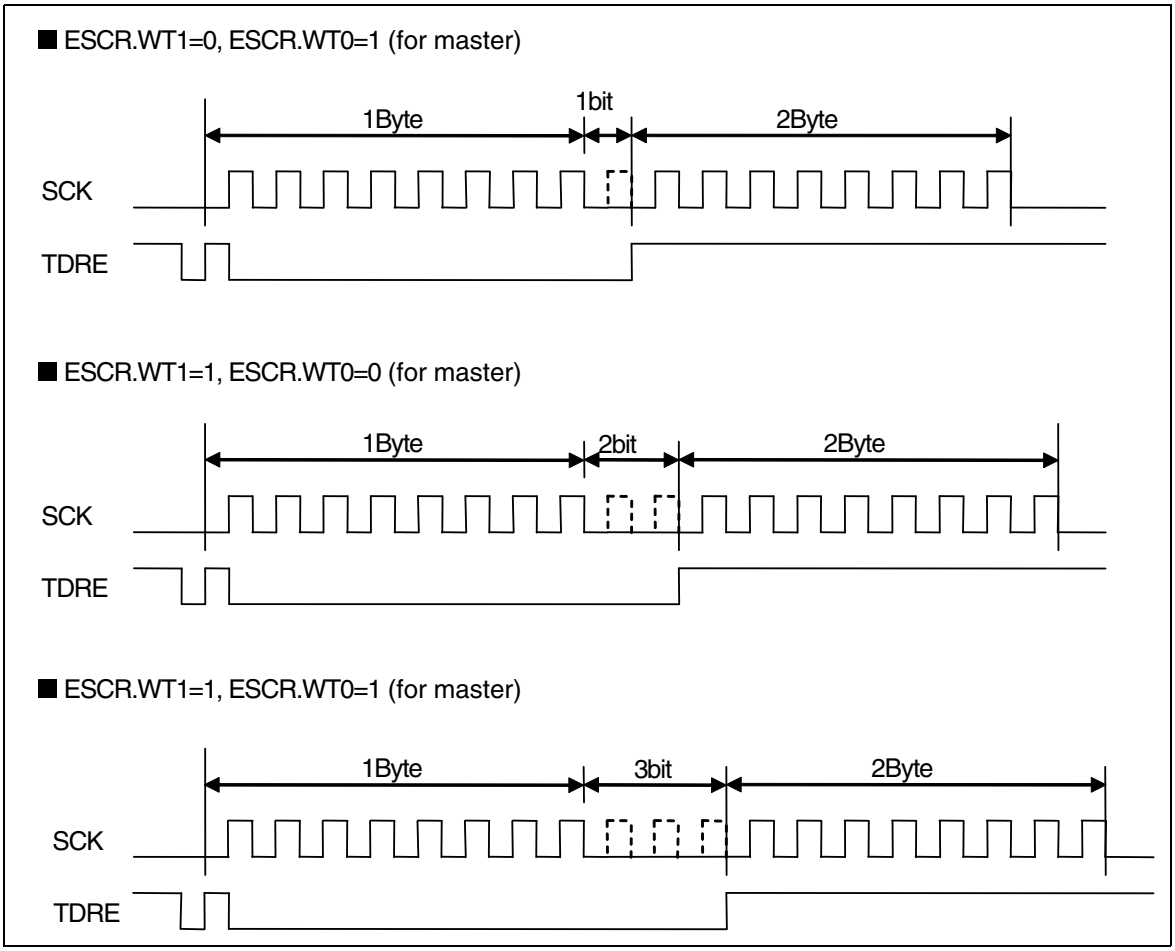
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE="1") and enable transmission/reception operation (SCR:TXE, RXE="1").

(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

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 (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 resulted 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.

(Notes) 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

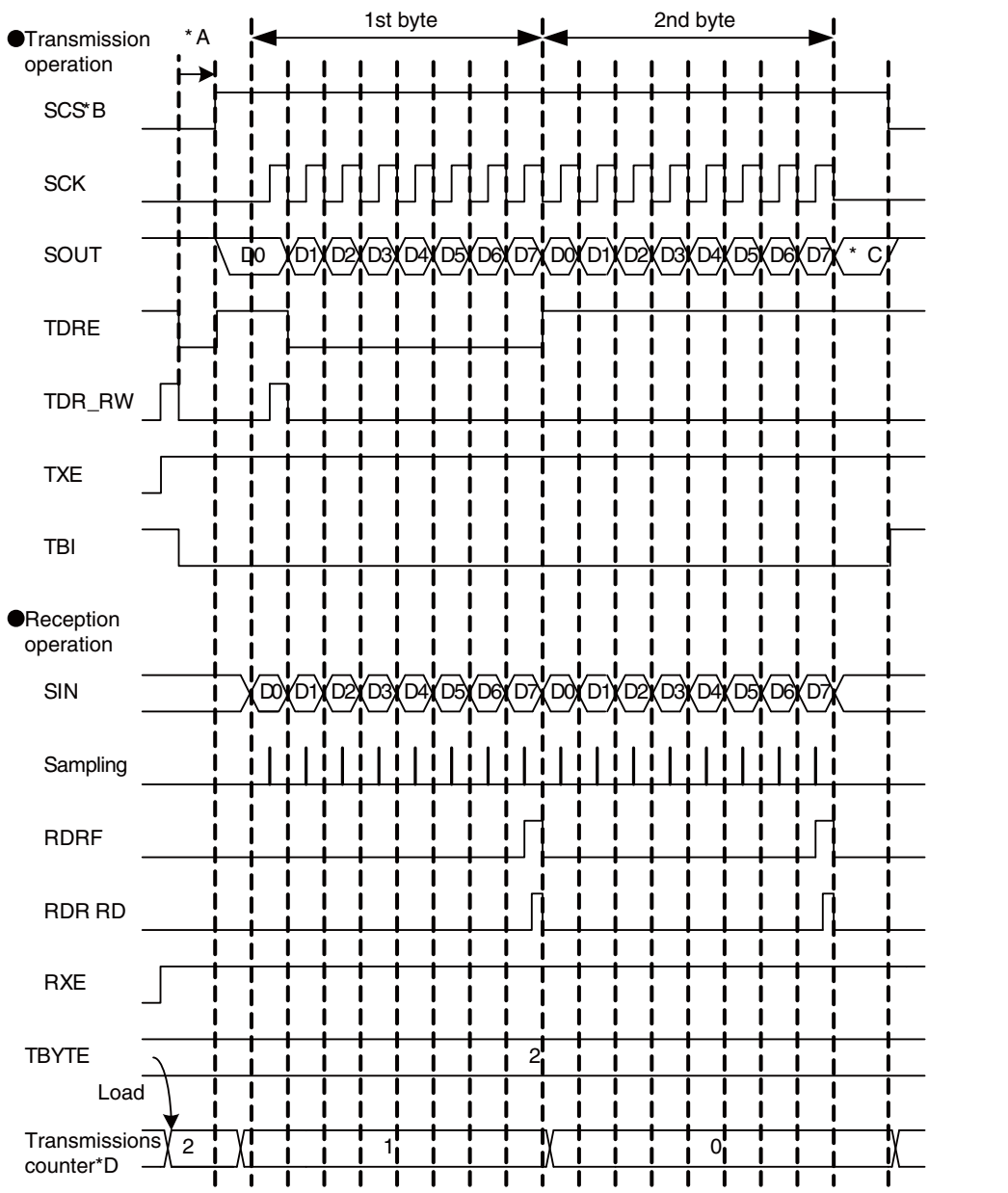
(1) Disable serial data output (SMR:SOE="0"), enable reception interrupt (SCR:RIE="1"), enable reception operation (SCR: RXE="1"), and enable transmission operation (SCR:TXE="1"). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the falling edge of serial clock 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, output transmission data in synchronization with the rising edge of the serial clock.

■ SPI Transfer (II) Timing Chart (When Serial Chip Selection Pin Used)

Figure 35.6-15 SPI Transfer (II) Timing Chart (When Serial Chip Selection Pin Used)



*A: More than 4 machine cycles are necessary after writing to TDR for slave transmission (MS=1, SCKE=0, SOE=1)
*B: SCS output when MS=0, SCS input when MS=1
*C: D7 when SCR:MS=0
D0 of the third byte when it is SCR:MS=1 and TRDE is "L"
D7 when it is SCR:MS=1 and TDRE is "H"
*D: Internal counter to be count transmission bytes

[1] Master operation (SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

*: n is the serial chip selection 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 selection pin (SCS) become active at the same time the first bit is output, and the transmission operation starts after the setup time of the serial chip selection pin has elapsed. 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 selection pin has elapsed, the serial chip selection pin (SCS) become inactive. However, if the serial chip selection active level (SCSCR:SCAM="1") is maintained at this time, the serial chip selection 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 selection pin (SCS) active and starts the reception operation after the setup time of that pin has elapsed. Starting the reception operation samples the reception 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 reception data (RDR) can be read.

(3) Reading the reception 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 selection pin has elapsed, the serial chip selection pin (SCS) become inactive. However, if the serial chip selection active level (SCSCR:SCAM="1") is maintained at this time, the serial chip selection 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 selection pin (SCS) become active at the same time the first bit is output, and the transmission/reception operation starts after the setup time of the serial chip selection pin has elapsed. 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.

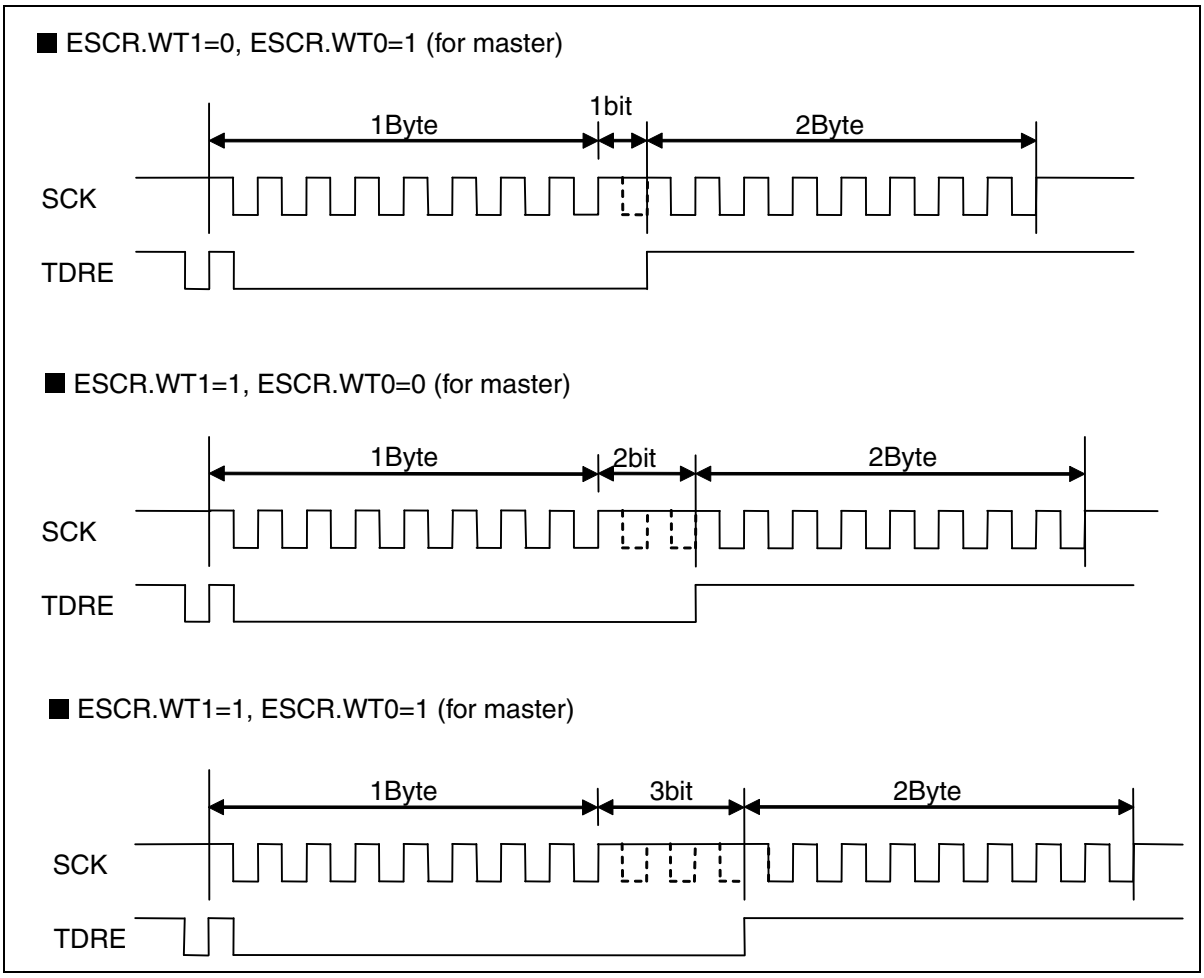
(3) Reception data is sampled by the rising edge of the serial clock (SCK) output. When the last bit of reception data is received, SSR:RDRF="1" is set, and when reception interrupt is enabled (SCR:RIE="1"), a reception interrupt request is output. At this time, the reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

(4) 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 selection pin has elapsed, the serial chip selection pin (SCS) become inactive. However, if the serial chip selection active level (SCSCR:SCAM="1") is maintained at this time, the serial chip selection pin (SCS) maintains its active state.

● Successive data transmission or reception wait operation

If setting other than (ESCR:WT1, ESCR:WT0)=(0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation (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 selection 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 selection pin (SCS) becomes inactive, the transmission operation is terminated and the serial output pin (SOUT) becomes "H".

(Notes) 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 selection 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 selection 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 selection 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 selection pin (SCS) becomes inactive, the transmission/reception operation is terminated and the serial output pin (SOUT) becomes "H".

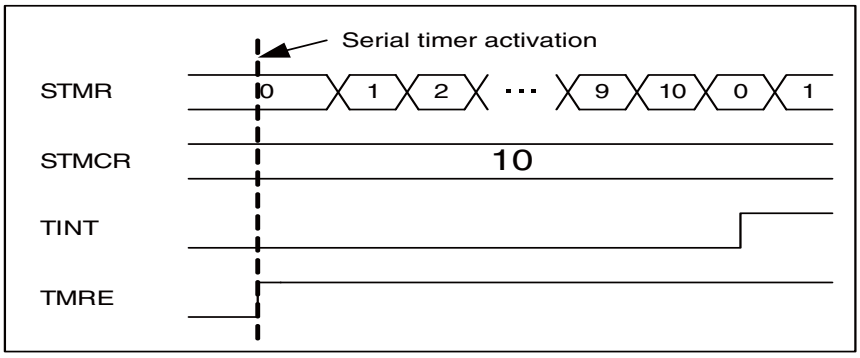
35.6.2.5 Operation of Serial Timer

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, set "1" in 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 35.6-16 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 35.6-17 Timer Operation (STMCR="10")

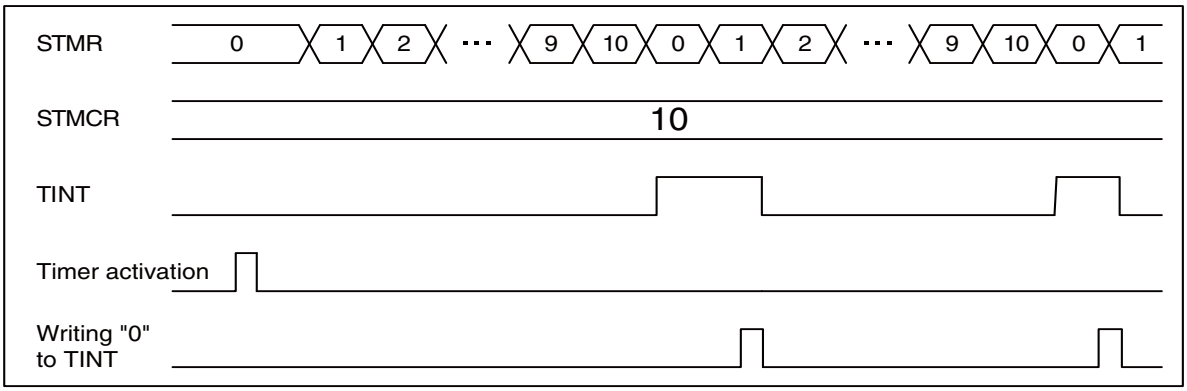


Figure 35.6-18 Flowchart of Serial Timer Initial Setting

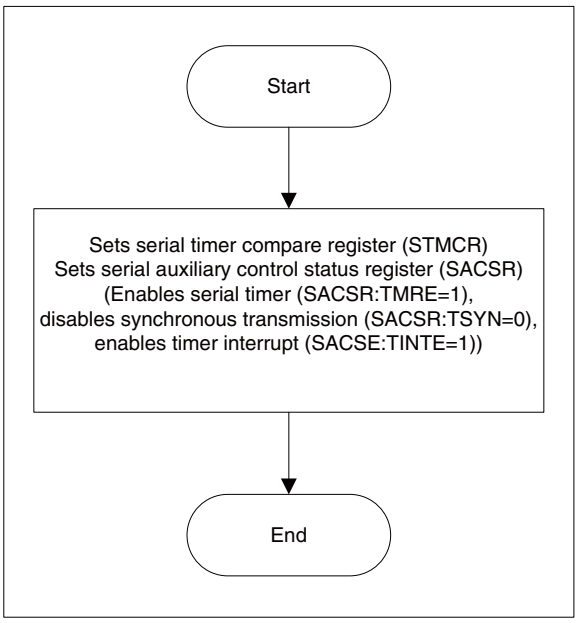
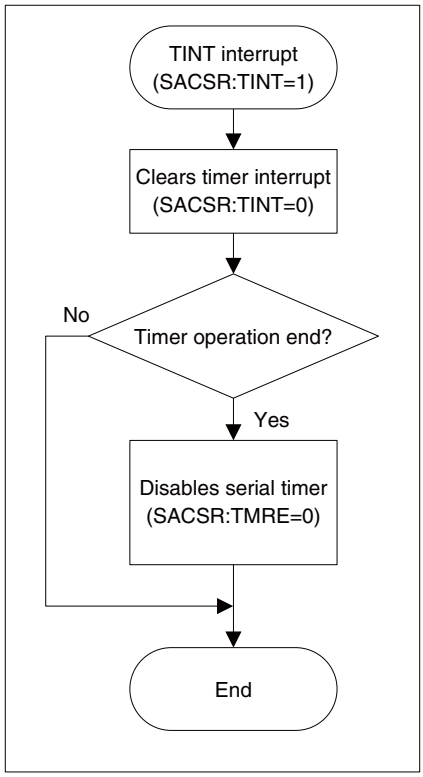


Figure 35.6-19 Flowchart of Serial Timer Interrupt Processing



(Note) With the timer compare register (STMCR) set to "0000"h and synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) is fixed at "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000"b while the timer is running.

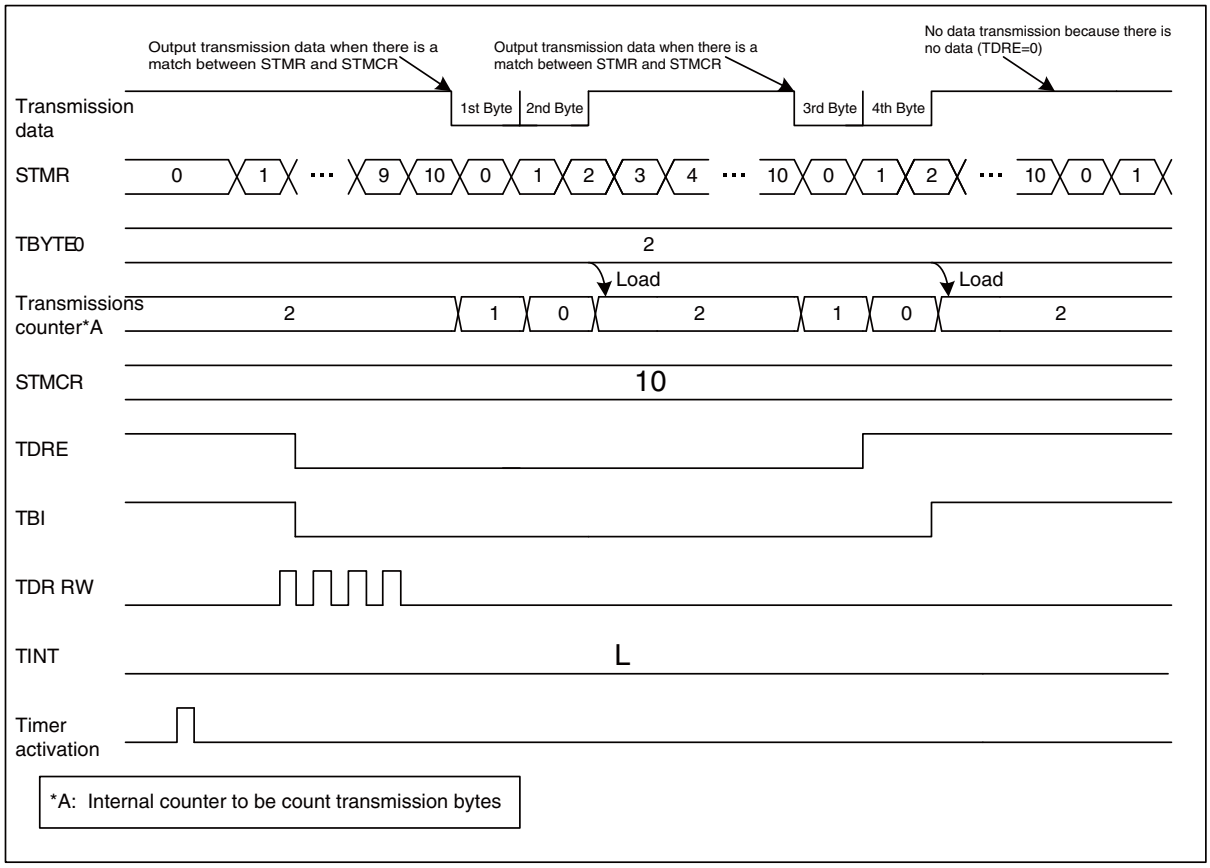
● Timer-synchronized Transmission Operation

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 35.6-20 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 selection 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, the transmission data will not be sent until the serial timer register (STMR) matches the serial timer comparison register (STMCR) next time.

However, 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 be stopped and the next transmission will be started after sending as much data as TBYTE0.

- Transmission operation in progress (SSR:TBI="0")

If the 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.

Figure 35.6-21 Flowchart of Initial Setting for Timer-synchronized Transmission

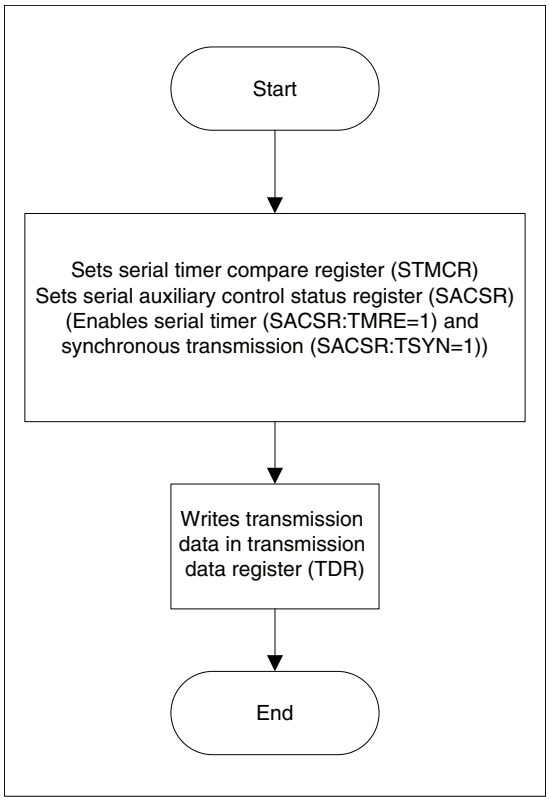
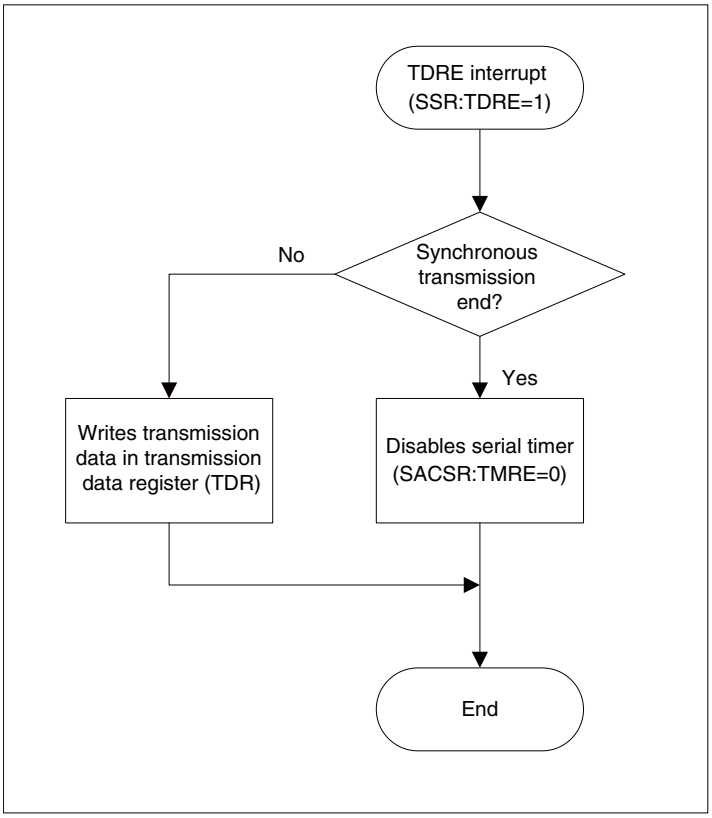


Figure 35.6-22 Flowchart of Interrupt Processing for Timer-synchronized Transmission



(Notes)

- 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 (SACSR:TBEEN="1"), a chip selection error (SACSR:CSE="1") occurs. When the chip selection error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmission data register (TDR) does not start transmission operation.
  - When the transfer byte error is disabled (SACSR:TBEEN="0"), transmission operation will be stopped until transmission data is written to the transmission data register (TDR). Transmission operation will be restarted when transmission data is written to the transmission data register (TDR).

35.6.2.6 Operation of Serial Chip Selection

This section explains the operation of the serial chip selection.

● Operation of master mode (SCR:MS="0")

In master mode (SCR:MS="0"), the serial chip selection pin operates as shown below:

(1) With the serial chip selection operation enabled (SCSCR:CSENn="1") and transmission enabled (SCR:TXE="1"), writing transmission data makes the serial chip selection pin active.

(2) The transmission/reception operation starts after the setup time of the serial chip selection pin has elapsed.

(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 selection pin has elapsed, the serial chip selection pin become inactive.

Figure 35.6-23 Operation of Serial Chip Selection (Master Transmission (MS="0"), Normal Transfer (SPI="0"), SCINV="0")

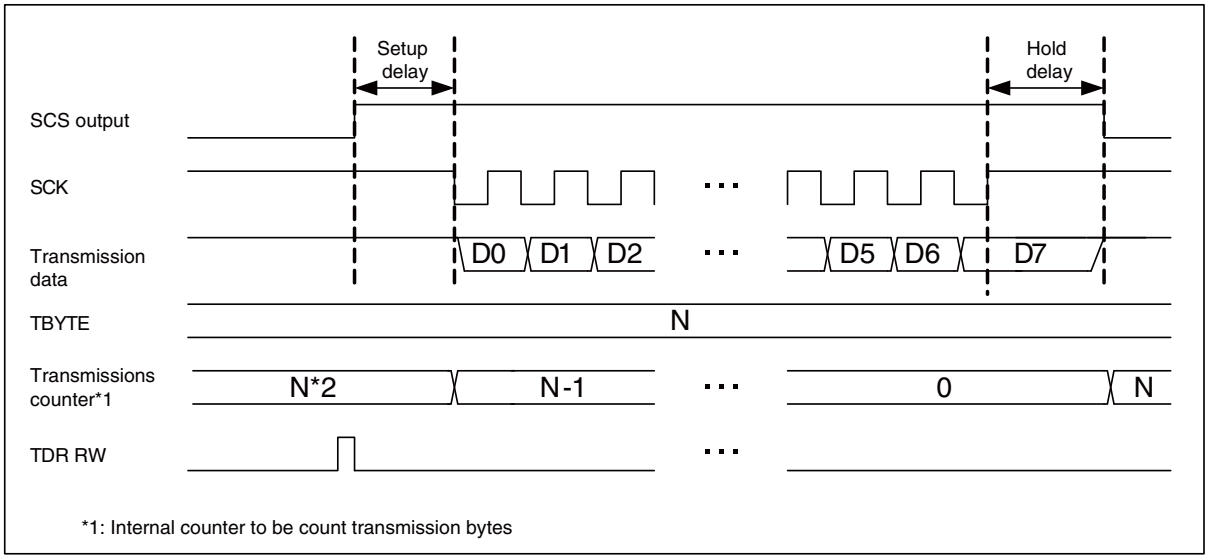
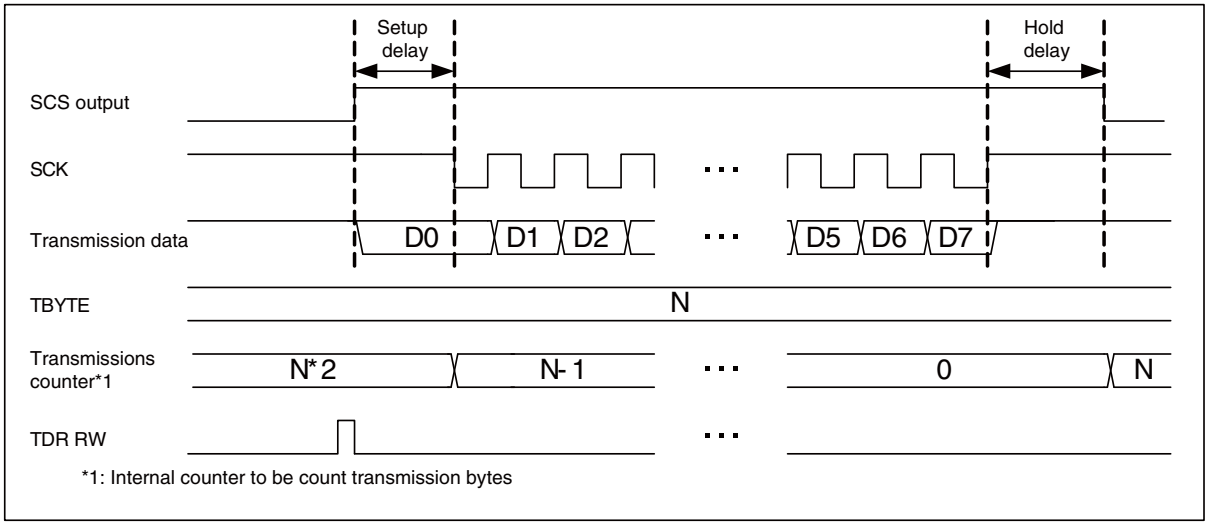


Figure 35.6-24 Operation of Serial Chip Selection (Master Transmission (MS="0"), SPI Transfer (SPI="1"), SCINV="0")

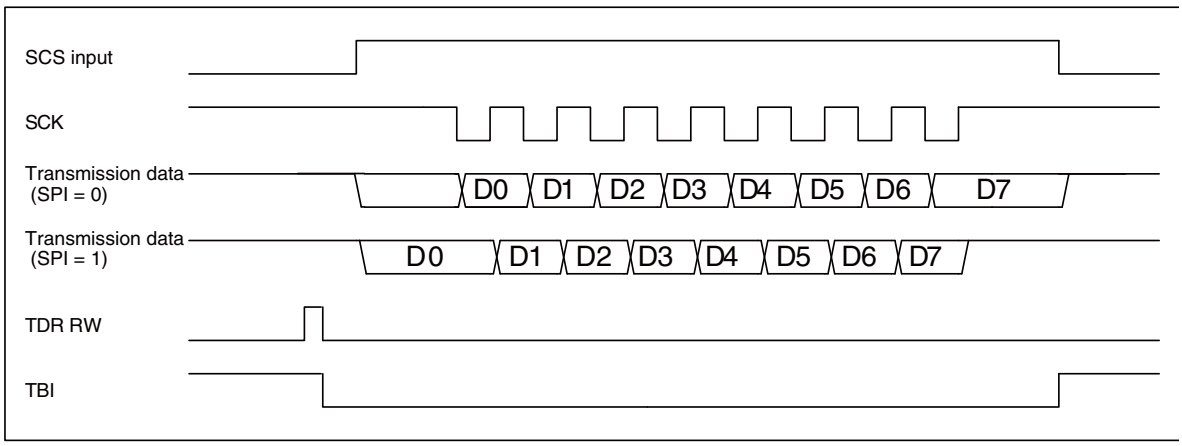


(Notes)
- If the transmission is disabled (SCR:TXE="0") and the software reset is set (SCR:UPCL=1) when the serial chip selection pin is active, the serial chip selection pin become inactive.
- If the transmission data is written to the hold delay time of the serial chip selection pin, the pin does not become inactive and the next transmission data is sent.
- When the serial chip selection pin does not maintain its active state (SCSCR:SCAM="0"), it becomes inactive. Then, after the deselect time elapses, 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 selection pin.
- If the transmit data register (TDR) has no valid transmission data (SSR:TDRE="1") when 1 frame transmission is completed during sending frames less than TBYTE setting, the following operation will be performed.
  - When the transfer byte error is enabled (SACSR:TBEEN="1"), a chip selection error (SACSR:CSE="1") occurs. After the hold delay time has elapsed since the chip selection error (SACSR:CSE="1") occurred, the serial chip selection pin become inactive. When the chip selection error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmission data register (TDR) does not start transmission operation.
  - When the transfer byte error is disabled (SACSR:TBEEN="0"), transmission operation will be stopped until transmission data is written to the transmission data register (TDR). At this time, the serial chip selection pin is active. Transmission operation will be restarted when transmission data is written to the transmission data register (TDR).

● Operation of slave mode (SCR:MS="1")

When the serial chip selection pin 0 (SCS0) is enabled (SCSCR:CSEN0="1") and the serial chip selection pin input become active, the transmission or reception operation is performed in synchronization to the serial clock (SCK). Then, when the serial chip selection pin input becomes inactive, the transmission or reception operation is terminated.

Figure 35.6-25 Operation of Serial Chip Selection in Slave Mode (Slave Transmission, SCINV="0")



(Notes)
- It does not operate even if the serial clock is input when the serial chip selection pin input is inactive.
- If the serial chip selection input becomes inactive before sampling a bit at the end during reception operation, the data being received is deleted.
- If the serial chip selection input becomes inactive during transmission operation, the data being transmitted is deleted and the chip selection error occurs (SACSR:CSE).
- If input of the serial chip selection pin becomes inactive with TDR empty (SSR:TDRE="1"), 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 selection pin.

● Timing Adjustment of Serial Chip Selection

When the serial chip selection 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 selection timing register (SCSTR3 to 0).

- Setup delay time

Time from the instant when the serial chip selection pin becomes active to the instant when the serial clock is output. See Figure 35.6-26 and Figure 35.6-27 for the specification of the setup delay time.

It can be adjusted by the chip selection 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 selection pin becomes inactive. See Figure 35.6-26 and Figure 35.6-27 for the specification of the hold delay time.

It can be adjusted by the chip selection hold delay bit (SCSTR1:CSHD7 to CSHD0).

- Deselection time

Minimum time from the instant when the serial chip selection pin becomes inactive to the next instant when the pin turns active. Even if transmission data is written to the transmission data register (TDR) during the deselection time, the serial chip selection pin does not become active until the deselection time end. See Figure 35.6-26 and Figure 35.6-27 for the specification of the deselection time.

Figure 35.6-26 Timing Adjustment (Normal Transfer (SPI="0"), SCINV="0")

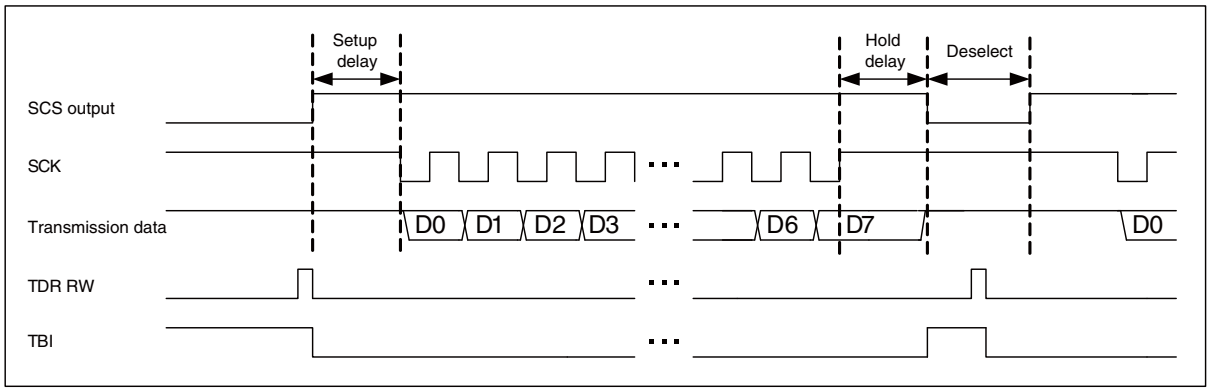
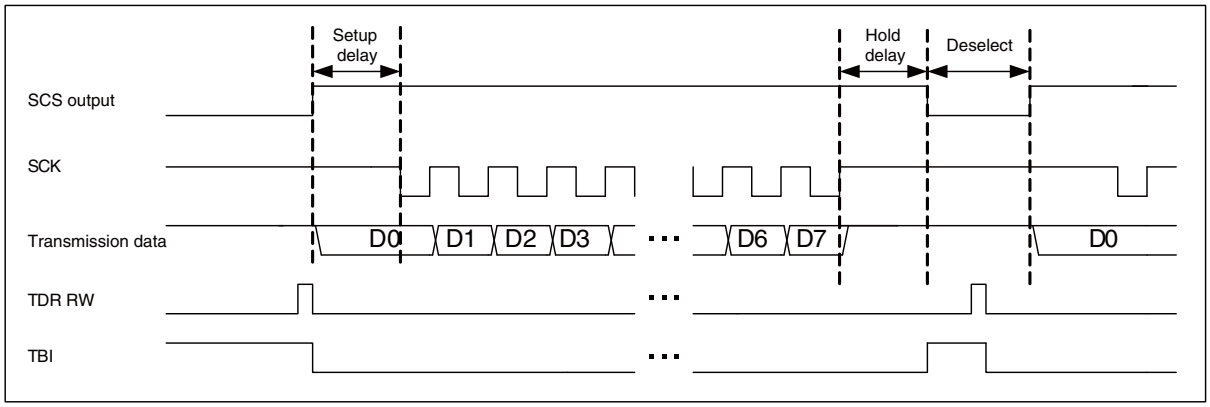


Figure 35.6-27 Timing Adjustment (SPI Transfer (SPI="1"), SCINV="0")



(Note) With the normal transfer (SPI=0) and no hold delay time (SCSTR1:CSHD7 to CSHD0="00"h), the chip selection pin may become inactive before the sampling of the last bit. In such cases, adjust by increasing the value of SCSTR1:CSHD7 to CSHD0.

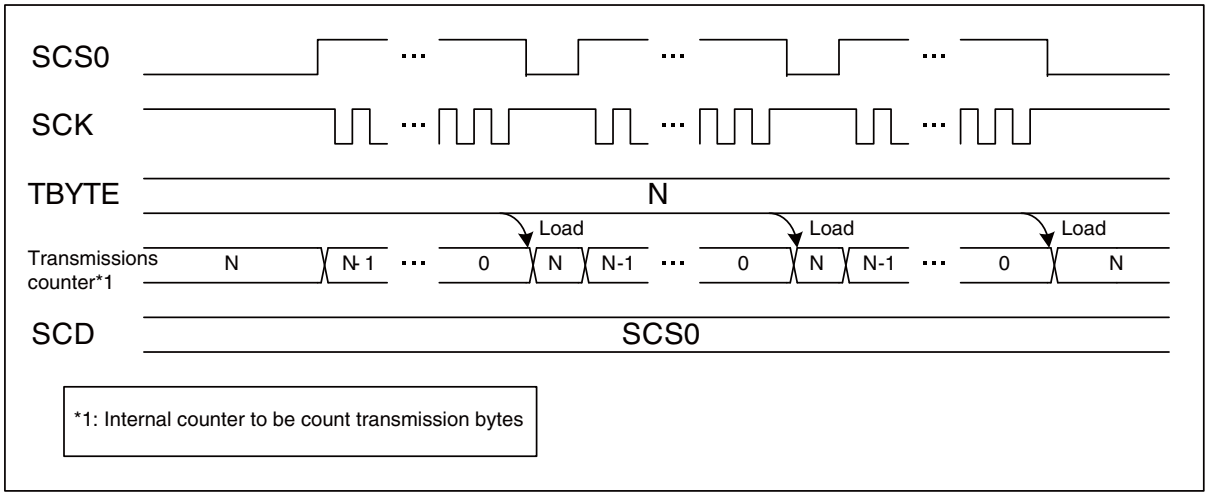
● Single Operation of Chip Selection Pin (Only Valid in Master Mode (SCR:MS="0"))

If the serial chip selection start bit (SCSCR:SST1 and SST0) and the serial chip selection end bit (SCSCR:SED1 and SED0) are equal, only the specified serial chip selection bit operates.

When the serial chip selection does not maintain its active state (SCSCR:SCAM="0"), the serial chip selection pin become inactive for each the data transmission/receptions for the number of times set with TBYTE.

See "Operation of Serial Chip Selection to Maintain Active" for the operation of the serial chip selection pin when the serial chip selection maintains its active state (SCSCR:SCAM="1").

Figure 35.6-28 Single Operation of Chip Selection (SST1 and SST0="0", SED1 and SED0="0", CSEN0="1", SCAM="0")



(Note) At the single operation, timing adjustment of the serial chip selection pin (setup time, hold time, deselection time) is valid.

● Operation of Serial Chip Selection to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))

The serial chip selection pin is maintained in the active state when the transmission operation has started with the serial chip select active retain bit (SCSCR:SCAM) set to "1".

Table 35.6-2 Serial Chip Select Active Retain Bit (SCSCR:SCAM)

Current State	Current SCSCR:SCAM Bit	Current SSR:TDRE Bit	Next State
Transmission in progress (Number of transmission times < TBYTE)	0	-	The serial chip selection pin is kept active until the transmission of frames reaches the number of times set in TBYTE.
	1		
Frame transmission completed by number of times set in TBYTE	0		After the hold delay time elapses, the serial chip selection pin becomes inactive.
	1	1	Serial chip select is maintained in the active state.
		0	The transmission operation continues with serial chip select in the active state. Once again, the serial chip selection pin is kept active until the transmission of frames reaches the number of times set in TBYTE.
Hold delay in progress	0	-	After the hold delay time elapses, the serial chip selection pin becomes inactive.
	1 ⇒ 0	1	The hold delay operation stops, and serial chip select is maintained in the active state.
		0	The hold delay operation stops, and the transmission operation resumes with serial chip select in the active state. Once again, the serial chip selection pin is kept active until the transmission of frames reaches the number of times set in TBYTE.
Serial chip select being maintained in active state	1 ⇒ 0	-	After the hold delay time elapses, the serial chip selection pin becomes inactive.
	1	1	Serial chip select continues to be maintained in the active state.
		0	The transmission operation resumes with serial chip select in the active state. Once again, the serial chip selection pin is kept active until the transmission of frames reaches the number of times set in TBYTE.
Occurrence of chip select error (SACSR:CSE=1)	-	-	After the hold delay time elapses, the serial chip selection pin becomes inactive regardless of the SCAM setting.
Software reset execution (SCR:UPCL="1")			The serial chip selection pin becomes inactive immediately regardless of the SCAM setting.
Transmission disabled (SCR:TXE="0")			

(Notes) If the transmission 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 selection pin is not retained and becomes inactive after the hold delay time has elapsed, and the chip selection error (SACSR:CSE="1") occurs.

● Format Setting of Serial Chip Selection Pin

Active level of the chip selection for each serial chip selection 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 35.6-3.

Table 35.6-3 Format Setting of Serial Chip Selection Pin

Condition		Active level of chip selection	Serial clock inversion	SPI setting	Data direction	Data length
Chip selection format enabled (SCR:CSFE="1") and master mode (SCR:MS="0")	Serial chip selection pin 0 output	SCSCR0: SCLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Chip selection format disabled (SCR:CSFE="0")		SCSCR0: SCLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Slave mode (SCR:MS="1")						
Chip selection unused (SCSCR:CSEN3-0="0000"b)						

35.6.2.7 Test Mode

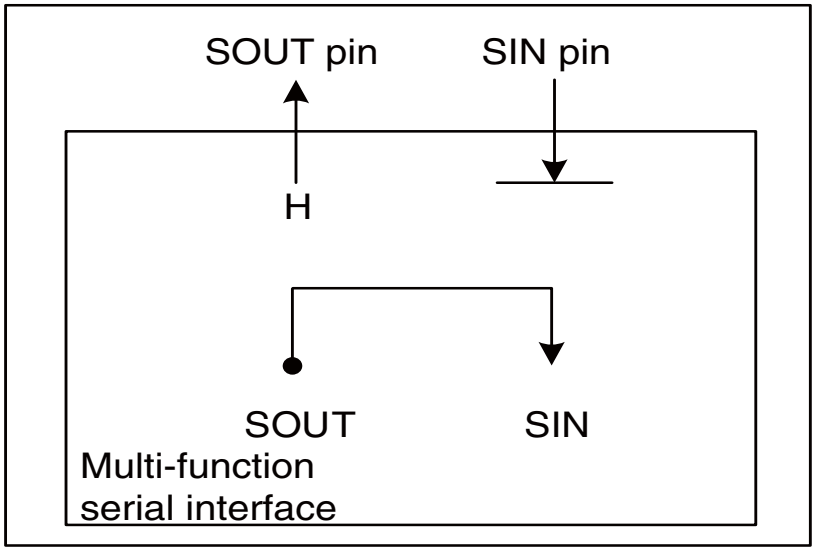
This section explains the operation of the test mode.

● Serial Test Mode

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 35.6-29 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").

35.6.2.8 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 $= (20 \times 1{,}000{,}000) / 153600 - 1 = 129$

Baud rate (calculated) $= (20 \times 1{,}000{,}000) / (129 + 1) = 153846$ bps

Error (%) $= (153846 - 153600) / 153600 \times 100 = 0.16(\%)$

(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 SMR:SCINV bit and SCR:SPI bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  If in normal transfer (SCR:SPI="0") the mark level of the serial clock is "H" (SMR:SCINV="0") or if in SPI transfer (SCR:SPI="1") the mark level of the serial clock is "L" (SMR:SCINV="1"), the "H" width of the serial clock is longer by one cycle of the bus clock.
  If in normal transfer (SCR:SPI="0") the mark level of the serial clock is "L" (SMR:SCINV="1") or if in SPI transfer (SCR:SPI="1") the mark level of the serial clock is "H" (SMR:SCINV="0"), 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 35.6-4 Reload Values and Baud Rate

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)

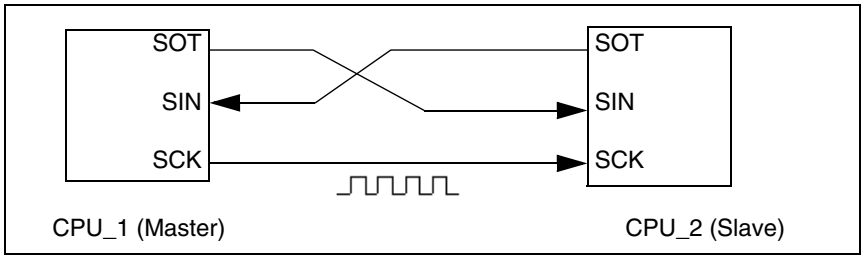
35.6.3 Setup Procedure and Program Flow

CSIO (clock synchronous serial interface) enables serial bidirectional transmission.

■ CPU Interconnection

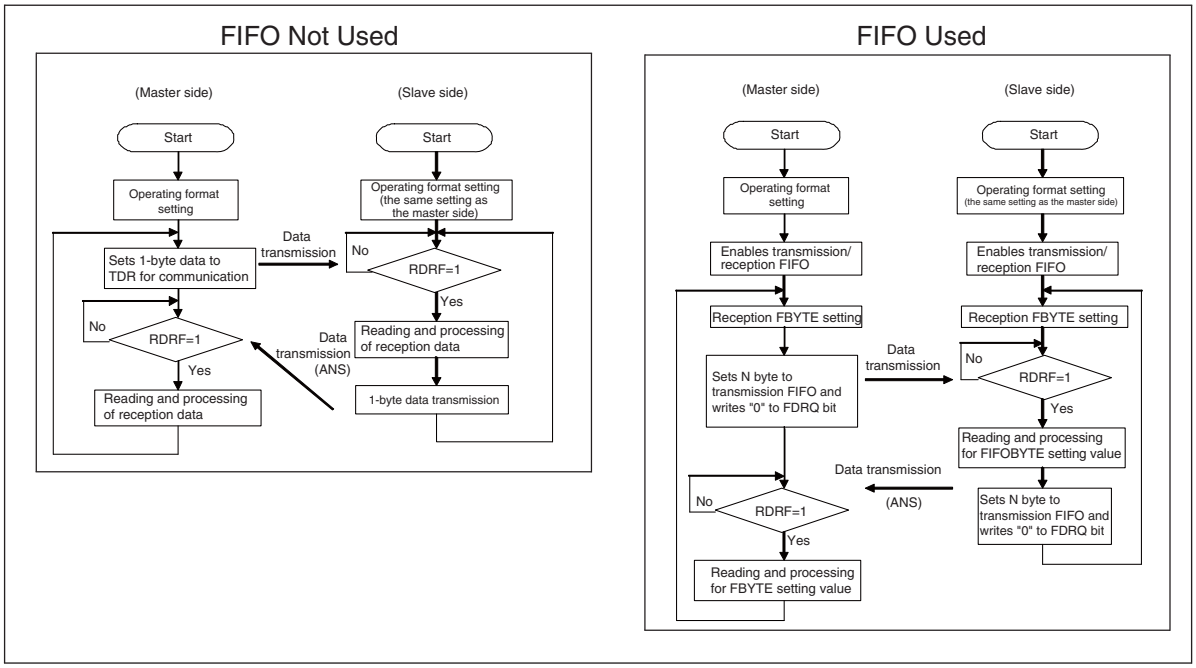
For CSIO (clock synchronous serial interface), select bidirectional communication. Connect 2 CPUs with each other as shown in Figure 35.6-30.

Figure 35.6-30 Example of Connection between CSIO Chips



■ Flowchart

Figure 35.6-31 Flowchart Example



35.7 Operation of LIN Interface (v2.1)

35.7.1 Interrupt of LIN Interface (v2.1)

The interrupts for the LIN interface (v2.1) include reception, transmission, and status interrupts. 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
- LIN Sync field detection
- Matching values of the serial timer compare register (STMCR) and the serial timer register (STMR)

35.7.1.1 List of Interrupts of LIN Interface (v2.1)

Table 35.7-1 indicates how LIN Interface (v2.1) interrupt control bits relate to interrupt factors.

Table 35.7-1 Interrupt Control Bits and Interrupt Factors for LIN Interface (v2.1)

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	Flaming error		
Transmission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*1
	TBI	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), write "1" to the Lin 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	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request
Status	LBD	SSR	Lin break field detection	ESCR:LBIE	Writing "0" to the SSR:LBD
	SFD	SACSR	SyncField is detected	SACSR: SFDE	Writing "0" to SyncField 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 *2	ICP	ICS	1st falling edge of Lin Sync Field	ICS:ICE0	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Sync Field		

*1: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

*2: An interrupt does not generate by Input Capture when the auto baud rate adjustment function is enable (SACSR.AUTE="1").

35.7.1.2 Reception Interrupts and Flag Setting Timing

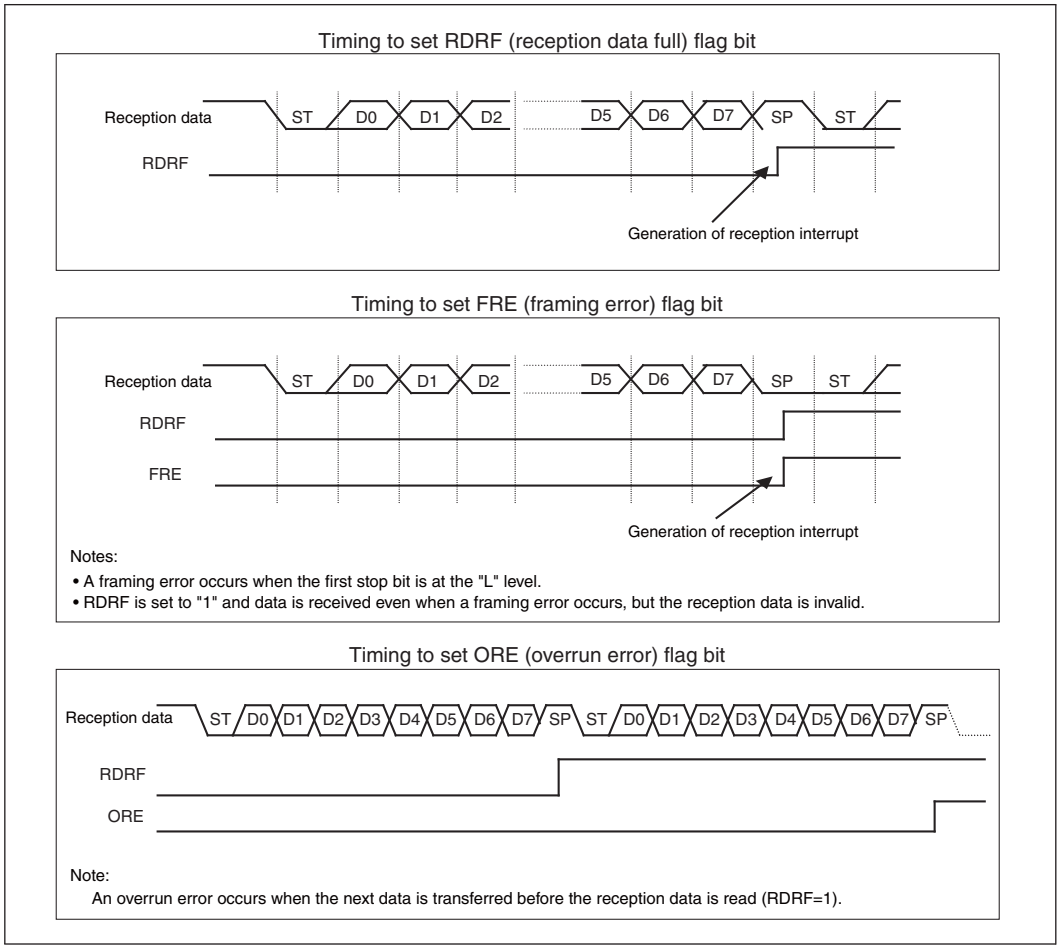
Reception interrupts occur when the reception is completed (SSR:RDRF), when a reception error occurs (SSR:ORE, FRE), or when Lin break field is detected.

■ 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 35.7-1 Timing of Flag Bit Setting

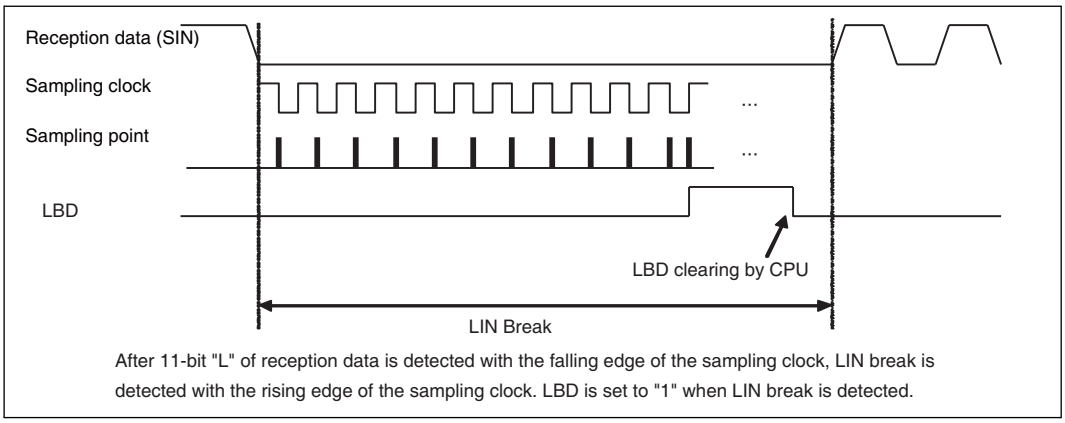


(Note) If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the data may not be received with the edge disabled. It is recommended to leave a space between frames if successive frames are to be output.

■ 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 35.7-2 Timing of LIN Break field Flag (LBD) Setting



35.7.1.3 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.

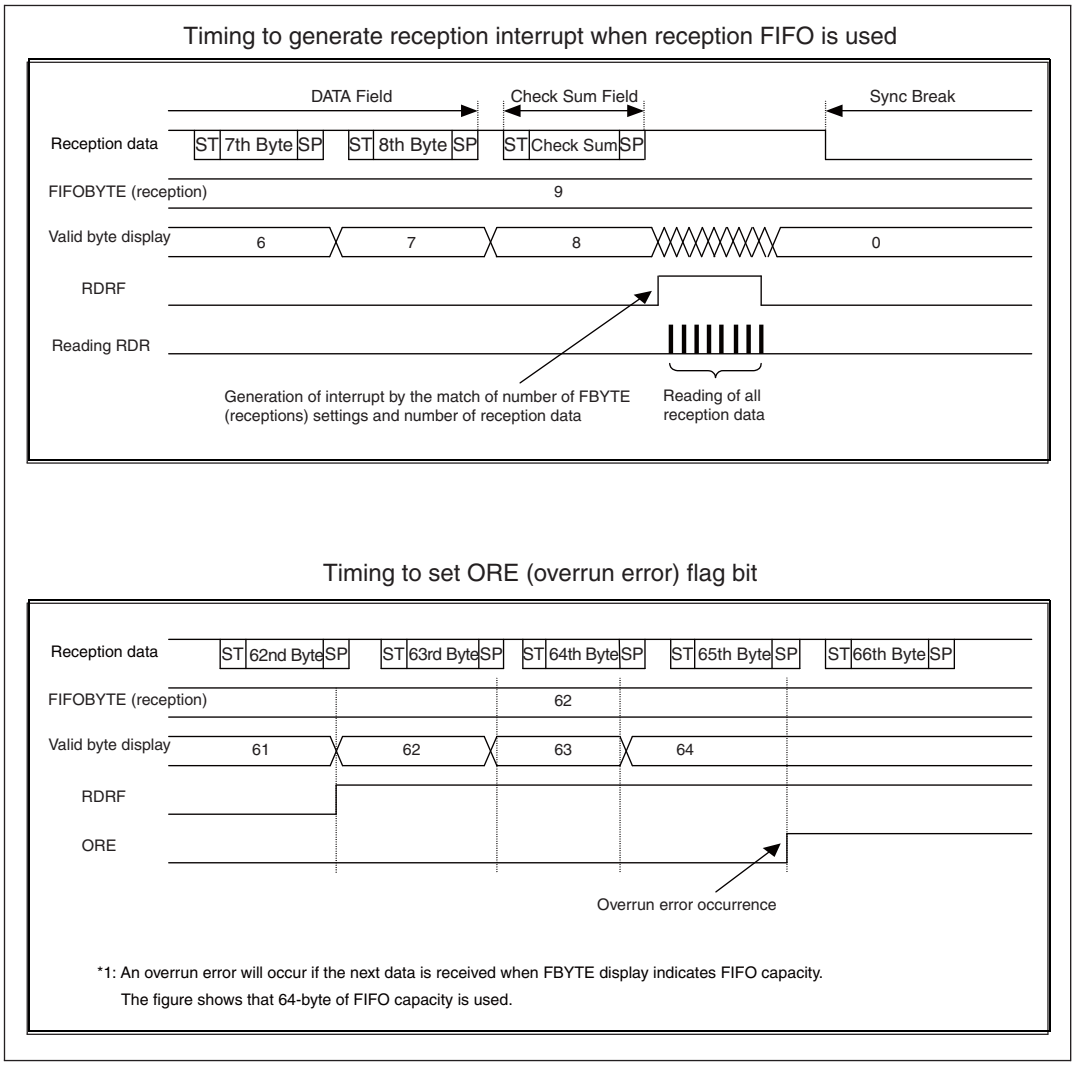
■ Reception Interrupts when Using Reception FIFO and Flag Setting Timing

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="1") 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 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 35.7-3 Timing of Interrupt Generation



35.7.1.4 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).

■ Transmission Interrupts and Flag Setting Timing

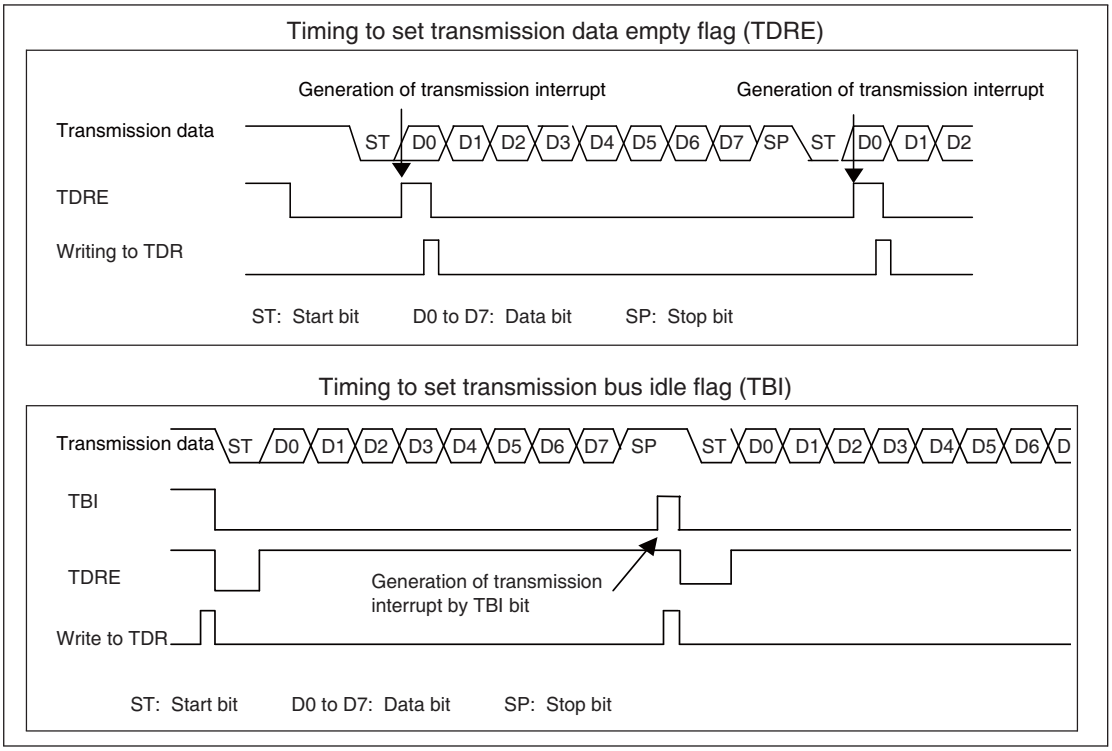
● Timing of transmission data empty flag (TDRE) setting

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 is a read-only bit. Writing data to the transmission data register (TDR) clears the SSR:TDRE bit to "0".

● Timing of transmission bus idle flag (TBI) setting

When the transmit data register is empty (SSR:TDRE="1") and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE="1") at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the TBI bit and the transmission interrupt request will be cleared.

Figure 35.7-4 Timing of TDRE and TBI Setting



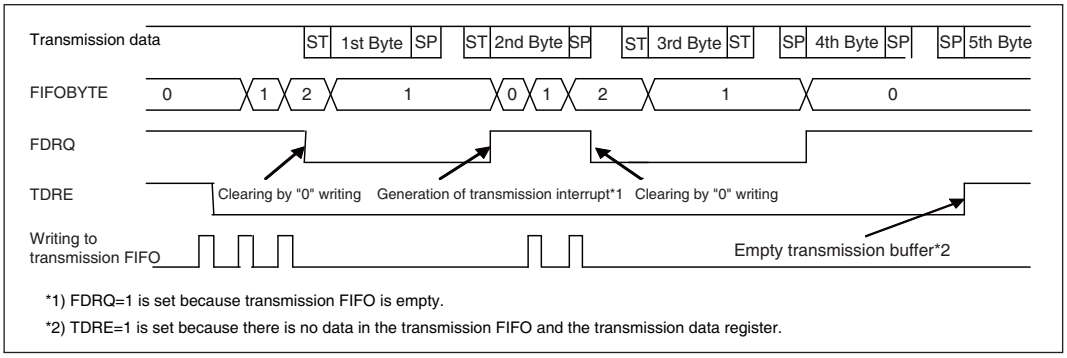
35.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

When the transmission FIFO is used, an interrupt will occur if there is no data in the transmission FIFO.

■ Transmission Interrupts When Using Transmission FIFO and Flag Setting Timing

- When there is no data in the transmission FIFO, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt occurs.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE).
  When FBYTE=0x00, there is no data in the transmission FIFO.

Figure 35.7-5 Timing of Transmission Interrupts when Using Transmission FIFO

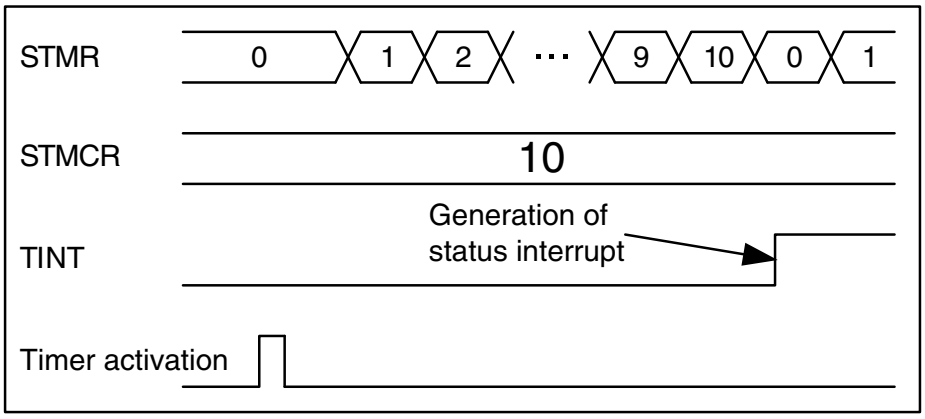


35.7.1.6 Timer Interrupt and Flag Setting Timing

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 35.7-6 Timer Interrupt Timing

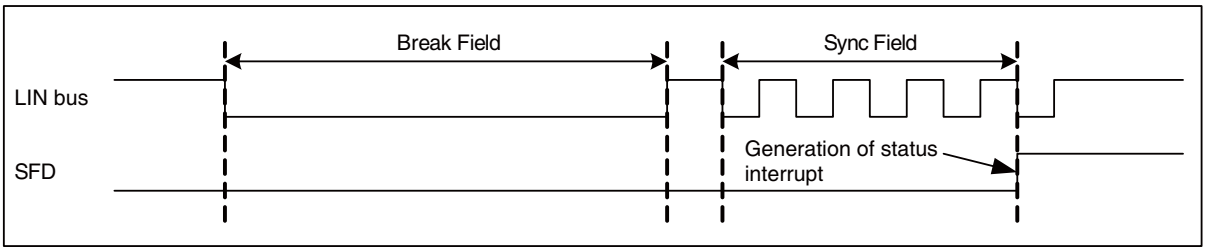


35.7.1.7 SyncField Detection Interrupts and Flag Setting Timing

SyncField detection interrupt occurs when the detection of SyncField is completed.

- When the auto baud rate adjustment is enabled (SACSR:AUTE="1") and the fifth falling of LIN bus on SyncField is detected, the SyncField detection flag (SACSR:SFD) is set to "1".
  If the SyncField interrupt is enabled (SACSR:SFDE="1") at this time, a status interrupt occurs.

Figure 35.7-7 SyncField Detection Interrupt Timing



35.7.2 Operation of LIN Interface (v2.1)

The LIN interface (v2.1) operates for the master/slave bidirectional LIN communication.

35.7.2.1 Master Operations

■ Selecting Master Operation

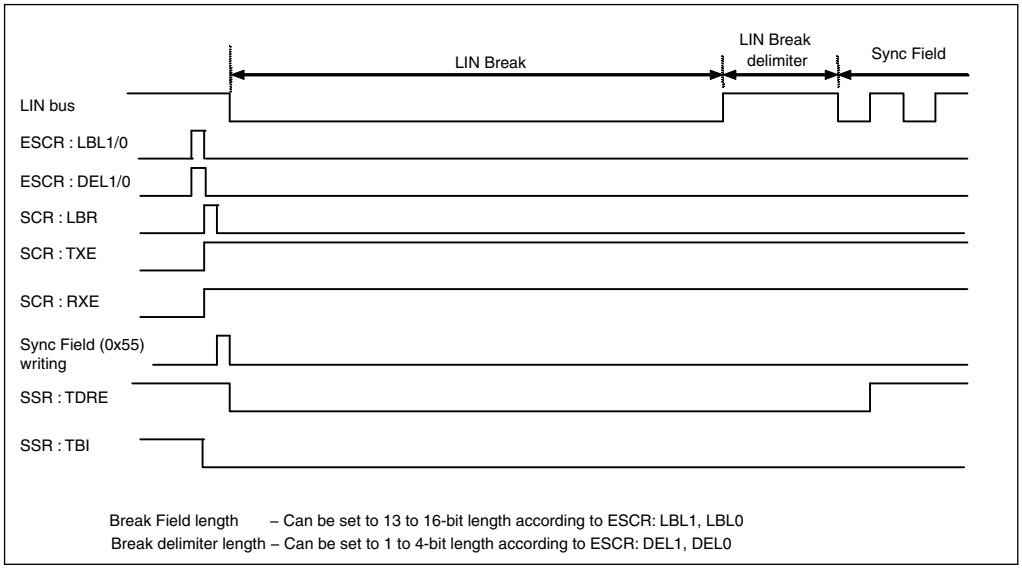
To make the LIN interface (v2.1) work as the master device, set the SCR:MS bit to "0".

■ From LIN Break Field Transmission to Sync 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 the transmission operation (SCR:TXE=1) and setting "1" in the SCR:LBR bit (LIN Break field setting bit).
- The Sync 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 Break field part is not received.

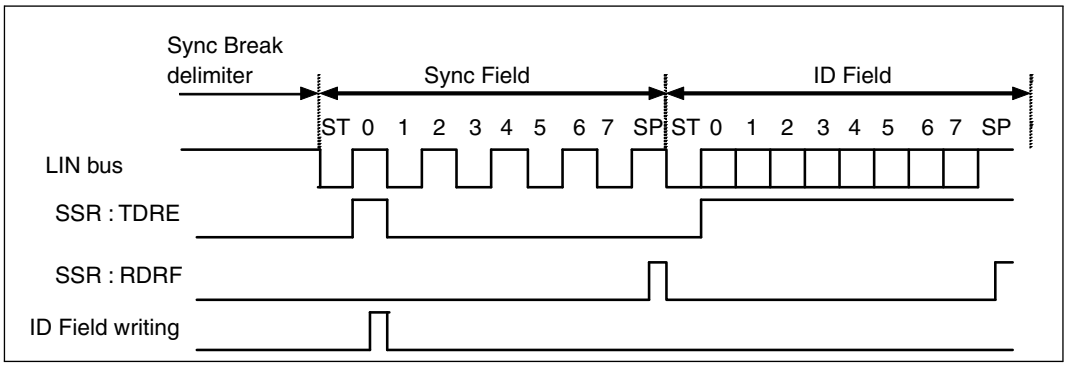
Figure 35.7-8 From LIN Break Field to Sync Field Transmission



■ From Sync Field Transmission to ID Field Transmission

- When the first bit of the Sync Field (0x55) is transmitted, the SSR:TDRE (transmission data empty) bit is set to "1". If the transmission interrupt is enabled (SCR:TIE="1") at this time, a transmission interrupt occurs.
- When this interrupt occurs, the ID Field can be written to the transmit data register (TDR).
- When a reception interrupt occurs, the received data will be compared with the transmitted data to confirm that no error has occurred.
- The ID Field is output in an LSB-first fashion with a data length of 8 bits.

Figure 35.7-9 From Sync 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.

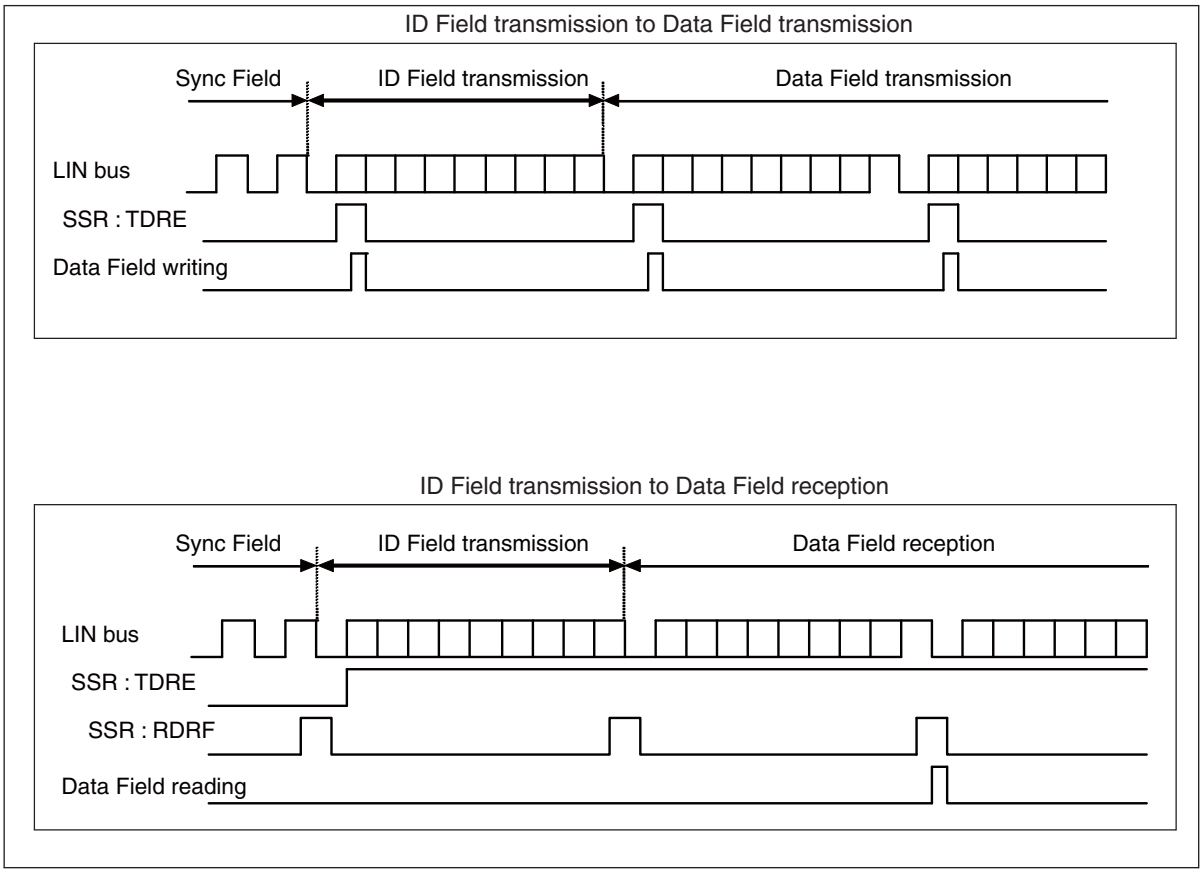
● In the case of Data Field transmission:

- 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 35.7-10 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, when the error occurs due to adding the checksum of data at the end, send it again).
- 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.

■ Master Operation Timing Chart (FIFO Unused)

Figure 35.7-11 LIN Bus Timing (at the Time of Data Field Transmission without Using FIFO)

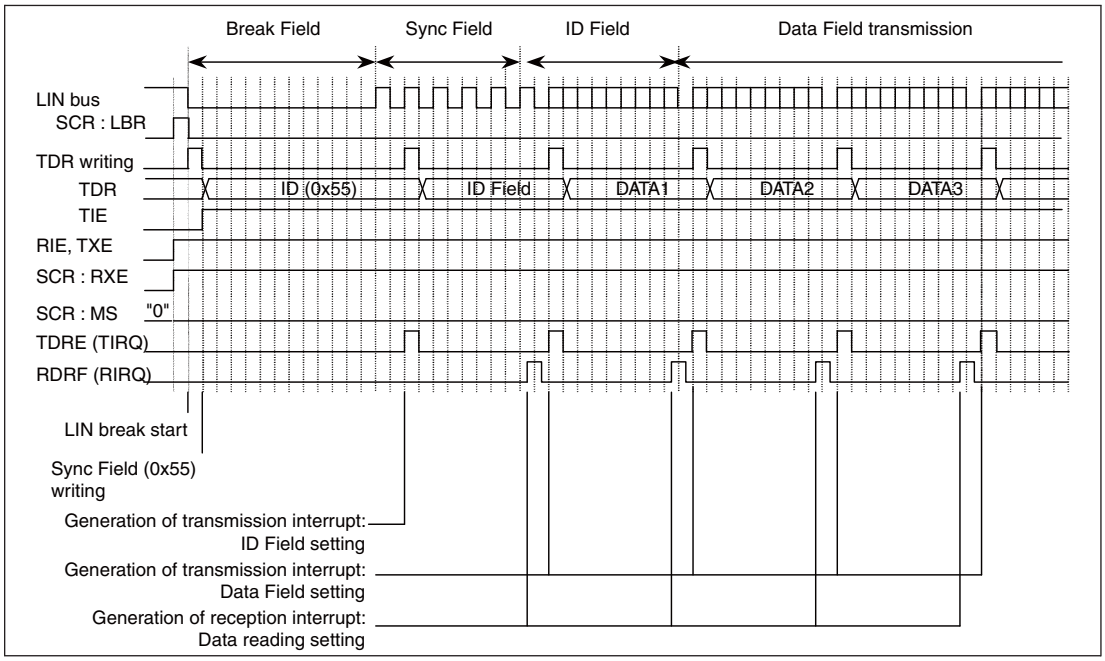
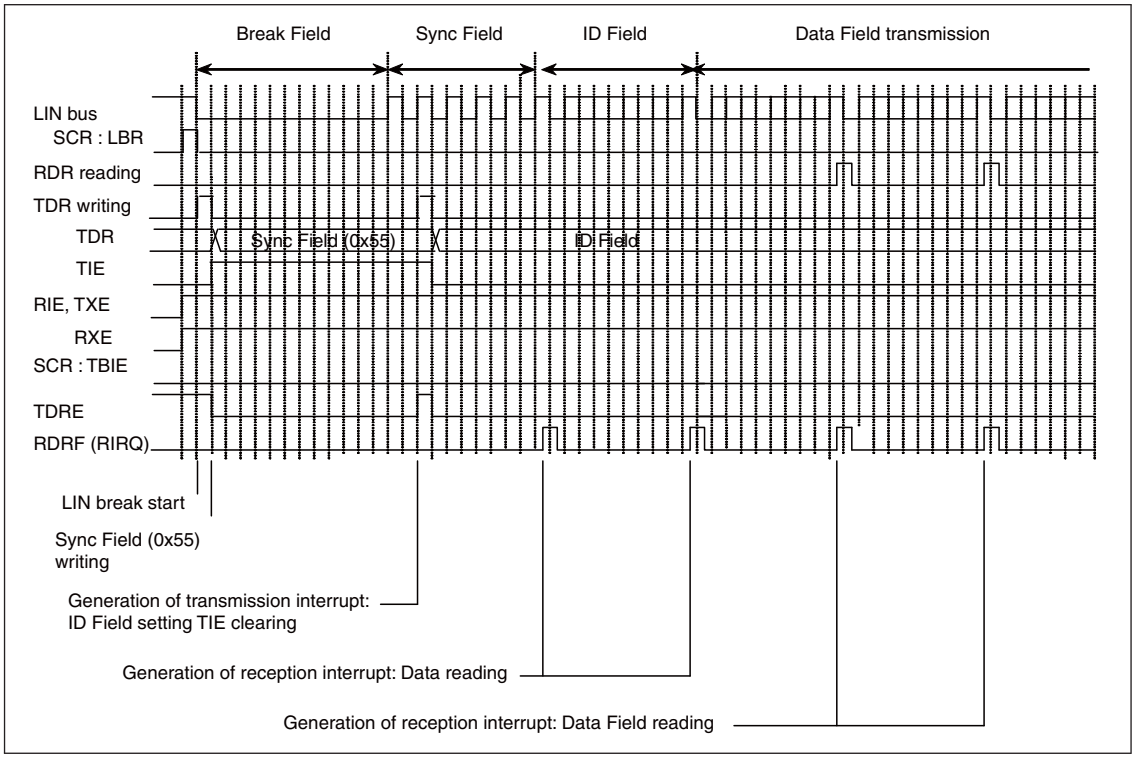


Figure 35.7-12 LIN Bus Timing (at the Time of Data Field Reception without Using FIFO)



■ Master Device Operation Timing Chart (FIFO Used)

Figure 35.7-13 LIN Bus Timing (at the Time of Data Field Transmission when Using FIFO)

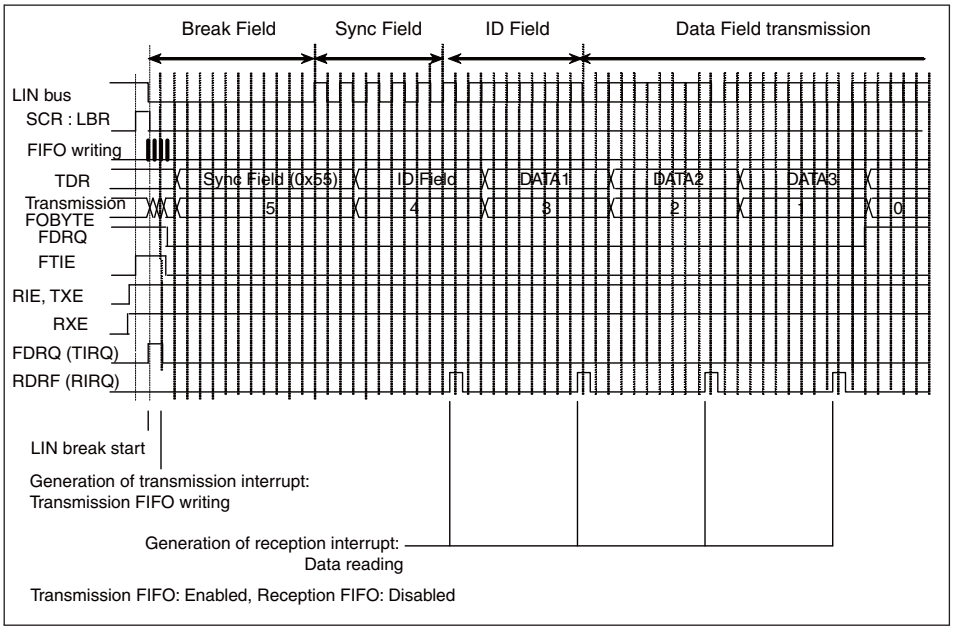
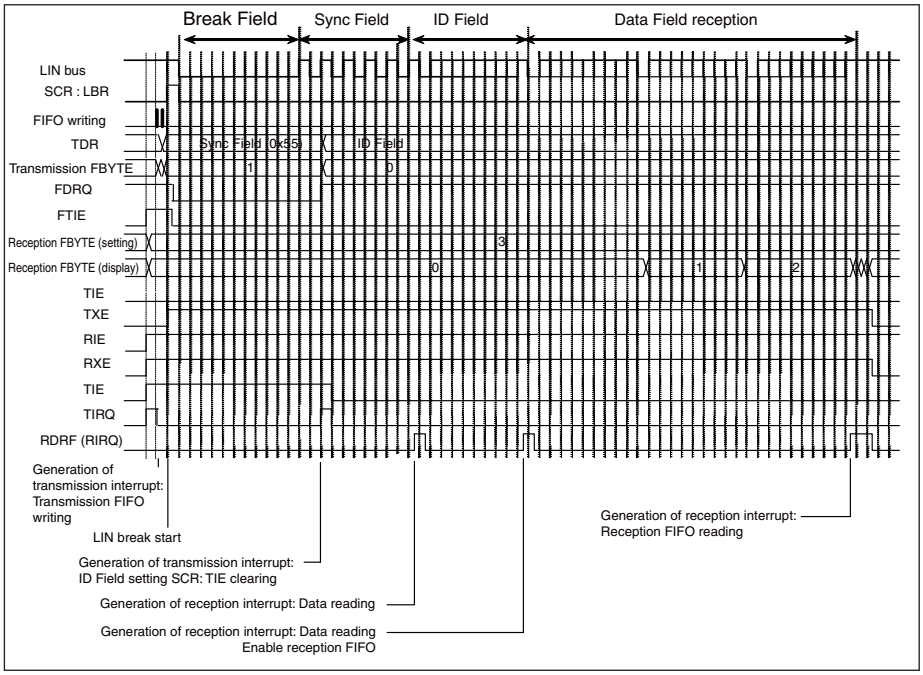


Figure 35.7-14 LIN Bus Timing (at the Time of Data Field Reception when Using FIFO)



35.7.2.2 Slave Operations

■ Selecting Slave Operation

To make the LIN interface (v2.1) work as the slave device, set the SCR:MS bit to "1".

■ Break Field Reception to Sync Field Reception

(1) Enable automatic baud rate adjustment (SACSR:AUTE="1").

(2) When LIN Break Field is input, LIN Break Field will be detected (SSR:LBD="1") at eleventh bit.
At this time, when ESCR:LBIE bit is set to "1", a reception interrupt will be generated.

(3) When the first falling edge of Sync Field is detected by the LIN interface (v2.1), serial timer register (STMR) will be initialized to "0".

(4) When fifth falling edge of Sync field is detected, sync field detection flag (SACSR:SFD) will be set to "1".

(5) When 5th falling edge of the Sync field is detected, the Sync field detection flag (SACSR:SFD) is set to "1". At this time, the following is checked to confirm automatic baud rate adjustment.

- If automatic baud rate adjustment is performed when the Sync field is detected (SACSR:SFD="1"), the value read from the serial timer register (STMR) is equal to the value read from the baud rate generator register (BGR).
- If automatic baud rate adjustment is not performed when the Sync field is detected (SACSR:SFD="1"), the value read from the serial timer register (STMR) is not equal to the value read from the baud rate generator register (BGR).

Figure 35.7-15 LIN Break Field Reception to Sync Field Reception (in case where STMR is SFUR or smaller and SFLR or larger)

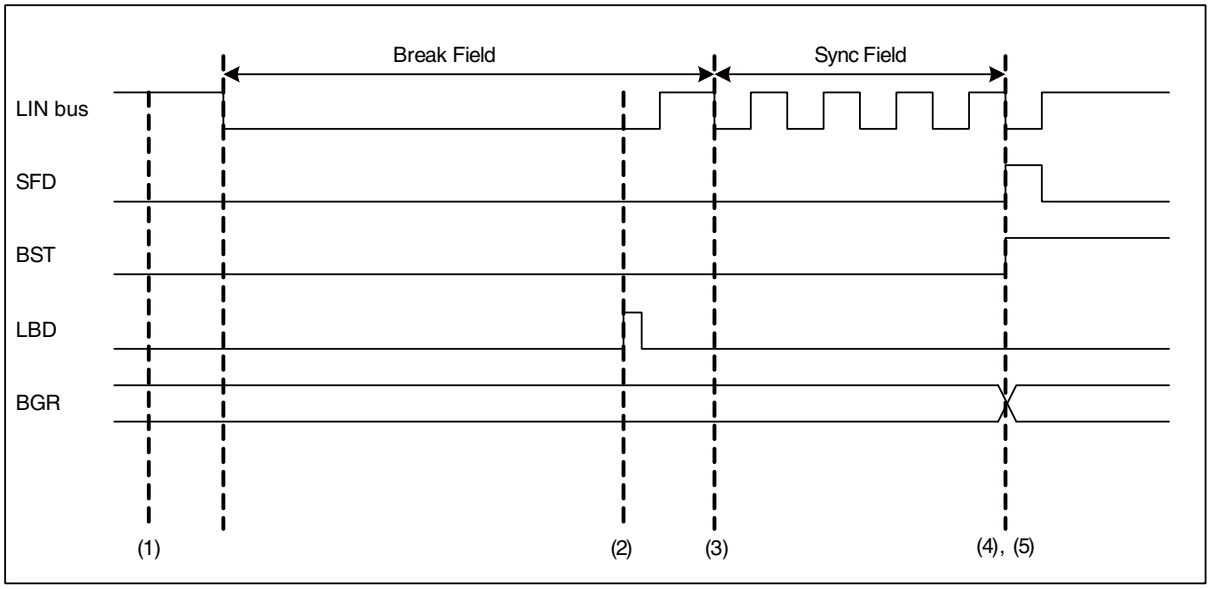
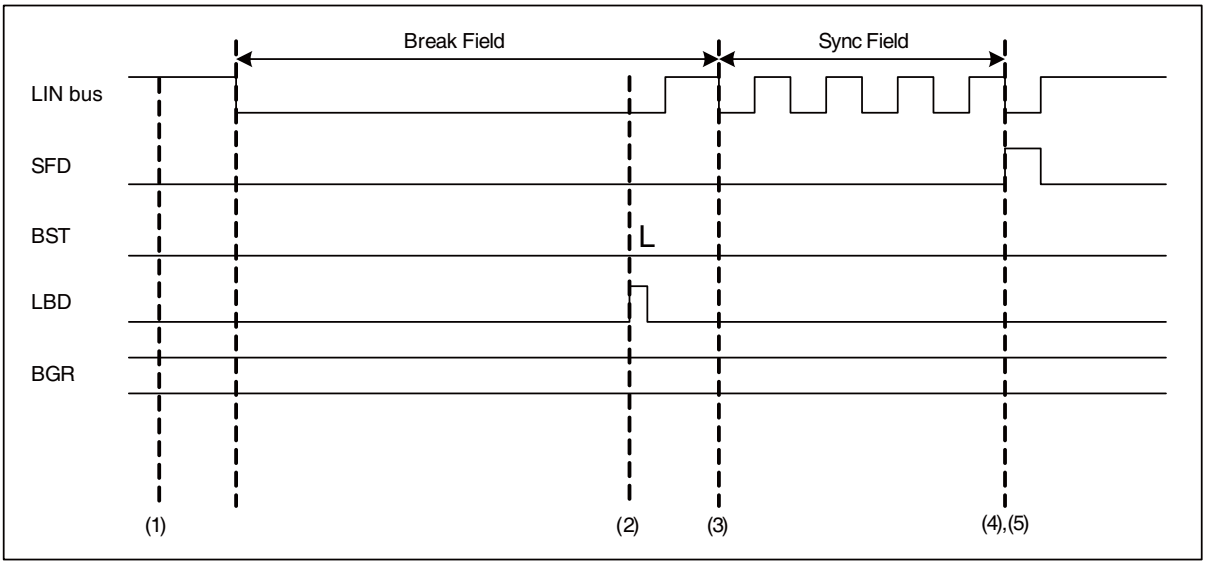


Figure 35.7-16 LIN Break Field Reception to Sync Field Reception (in case where STMR is larger than SFUR or smaller than SFLR)



(Note) While in LIN Break Field and Sync field, set reception disabled (SCR:RXE="0").

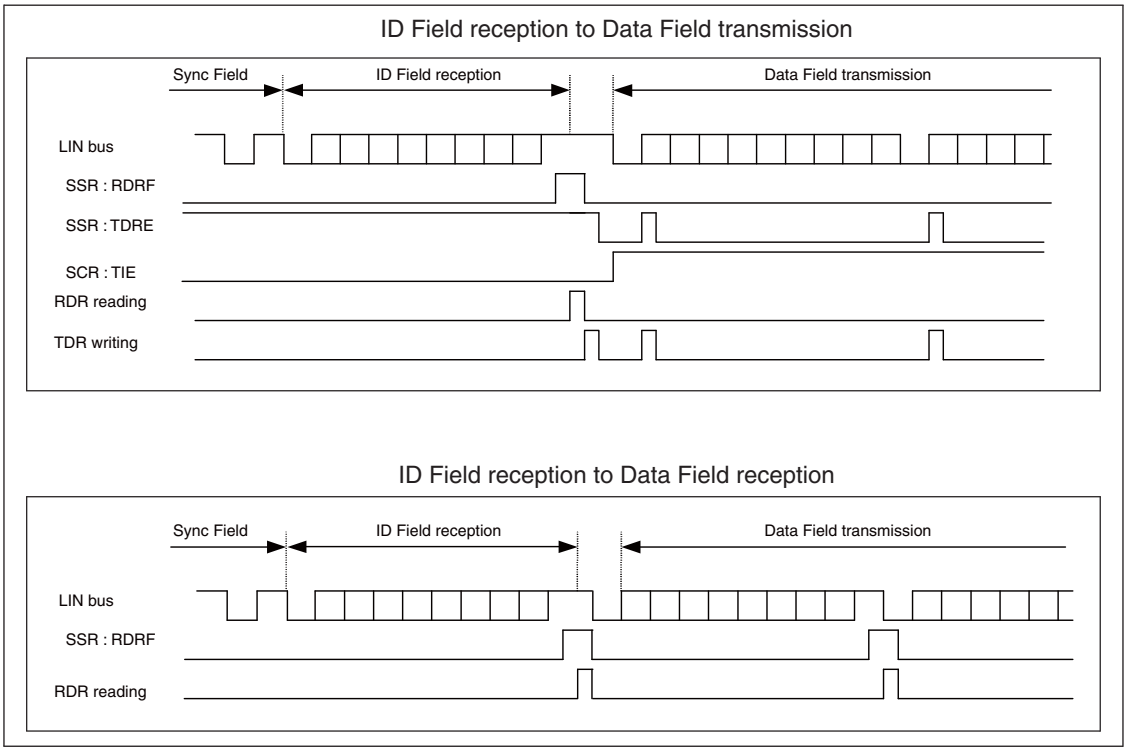
■ From ID Field Reception to Data Field Transmission/Reception

After the ID Field is received, you can select 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 35.7-17 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, when the error occurs due to adding the checksum of data at the end, send it again).
- 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.

■ Slave Operation Timing Chart

Figure 35.7-18 LIN Bus timing (DATA Field transmitted: FIFO unused, AUTE=1)

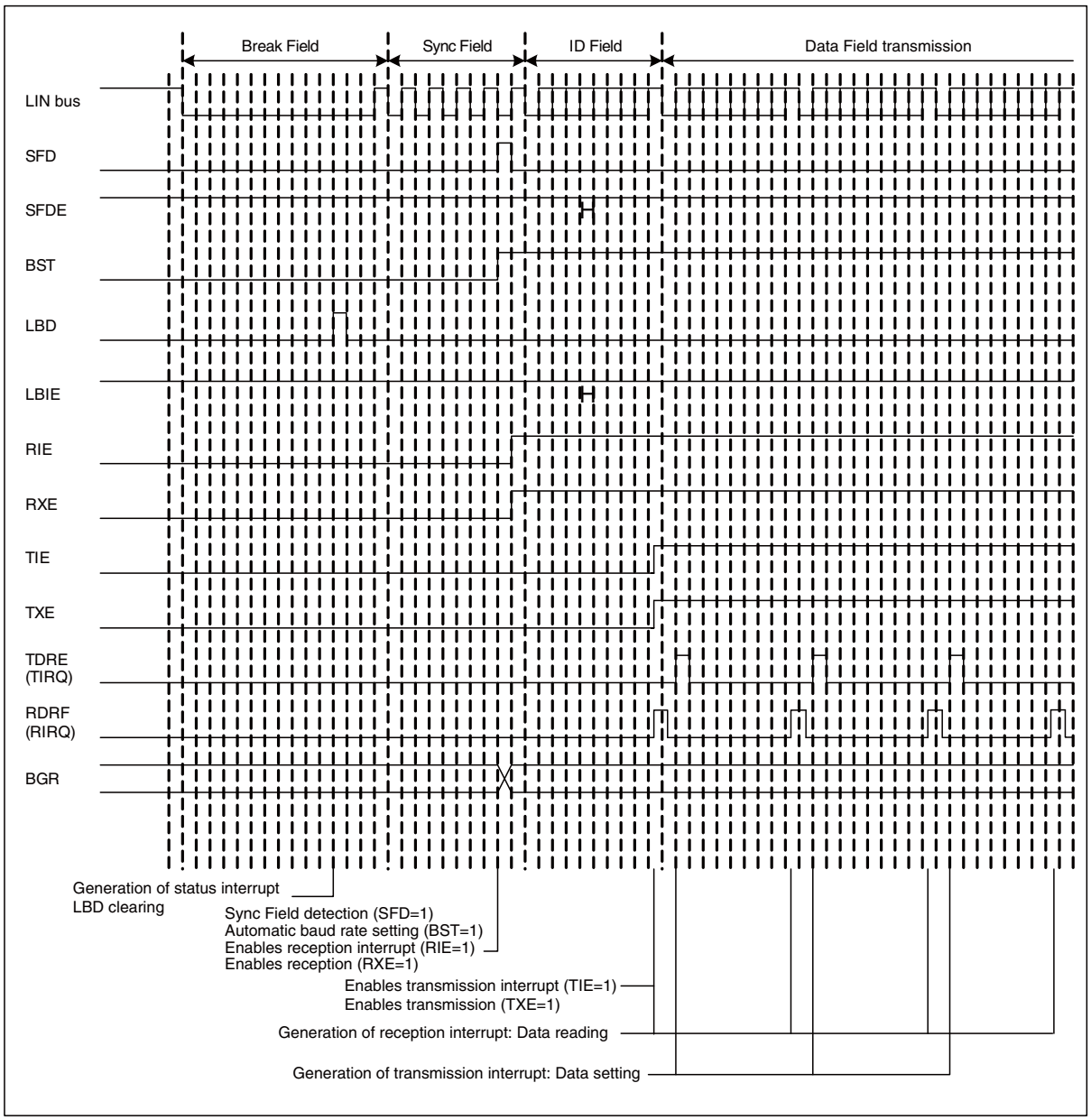
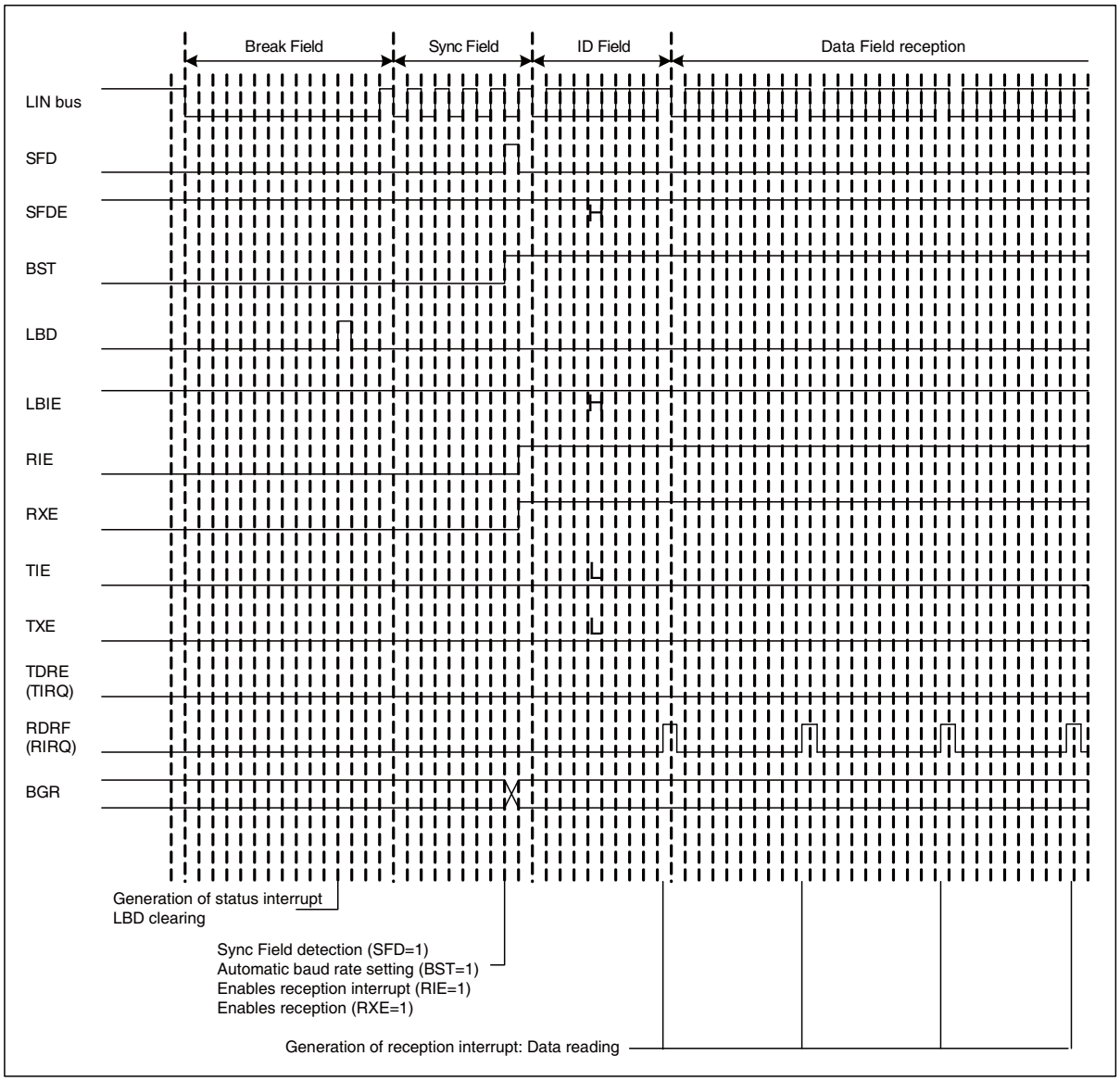


Figure 35.7-19 LIN Bus Timing (DATA Field received: FIFO unused, AUTE=1)



■ FIFO Used

Figure 35.7-20 LIN Bus Timing (DATA Field transmitted: FIFO used, AUTE=1)

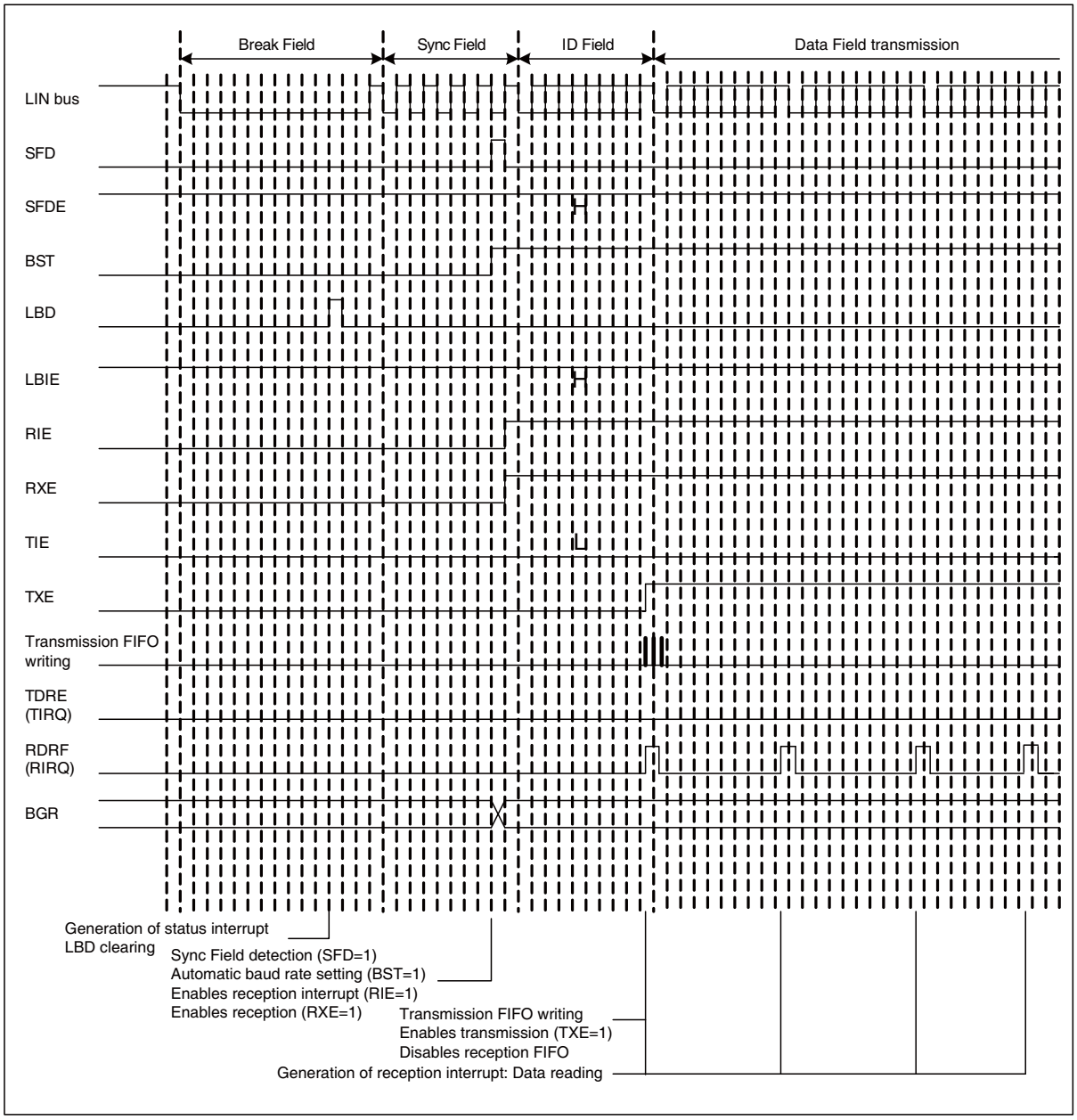
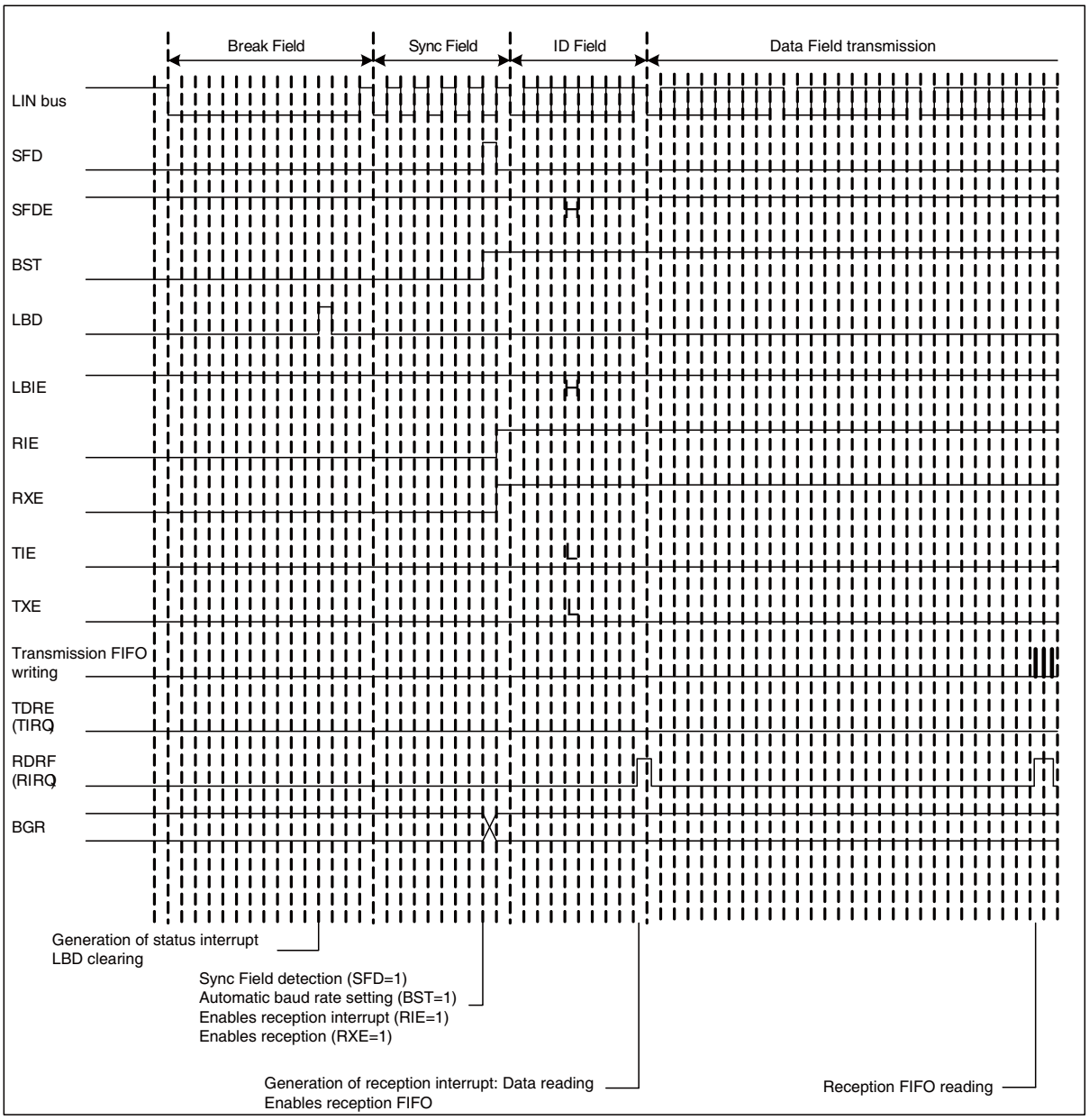


Figure 35.7-21 LIN Bus Timing (DATA Field received: FIFO used, AUTE=1)



35.7.2.3 Serial Timer Operation

The serial timer can use the timer function.

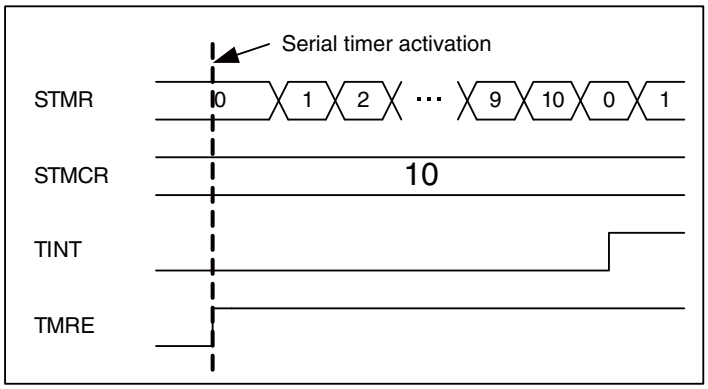
● How to Start the Serial Timer

There are 2 methods for starting the serial timer: one method sets the serial timer enable bit (SACSR:TMRE) to "1", and the other method uses the Sync field.

• Starting using the serial timer enable bit (SACSR:TMRE)

Setting the serial timer enable bit (SACSR:TMRE) to "1" starts the serial timer, and the serial timer register (STMR) starts counting from "0".

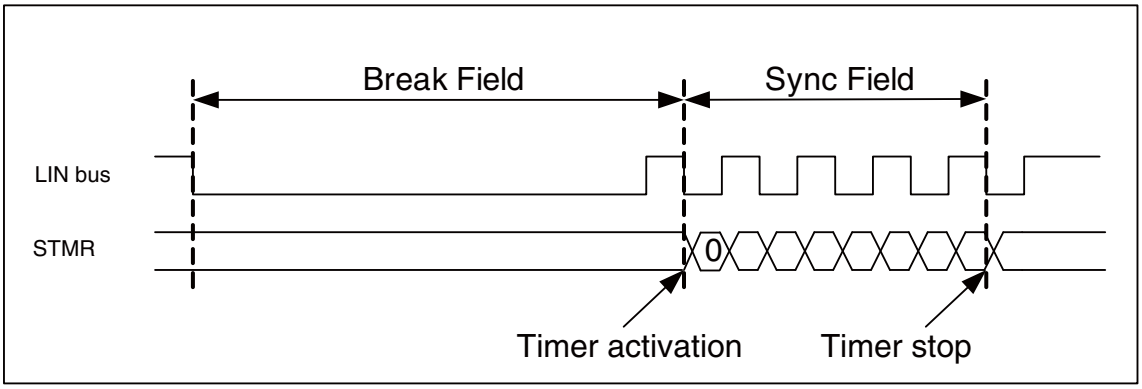
Figure 35.7-22 Starting Using the Serial Timer Enable Bit (STMCR="10", SACSR:TSYNE="0"))



• Starting with reception of the Sync field

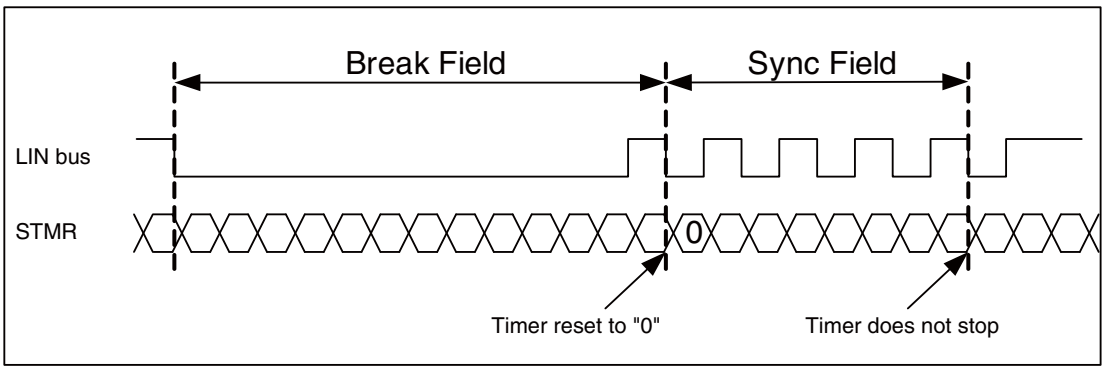
When the serial timer is stopped and the automatic baud rate adjustment bit (SACSR:AUTE) is "1", the detection of the 1st falling edge of the Sync field by the LIN interface (v2.1) starts the serial timer. Then, the serial timer register (STMR) starts counting from "0".

Figure 35.7-23 Starting with Sync Field Reception during a Serial Timer Stop (SACSR:AUTE="1", TMRE="0")



If the serial timer is running and the automatic baud rate adjustment bit (SACSR:AUTE) is "1" when the LIN interface (v2.1) detects the 1st falling edge of the Sync field, then the serial timer register (STMR) starts counting from "0".

Figure 35.7-24 Starting with Sync Field Reception during Serial Timer Operation (SACSR:AUTE="1", TMRE="1")



● How to Stop the Serial Timer

If any of the following conditions is met, the serial timer stops.

- When the automatic baud rate adjustment bit (SACSR:AUTE) is "0", resetting the serial timer enable bit (SACSR:TMRE) to "0" stops the serial timer. At this time, the value in the serial timer register (STMR) is retained.
- When the automatic baud rate adjustment bit (SACSR:AUTE) is "1" and the serial timer enable bit (SACSR:TMRE) is "1", resetting the serial timer enable bit (SACSR:TMRE) to "0" anytime except during Sync field reception stops the serial timer. At this time, the value in the serial timer register (STMR) is retained.
- When the automatic baud rate adjustment bit (SACSR:AUTE) is "1" and the serial timer enable bit (SACSR:TMRE) is "0", the detection of the 5th falling edge of the Sync field by the LIN interface (v2.1) stops the serial timer. At this time, the value in the serial timer register (STMR) is retained.

(Note) If the automatic baud rate adjustment bit (SACSR:AUTE) is "1" and the serial timer enable bit (SACSR:TMRE) is "1", the serial timer does not stop and continues running even when the LIN interface (v2.1) detects the 5th falling edge of the Sync field.

● Timer Operation

When the serial timer register (STMR) and serial timer compare register (STMCR) have matching values, the timer interrupt flag (SACSR:TINT) is set to "1", and the serial timer register (STMR) is reset to "0".

Figure 35.7-25 Timer Operation (STMCR="10")

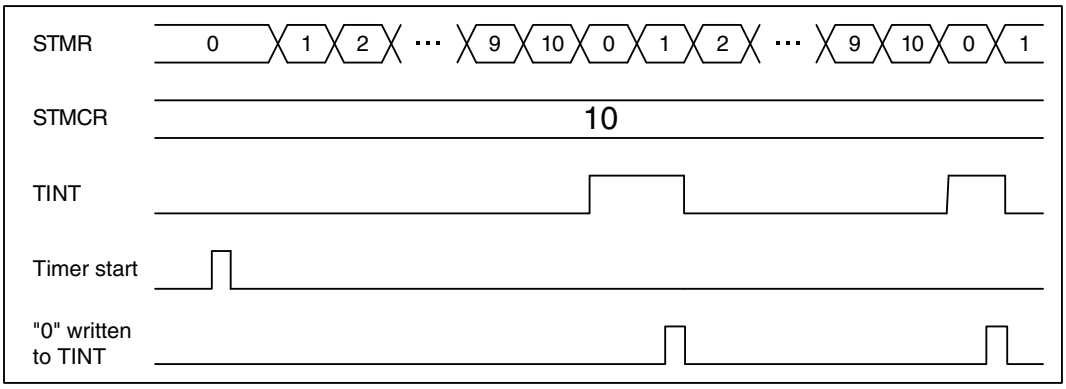


Figure 35.7-26 Flowchart of Serial Timer Initial Setting

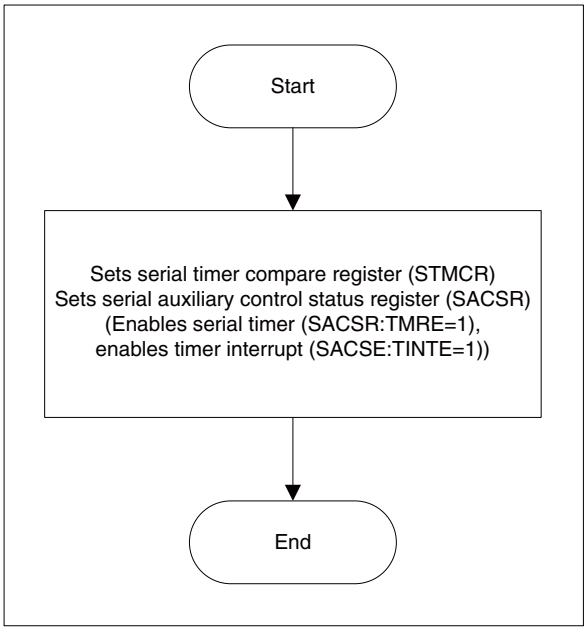
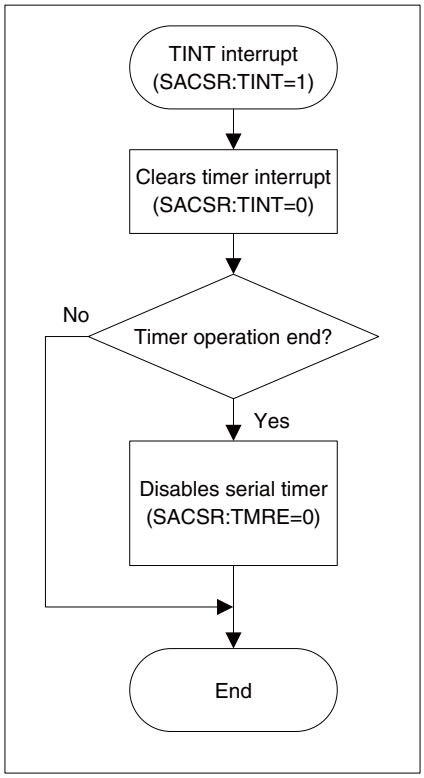


Figure 35.7-27 Flowchart of Serial Timer Interrupt Processing



(Notes)
- With "0000"h set in the serial timer compare register (STMCR), the timer interrupt flag (SACSR:TINT) is fixed at "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000"b while the timer is running.
- If the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1" when the Sync field is received, the serial timer register (STMR) is reset to "0".

35.7.2.4 Test Mode

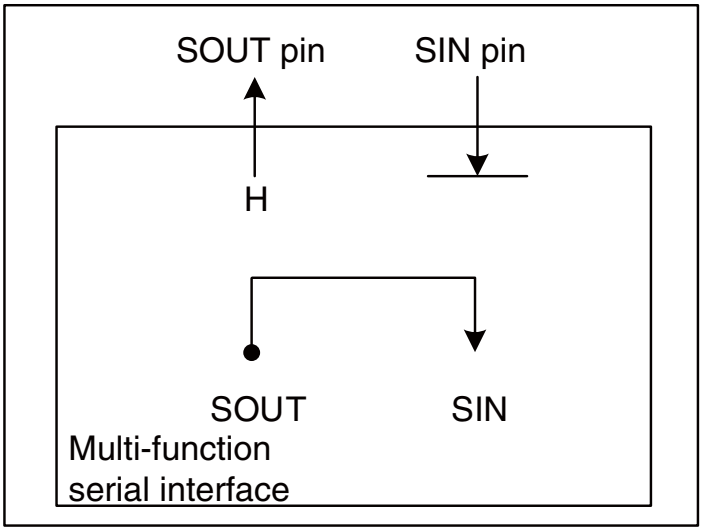
This section explains test mode operation.

● Serial Test Mode

When serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, so SIN can directly receive data transmitted by SOT.

When serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed at "H" and data input to the SIN pin is ignored.

Figure 35.7-28 Serial Test Mode



(Note) You can change the serial test mode enable bit (SACSR:STST) only when transmission and reception are disabled (SCR:TXE=RXE="0").

35.7.2.5 LIN Interface (v2.1) Baud Rate Selection/Setting

The LIN interface (v2.1) can use:

- Baud rates obtained when the dedicated baud rate generator (reload counter) divides the frequency of the internal clock
- Baud rates obtained when the dedicated baud rate generator (reload counter) divides the frequency of the external clock

The setting method is the same as for UART (mode 0/1). See Section "35.5.2.13 UART Baud Rate Selection/Setting".

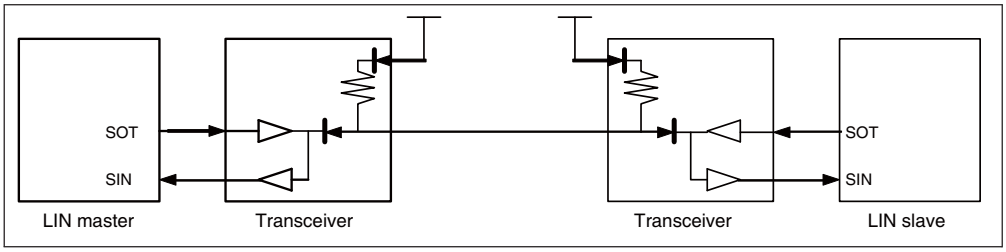
35.7.3 Setup Procedure and Program Flow

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

Figure 35.7-29 shows a communication system that contains 1 LIN master and 1 LIN slave. The LIN interface (v2.1) can work as a LIN master or LIN slave.

Figure 35.7-29 Example of LIN Bus System Communication



■ Flowchart Example

● Master Operation

Figure 35.7-30 Example of a Flowchart in LIN Communication Master Mode (without Using FIFO)

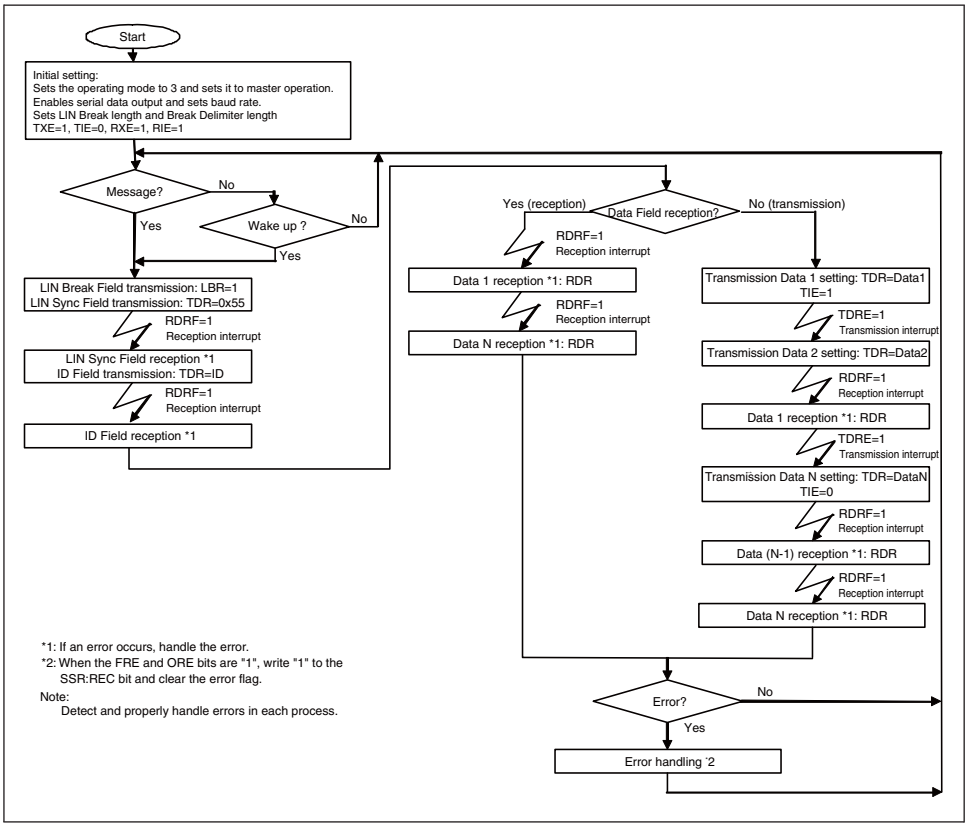
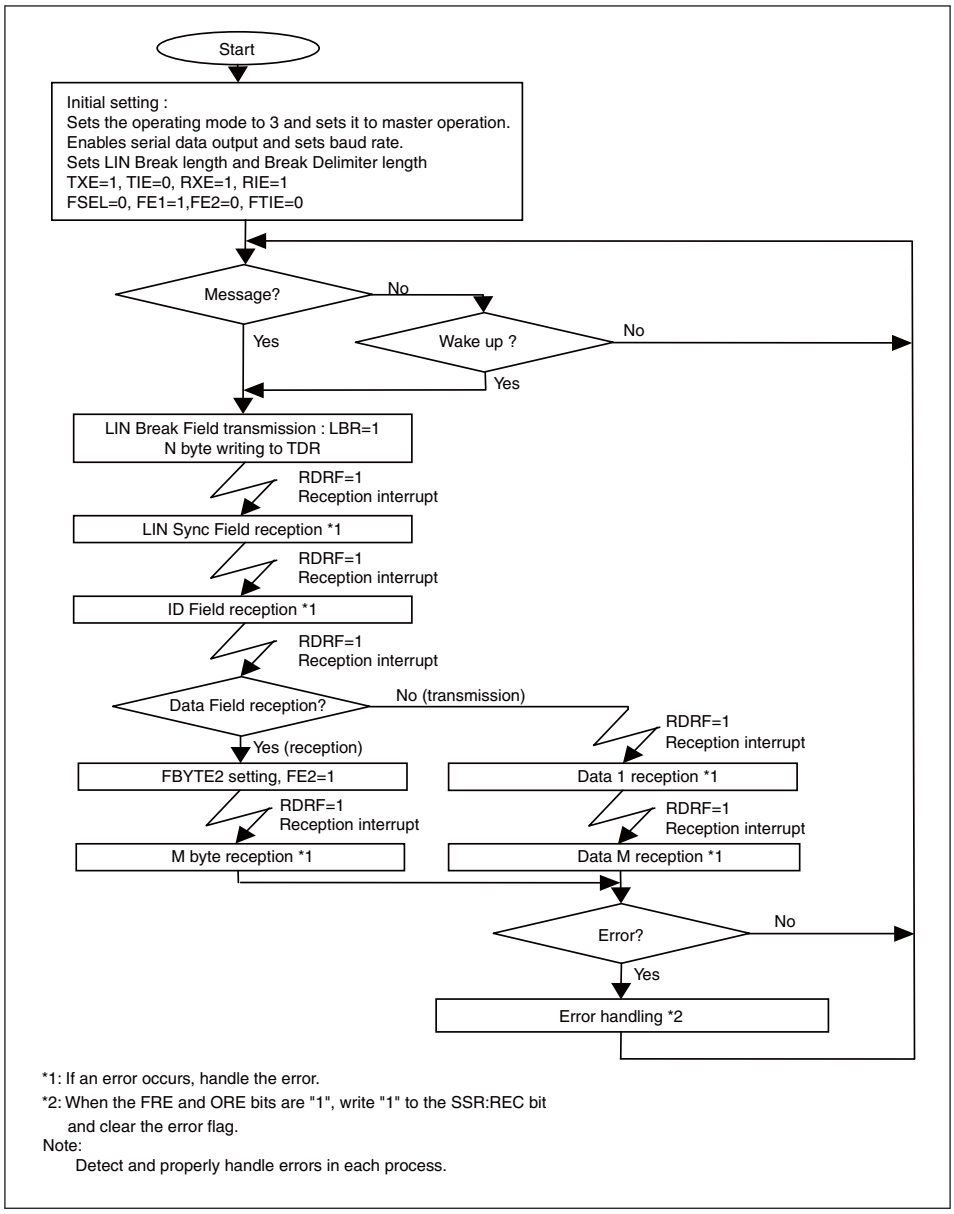


Figure 35.7-31 Example of a Flowchart in LIN Communication Master Mode (Using FIFO)



*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 35.7-32 Example of a Flowchart in LIN Communication Slave Mode (without Using FIFO, Auto Baud Rate Adjustment Enabled (SACSR:AUTE=1))

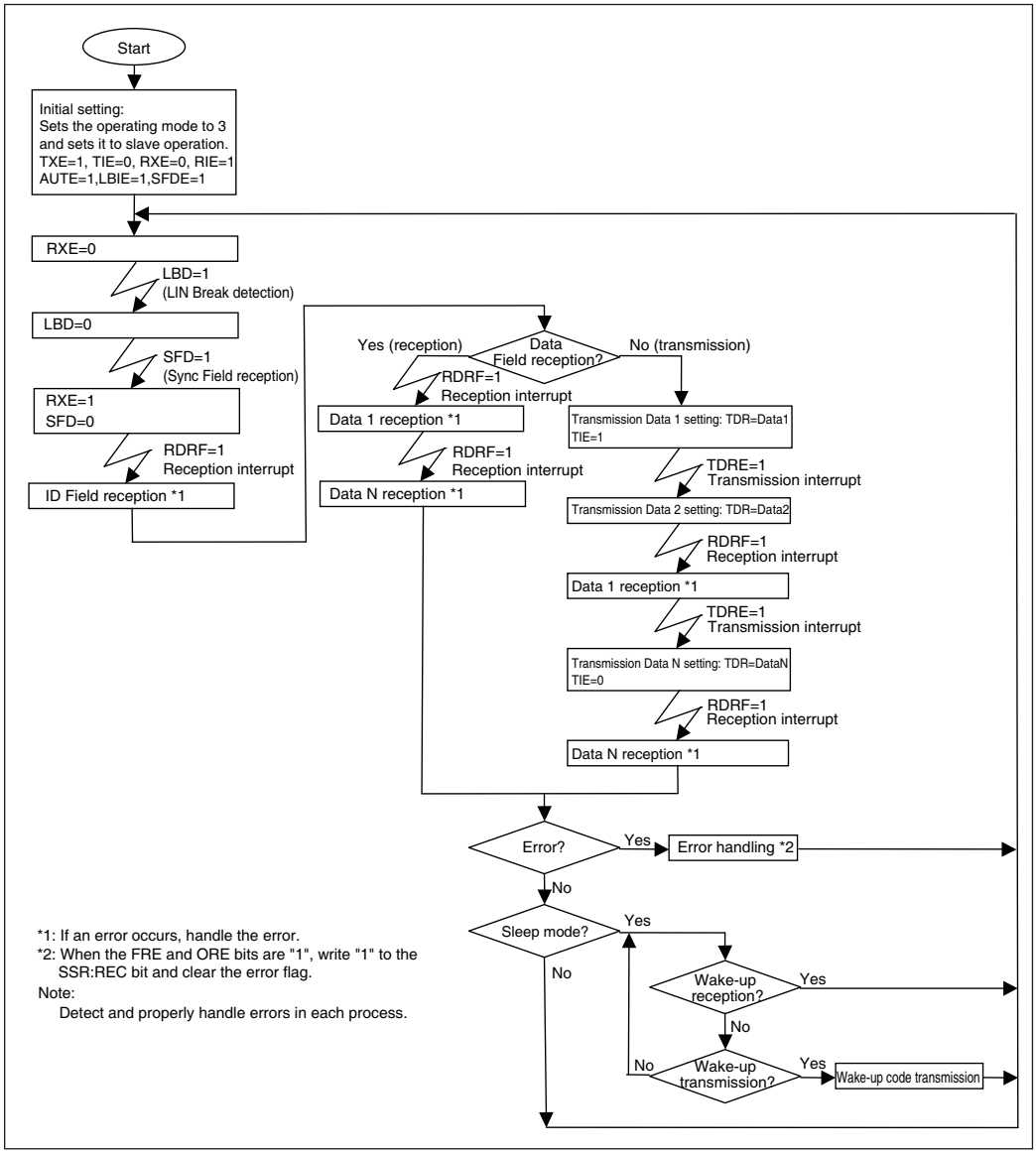
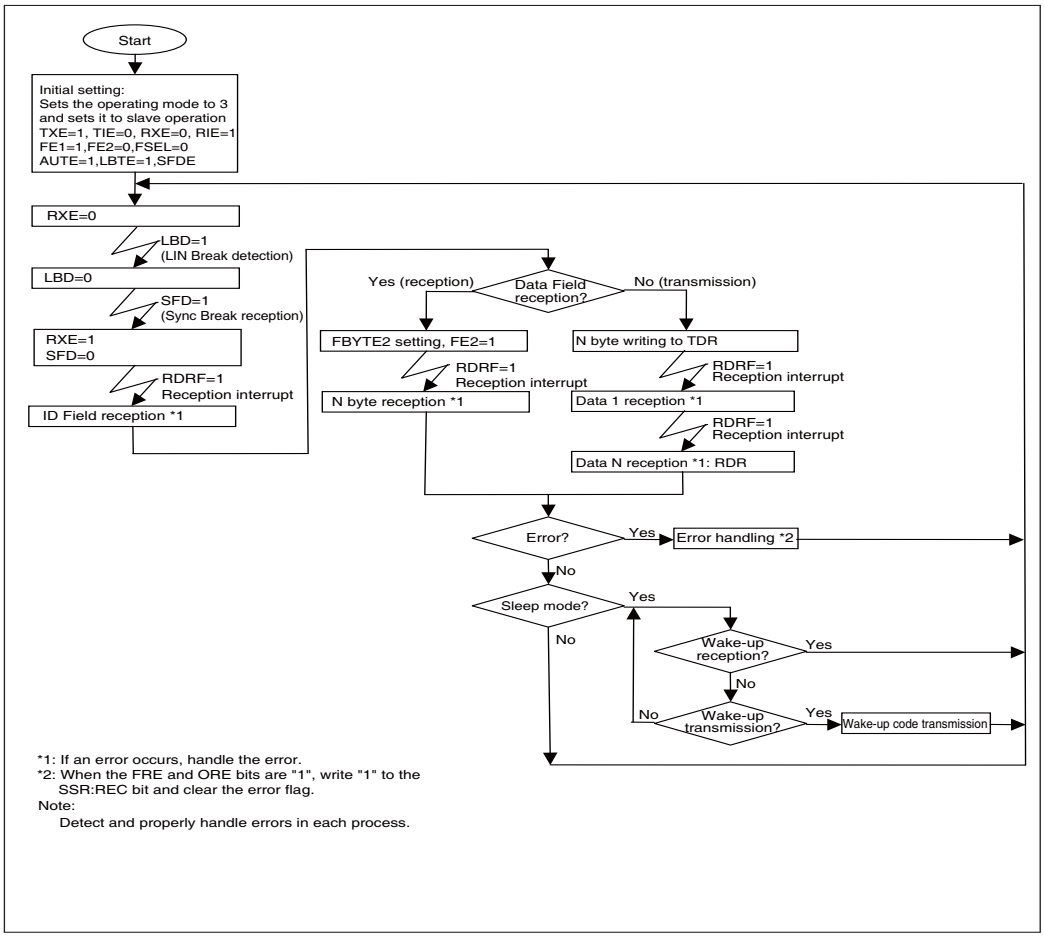


Figure 35.7-33 Example of a Flowchart in LIN Communication Slave Mode (Using FIFO)



35.8 Operation of I2C

35.8.1 Interrupts of I2C

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
- Matching values of the serial timer compare register (STMCR) and the serial timer register (STMR)

35.8.1.1 List of Interrupts of I2C Interface

Table 35.8-1 indicates how I2C interface interrupt control bits relate to interrupt factors.

Table 35.8-1 I2C Interface Interrupt Control Bits and Interrupt Factors (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Status	INT	IBCR	After transmission and reception of the first byte *1 (except in case of master operation with SSR:DMA=1)	IBCR:INTE	Writing "0" to the interrupt flag bit (IBCR:INT)
			After data transmission and reception *1 (SSR:DMA=0)		
			Bus error detected		
			Arbitration lost detected		
			Reserved address detected		
			NACK reception		
			Reception FIFO full during slave reception (SSR:DMA=0)		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)

Table 35.8-1 I2C Interface Interrupt Control Bits and Interrupt Factors (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Transmission	TDRE	SSR	Transmission register is empty	SMR:TIE	Write to the transmit data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission) *2
			Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit or the transmission FIFO is full
	TBI (SSR: DMA="1")	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission) *3
			Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)		

*1: No interrupt occurs if normal data can be transmitted/received and SSR: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".

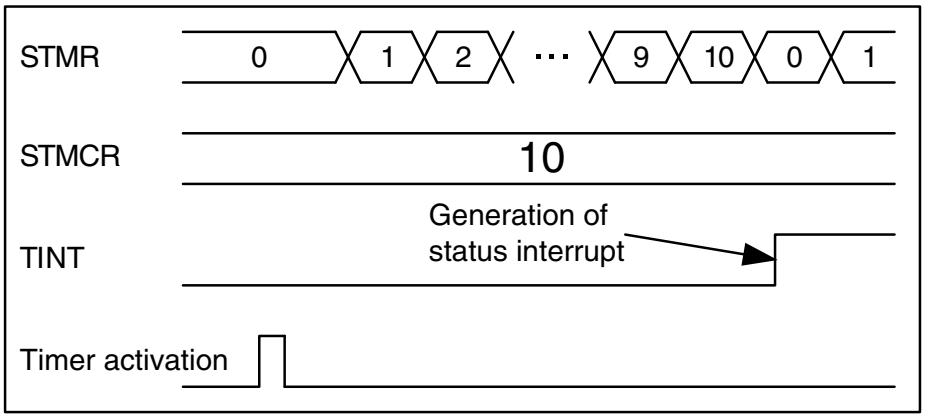
*3: Set the SSR:TBIE bit to "1" after the SSR:TBI bit is cleared to "0".

35.8.1.2 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 35.8-1 Timer Interrupt Generation Timing



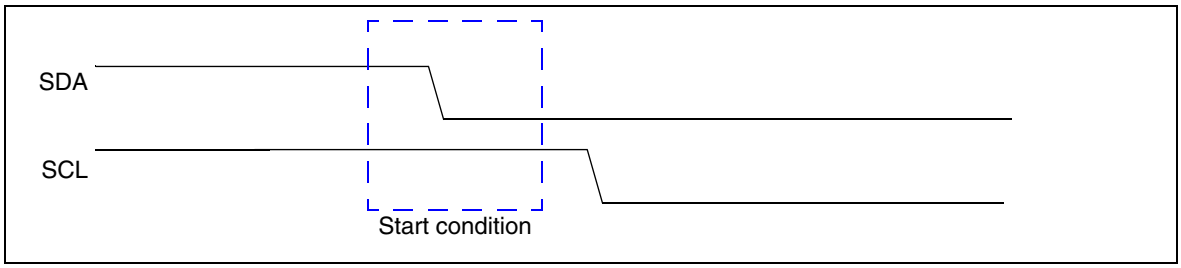
35.8.2 Operation for I2C Interface Communication

The I2C interface handles communication using two bidirectional bus lines, a serial data line (SDA), and a serial clock line (SCL).

35.8.2.1 I2C Bus Start Condition

The condition for the I2C bus to be activated is as follows:

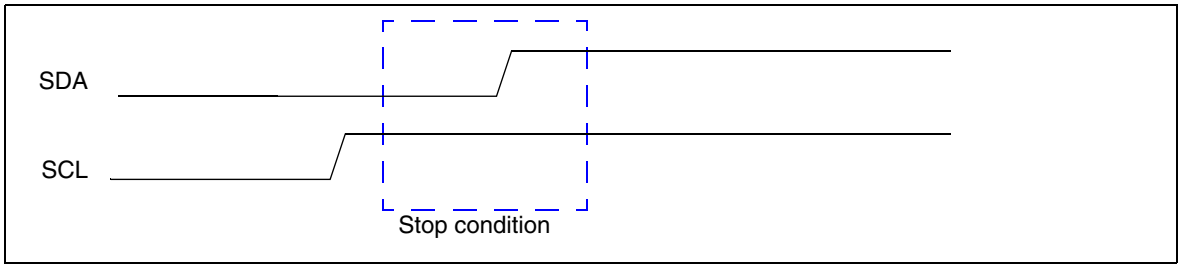
Figure 35.8-2 Start Condition



35.8.2.2 I2C Bus Stop Condition

The condition for the I2C bus to stop is as follows:

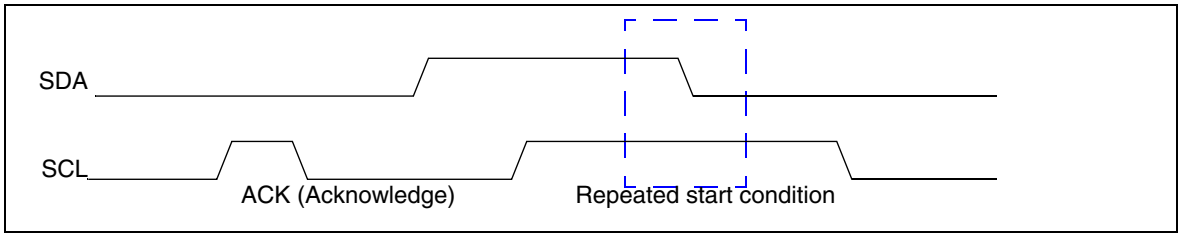
Figure 35.8-3 Stop Condition



35.8.2.3 I2C Bus Repeated Start Condition

The condition for the I2C bus to initiate a repeated start is as follows:

Figure 35.8-4 Repeated Start Condition



35.8.2.4 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".

35.8.2.5 Serial Timer Operations

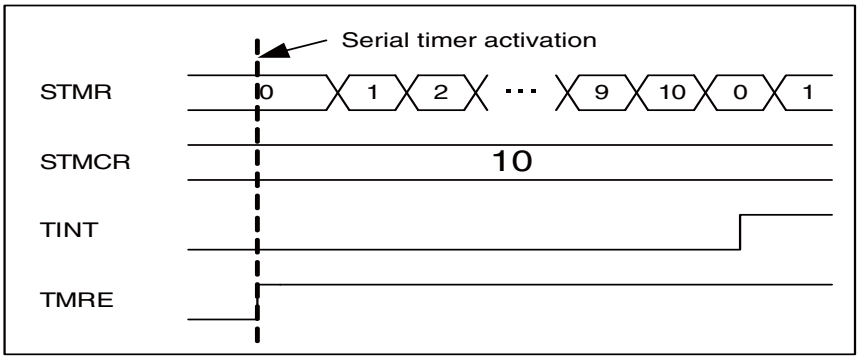
Serial timer can be used as a timer feature.

● Starting Serial Timer

To start the serial timer, set "1" in the serial timer enable bit (SACSR:TMRE).

Setting "1" to serial timer enable bit (SACSR:TMRE) will start the serial timer and the serial timer register (STMR) will start counting from "0".

Figure 35.8-5 Starting 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 35.8-6 Timer Operation (STMCR="10")

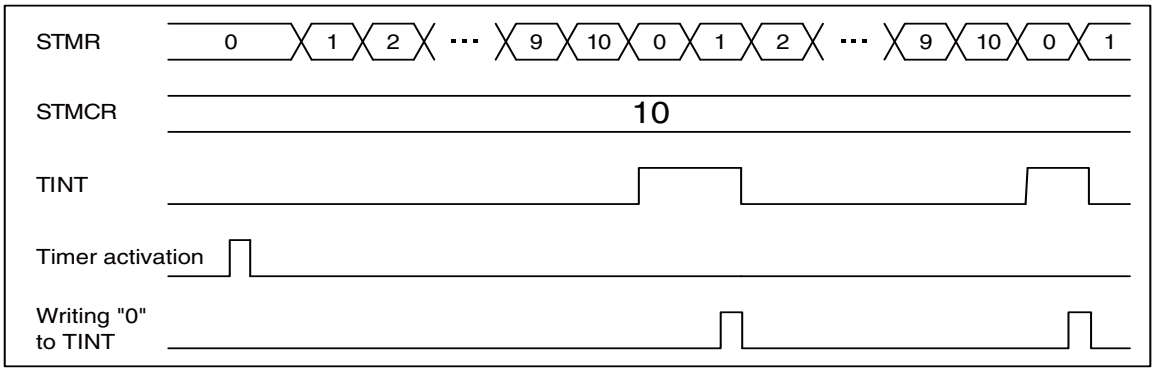


Figure 35.8-7 Flowchart of Serial Timer Initial Setting

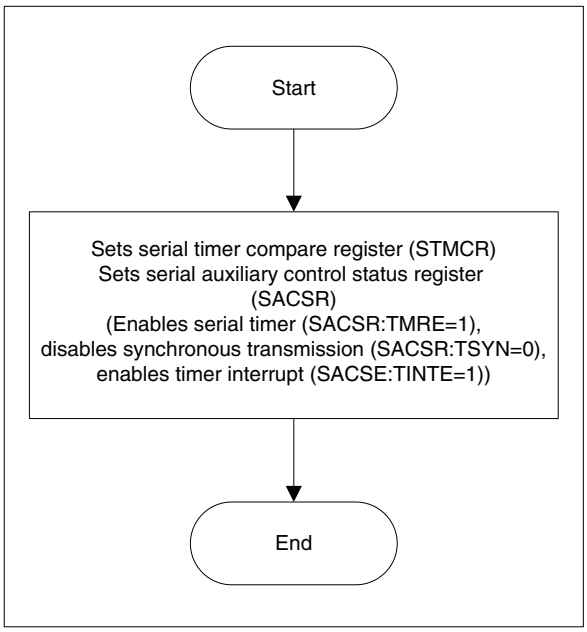
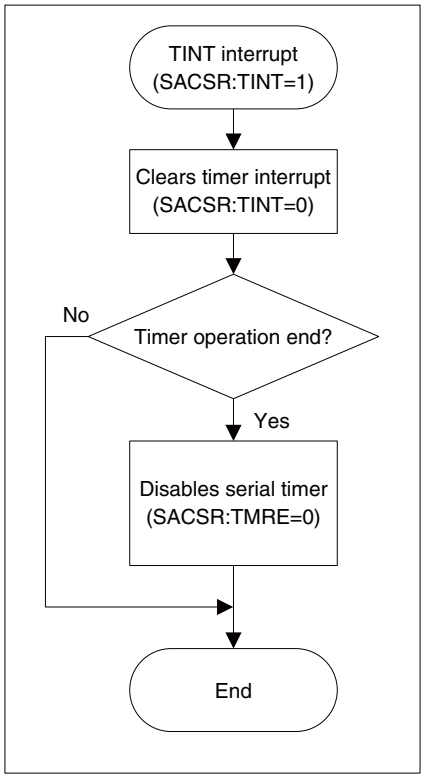


Figure 35.8-8 Flowchart of Serial Timer Interrupt Processing



(Note) With "0000"h set in the timer compare register (STMCR), the timer interrupt flag (SACSR:TINT) is fixed at "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000"b while the timer is running.

35.8.2.6 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:

$V = (16 \times 1{,}000{,}000) / 400{,}000 - 1 = 39$

Therefore, the baud rate is

$b = (16 \times 1{,}000{,}000) / (39+1) = 400\text{kbps}$

(Notes)
- Write to the baud rate generator (BGR) in 16-bit access mode.
- Configure the baud rate generator register (BGR) 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 35.8-2 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.

35.8.3 I2C Master Mode

In master mode, a start condition is generated on the I2C bus, which then receives the clock. If I2C bus is in the idle state (SCK="H", SOT="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".

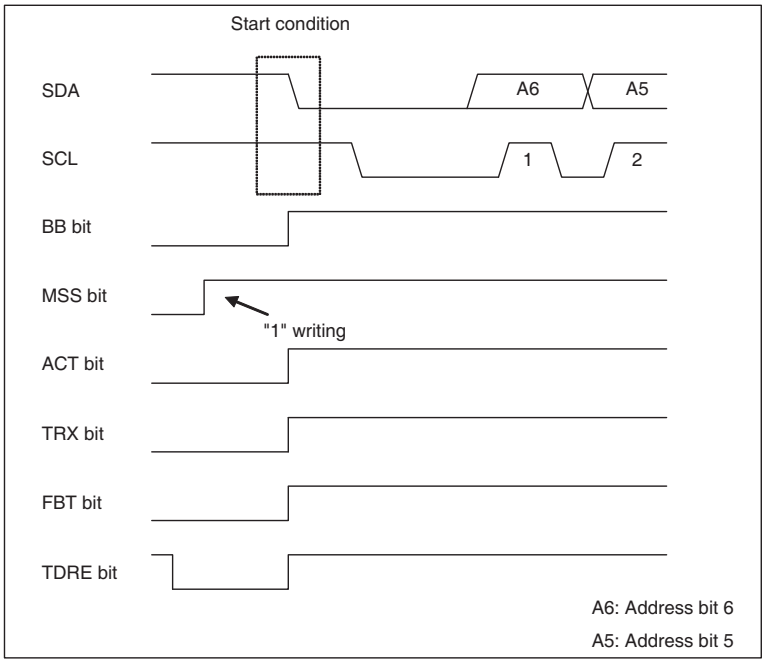
35.8.3.1 Start Condition Generation

A start condition is output if:

- When SDA="H", SCL="H", ISMK:EN="1", and IBCR:BB="0" is written to the IBCR:MSS bit

If a start condition is output to I2C bus the IBCR:ACT bit is set to "1". Then, once the start condition is received, the IBSR:BB bit is set to "1", indicating that I2C bus is on the communication. (See Figure 35.8-9)

Figure 35.8-9 Relationship between Start Condition Output and Various Bits



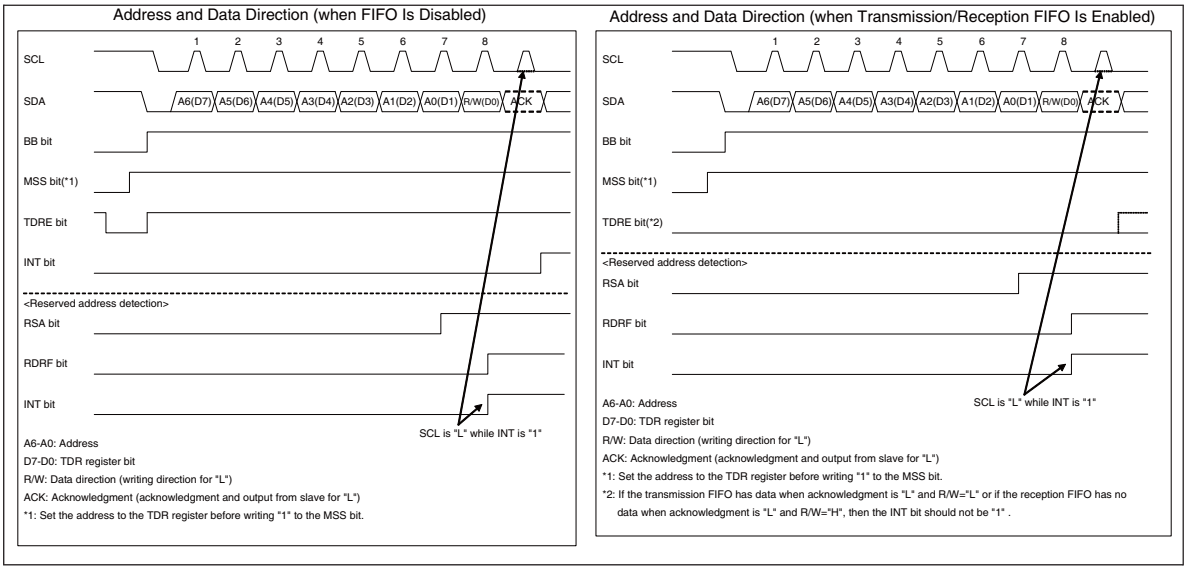
(Note) Bus clock should be used at 8MHz or more in operating mode 4 (I2C mode) and setting a baud rate generator at 400kbps or more is prohibited.

35.8.3.2 Slave Address Output

After a start condition is output, the data contained in the TDR register is output as the address, beginning with bit7. If FIFO is enabled, the data first written in the TDR register is output. Bit0 is used to indicate the data direction bit (R/W). If the data direction bit (R/W) is "0", the data is in the write direction (from master to slave). Set the address for the TDR register before "1" is written to IBCR:MSS or IBCR:SCC.

Figure 35.8-10 shows the address and data direction output timing.

Figure 35.8-10 Address and Data Direction



35.8.3.3 Acknowledge Reception by Transmitting First Byte

After the data direction bit (R/W) is output, the I2C interface receives an acknowledge from the slave. The operation varies depending on whether FIFO is enabled or disabled, as indicated in the following table:

Table 35.8-3 Operation after Acknowledge Reception (when DMA mode is disabled) (IBSR:RSA="0", SSR:DMA="0")

Transmission FIFO operation	Reception FIFO operation	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge reception	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	

Table 35.8-4 Operation after Acknowledge Reception (when DMA mode is enabled) (IBSR:RSA="0", SSR:DMA="1")

Transmission FIFO operation	Reception FIFO operation	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge reception	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	

■ 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 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"), set data including the slave address, etc. in the transmission FIFO.
  - For data reception from the slave (data direction bit = "1"), configure the FIFO byte count register to specify the number of bytes to be received, write to the transmit data register using the slave address, data direction bit, and number of dummy data to be received.
- If the IBSR:RSA bit is "0", the master, after receiving ACK as an acknowledge, does not set the interrupt flag (IBCR:INT) to "1", but transmits/receives data according to the data direction bit (not waited). If NACK is received, the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" is written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

■ 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"), set data including the slave address, etc. in the transmission FIFO.
  - For data reception from the slave (data direction bit ="1"), configure the FIFO byte count register to specify the number of bytes to be received, write to the transmit data register using the slave address, data direction bit, and number of dummy data to be received.
- If the IBCR:RSA bit is "0", after receiving ACK as an acknowledge, does not set the interrupt flag (IBCR:INT) to "1", but transmits/receives data according to the data direction bit (not waited). If NACK is received, the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" is written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

Figure 35.8-11 Acknowledge (If FIFO Is Disabled, IBSR:RSA="0", and the Response Is ACK)

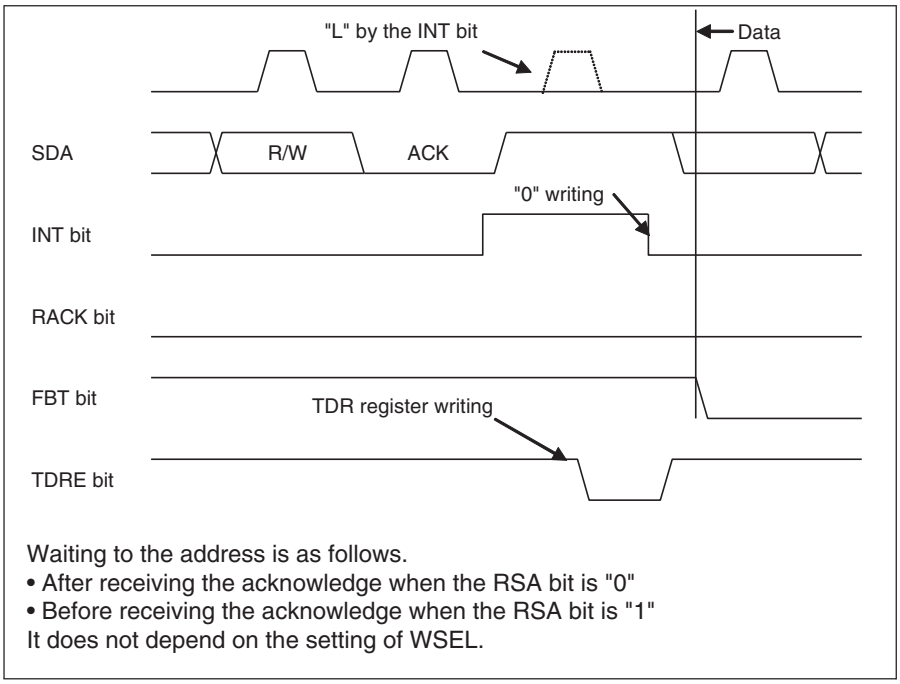


Figure 35.8-12 Acknowledge (If FIFO Is Disabled, IBSR:RSA="0", and the Response Is NACK)

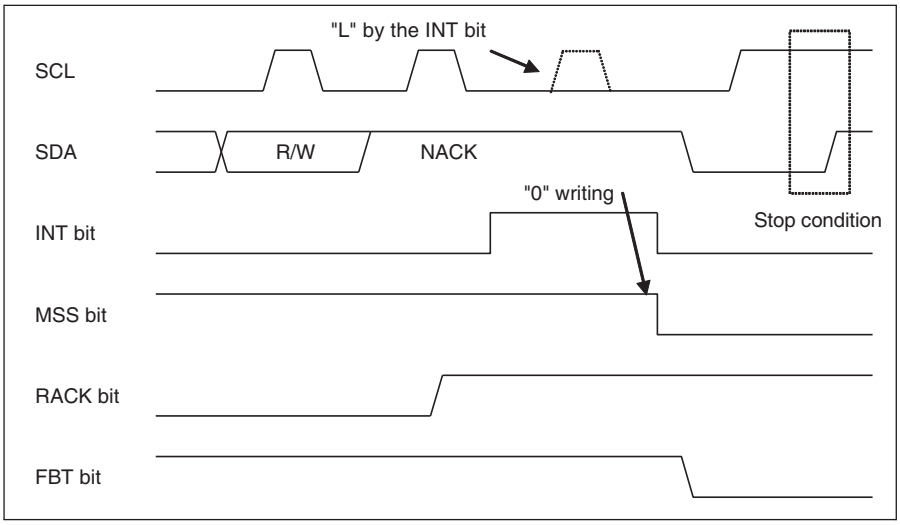


Figure 35.8-13 Acknowledge (If FIFO Is Disabled, IBSR:RSA="1", and the Response Is ACK)

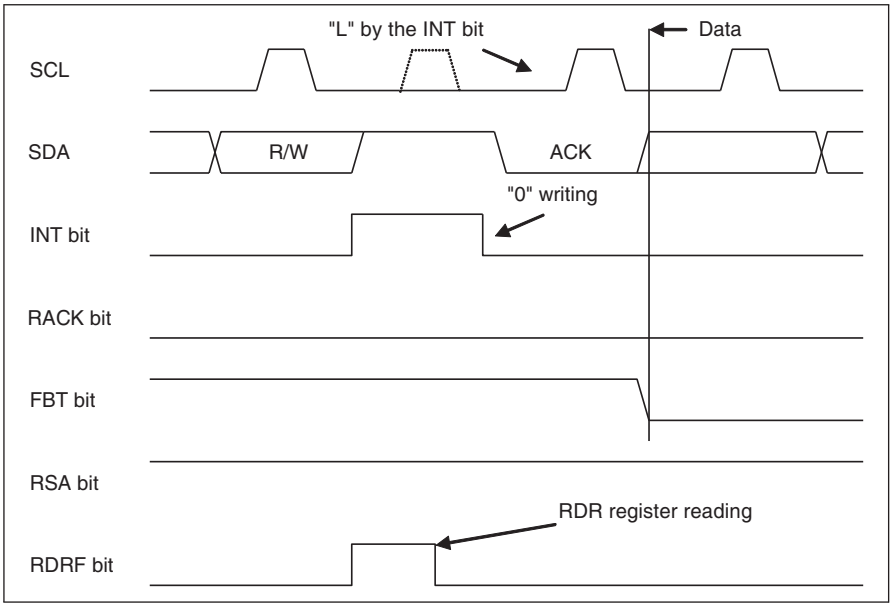


Figure 35.8-14 Acknowledge (If FIFO Is Disabled, IBSR:RSA="1", and the Response Is NACK)

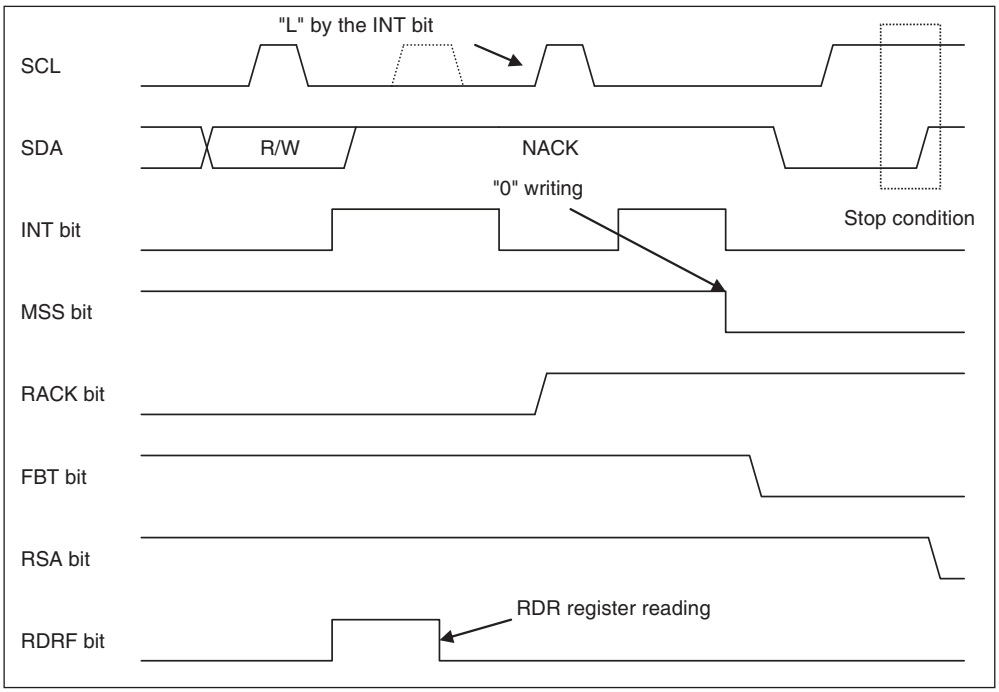
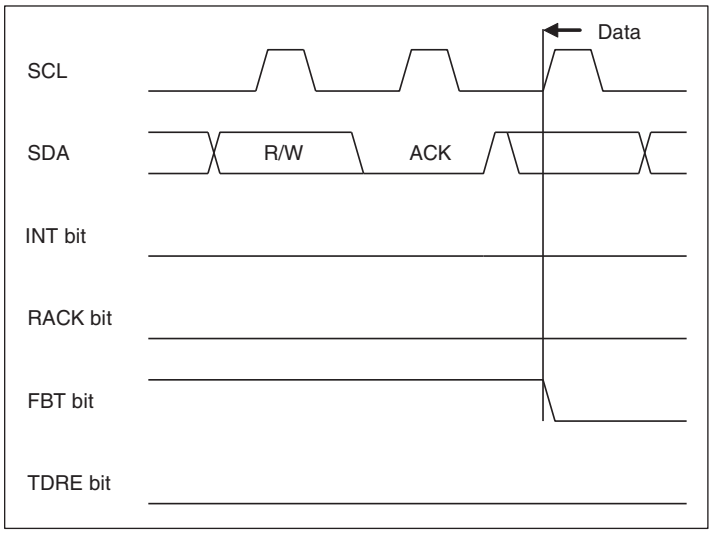


Figure 35.8-15 Acknowledge (If FIFO Is Enabled, Transmission FIFO Data Exists, No Reception FIFO Data Exists, IBSR:RSA=0, and the Response Is ACK)



35.8.3.4 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 35.8-5 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 35.8-6 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 that is not at the reserved address

- If transmission FIFO is disabled

(1) Set the slave address (including the data direction bit) in the TDR register, and set the IBCR:MSS bit to "1".
(2) Transmit the slave address and receive ACK. The interrupt flag (IBCR:INT) is set to "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) is set to "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) is set to "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) is set to "1".
(3) The data transmitted to the TDR register is written and release the I^2C bus from waiting state.
(4) Put the I^2C bus in a wait by setting the transmission bus idle flag (SSR:TBI) to "1", after receiving an acknowledge upon the transmission of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1".
(5) The data transmitted to the TDR register is written and release the I^2C bus from waiting state.
(6) Put the I^2C bus in a wait by setting the transmission bus idle flag 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) is set to "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) The data transmitted to the TDR register is written 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 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 (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) is set to "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 I2C 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.

1. Write "1" to the IBCR:INT bit.
2. Confirm that the IBCR:INT bit is set to "1".
3. Write a slave address in TDR.
4. Set "1" in the IBCR:SCC bit.

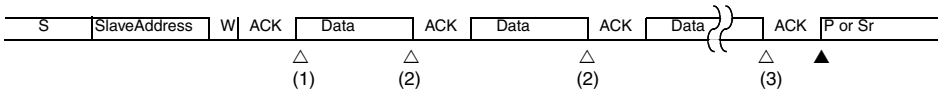
(Notes)

- 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 enable(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 disabeld (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 disabeld (SSR:DMA="0"), the interrupt flag (IBCR:INT) will not become "1" but only SSR:RDRF is set to "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 is set to "1" (when reception FIFO is enabled or number of data set in the FBYTE register is received).

Figure 35.8-16 Master Transmission Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")

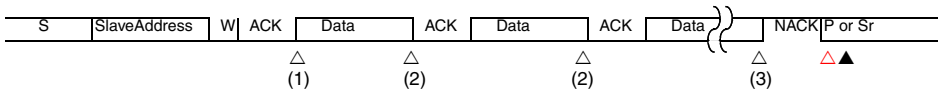
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 35.8-17 Master Transmission Interrupt (2)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", ACK Response)



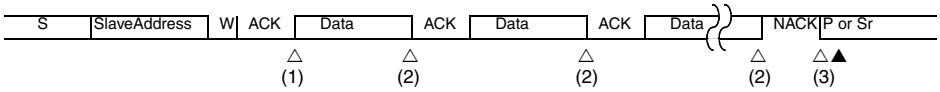
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 35.8-18 Master Transmission Interrupt (3)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)



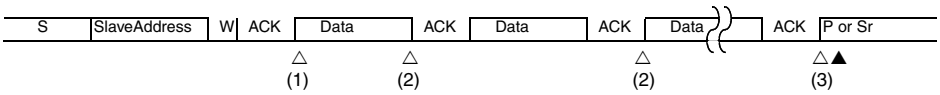
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 35.8-19 Master Transmission Interrupt (4)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", Intermediate NACK Response)



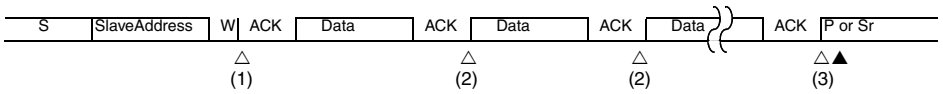
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "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 35.8-20 Master Transmission Interrupt (5)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1" → "0", IBSR:RSA="0", ACK Response)



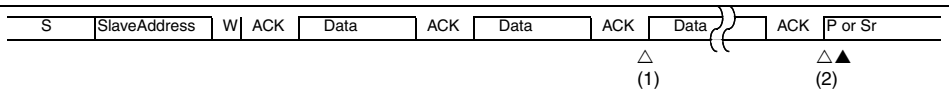
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write WSEL="0", INT = "0" after the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 35.8-21 Master Transmission Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")



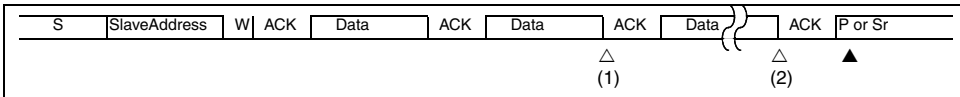
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception
- 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 35.8-22 Master Transmission Interrupt (7)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



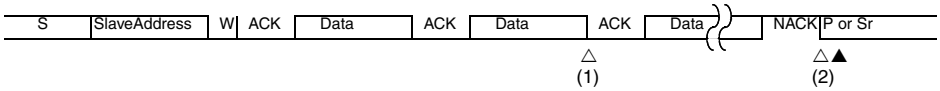
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to 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 35.8-23 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 35.8-24 Master Transmission Interrupt (9)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)

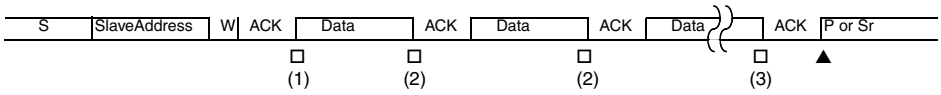


S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to the transmission FIFO
(2) An interrupt generated by NACK response
- Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 35.8-25 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 by TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission+ acknowledgment reception
- the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission + acknowledgment reception
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 35.8-26 Master Transmission Interrupt (11)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", ACK Response)

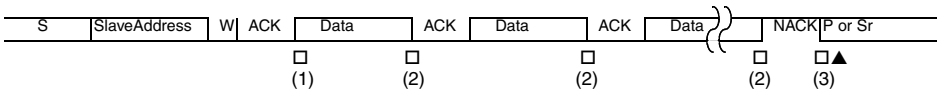


S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission+ acknowledgment reception
- the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 35.8-27 Master Transmission Interrupt (12)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)

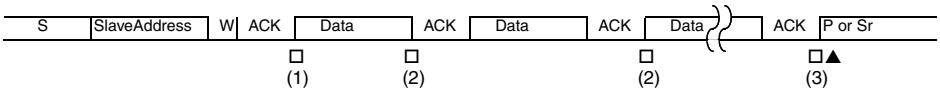
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE= "1"
□: Interrupt by TBIE= "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- The transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- The transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT, TBI)

Figure 35.8-28 Master Transmission Interrupt (13)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", Intermediate NACK Response)



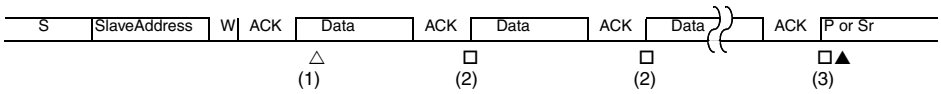
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after 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 35.8-29 Master Transmission Interrupt (14)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1"→"0", IBSR:RSA="0", ACK Response)



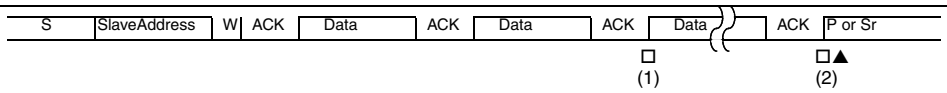
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- The transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write WSEL="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 (TBIE)

Figure 35.8-30 Master Transmission Interrupt (15)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")



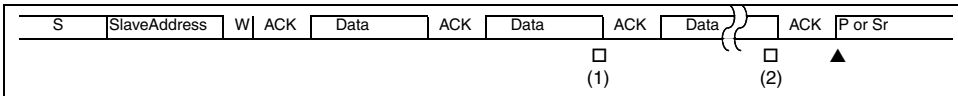
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE= "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + 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
- 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, TBI)

Figure 35.8-31 Master Transmission Interrupt (16)-when FIFO is Enabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



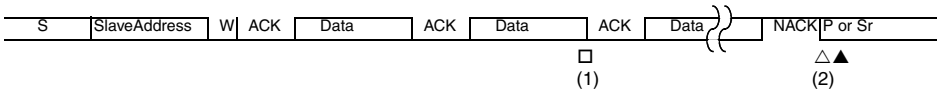
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated because the transmission FIFO is empty
- Write INT = "0" after the transmission data is written to the transmission 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 35.8-32 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 by 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 35.8-33 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 by 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"

35.8.3.5 Data Reception by Master System

■ When DMA mode is disable (SSR:DMA="0")

If the data direction bit (R/W) is "1", data sent from the slave device will be received.

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 when IBCR:WSEL bit is "1" means before a wait and when IBCR:WSEL bit is "0" means after a wait, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register.
- 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 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 when received a NACK even if the setting is for ACKE.
- 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 35.8-7 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) is set to "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 sets are received.
  (5) After reception of the last data, send a NACK response, set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" in order to generate a stop condition or a repeated start condition.

- If the send/receive FIFO operation is enabled
  (1) Set the receive data count to the FBYTE register.
  (2) Write the slave address (including the data direction bit) and the dummy data (for the receive data size) into the TDR register.
  (3) Set the IBCR:MSS bit to "1".
  (4) Respond with an ACK and continue data reception when the SSR:TDRE bit is kept "0". When receiving the specified bytes of data (set by FBYTE), set the SSR:RDRF bit to "1". When the SSR:RDRF bit is set to "1", read the RDR register.
  (5) If the SSR:TDRE bit is set to "1" and if IBCR:WSEL="0", send a NACK response. If IBCR: WSEL="1", set the interrupt flag to "1" immediately after 1 byte of data reception in order to wait the I^2C bus.
  (6) If IBCR:WSEL="1", set the IBCR:ACKE bit to "0". If IBCR:WSEL="0", the IBCR:ACKE bit needs not be set. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" in order to generate a stop condition or a repeated start condition.

■ When DMA mode is enable (SSR:DMA="1")

If the data direction bit (R/W) is "1", data sent from the slave device will be received.

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 when IBCR:WSEL bit is "1" means before a wait and when IBCR:WSEL bit is "0" means after a wait, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register.
- 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 when IBCR:WSEL bit is "1" means before a wait and when IBCR:WSEL bit is "0" means after a wait, the master device responds with ACK or NACK by the ACKE bit setting of IBCR register.

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 when received a NACK even if the setting is for ACKE bit.
- When 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 35.8-8 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) is set to "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 RDR register is read, and the data of the dummy is written 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 transmit/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". When 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 wait the I^2C bus. If IBCR:WSEL="1", set the transmission bus idle flag (SSR:TBI) to "1" immediately after 1 byte of data reception in order to wait the 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(*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", I^2C bus will be waited immediately after one byte data is received when IBCR:WSEL="1".

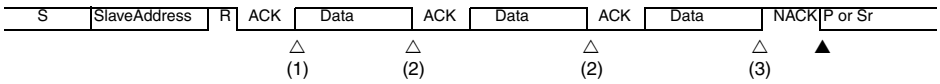
(Notes)
- If the 7-bit slave address detection is enabled (ISBA:SAEN="1"), you cannot specify a 7-bit slave address in the master mode.
- If SSR:TDRE is "0", an acknowledge signal will be sent based on the IBCR:ACKE bit setting and the subsequent process will be executed even if an overrun error occurs.
- 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 is set to "1" after the last bit is received and the interrupt flag (IBCR:INT) is set to "1" after ACK is transmitted.

Figure 35.8-34 Master Reception Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")

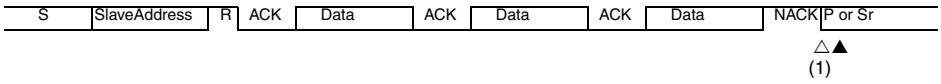
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
 - Interrupt is cleared to "0" by writing INT = "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 35.8-35 Master Reception Interrupt (2)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")



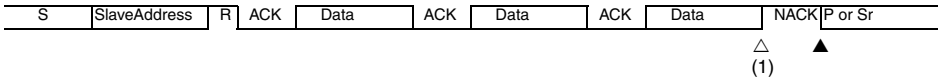
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by slave address transmission, direction bit transmission and acknowledgement reception:
 - Clears interrupt to "0" after writing "0" to INT
(2) Generation of interrupt by 1-byte reception:
 - Writes "0" to INT after reading reception data
(3) Generation of interrupt by 1-byte reception:
 - Sets "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 35.8-36 Master Reception interrupt (3)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBCR:ACKE="0", IBSR:RSA="0")



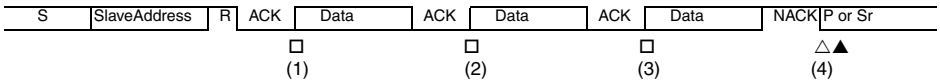
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by TDRE="1":
- Sets MSS="0", MSS="1" or SCC="1" after reading all data from reception FIFO

Figure 35.8-37 Master Reception interrupt (4)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="1", IIBSR:RSA="0")



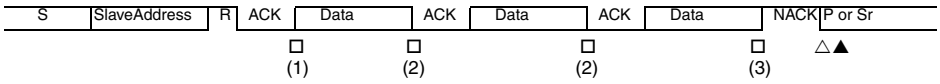
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) Generation of interrupt by TDRE="1":
- Sets "0" to ACKE and sets MSS="0" or MSS="1", and SCC="1" after reading all data from reception FIFO

Figure 35.8-38 Master Reception interrupt (5)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



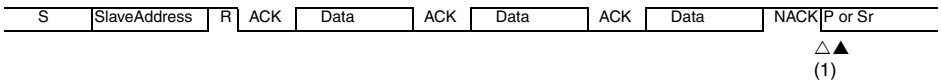
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- the data of the dummy is written in the TDR register.
(2) An interrupt generated by 1byte reception + acknowledgment transmission
- the data of the dummy is written in the TDR register after reading reception data
(3) An interrupt generated by 1byte reception + acknowledgment transmission
- The data of the dummy is written in the TDR register after reading reception data and set ACKE= "0"
(4) An interrupt generated by 1byte reception + acknowledgment transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 35.8-39 Master Reception interrupt (6)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



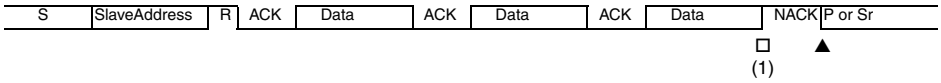
S: Start condition
R: Data direction bit (Read direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
□: Interrupt by TBIE= "1"
(1) Generation of interrupt by slave address transmission + direction bit transmission + acknowledgment reception
- the data of the dummy is written in the TDR register.
(2) An interrupt generated by 1 byte reception
- the data of the dummy is written in the TDR register after reading reception data
(3) An interrupt generated by 1 byte reception
- Set ACKE= "0", MSS= "0" or MSS= "1", and SCC= "1" after reception data reading
*: Interrupt flag(INT,TBI) generated, TDRE bit is "1"

Figure 35.8-40 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 35.8-41 Master Reception interrupt (8)-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 CNDE = "1"
□: Interrupt by 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

35.8.3.6 Arbitration Lost

Suppose data from the master device conflicts with data from another master device, and the transmitted data is different from the received data. Such a case is determined to be arbitration lost. At this time, the IBCR:MSS bit is set to "0" and the IBSR:AL bit is set to "1" so that the device can operate in 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="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.

35.8.3.7 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".

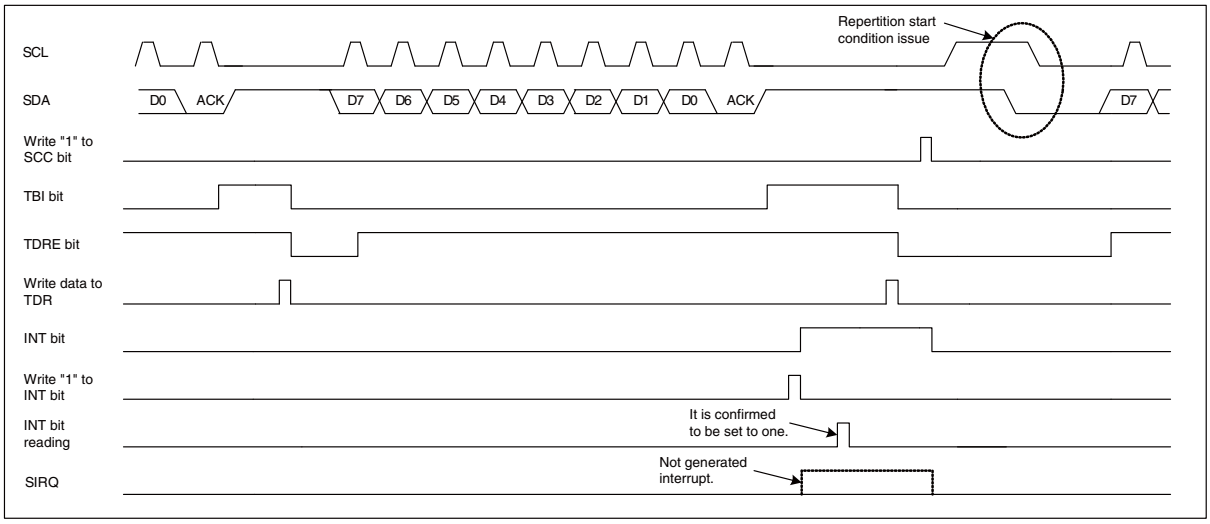
35.8.3.8 Repetition Start Condition Issue when DMA Mode Enabled (SSR:DMA="1")

When the transmission bus is idle (SSR:TBI="1") 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 status 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 35.8-42 Repetition Start Condition Issue when DMA Mode is Enabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0", ACK Response)



35.8.4 I2C Slave Mode

If the (repeated) start condition is detected and if a combination of ISBA register and ISMK register settings match the received address, an ACK response is sent and the slave mode operation starts.

(Note) Suppose the repeated start condition is detected during an address data transfer after the detection of a start condition or during a transfer of bits 2 to 9 (acknowledge bits). In such a case, a bus error is detected (IBCR:BER="1") and reception is interrupted, so the next data cannot be received.
In this case, after clearing the interrupt flag (IBCR:INT), the master must retransmit the start condition.

35.8.4.1 Detection of Slave Address Matching

After 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 35.8-9 Operations Immediately After Acknowledgment to Slave Address

Transmission FIFO operation	Reception FIFO operation	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledgement	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
				1		
Disabled	Enabled	-	Without data	0	The IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
				1		
Enabled	Enabled	-	Without data	0	The IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	

- Reserved address detected

If the first byte already matches the reserved address ("0000xxxx" or "1111xxxx"), the IBCR:INT bit is set to "1" and the I2C 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 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 ACK is output.

35.8.4.2 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.

35.8.4.3 Reception by Slave Device

If the slave address matches and if the data direction bit is "0", it indicates the data reception in the slave mode. The following gives an example of procedure of data reception in the slave mode.

■ When DMA mode is disable (SSR:DMA="0")

- If the receive FIFO operation is disabled

(1) After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to wait the I2C bus. You can determine the interrupt occurrence due to slave address matching shown by IBCR:MSS bit, IBCR:ACT bit and IBSR:FBT bit. Set the IBCR:ACKE bit to "1" and set the interrupt flag (IBCR:INT) to "0" to release the I2C bus from the waiting state. See Table 35.8-9.

(2) After receiving one byte of data, set the interrupt flag (IBCR:INT) to "1" based on the IBCR:WSEL setting, and wait the I2C bus.

(3) Read the received data from the RDR register, set the IBCR:ACKE bit, and set the interrupt flag (IBCR:INT) to "0" to release the I2C bus from the waiting state.

(4) Repeat Steps (2) and (3) until the stop condition or the repeated start condition is detected.

- If the 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 I2C bus is waited. When the stop condition or the repeated start condition is detected, the IBSR:SPC bit and IBSR:RSC bit are set to "1" but the interrupt flag (IBCR:INT) is not set to "1" (and the I2C bus is not waited). If the value set in the FBYTE register matches the number of received data, the reception FIFO sets the SSR:RDRF bit to "1". During this time, if the SMR:RIE bit is "1", a reception interrupt occurs.

(2) If the interrupt flag (IBCR:INT) is set to "1", the received data is read from the RDR register. After reading all data sets, set the interrupt flag to "0" and release the I2C bus from the waiting state. When the stop condition or the repeated start condition is detected, all of the received data sets are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".

■ When DMA mode is enable (SSR:DMA="1")

- If the receive FIFO operation is disabled

(1) After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to wait the I^2C bus. You can determine the interrupt occurrence due to slave address matching shown by IBCR:MSS bit, IBCR:ACT bit and IBSR:FBT bit. Set the IBCR:ACKE bit to "1" and set the interrupt flag (IBCR:INT) to "0" to release the I^2C bus from the waiting state. See Table 35.8-9.

(2) Receive data full flag (SSR:RDRF) is set in "1" immediately after the reception 1 byte after the data of one byte is received. The I^2C bus is waited for IBCR:WSEL="1" immediately after the reception 1 byte after the acknowledge is transmitted for IBCR:WSEL ="0" in the place where receive data full flag (SSR:RDRF) is set in "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^2C 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

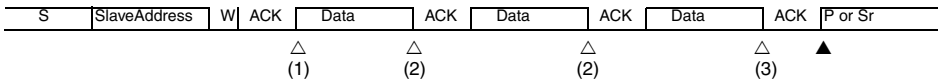
(1)The interrupt flag (IBCR:INT) becomes "1" and waits for the I^2C bus by detecting NACK. The I^2C bus is waited for when reception FIFO becomes full. The IBSR:SPC bit and the IBSR:RSC bit are made "1" when the stop condition and the repetition start condition are detected and interrupt flag (IBCR:INT) does not become "1" (none waiting of the I^2C bus). When a set value of the FBYTE register is corresponding to the received number of data, reception FIFO makes the SSR:RDRF bit "1". When the SMR:RIE bit is "1" at that time, the reception interrupt is generated.

(2) If the interrupt flag (IBCR:INT) is set to "1", the received data is read from the RDR register. After reading all data sets, set the interrupt flag to "0" and release the I^2C 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 sets are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".

Figure 35.8-43 Slave Reception Interrupt (1)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="0")

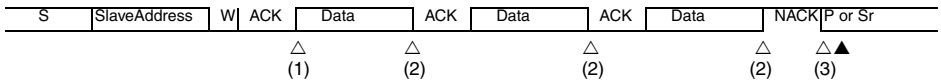
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt 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 35.8-44 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 35.8-45 Slave Reception Interrupt (3)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0")

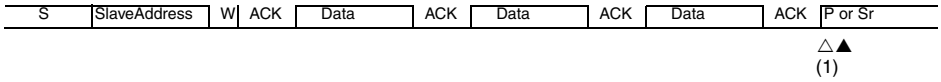


S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt 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 35.8-46 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 condition
- Read all data from the reception FIFO

Figure 35.8-47 Slave Reception Interrupt (5)-when FIFO is Enabled (SSR:DMA="0", IBSR:RSA="0")

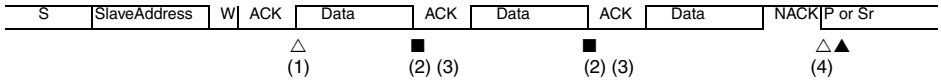


S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the reception FIFO gets full
- Read all data from the reception FIFO, and write INT = "0"

Figure 35.8-48 Slave Reception Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")

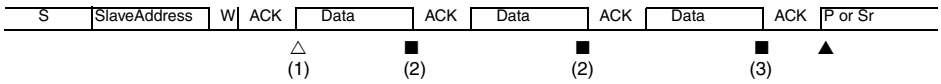
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by matching the reserved address ("0000XXXX" or "1111XXXX")
- Read the reception data, and write ACKE = "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 35.8-49 Slave Reception Interrupt (7)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="0")



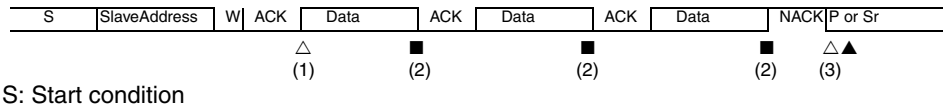
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt by RIE= "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- ACKE = "1", INT= "0" write
(2) An interrupt generated by 1 byte reception (I^2C bus is no wait)
- reception data is read from the reception buffer.
(3) I^2C bus wait by ACK response
- reception data is read from the reception buffer.
(4) An interrupt generated by 1 byte reception + NACK response
- Reception data is read from the reception buffer.INT="0" write

Figure 35.8-50 Slave Reception Interrupt (8)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



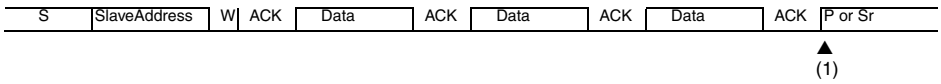
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt by RIE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- ACKE="1", INT="0" write
(2) An interrupt generated by 1 byte reception
- reception data is read from the reception buffer.
(3) An interrupt generated by 1 byte reception
- After reception data is read from the reception buffer.

Figure 35.8-51 Slave Reception Interrupt (9)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0")



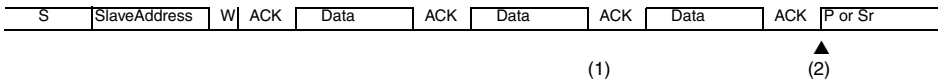
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition
Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
■: Interrupt by RIE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- ACKE="1", INT="0" write
(2) An interrupt generated by 1 byte reception
- reception data is read from the reception buffer.
(3) An interrupt generated by NACK response
- INT="0" write

Figure 35.8-52 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 condition
- Read all data from reception FIFO

Figure 35.8-53 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 condition
- Read all data from reception FIFO

Figure 35.8-54 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 by RIE = "1"
(1) Interrupt generated by reserved address ("0000xxxx" or "1111xxxx") is match
- Receive data is read. ACKE="1", INT="0" write
(2) An interrupt generated by 1 byte reception + acknowledge output
- Receive data is read.
(3) An interrupt generated by 1 byte reception + acknowledge output
- Receive data is read.

35.8.4.4 Transmission by Slave Device

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 35.8-9).

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.

35.8.5 Example of I2C Flowchart

Figure 35.8-55 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is disable (SSR:DMA="0")) 1/3

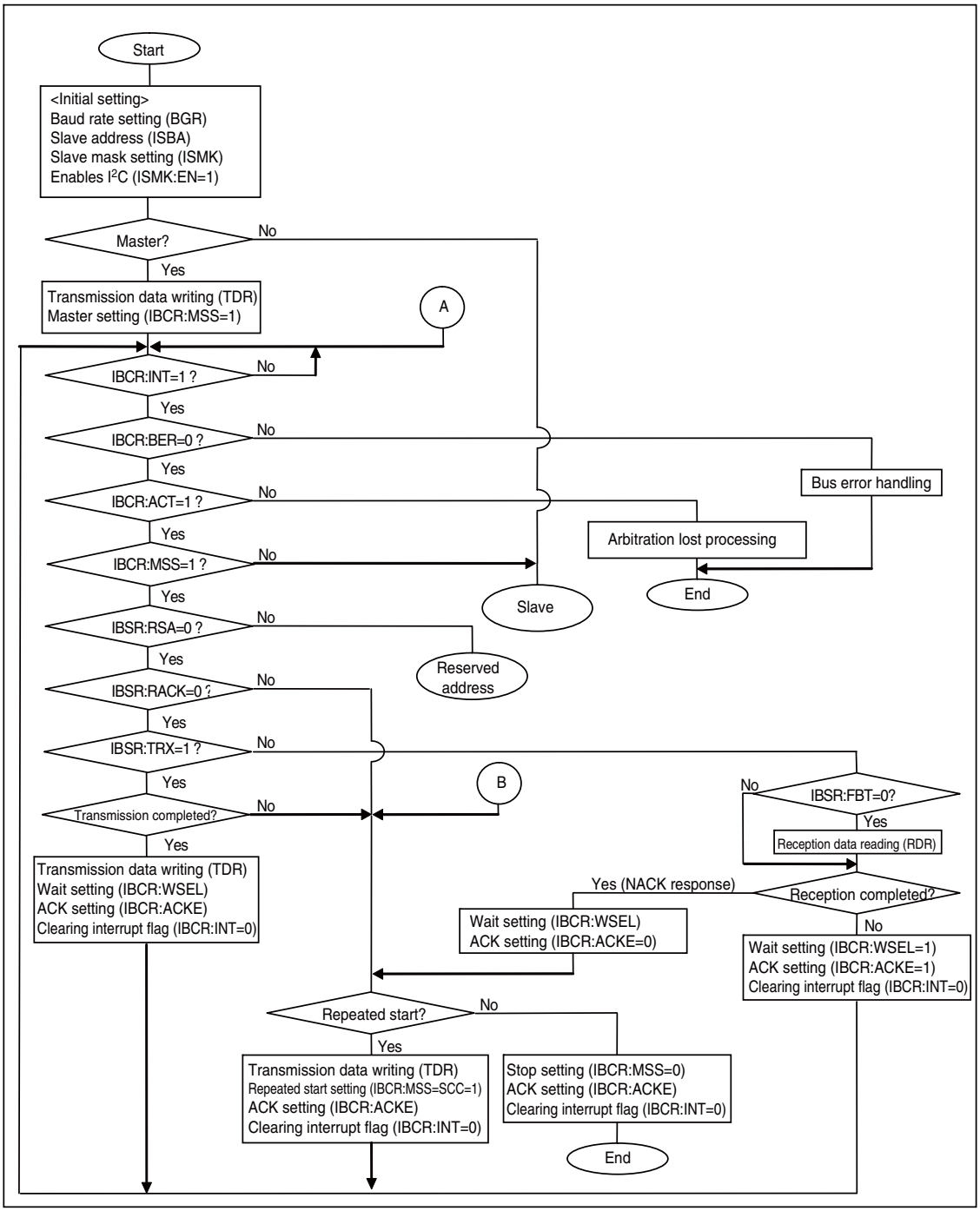


Figure 35.8-56 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is disable (SSR:DMA="0")) 2/3

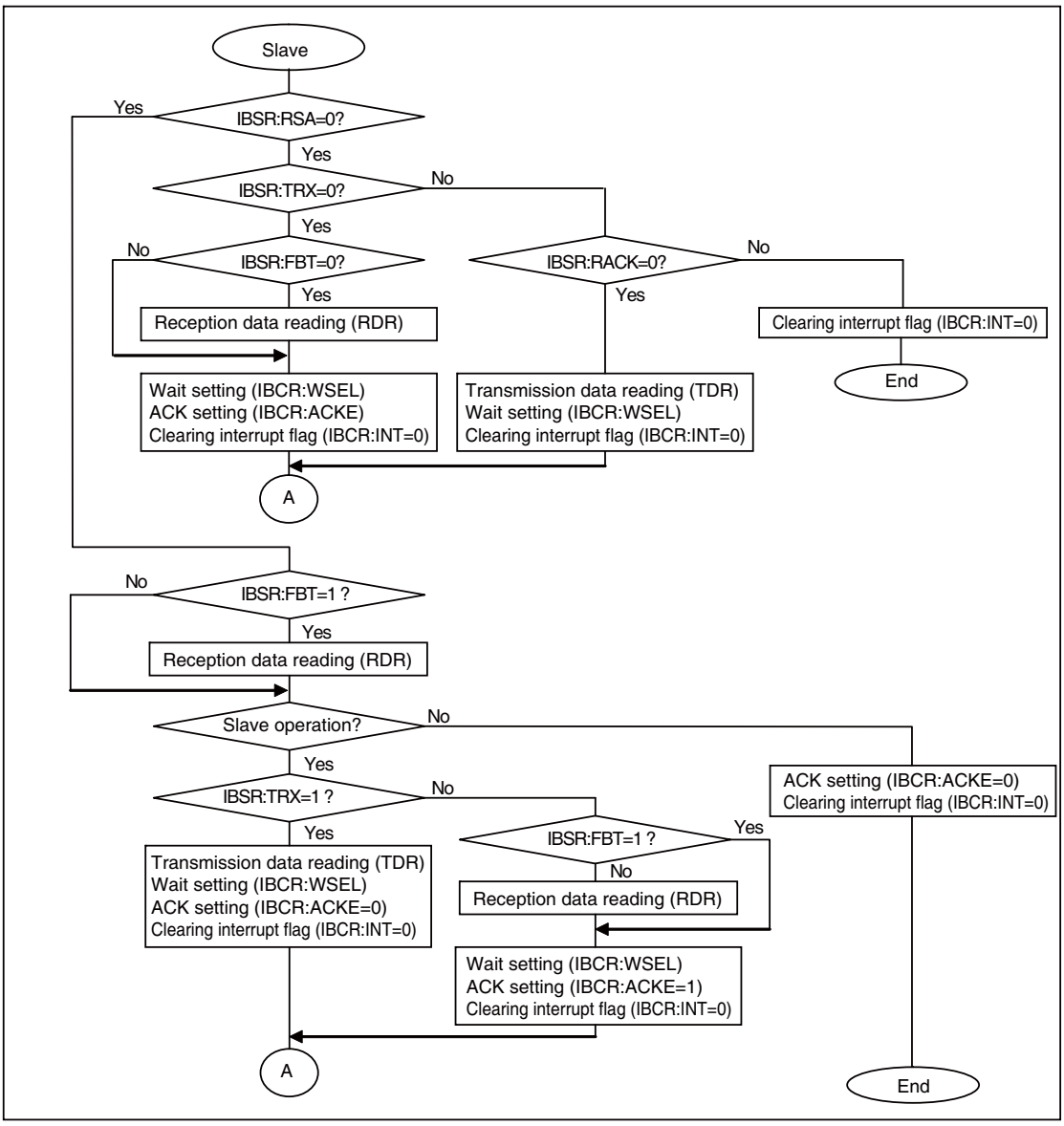


Figure 35.8-57 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is disable (SSR:DMA="0")) 3/3

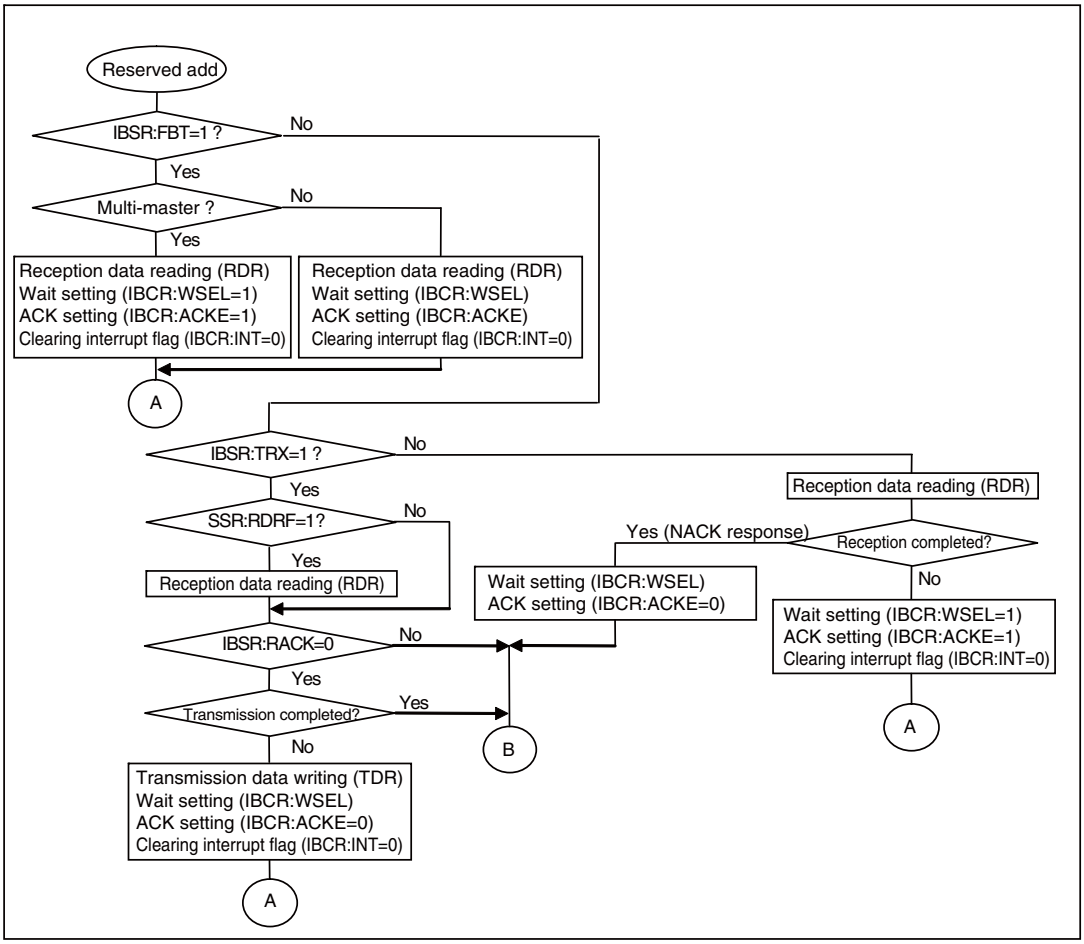


Figure 35.8-58 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is enable (SSR:DMA="1")) 1/4

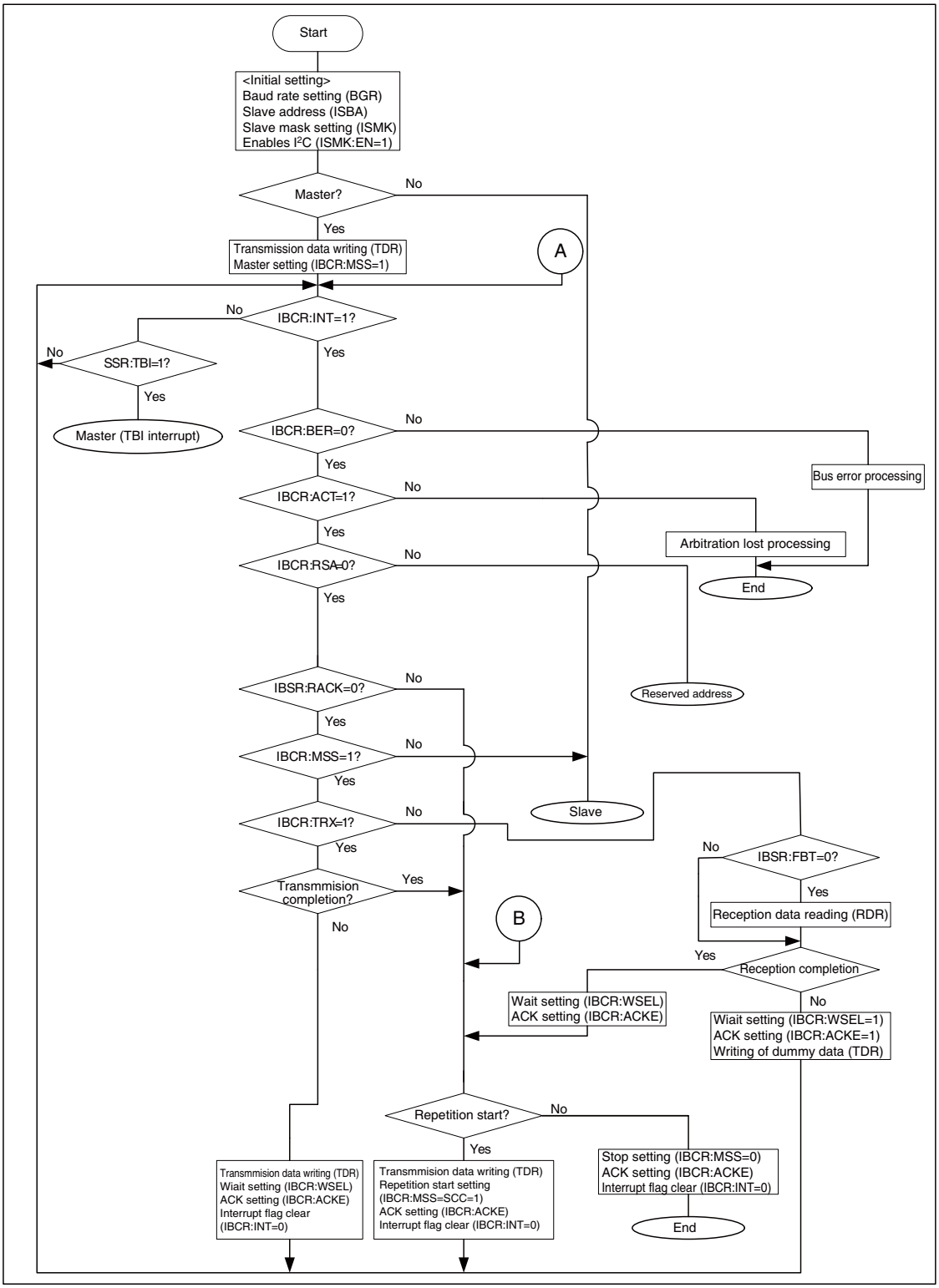


Figure 35.8-59 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is enable (SSR:DMA="1")) 2/4

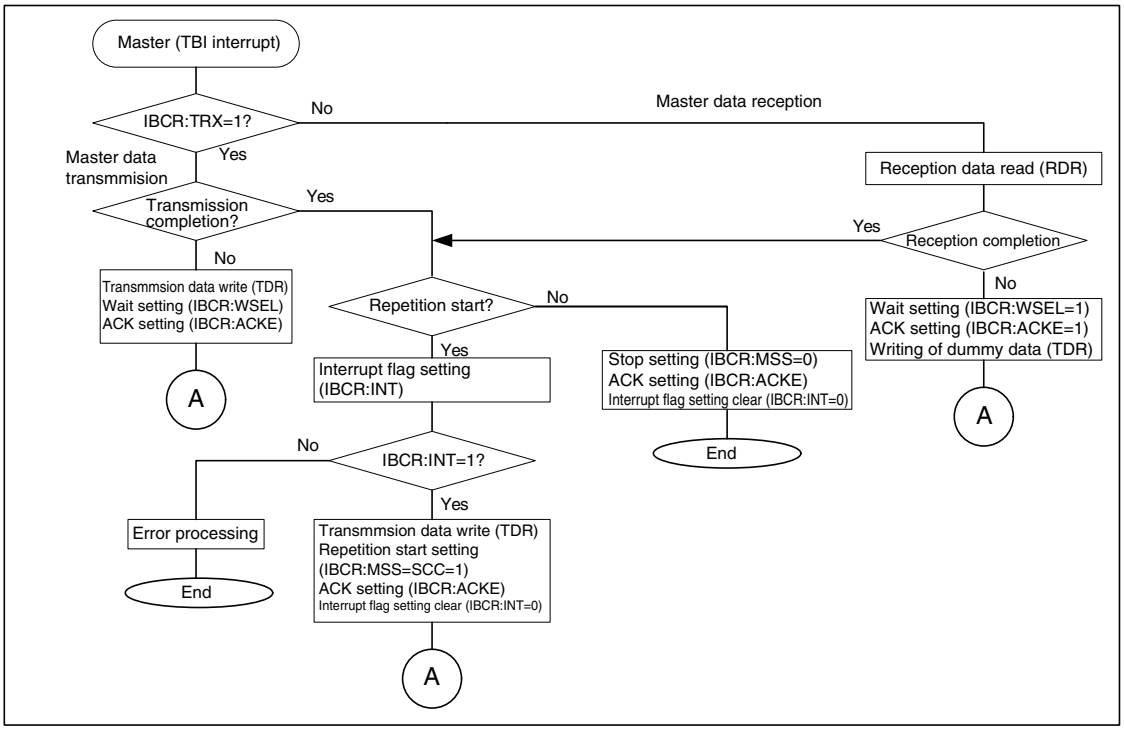


Figure 35.8-60 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is enable (SSR:DMA="1")) 3/4

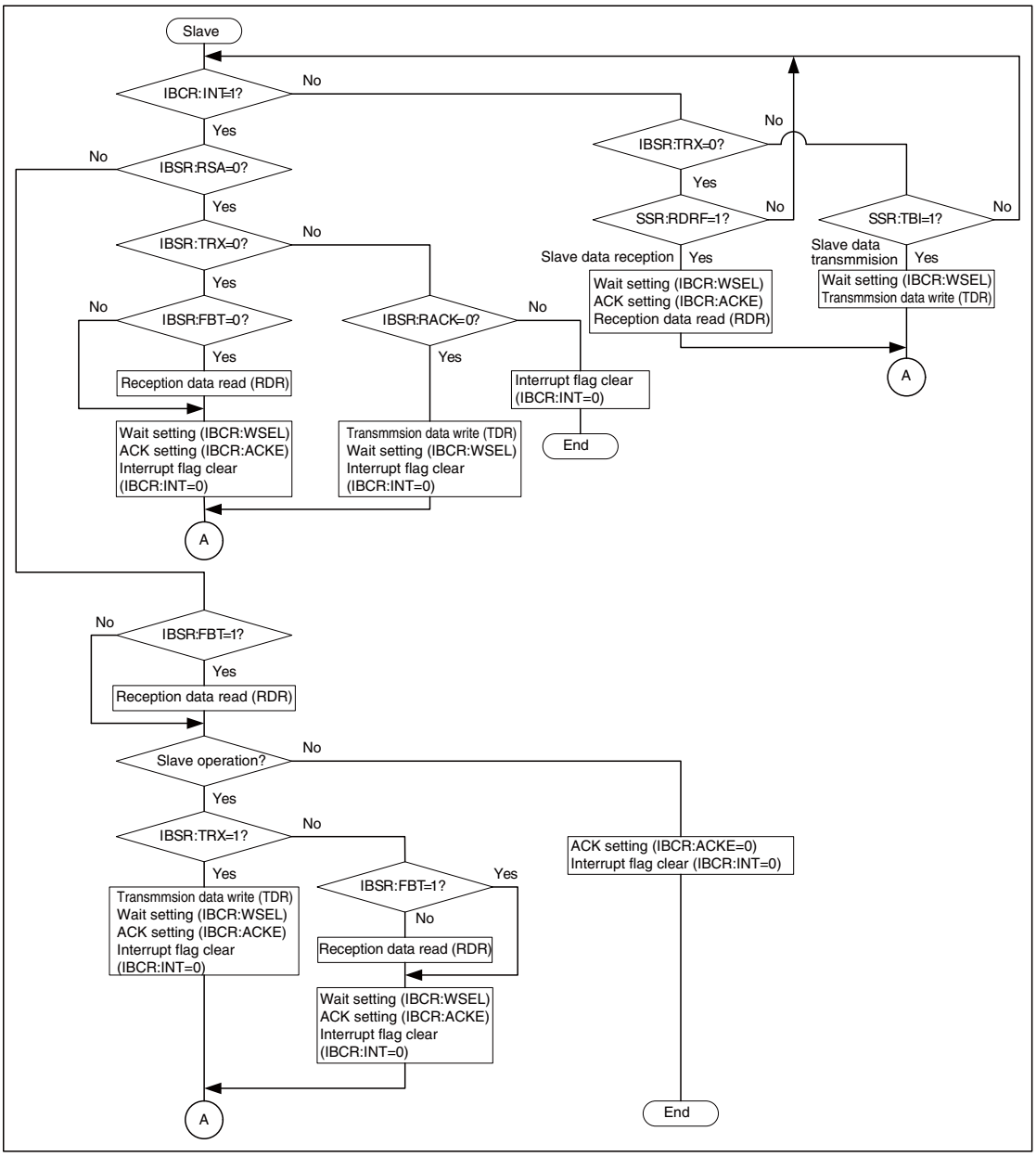


Figure 35.8-61 Example of I2C Flowchart (FIFO Memory Not Used) (When DMA mode is enable (SSR:DMA="1")) 4/4

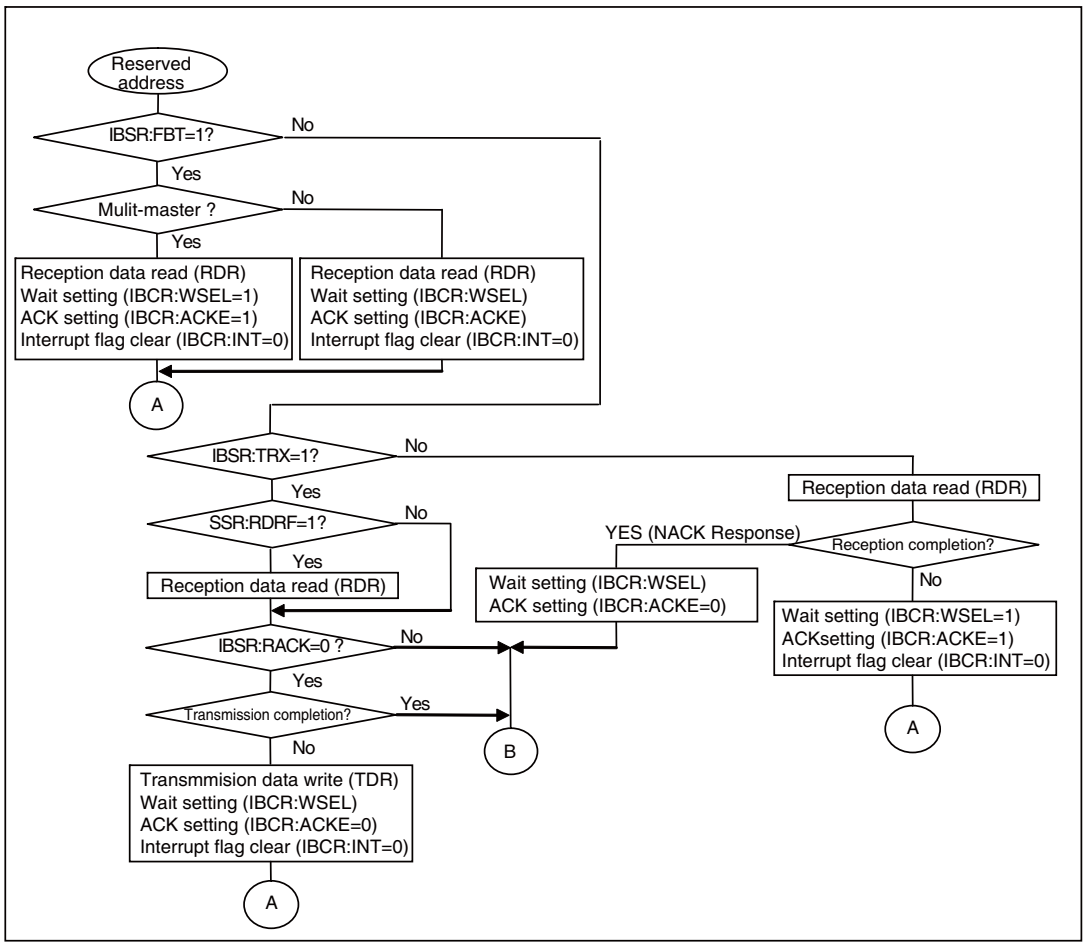
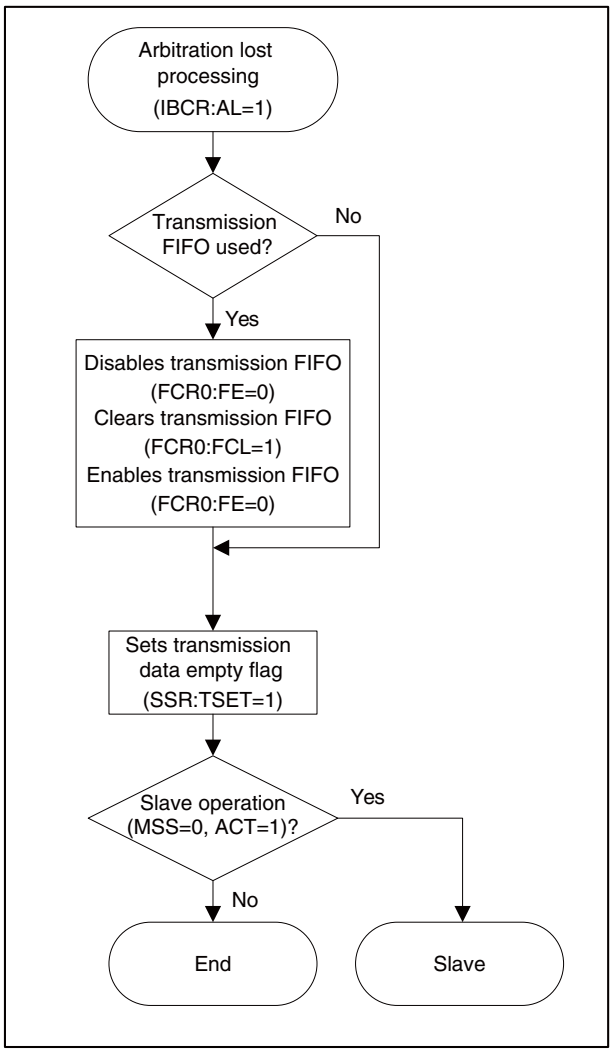


Figure 35.8-62 Example of Arbitration Lost Processing Flowchart



(Note) These flows indicate the operation setting outline in the I2C mode. It is necessary to do processing, considering error processing, etc., in accordance with applications.

CHAPTER 36 CAN

36.1 Overview

This product includes 2 CAN channels.
MB91F583AS/F584AS/F585AS includes one CAN channel.

The CAN is based on the CAN protocol ver. 2.0A/B, which is a standard protocol for serial communication and is widely used for automobiles, FA, and other industrial fields.

36.2 Features

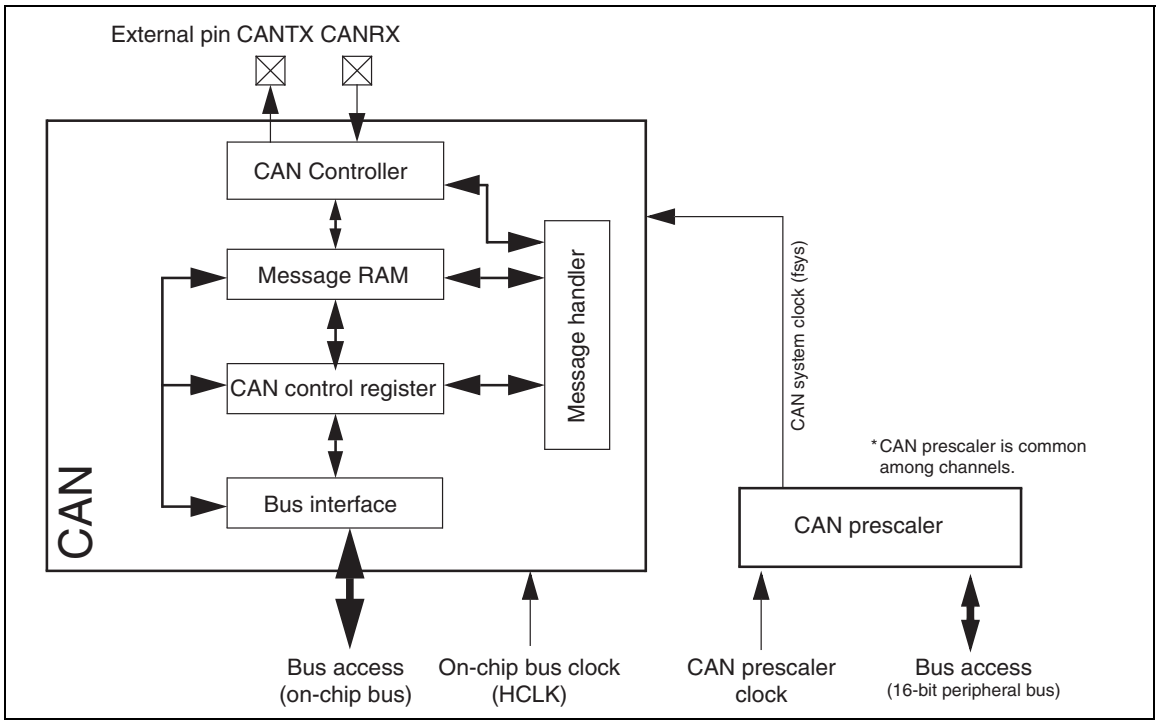
The CAN of this product has the following features:

- CAN protocol ver. 2.0A/B is supported.
- Bit rates up to 1 Mbit/s are supported.
- An identification mask is applied to each message object.
- Programmable FIFO mode is supported.
- Maskable interrupts are supported.
- Programmable loopback mode for self-test operation is supported.
- Data can be written to and read from a message buffer using an interface register.
- 2 channels of 64 transmission/ reception message buffers are supported. Since the support varies with the product type and channels, see "CHAPTER 2 OVERVIEW".

36.3 Configuration

Figure 36.3-1 is a block diagram of the CAN.

Figure 36.3-1 Block Diagram of CAN (for one channel)



Code: FC28-1v4-91580MS-1-E

■ CAN Controller

The CAN controller controls the CAN protocol and serial registers for serial/parallel conversion to transfer the transmission/reception message.

■ Message RAM

Holds message objects.

■ Message Handler

Controls message RAM and the CAN controller.

■ CPU Interface

Controls the interface with the FR internal bus.

■ CAN Prescaler

Generates CAN system clocks (fsys).

36.4 Registers

The CAN includes the following registers:

- CAN control register (CTRLR)
- CAN status register (STATR)
- CAN error counter (ERRCNT)
- CAN bit timing register (BTR)
- CAN interrupt register (INTR)
- CAN test register (TESTR)
- CAN prescaler extension register (BRPER)
- IFx command request registers (IFxCREQ)
- IFx command mask registers (IFxCMSK)
- IFx mask registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx arbitration registers 1, 2 (IFxARB1, IFxARB2)
- IFx message control register (IFxMCTR)
- IFx data registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
- CAN transmission request registers 1, 2 (TREQR1, TREQR2)
- CAN new data update registers 1, 2 (NEWDT1, NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1, INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1, MSGVAL2)
- CAN clock prescaler register (CANPRE)

The CAN register is given an address space of 256 bytes (64 words) and accessible in byte or word mode. The CPU accesses the message RAM via a message interface register.

■ List of Base Addresses (Base-addr) and External Pins

Channel number	Base-addr	External pin name	
		CANTX	CANRX
0	0x2000	TX0	RX0
1	0x2100	TX1	RX1

■ List of Overall Control Register

Table 36.4-1 List of Overall Control Register

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 00_H	CAN control register (CTRLR)		CAN status register (STATR)		STATR: BOff, EWarn, EPass=Read Only RxOk, TxOk, LEC = Read/ Write
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Reserved bit	See the CTRLR.	Reserved bit	See the STATR.	
	Reset:00_H	Reset:01_H	Reset:00_H	Reset:00_H	
Base-addr + 04_H	CAN error counter (ERRCNT)		CAN bit timing register (BTR)		ERRCNT: Read only BTR: Data can be written to this register 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: Data can be written to this register 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]	
	Int-Id[15:8]	Int-Id[7:0]	Reserved bit	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 bit		BRPER: When CCE(CTLR) = 1, data can be written to this register.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Reserved bit	BRPE[3:0]	-	-	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	

■ List of Message Interface Register

Table 36.4-2 List of Message Interface Register (1 / 3)

Address	Register				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 bit	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 bit		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	See the IF1MCTR.	See the IF1MCTR.	-	-	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 20_H	IF1 data A register 1 (IF1DTA1)		IF1 data A register 2 (IF1DTA2)		Byte order: Big endian
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data [0]	Data [1]	Data [2]	Data [3]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	

Table 36.4-2 List of Message Interface Register (2 / 3)

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 24_H	IF1 data B register 1 (IF1DTB1)		IF1 data B register 2 (IF1DTB2)		Byte order: Big endian
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data [4]	Data [5]	Data [6]	Data [7]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 30_H	IF1 data A register 2 (IF1DTA2)		IF1 data A register 1 (IF1DTA1)		Byte order: Little endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data [3]	Data [2]	Data [1]	Data [0]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 34_H	IF1 data B register 2 (IF1DTB2)		IF1 data B register 1 (IF1DTB1)		Byte order: Little endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data [7]	Data [6]	Data [5]	Data [4]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 40_H	IF2 command request register (IF2CREQ)		IF2 command mask register (IF2CMSK)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Busy	Mess. No. [5:0]	Reserved bit	See the IF2CMSK.	
	Reset:00_H	Reset:01_H	Reset:00_H	Reset:00_H	
Base-addr + 44_H	IF2 mask register 2 (IF2MSK2)		IF2 mask register 1 (IF2MSK1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MXtd. MDir, Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset:FF_H	Reset:FF_H	Reset:FF_H	Reset: FF_H	
Base-addr + 48_H	IF2 arbitration register 2 (IF2ARB2)		IF2 arbitration register 1 (IF2ARB1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal, Xtd, Dir, ID[28:24]	ID[23:16]	ID[15:8]	ID[7:0]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	

Table 36.4-2 List of Message Interface Register (3 / 3)

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + $4C_H$	IF2 message control register (IF2MCTR)		Reserved bit		
	bit[15:8]	bit[7:0]	bit[7:0]	bit[15:8]	
	See the IF2MCTR.	See the IF2MCTR.	-	-	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 50_H	IF2 data A register 1 (IF2DTA1)		IF2 data A register 2 (IF2DTA2)		Byte order: Big endian
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data [0]	Data [1]	Data [2]	Data [3]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 54_H	IF2 data B register 1 (IF2DTB1)		IF2 data B register 2 (IF2DTB2)		Byte order: Big endian
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data [4]	Data [5]	Data [6]	Data [7]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 60_H	IF2 data A register 2 (IF2DTA2)		IF2 data A register 1 (IF2DTA1)		Byte order: Little endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data [3]	Data [2]	Data [1]	Data [0]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr + 64_H	IF2 data B register 2 (IF2DTB2)		IF2 data B register 1 (IF2DTB1)		Byte order: Little endian
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data [7]	Data [6]	Data [5]	Data [4]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	

■ List of Message Handler Register

Table 36.4-3 List of Message Handler Register (1 / 2)

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 80_H	CAN transmission request register 2 (TREQR2)		CAN transmission request register 1 (TREQR1)		TREQR1, 2:Read only
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	TxRqst[32:25]	TxRqst[24:17]	TxRqst[16:9]	TxRqst[8:1]	
	Reset:00_H	Reset:00_H	Reset:00_H	Reset:00_H	
Base-addr +84_H	Reservation area for support the number of message buffers is 32 or more (Refer to CAN transmission request register (TREQR1, TREQR2)) TREQR3, TREQR4: 64 message buffer is supported				TREQR3, 4:Read only
Base-addr +88_H Base-addr +$8C_H$	Reservation area				
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	Reservation area for support the number of message buffers is 32 or more (Refer to CAN new data update register (NEWDT1, NEWDT2)) NEWDT3, NEWDT4: 64 message buffer is supported				NEWDT3, 4: Read only
Base-addr +98_H Base-addr +$9C_H$	Reservation area				
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 + $A4_H$	Reservation area for support the number of message buffers is 32 or more (Refer to CAN interrupt pending register (INTPND1, INTPND2)) INTPND3, INTPND4: 64 message buffer is supported				INTPND3, 4: Read only
Base-addr + $A8_H$ Base-addr + AC_H	Reservation area				
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	

Table 36.4-3 List of Message Handler Register (2 / 2)

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + $B4_H$	Reservation area for support the number of message buffers is 32 or more (Refer to CAN message valid register (MSGVAL1, MSGVAL2)) MSGVAL3, MSGVAL4: 64 message buffer is supported				MSGVAL3, 4: Read only
Base-addr + $B8_H$ Base-addr + BC_H	Reservation area				

■ Clock Prescaler Register

Table 36.4-4 Clock Prescaler Register

Address	Register				Note
	+0	+1	+2	+3	
00_04A4_H	CANPRE	-	-	-	CAN Prescaler
	bit[4:0]	-	-	-	
	CANPRE[4:0]	-	-	-	
	Reset:00_H	-	-	-	

■ Overall Control Register

- CAN control register (CTRLR)
- CAN status register (STATR)
- CAN error counter (ERRCNT)
- CAN bit timing register (BTR)
- CAN interrupt register (INTR)
- CAN test register (TESTR)
- CAN prescaler extension register (BRPER)

■ Message Interface Register

- IFx command request register (IFxCREQ)
- IFx command mask register (IFxCMSK)
- IFx mask registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx arbitration registers 1, 2 (IFxARB1, IFxARB2)
- IFx message control register (IFxMCTR)
- IFx data registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

■ Message Handler Register

- CAN transmission request registers 1, 2 (TREQR1, TREQR2)
- CAN data update registers 1, 2 (NEWDT1, NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1, INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1, MSGVAL2)

■ Prescaler Register

- CAN clock prescaler register (CANPRE)

36.4.1 Overall Control Register

Overall control register controls the CAN protocol and operation modes and provides status information.

■ Overall Control Register

- 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)

■ CAN Control Register (CTRLR)

This register controls the operation mode of the CAN controller.

- **CAN Control Register (upper byte): Address Base+00$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **CAN Control Register (lower byte): Address Base+01$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Test	CCE	DAR	Reserved	EIE	SIE	IE	Init	
0	0	0	0	0	0	0	1	Initial value
R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8] Reserved bits

The read value is always "0". For writing, set "0" to these bits.

[bit7] Test: Test mode enable bit

Test	Function
0	Normal operation [Initial value]
1	Test mode

(Note) Set "1" to the test bit only when the Init bit is "1".

[bit6] CCE: Bit timing register write enable bit

CCE	Function
0	Disables the writing to the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER).[Initial value]
1	Enables the writing to the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER). This bit is valid when the Init bit is "1".

[bit5] DAR: Automatic retransmission disable bit

DAR	Function
0	Enables the automatic retransmission of the message when CAN loses the arbitration or when an error is detected. [Initial value]
1	Disables automatic retransmission.

The CAN controller retransmits the frame automatically when it loses the arbitration or when an error is detected during transfer. To enable automatic retransmission, set "0" to the DAR bit. In order to operate CAN in Time Triggered CAN environments, "1" needs to be set to the DAR bit.

(Notes) When "1" is set to the DAR bit, the values for the TxRqst and NewDat bits of the message objects are as follows: (For message objects, see Section "36.4.3 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 transmission, set "1" to the TxRqst bit.
- When the DAR bit for 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". For writing, set "0" to this bit.

[bit3] EIE: Error interrupt code enable bit

EIE	Function
0	Disables the interrupt code setting to the CAN interrupt register (INTR) with the bit change for Boff or EWarn of the CAN status register (STATR). [Initial value]
1	Enables the status interrupt code setting to the CAN interrupt register (INTR) with the bit change for Boff or EWarn of the CAN status register (STATR).

[bit2] SIE: Status interrupt code enable bit

SIE	Function
0	Disables the interrupt code setting to the CAN interrupt register (INTR) with the bit change for TxOk, RxOk or LEC of the CAN status register (STATR). [Initial value]
1	Enables the status interrupt code setting to the CAN interrupt register (INTR) with the bit change for TxOk, RxOk or LEC of the CAN status register (STATR). The bit change for TxOk, RxOk and LEC generated by the writing from CPU is not set to the CAN interrupt register (INTR).

[bit1] IE: Interrupt enable bit

IE	Function
0	Disables interrupt.[Initial value]
1	Enables interrupt.

[bit0] Init: Initialization bit

Init	Function
0	Operates after the initialization release of the CAN controller.
1	Initialize the CAN controller and stops the operation. [Initial value]

(Notes)

- The bus-off recovery sequence cannot be shortened with the Init bit setting/release. When a device is in the bus-off state, the CAN controller itself sets "1" to the Init bit and stops all bus operations. When the Init bit is cleared to "0" in the bus-off state, the bus operation is stopped until the bus-idle continues 129 times (11-bit recessive is regarded as one time). The error counter is reset after the execution of the bus-off recovery sequence.
- When the Init bit is set to "1" and then to "0" during the bus-off recovery sequence, the bus-off recovery sequence runs from the beginning (129 times regarding 11-bit recessive as one time).
- To set the CAN bit timing register (BTR), set "1" to the Init and CCE bits.
- When setting the INIT bit of the CAN control register (CTRLR) to "1", set the INIT bit of the CAN control register (CTRLR) to "1" after transmission is completed. When setting the INIT bit of the CAN control register to "1" during transmission, and then setting the INIT bit to "0" to transmit:
  - First set the INIT bit to "1".
  - Second make the message buffer with the transmission request bit (TxRqst) = "1" cancel transmission (set the TxRqst bit to "0").
  - Third set the INIT bit to "0"..

  After a 2-bit time passes, set the transmission request bit (TxRqst) of the message buffer to "1".
- Before transiting to the low-power consumption mode (Stop mode or Clock mode) or before changing the source clock supplied to CAN controller, CAN controller must be initialized by setting Init bit to "1".
- To change the clock divide ratio which supplies to the CAN interface by the following registers, set "1" to the Init bit and stop the CAN controller.
  - CAN bit timing register (BTR)
  - CAN prescaler extension register (BRPER)
  - CAN prescaler register (CANPRE)

■ CAN Status Register (STATR)

This register displays the CAN and CAN bus statuses.

- **CAN Status Register (upper byte): Address Base+02$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **CAN Status Register (lower byte): Address Base+03$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
BOff	EWarn	EPass	RxOk	TxOk	LEC[2:0]			
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,W	R,W	R,WX	R,WX	R,WX	Attribute

[bit15 to bit8] Reserved bits

The read value is always "0". For writing, set "0" to these bits.

[bit7] BOff: Bus-off bit

BOff	Function
0	Indicates the CAN controller is not in the bus-off state. [Initial value]
1	Indicates the CAN controller is in the bus-off state.

[bit6] EWarn: Warning bit

EWarn	Function
0	Indicates both the transmission and reception counters are below 96. [Initial value]
1	Indicates the transmission or reception counter is 96 or more.

[bit5] EPass: Error passive bit

EPass	Function
0	Indicates both the transmission and reception counters are below 128 (error active state). [Initial value]
1	Indicates the reception counter is RP bit =1 and the transmission counter is 128 or more (error passive state).

[bit4] RxOk: Successful message reception bit

RxOk	Function
0	Indicates successful message communication is not performed on the CAN bus or the bus is in the idle state.[Initial value]
1	Indicates successful message communication is performed on the CAN bus.

[bit3] TxOk: Successful message transmission bit

TxOk	Function
0	Indicates the bus is in the idle state or successful message transmission is not performed.[Initial value]
1	Indicates successful message transmission is performed.

(Note) The RxOk and TxOk bits are cleared only with "0" writing.

[bit2 to bit0] LEC2 to LEC0 Last error code bits

LEC2 to LEC0	State	Function
000	Normal	Indicates transmission or reception is performed successfully. [Initial value]
001	Stuff error	Indicates more than 6 bits of dominant or recessive is detected continuously in a message.
010	Form error	Indicates the fixed format segment of a received frame is detected as incorrect.
011	Ack error	Indicates the transmission message is not acknowledged by other nodes.
100	Bit1 error	Indicates dominant was detected even though recessive was sent with the message transmission data other than arbitration field.
101	Bit0 error	Indicates recessive was detected even though dominant was sent with the message transmission data. This bit is set every time 11 bits of recessive is detected during the bus recovery. Reading this bit allows the monitoring of the bus recovery sequence.
110	CRC error	Indicates that CRC data and CRC result calculated for a received message did not match.
111	Undetected	Indicates no transmission or reception is performed during the period when LEC reads "111_B" after "111_B" is set to the LEC bit. (bus idle status)

The LEC bit holds the code that indicates the last error occurred on the CAN bus. This bit is cleared to "0" when a message transfer (reception/transmission) completes without error. The undetected code "111_B" can be used for checking the code update.

(Notes)

- The status interrupt code (8000_H) is set to the CAN interrupt register (INTR) if the BOff or EWarn bit is changed when the EIE bit is "1" or if RxOk, TxOk or the LEC bit is changed when the SIE bit is "1".
- The flag values for the RxOk and TxOk bits are updated with the program writing, and thus the RxOk and TxOk bit values set by the CAN controller are changed. When using RxOk or TxOk bit, these bits needs to be cleared within (45 × 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).

■ CAN Error Counter (ERRCNT)

This register 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)**

15	14	13	12	11	10	9	8	bit
RP	REC 6 to REC0							
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

- **CAN Error Counter Register (lower byte): Address Base+05$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
TEC 7 to TEC0								
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] RP: Reception error passive display

RP	Function
0	The reception error counter indicates that it is not the error passive state.[Initial value]
1	The reception error counter indicates that the error passive state that is defined in the CAN specification has been reached.

[bit14 to bit8] REC6 to REC0: Reception error counter

These are reception error counter values. 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 REC6 to REC0=127 is incremented by 8 for reception error, the result is RP=1 and REC6 to REC0=127.
When REC6 to REC0=126 is incremented by 8 for reception error, the result is RP=1 and REC6 to REC0=126.
When REC6 to REC0=119 is incremented by 8 for reception error, the result is RP=0 and REC6 to REC0=127.

[bit7 to bit0] TEC7 to TEC0: Transmission error counter

These are transmission error counter values. 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 TEC7 to TEC0=255 is incremented by 8 for transmission error, the result is Init=1 and TEC7 to TEC0=255.
When TEC7 to TEC0=254 is incremented by 8 for transmission error, the result is Init=1 and TEC7 to TEC0=254.
When TEC7 to TEC0=247 is incremented by 8 for transmission error, the result is Init=0 and TEC7 to TEC0=255.

■ CAN Bit Timing Register (BTR)

This register sets the prescaler and bit timing.

- **CAN Bit Timing Register (upper byte): Address Base+06$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	TSeg2			TSeg1				
0	0	1	0	0	0	1	1	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- **CAN Bit Timing Register (lower byte): Address Base+07$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
SJW		BRP						
0	0	0	0	0	0	0	1	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Reserved bit

The read value is always "0". For writing, set "0" to this bit.

[bit14 to bit12] TSeg2: Time segment 2 setting bits

Valid setting values are 0 to 7. TSeg2+1 bit value is time segment 2.

Time segment 2 corresponds to the phase buffer segment (PHASE_SEG2) based on the CAN specification.

[bit11 to bit8] TSeg1: Time segment 1 setting bits

Valid setting values are 1 to 15. 0 cannot be set. TSeg1+1 bit value is time segment 1.

Time segment 1 corresponds to the propagation segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) based on the CAN specification.

[bit7, bit6] SJW: Resynchronization jump width setting bits

Valid setting values are 0 to 3. The SJW+1 bit value is the resynchronization jump width.

[bit5 to bit0] BRP: Baud rate prescaler setting bits

Valid setting values are 0 to 63. The BRP+1 bit value is the baud rate prescaler.

Divides frequency for system clock (fsys) and determines the basic unit time (tq) of the CAN controller.

(Note) When "1" is set to the CCE and Init bits of the CAN control register (CTRLR), set the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER).

■ CAN Interrupt Register (INTR)

This register displays the message interrupt and status interrupt codes.

- **CAN Interrupt Register (upper byte): Address Base+08$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
IntId 15 to IntId 8								
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

- **CAN Interrupt Register (lower byte): Address Base+09$_H$ (Access: Half-word, Word)**

7	6	5	4	3	2	1	0	bit
IntId 7to IntId 0								
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

IntId	Function
0000_H	No interrupt
0001_H to 0040_H	The message object number is indicated as an interrupt factor. (Message interrupt code)
0041_H to $7FFF_H$	Unused
8000_H	Indicates interrupts with the change of the CAN status register (STATR). (Status interrupt code)
8001_H to $FFFF_H$	Unused

If more than one interrupt codes are pending, the CAN interrupt register (INTR) will indicate the interrupt code of the highest priority. If a higher-priority interrupt code is generated when an interrupt code is set to the CAN interrupt register (INTR), the CAN interrupt register (INTR) is updated to the higher-order interrupt code.

Higher orders are given to the status interrupt code (8000_H), message interrupt (0001_H, 0002_H, 0003_H, ..., 0020_H) in descending order.

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 Section "36.4.3 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.

■ CAN Test Register (TESTR)

This register monitors the test mode setting and RX pins. For operation, see Section "36.5.7 Test Mode".

- **CAN Test Register (upper byte): Address Base+0A$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **CAN Test Register (lower byte): Address Base+0B$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	Reserved	
r	0	0	0	0	0	0	0	Initial value
R,WX	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0	Attribute

The level on the CAN bus is displayed as the initial value (r) of Rx for bit7.

[bit15 to bit8] Reserved bits

The read value is always "0". For writing, set "0" to these bits.

[bit7] Rx: Pin monitor bit

Rx	Function
0	Indicates the CAN bus is dominant.
1	Indicates the CAN bus is recessive.

[bit6, bit5] Tx1, Tx0: 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 bits

The read value is always "0". For writing, set "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 other than "00".

■ CAN Prescaler Extension Register (BRPER)

This register 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)**

15	14	13	12	11	10	9	8	bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **CAN Prescaler Extension Register (lower byte): Address Base+0D$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	Reserved	Reserved	Reserved	BRPE				
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit4] Reserved bits

The read value is always "0". For writing, set "0" to these bits.

[bit3 to bit0] BRPE: Baud rate prescaler extension bits

The baud rate prescaler can be extended up to 1023 by combining the BRP and BRPE bits of the CAN bit timing register (BTR).

The {BRPE (MSB:4 bit), BRP (LSB:6 bit)} + 1 value is the prescaler of the CAN controller.

36.4.2 Message Interface Register

Provides two pairs of message interface registers to control access from the CPU to the message RAM.

There are two pairs of message interface registers used to control access from the CPU to the message RAM. These two pairs of registers avoid conflict between CPU access to the message RAM and access from the CAN controller by buffering transferred data (message object). The message object (for message object, see Section "36.4.3 Message Object") transfers messages between the message interface register and the message RAM.

The functions for two pairs of message interface registers are the same expect the test basic mode, and these registers can operate independently. For example, the message interface register of IF2 can be used for readout from the message RAM while the message interface register of IF1 is being written into the message RAM. Table 36.4-1 shows two-pairs of message interface registers.

The message interface register consists of the command register (command request, command mask register) and the message buffer register (mask, arbitration, message control and data register) 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 36.4-5 IF1, IF2 Message Interface Registers (1 / 2)

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 + $1E_H$	Reserved	Base + $4E_H$	Reserved
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
Base + 28_H	Reserved	Base + 58_H	Reserved
Base + $2A_H$	Reserved	Base + $5A_H$	Reserved
Base + $2C_H$	Reserved	Base + $5C_H$	Reserved
Base + $2E_H$	Reserved	Base + $5E_H$	Reserved
Base + 30_H	IF1 data A2 (little endian)	Base + 60_H	IF2 data A2 (little endian)
Base + 32_H	IF1 data A1 (little endian)	Base + 62_H	IF2 data A1 (little endian)

Table 36.4-5 IF1, IF2 Message Interface Registers (2 / 2)

Address	IF1 register set	Address	IF2 register set
Base + 34_H	IF1 data B2 (little endian)	Base + 64_H	IF2 data B2 (little endian)
Base + 36_H	IF1 data B1 (little endian)	Base + 66_H	IF2 data B1 (little endian)

■ IFx Command Request Register (IFxCREQ)

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 test basic mode.

- **IFx Command Request Register (upper byte): Address Base+10_H & Base+40_H(Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
BUSY	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **IFx Command Request Register (lower byte): Address Base+11_H & Base+41_H(Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Message Number								
0	0	0	0	0	0	0	1	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Immediately after the message number is written into the IFx command request register (IFxCREQ), the message transfer between the message RAM and the message buffer register (mask, arbitration, message control and data register) is started. This writing operation indicates that "1" is set to the BUSY bit and a message is being transferred. When the transfer is completed, the BUSY bit is reset to "0".

When "1" is set to the BUSY bit, the CPU will be kept waiting until the BUSY bit becomes "0" if the CPU accesses to the message interface register (3 to 6 clock cycles after writing into the command request register).

The BUSY bit is used differently in the basic mode for tests. The IF1 command request register is used as a transmission message, and setting "1" to the BUSY bit directs message transmission start. When the message transfer is completed successfully, the BUSY bit is reset to "0". In addition, resetting the BUSY bit to "0" aborts message transfer at any time.

The IF2 command request register is used as a reception message, and setting "1" to the BUSY bit stores the received message in the IF2 message interface register.

[bit15] Busy: Busy flag bit

- Other than test basic mode

BUSY	Function
0	Indicates that data is not being transferred between the message interface register and the message RAM.[Initial value]
1	Indicates that data is being transferred between the message interface register and the message RAM.

- Test basic mode

IF1 command request register

BUSY	Function
0	Disables the message transmission.
1	Enables the message transmission.

IF2 command request register

BUSY	Function
0	Disables the message reception.
1	Enables the message reception.

[bit14 to bit8] Reserved bits

The read value is always "0". For writing, set "0" to these bits.

[bit7 to bit0] Message Number: Message Number

Message Number	Function
00_H	Setting is not allowed. 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 is not allowed. If this value is set, it is interpreted as 01_H to $3F_H$ and the value interpreted is read out.

(Note) 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 Section "36.5.7 Test Mode" for the details of basic mode.)

■ IFx Command Mask Register (IFxCMSK)

This register sets which data to be updated by controlling the direction of transfer between message interface register and message RAM. The register becomes invalid in test basic mode.

- **IFx Command Mask Register (upper byte): Address Base+12$_H$ & Base+42$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

- **IFx Command Mask Register (lower byte): Address Base+13$_H$ & Base+43$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
WR/RD	Mask	Arb	Control	CIP	TxRqst/ NewDat	Data A	Data B	
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] Reserved bits

The read value is always "0". For writing, set "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 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 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 indefinite after reset. Reading data from message RAM is disabled while data in message RAM is indefinite.

Bit6 to bit0 of IFx command mask register (IFxCMSK) has different meanings depending on the settings of transfer direction (WR/RD bit).

When the transfer direction is write (WR/RD=1)

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicates not updating the mask data (ID mask + MDir + MXtd) of message object*. [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 updating 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] DataA: Data 0 to Data 3 update bit

Data A	Function
0	Indicates not updating Data 0 to Data 3 of message object*. [Initial value]
1	Indicates updating Data 0 to Data 3 of message object*.

[bit0] DataB: Data 4 to Data 7 update bit

Data B	Function
0	Indicates not updating Data 4 to Data 7 of message object*. [Initial value]
1	Indicates updating Data 4 to Data 7 of message object*.

*: See Section "36.4.3 Message Object".

(Notes) • When the TxRqst/NewDat bit of IFx command mask register (IFxCMSK) is set to "1", the TxRqst bit settings of IFx message control register (IFxMCTR) becomes invalid.
• The register becomes invalid in test basic mode.

When the transfer direction is read (WR/RD=0)

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicates not transferring data (ID mask +MDir + MXtd) from message object* 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 bits

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] DataA: Data 0 to Data 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] DataB: Data 4 to Data 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.

*1 See Section "36.4.3 Message Object".

(Notes)
- It is possible to reset the IntPnd and NewDat bits to "0" by reading access to message object. However, for the IntPnd and NewDat bits of IFx message control register (IFxMCTR), the IntPnd and NewDat bits prior to being reset by reading access will be stored.
- It becomes invalid in test basic mode.

■ IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2)

These registers are used to write/read message object mask data of message RAM. In test basic mode, the configured mask data becomes invalid.

See Section "36.4.3 Message Object" for functions of each bit.

- **IFx Mask Register 2 (upper byte): Address Base+14$_H$ & Base+44$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
MXtd	MDir	Reserved	Mask28 to Mask24					
1	1	1	1	1	1	1	1	Initial value
R/W	R/W	R1,W1	R/W	R/W	R/W	R/W	R/W	Attribute

- **IFx Mask Register 2 (lower byte): Address Base+15$_H$ & Base+45$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Mask23 to Mask16								
1	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

- **IFx Mask Register 1 (upper byte): Address Base+16$_H$ & Base+46$_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Mask15 to Mask8								
1	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

- **IFx Mask Register 1 (lower byte): Address Base+17$_H$ & Base+47$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Mask7 to Mask0								
1	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

See Section "36.4.3 Message Object" for the description of each bit.

For the reserved bit (bit13 of IFx mask register 2), "1" is read out. For writing, set "1" to this bit.

■ IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2)

These registers are used to write/read message object arbitration data of message RAM. They become invalid in test basic mode.

See Section "36.4.3 Message Object" for functions of each bit.

- **IFx Arbitration Register 2 (upper byte): Address Base+18$_H$ & Base+48$_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
MsgVal	Xtd	Dir	ID28 to ID24					
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

- **IFx Arbitration Register 2 (lower byte): Address Base+19$_H$ & Base+49$_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
ID23 to ID16								
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

- **IFx Arbitration Register 1 (upper byte): Address Base+1A$_H$ & Base+4A$_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
ID15 to ID8								
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

- **IFx Arbitration Register 1 (lower byte): Address Base+1B$_H$ & Base+4B$_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
ID7 to ID0								
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

See Section "36.4.3 Message Object" for the description of each bit.

(Note) If the MsgVal bit of the message object is cleared to "0" while 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.

■ IFx Message Control Register (IFxMCTR)

These registers are used for reading/writing message object control data in message RAM. IF1 message control register will be disabled in test basic mode. NewDat and MsgLst of IF2 message control register will operate normally and the DLC bit will display the DLC of message received. Other control bits will operate as disabled ("0").

See Section "36.4.3 Message Object" for functions of each bit.

- **IFx Message Control Register (upper byte): Address Base+1C$_H$ & Base+4C$_H$**
  (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst	
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

- **IFx Message Control Register (lower byte): Address Base+1D$_H$ & Base+4D$_H$**
  (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
EoB	Reserved	Reserved	Reserved	DLC3 to DLC0				
0	0	0	0	0	0	0	0	Initial value
R/W	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	Attribute

See Section "36.4.3 Message Object" for the description of each bit.

(Notes) TxRqst, NewDat, and IntPnd bits operate differently depending on the settings of WR/RD bit in IFx command mask register (IFxCMSK).

○ If the transfer direction is "write" (IFx command mask register (IFxCMSK): WR/RD=1).
- 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).
- IntPnd bit before it has been reset will be stored to this register when the message object and IntPnd bit of the CAN interrupt pending register (INTPND) are reset by a write operation to the IFx command request register (IFxCREQ) after setting CIP bit of the IFx command mask register (IFxCMSK) to "1".
- NewDat bit before it has been reset will be stored to this register when the message object and NewDat bit of the CAN data update register are reset by a write operation to the IFx command request register (IFxCREQ) after setting TxRqst/NewDat bit of the IFx command mask register (IFxCMSK) to "1".

■ IFx Data Registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

These registers are used for reading/writing 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:**

15	14	13	12	11	10	9	8	bit
7	6	5	4	3	2	1	0	
Data								
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	Attributes

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, the remaining bytes of the data register will be written by undefined data.
- 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.

36.4.3 Message Object

Message RAM has 64 message objects. In order to prevent conflict between accesses to message RAM from CPU and CAN controller, 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 message objects.

■ Configuration of Message Object

Table 36.4-6 shows the configuration of the message object.

Table 36.4-6 Configuration of Message Object

UMask	Msk 28-0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28 to ID0	Xtd	Dir	DLC 3-0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

(Note) Message object will not be initialized by Init bit of CAN control register (CTRLR) or hardware reset. In the case of hardware reset, after its release, initialize message RAM by the CPU or set MsgVal of message RAM to "0".

■ Functions of Message Object

When transmitting a message, ID28 to ID0, Xtd and Dir bits will be used as the ID and type of the message. When receiving a message, Msk28 to Msk0, MXtd and MDir bits will be used in the acceptance filter.

ID, IDE, RTR, DLC and DATA for data frame or remote frame passing through the acceptance filter will be stored in the ID28 to ID0, Xtd, Dir, DLC3 to DLC0, Data7 to Data0 of the message objects. Xtd indicates whether the message object is an extension frame or standard frame, a 29-bit ID (extension frame) is received if Xtd is "1", and an 11-bit ID (standard frame) is 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 Section "36.5.3 Message Reception Operation" for details.)

- **MsgVal: Valid message bit**

MsgVal	Function
0	Message object is disabled. Message transmission/reception will not be performed.
1	Message object is enabled. Message transmission/reception is possible.

(Notes)
- Be sure to initialize MsgVal bit of the message object before resetting 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 MsgVal bit is cleared to "0" while the transmission is in progress, TxOk bit of the CAN status register (STATR) will become "1" as soon as the transmission ends. However, the message object and TxRqst bit of the CAN transmission request register (TREQR) will not be cleared to "0". So be sure to clear 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	Use Msk28 to Msk0, MXtd, and MDir.

(Notes) • Change the UMask bit while Init bit of the CAN control register (CTRLR) is "1" or while MsgVal bit is "0".
• When Dir bit is "1" and RmtEn bit is "0", it will operate differently depending on the UMask setting.
- If UMask is "1", 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, NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
- If UMask is "0", 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	Mask 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 message object.
1	A 29-bit ID (extended frame) is used for 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 bit in the received frame determines whether to compare it as a standard frame ID or an extended frame ID.
1	Compare the values between those set to the Xtd of the message object and those for the IDE bit in the received frame.

(Note) If an 11-bit ID (standard frame) is set to the message object, ID of the received data frame will be written to ID28 to ID18. Msk28 to Msk18 are used for ID masks.

- **Dir: Message direction bit**

Dir	Function
0	Indicates the reception direction. The remote frame will be transmitted when TxRqst is set to "1", and the data frame that has passed through the acceptance filter will be received when it is set to "0".
1	Indicates the transmission direction. Data frame will be transmitted when TxRqst is set to "1". If TxRqst is "0" and RmtEn is set to "1", CAN controller sets its TxRqst to "1" by receiving the remote frame that has passed through the acceptance filter.

- **MDir: Message direction mask bit**

MDir	Function
0	Mask the message direction bit (Dir) in the acceptance filter.
1	Does not mask the message direction bit (Dir) in the acceptance filter.

(Note) Always set MDir bit to "1".

- **EoB: End of Buffer bit (see Section "36.5.4 FIFO buffer function" for details)**

EoB	Function
0	Message object is used as FIFO buffer and is not the final message.
1	Single message object or the final message object of FIFO buffer.

(Notes)
- EoB bit is used to configure the FIFO buffer of 2 to 32 messages.
- Always set 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) MsgLst bit is only enabled when Dir bit is "0" (reception direction).

- **RxIE: Reception interrupt flag enable bit**

RxIE	Function
0	IntPnd remains unchanged after successful frame reception.
1	IntPnd is set to "1" after successful frame reception.

- **TxIE: Transmission interrupt flag enable bit**

TxIE	Function
0	IntPnd remains unchanged after successful frame transmission.
1	IntPnd is set to "1" after successful frame transmission.

- **IntPnd: Interrupt pending bits**

IntPnd	Function
0	No interrupt factor exists.
1	Interrupt factor exists. If no other high priority interrupt exists, IntId bit of the CAN interrupt register (INTR) will indicate this message object.

- **RmtEn: Remote enable**

RmtEn	Function
0	TxRqst remains unchanged by remote frame reception.
1	TxRqst will be set to "1" if a remote frame is received while Dir bit is "1".

(Notes) • When Dir bit is "1" and RmtEn bit is "0", it will operate differently depending on the UMask setting.
- If UMask bit is "1", 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, NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
- If UMask is "0", 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**

DLC 3-0	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 DLC bit when the data frame is received.

- **Data 0 to Data7: Data 0 to Data7**

Data	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.

36.4.4 Message Handler Register

All message handler registers are for reading only. The TxRqst, NewDat, IntPnd, MsgVal, and IntId bits of message object are used to display a status.

■ Message Handler Register

- CAN transmission request registers 1, 2 (TREQR1, TREQR2)
- CAN data update registers 1, 2 (NEWDT1, NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1, INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1, MSGVAL2)

■ CAN Transmission Request Registers1, 2 (TREQR1, TREQR2)

These registers display the TxRqst bit of all message objects. It is possible to check which message objects transmission request is pending by reading the TxRqst bit.

- **CAN Transmission Request Register 2 (upper byte): Address Base+ 80_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
TxRqst32 to TxRqs25								
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

- **CAN Transmission Request Register 2 (lower byte): Address Base+ 81_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
TxRqst24 to TxRqst17								
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

- **CAN Transmission Request Register 1 (upper byte): Address Base+ 82_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
TxRqst16 to TxRqst9								
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

- **CAN Transmission Request Register 1 (lower byte): Address Base+ 83_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
TxRqst8 to TxRqst1								
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

- **TxRqst32 to TxRqst1: Transmission request bits**

TxRqst32-1	Function
0	Indicates the transmission idle state (neither transmission is in progress nor in the transmission wait state).
1	Indicates that transmission is in progress or in the transmission wait state.

Set/reset conditions of the TxRqst bit are shown below.

● Set condition

It is possible to set the TxRqst of a specific object by setting "1" to the WR/RD of IFx command mask register (IFxCMSK) and "1" to the TxRqst while writing data to the IFx command request register (IFxCREQ).

When WR/RD of IFx command mask register (IFxCMSK) is set to "1", TxRqst is set to "0", Control is set to "1", and 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 WR/RD of IFx command mask register (IFxCMSK) is set to "1", TxRqst is set to "0", Control is set to "1", and 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 regarding the transmission request bit of the CAN macro that equips 64 message buffers.

Table 36.4-7 Transmission Request Bit of the CAN Macro that equips 64 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
TREQR 4 & 3	TxRqst 64-33 (address 84_H)	TxRqst64-57	TxRqst56-49	TxRqst48-41	TxRqst40-33

(Notes)
- When the message buffer with the lowest priority is used for transmission and TXRqst is set to "1" and then to "0" to cancel transmission, setting 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 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 TxRqst to "1" again.
- When 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 TxRqst bit is "0".

■ CAN Data Update Registers1, 2 (NEWDT1, NEWDT2)

These registers display the NewDat bit of all message objects. It is possible to check which message objects data has been updated by reading the NewDat bit.

- **CAN Data Update Register 2 (upper byte): Address Base + 90_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
NewDat32 to NewDat25								
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

- **CAN Update Data Register 2 (lower byte): Address Base+91_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
NewDat24 to NewDat17								
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

- **CAN Date Update Register 1 (upper byte): Address Base+92_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
NewDat16 to NewDat9								
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

- **CAN Update Data Register 2 (lower byte): Address Base+93_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
NewDat8 to NewDat1								
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

- **NewDat32 to NewDat1: Data update bits**

NewDat32-1	Function
0	Indicates no valid data exists
1	Indicates valid data exists

Set/reset conditions of the NewDat bit are shown below.

● Set condition

When WR/RD of IFx command mask register (IFxCMSK) is set to "1", Control is set to "1", and NewDat of the IFx message control register (IFxMCTR) is set to "1", it is possible to set the bit of 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", RmtEN is set to "0", and 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 WR/RD of IFx command mask register (IFxCMSK) is set to "0" and 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 WR/RD of IFx command mask register (IFxCMSK) is set to "1", Control is set to "1", and 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 regarding the data update bit of the CAN macro that equips 64 message buffers.

Table 36.4-8 Data Update Bit of the CAN Macro that equips 64 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
NEWDT 4 & 3	NewDat 64-33 (address 94_H)	NewDat64-57	NewDat56-49	NewDat48-41	NewDat40-33

■ CAN Interrupt Pending Registers1, 2 (INTPND1, INTPND2)

These registers display the IntPnd bit of all message objects. It is possible to check which message objects interrupt is pending by reading the IntPnd bit.

- **CAN Interrupt Pending Register 2 (upper byte): Address Base + $A0_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
IntPnd32 to IntPnd25								
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

- **CAN Interrupt Pending Register 2 (lower byte): Address Base + $A1_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
IntPnd24 to IntPnd17								
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

- **CAN Interrupt Pending Register 1 (upper byte): Address Base + $A2_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
IntPnd16 to IntPnd9								
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

- **CAN Interrupt Pending Register 1 (lower byte): Address Base + $A3_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
IntPnd8 to IntPnd1								
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

- **IntPnd32 to IntPnd1: Interrupt pending bits**

IntPnd32-1	Function
0	No interrupt factor exists
1	An interrupt factor exists

Set/reset conditions of the IntPnd bit are shown below.

● Set condition

If TxIE is set to "1", IntPnd bit will be set after the frame transmission has ended successfully.

If RxIE is set to "1", the bit will be set after the frame reception that has passed the acceptance filter completed successfully.

When WR/RD of IFx command mask register (IFxCMSK) is set to "1", Control is set to "1", and 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.

● Reset condition

When WR/RD of IFx command mask register (IFxCMSK) is set to "0" and 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 WR/RD of IFx command mask register is set to "1", Control is set to "1", and IntPnd of the IFx message control register 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 regarding the interrupt pending bit of the CAN macro that equips 64 message buffers.

Table 36.4-9 Interrupt Pending Bit of the CAN Macro that equips 64 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
INTPND 4 & 3	IntPnd 64-33 (address $A4_H$)	IntPnd64-57	IntPnd56-49	IntPnd48-41	IntPnd40-33

■ CAN Message Valid Registers1, 2 (MSGVAL1, MSGVAL2)

These registers display the MsgVal bit of all message objects. It is possible to check which message object is valid by reading the MsgVal bit.

- **CAN Message Valid Register 2 (upper byte): Address Base + $B0_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
MsgVal32 to MsgVal25								
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

- **CAN Message Valid Register 2 (lower byte): Address Base + $B1_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
MsgVal24 to MsgVal17								
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

- **CAN Message Valid Register 1 (upper byte): Address Base + $B2_H$**

(Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
MsgVal16 to MsgVal9								
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

- **CAN Message Valid Register 1 (lower byte): Address Base + $B3_H$**

(Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
MsgVal 8 to MsgVal1								
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

- **MsgVal32 to MsgVal1: Message valid bits**

MsgVal32-1	Function
0	Message object is invalid. Message will not be transmitted/received.
1	Message object is valid. Message transmission/reception is possible.

Set/reset conditions of the MsgVal bit are shown below.

● Set condition

When WR/RD of IFx command mask register is set to "1", 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 WR/RD of IFx command mask register is set to "1", 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 regarding the message valid bit of the CAN macro that equips 64 message buffers.

Table 36.4-10 Message Valid Bit of the CAN Macro that equips 64 Message Buffers

		addr + 0	addr + 1	addr + 2	addr + 3
MSGVAL 4 & 3	MsgVal 64-33 (address $B4_H$)	MsgVal64-57	MsgVal56-49	MsgVal48-41	MsgVal40-33

36.4.5 CAN Prescaler Register

This register sets the CAN system clock (fsys) generation prescaler. For details, see Section "36.5.6 Bit Timing and CAN System Clock (fsys) Generation". To change the value of this register, set the initialization bit (Init) in the CAN control register (CTRLR) to "1" to stop all the bus operations.

- **CAN Prescaler Register: Address $04A4_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved	Reserved		CPCKS	CANPRE3	CANPRE2	CANPRE1	CANPRE0	
0	0	0	0	0	0	0	0	Initial value
R/W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7] Reserved bit

Always write "0" to this bit.

[bit6, bit5] Reserved bits

The read value is always "0". When writing to these bits, set "0".

[bit4] CPCKS: CAN prescaler clock selection bits

CPCKS	CAN prescaler source clock
0	PLL clock (PLLCLK)/On-chip bus clock (HCLK)
1	Main clock (MCLK)

[bit3 to bit0] CANPRE3 to CANPRE0: CAN prescaler setting bits

CANPRE [3:0]	Function	Input CAN prescaler clock: 128MHz	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)	128MHz	80MHz	64MHz	48MHz
0001	Selects 1/2 period of the system clock as the CAN clock.	64MHz	40MHz	32MHz	24MHz
001x	Selects 1/4 period of the system clock as the CAN clock.	32MHz	20MHz	16MHz	12MHz
01xx	Selects 1/8 period of the system clock as the CAN clock.	16MHz	10MHz	8MHz	6MHz
1000	Selects 2/3 period of the system clock as the CAN clock. The clock duty is 67%.	85.3MHz	53.3MHz	42.7MHz	32MHz
1001	Selects 1/3 period of the system clock as the CAN clock.	42.7MHz	26.7MHz	21.4MHz	16MHz
1010	Selects 1/6 period of the system clock as the CAN clock.	21.3MHz	13.3MHz	10.7MHz	8MHz
1011	Selects 1/12 period of the system clock as the CAN clock.	10.7MHz	6.7MHz	5.4MHz	4MHz
110x	Selects 1/5 period of the system clock as the CAN clock.	25.6MHz	16.0MHz	12.8MHz	9.6MHz
111x	Selects 1/10 period of the system clock as the CAN clock.	12.8MHz	8.0MHz	6.4MHz	4.8MHz

(Notes)
- Change the CAN prescaler setting bits after setting the initialization bit of the CAN control register (CTRLR) to "1" and stopping all the bus operations.
- The clock to be supplied to the CAN interface using the register setting must be 16MHz or less.

36.5 Operation

CAN has the following functions:

- Message object
- Message transmission operation
- Message reception operation
- FIFO buffer function
- Interrupt function
- Bit timing
- Test mode
- Software initialization

36.5.1 Message Object

This section explains the message object and interface of message RAM.

■ Message Object

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 MsgVal bit (MsgVal=0). Set CAN bit timing register (BTR) and CAN prescaler extension register (BRPER) while Init bit of the CAN control register (CTRLR) is set to "1" and CCE bit is set to "1".

Message object can be set by setting the data to the message interface register (IFx mask register, IFx arbitration register, 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.

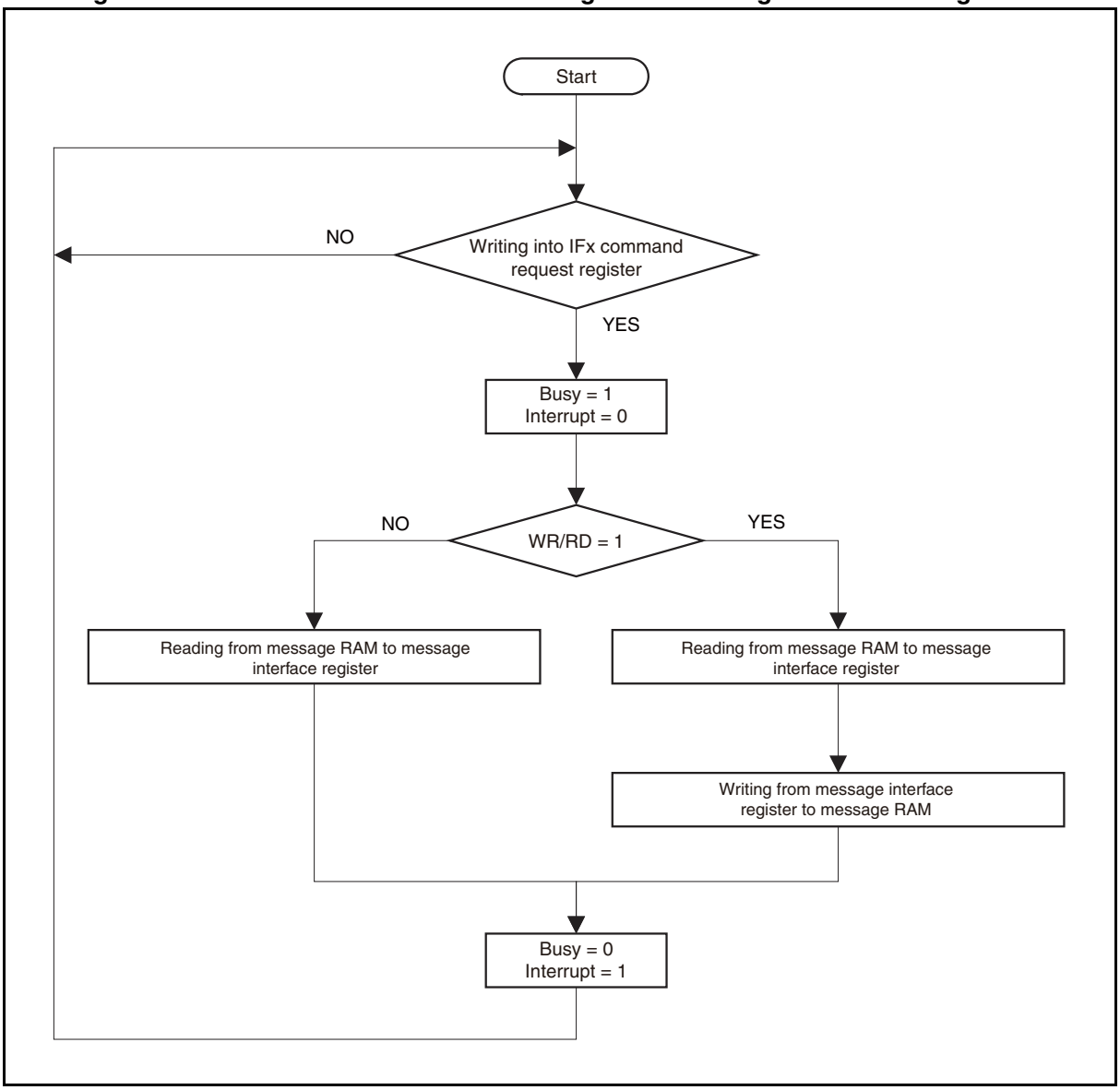
CPU reads the reception message via the message interface register and updates the transmission message. An interrupt is sent to the CPU according to the settings of the CAN control register (CTRLR) and IFx message control register (IFxMCTR) (message object).

■ Data Transmission/Reception with Message RAM

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. Busy bit will be cleared to "0" after the transfer completion (see Figure 36.5-1).

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 36.5-1 Data Transfer between Message Interface Register and Message RAM



36.5.2 Message Transmission Operation

This section explains the setting method and transmission operation of the transmission message object.

■ Message Transmission

If there is no data transfer between the message interface register and message RAM, MsgVal bit of the CAN message valid register (MSGVAL) and 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. NewDat bit of the message object will be reset to "0" at this time.

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 TxIE is set to "1", 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.

■ Transmission Priority

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 transmitted 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 TXRqst is set to "1" and then to "0" to cancel transmission, setting 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 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 TxRqst to "1" again.
- When 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 TxRqst bit is "0".

■ Transmission Message Object Setting

Table 36.5-1 shows the initialization method for the transmission object.

Table 36.5-1 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

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 TxIE bit is set to "1", IntPnd bit will be set to "1" after the transmission of the message object has ended successfully.

If RmtEn bit is set to "1", TxRqst bit will be set to "1" and the data frame will be transmitted automatically after receiving the matching remote frame.

Settings for data registers (DLC3 to DLC0, Data0 to Data7) are provided by the application.

When UMask=1, IFx mask register (Msk28 to Msk0, 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 TxRqst bit to "1"). See the heading "Remote frame" in Section "36.5.3 Message Reception Operation" for details.

(Note) Mask is not allowed for Dir bit of the IFx mask register.

■ Update of Transmission Message Object

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 TxRqst bit at the same time when a message number is written to the IFx command request register (IFxCREQ).

If NewDat bit and TxRqst bit are both "1", 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 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 TxRqst bit is "0".

36.5.3 Message Reception Operation

This section explains the setting method and reception operation of the reception message object.

■ Reception Message Acceptance Filter

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).

■ Reception Priority

Reception priority of a message object is determined by its message number. Message object 1 has the highest priority; and message object 32 (or the maximum equipped message object number) has the lowest priority. If 2 or more message objects match the acceptance filter, the one having the smaller message number will be the reception message object.

■ Data Frame Reception

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).

NewDat bit will be set to "1" when a new data is received. Reset NewDat bit to "0" when a message object is read by the CPU. If NewDat bit is already set to "1" when the message is received, the previous data will be treated as lost and MsgLst bit will be set to "1".

If RxIE bit is set to "1", IntPnd bit of the CAN interrupt pending register (INTPND) will be set to "1" when a message buffer is received. 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.

■ Remote Frame

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 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 TxRqst bit.)

2) Dir=1 (Transmission direction), RmtEn=0, UMask=0

Remote frame will be disabled without receiving the message, even if the received remote frame matches the message object. (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, 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 NewDat bit of this message object will be set to "1". Data field of the message object will be unchanged.

■ Reception Message Object Setting

Table 36.5-2 shows the initialization method for the reception message object.

Table 36.5-2 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

IFx arbitration register (ID28 to ID0 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 RxIE bit is set to "1", IntPnd bit will be set to "1" when the reception data frame is stored to the message object.

Data length code (DLC3 to DLC0) 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, IFx mask register (Msk28 to Msk0, 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 Section "36.5.3 Message Reception Operation" for details.

(Note) Dir bit of the IFx mask register cannot be set as a mask.

■ Reception Message Processing

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, 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.

NewDat bit indicates whether a new message has been received after the message object was last read.

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. MsgLst bit will not be reset automatically.

TxRqst bit will be clear to "0" automatically when a data frame matching the acceptance filter is received while the remote frame transmission is being processed.

36.5.4 FIFO buffer function

This section explains the configuration and operation of the FIFO buffer of the message object in the reception message processing.

■ Configuration of FIFO Buffer

Configuration of the reception message objects in the FIFO buffer is the same as that of other reception message objects, except for EoB bit (see Section "36.5.3 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). 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 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 EoB bit to "1" when FIFO buffer is not used.

■ Message Reception by FIFO Buffer

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.

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 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 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.

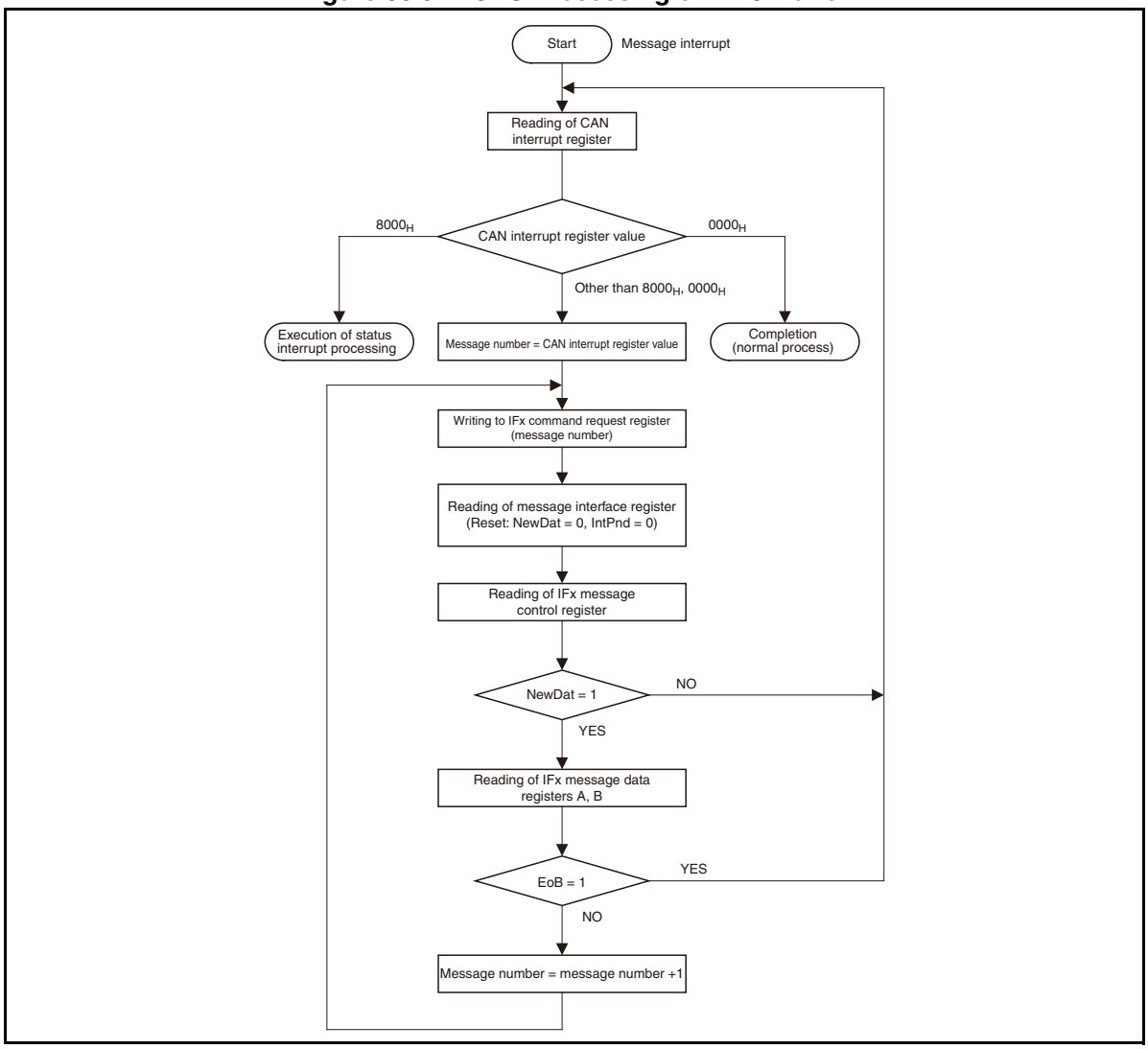
■ Reading from FIFO Buffer

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 WR/RD to "0" (read), set TxRqst/NewDat and 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.

Figure 36.5-2 shows the CPU processing method for the message objects that are linked in the FIFO buffer.

Figure 36.5-2 CPU Processing of FIFO Buffer



36.5.5 Interrupt Function

This section explains the processing of status interrupt (IntId=8000_H) and message interrupt (IntId=message number).

If 2 or more interrupts are pending, 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 IntPnd bit of the message object is cleared. Status interrupt will be cleared when the CAN status register (STATR) is read.

IntPnd bit of the CAN interrupt pending register (INTPND) indicates whether any interrupt exists. IntPnd bit will indicate "0" if there is no pending interrupt.

The interrupt signal to the CPU will become active when IndPnd bit becomes "1" while 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".

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 IE bit of the CAN control register (CTRLR).

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.

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.

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 Section "36.4.4 Message Handler Register".) 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. 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.

36.5.6 Bit Timing and CAN System Clock (fsys) Generation

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 quartz 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 36.5-4 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 36.5-3 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 "Figure 36.5-3 Schematic Diagram of CAN System Clock (fsys) Generation". 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 36.5-3 Schematic Diagram of CAN System Clock (fsys) Generation

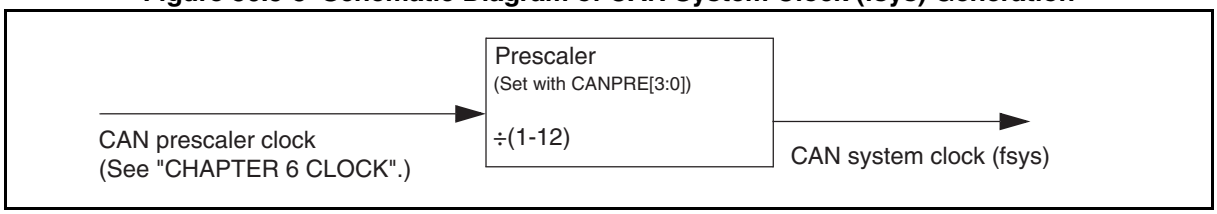


Figure 36.5-4 Bit timing

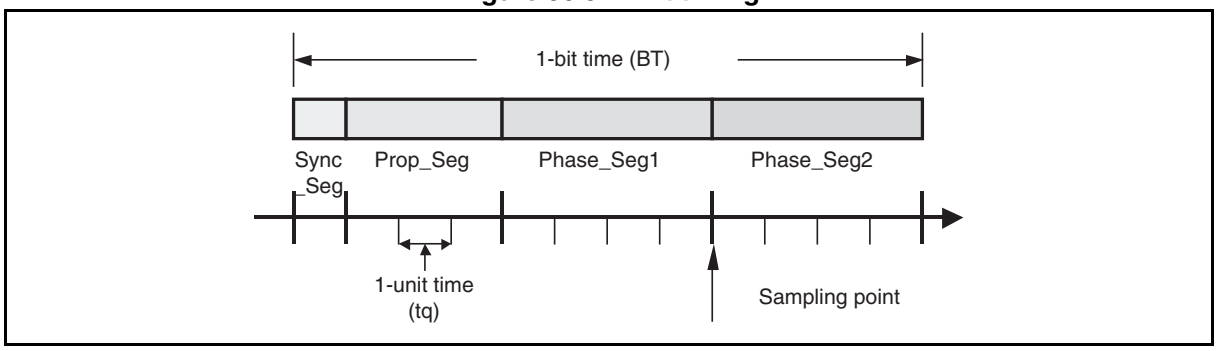


Table 36.5-3 CAN Bit Time Parameters

Parameter	Range	Function
BRP	[1-32]	Defines the time quantity tq.
Sync_Seg	1 tq	Synchronizes the bit time with a fixed-length system clock.
Prop_Seg	[1-8] tq	Compensates for physical delay time.
Phase_Seg1	[1-8] tq	Guarantees identification of edge-phase errors prior to the sample point. The bit time may be temporarily prolonged due to synchronization.
Phase_Seg2	[1-8] tq	Guarantees identification of edge-phase errors subsequent to the sample point. The bit time may be temporarily shortened due to synchronization.
SJW	[1-4] 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 Figure 36.5-5.

Figure 36.5-5 Bit Timing Effected by the CAN Controller

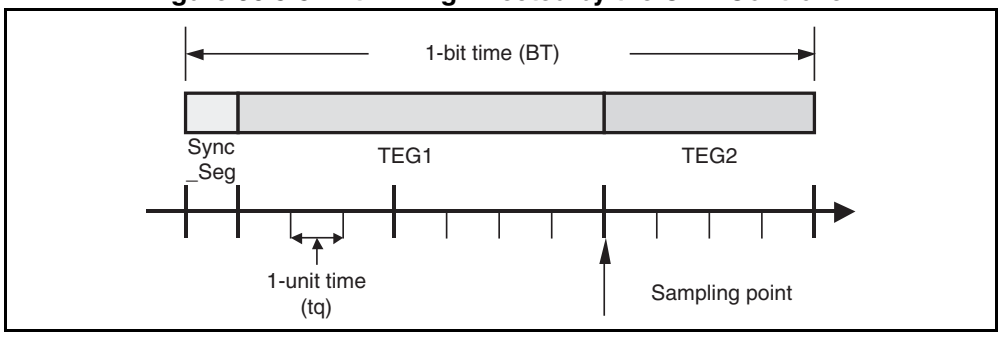


Table 36.5-4 CAN Controller Parameters

Parameter	Range	Function
BRPE, BRP	[0-1023]	Defines the time quantity tq. The prescaler can be extended up to 1024 using the bit timing and prescaler extension registers.
SYNS_SEG	1 tq	Synchronizes the bit time with the system clock. Fixed length
TSEG1	[1-15] tq	Time segment prior to the sample point. This corresponds to Prop_Seg and Phase_Seg1. This segment can be controlled using the bit timing register.
TSEG2	[0-7] tq	Time segment subsequent to the sample point. This corresponds to Phase_Seg2. This segment can be controlled using the bit timing register.
SJW	[0-3] 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

36.5.7 Test Mode

This section explains the test mode setting method and operation.

■ Test Mode Setting

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".

■ Silent Mode

The CAN controller enters silent mode when the Silent bit of the CAN test register (TESTR) is set to "1".

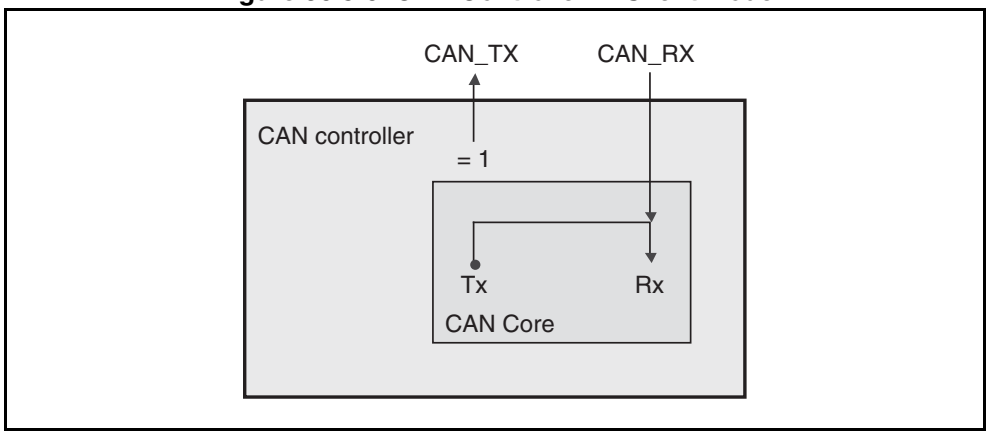
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs a recessive level to the CAN bus and does not send messages or ACKs.

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).

Figure 36.5-6 shows how signals CAN_TX and CAN_RX are connected to the CAN controller in silent mode.

Figure 36.5-6 CAN Controller in Silent Mode



■ Loopback Mode

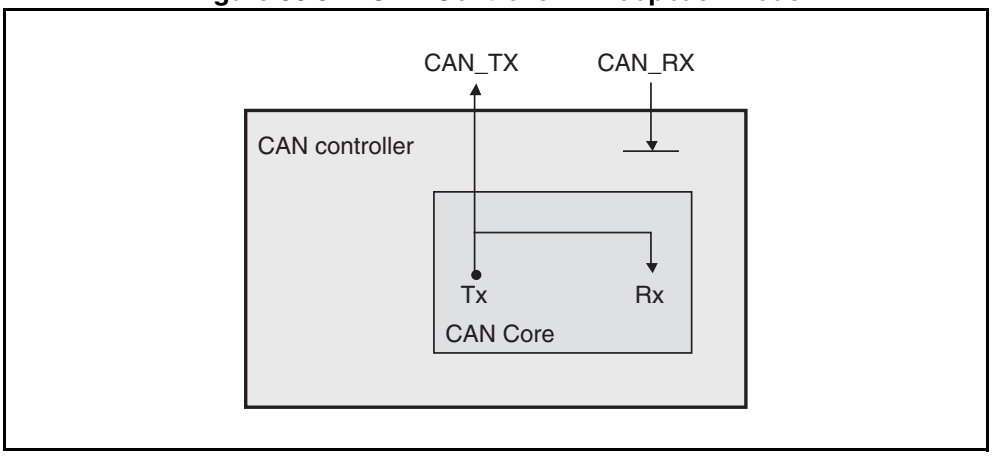
The CAN controller enters loopback mode when the LBack bit of the CAN test register (TESTR) is set to "1".

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.

Figure 36.5-7 shows how signals CAN_TX and CAN_RX are connected to the CAN controller in loopback mode.

Figure 36.5-7 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.

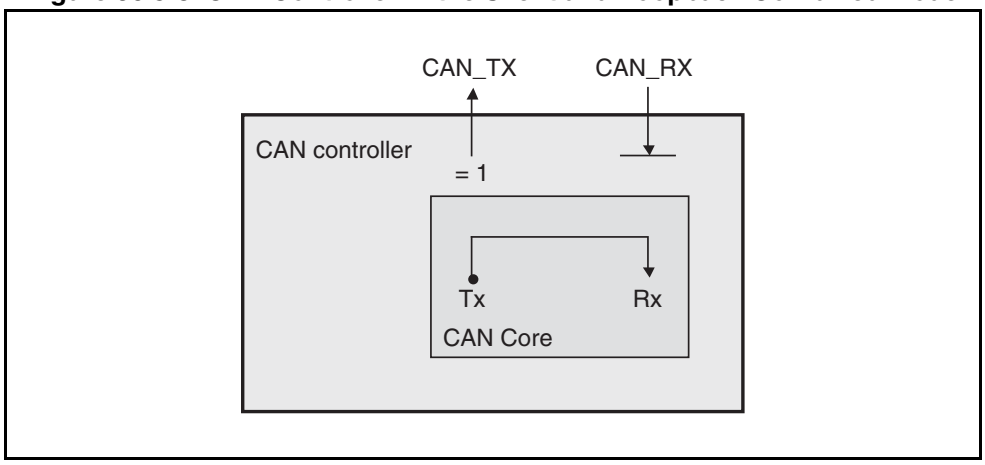
■ Combination of Silent and Loopback Modes

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.

Figure 36.5-8 shows how signals CAN_TX and CAN_RX are connected to the CAN controller in the silent and loopback combined mode.

Figure 36.5-8 CAN Controller in the Silent and Loopback Combined Mode



■ Basic Mode

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 DLC0 bits identify the received DLC, and the other control bits are read as "0".

■ Software Control of the CAN_TX Pin

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.

36.5.8 Software Initialization

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 11 consecutive recessive bits (indicating a bus-idling state) are observed 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 mask, ID, xtd, EoB, and/or RmtEn needs to be changed during ordinary operation, change it after invalidating the MsgVal bit.

36.5.9 CAN Wakeup Function

You can wake up with the receive operation of CAN by connecting CAN RX pin to an external interrupt pin.

■ Pins Used for CAN Wakeup Function

Since RX0 and INT1 pins, or RX1 and INT7 pins are shared, the wakeup function is available for these pins.

The relation of CAN wakeup function and RX pin/INT pin is shown in Table 36.5-5.

Table 36.5-5 Relation of CAN Wakeup Function and RX Pin/INT Pin

	RX Pin	Interrupt Function
CAN0	RX0	INT1
CAN1	RX1	INT7

■ CAN Wakeup Function

Return from sleep mode or standby mode by CAN reception data is enabled.

(Note) When the wakeup function is used, the external interrupt setting before transition to sleep mode or standby mode is required.

CHAPTER 37 FLEXRAY

37.1 Overview

The FlexRay controller communicates based on the FlexRay protocol specification v2.1. The bit rate will be set to 10 Mbit/s when the maximum system clock is specified.

The message buffer of up to 254 data byte length can be allocated for FlexRay network communication. The message storage area consists of single-port message RAMs having up to 128 of message buffers. The message handler performs the functions related to all message processing. These functions are shown below.

- Acceptance filter
- Message transfer between two FlexRay channel protocol controllers and the message RAM
- Transmission schedule management
- Provision of massage status information

The FlexRay controller registers can be accessed by the host. These registers are used for setting, controlling and monitoring the following:

- FlexRay channel protocol controller
- Message handler
- Global time unit
- System universal control
- Frame and symbol processing
- Network management
- Interrupt control
- Access to the message RAM via input/output buffers

37.2 Features

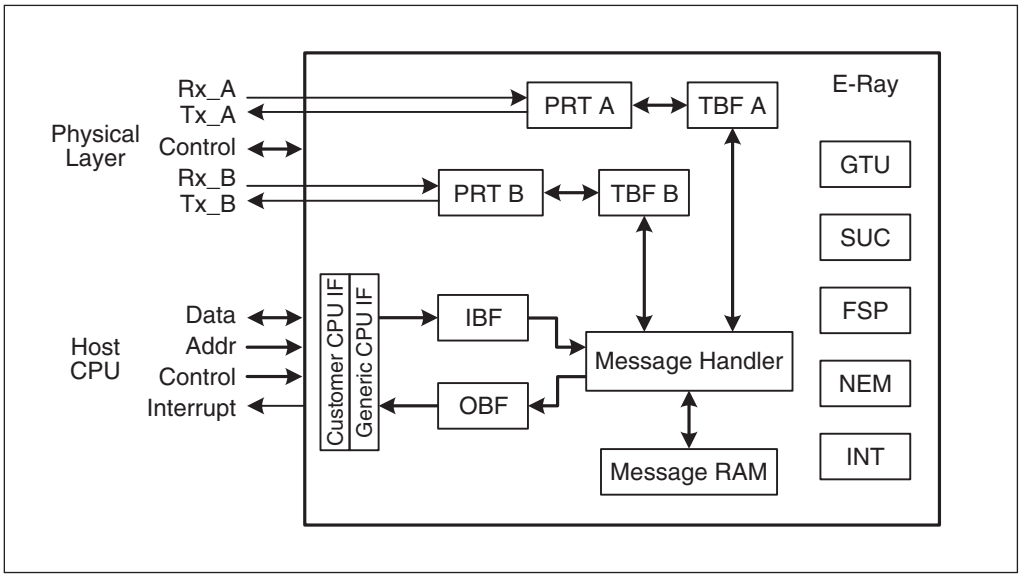
The FlexRay controller supports the following functions.

- Supports FlexRay protocol specification v2.1
- Up to 10 Mbit/s of bit rate for each channel
- Up to 128 message buffers can be allocated
- 8 Kbytes of message RAM (equivalent to the memory capacity described below)
  - For the maximum 48 bytes of data section, 128 message buffers
  - For the maximum 254 bytes of data section, 30 message buffers
- Variable length of message buffers
- A single configurable reception FIFO
- Each message buffer can be allocated as a part of reception buffer, transmission buffer or reception FIFO
- Host access to the message buffer via input and output buffers
  - Input buffer: Stores messages to be transferred to the message RAM
  - Output buffer: Stores messages read from the message RAM
- Filtering for slot counter, cycle counter and channels
- Maskable interrupts are supported
- Network management support

Code: eray-1v2.7-91580MS-1-E

■ Block diagram for FlexRay controller

Figure 37.2-1 Block diagram for FlexRay controller



● Functional description for each block

CPU interface (CIF: CPU Interface)

Connects the host CPU to the FlexRay controller.

Input buffer (IBF: Input Buffer)

This buffer is used for writing to message buffers configured in the message RAM.

The host CPU can write header and data sections from the input buffer to the specific message buffer.

The message handler transfers data from the input buffer to the message buffer selected by the message RAM.

Output buffer (OBF: Output Buffer)

This buffer is used for reading from the message buffers configured in the message RAM.

The message handler transfers data from the selected message buffer to the output buffer.

After this data transfer is completed, the host CPU can read the header and data sections of the message buffer from the output buffer.

Message handler (MHD: Message Handler)

The message handler controls data transfer between the following components.

- Input/output buffer and message RAM
- Temporary storage buffer RAM and message RAM for two FlexRay protocol controllers

Message RAM (MRAM: Message RAM)

The message RAM consists of single-port RAMs which store the configuration data (header and data) in the FlexRay message buffers (up to 128 pieces).

■ Temporary storage buffer RAM(TBF A/B: Transient Buffer RAM)

This RAM stores data sections for two messages.

■ FlexRay channel protocol controller (PRT A/B: FlexRay Channel Protocol Controller)

The FlexRay channel protocol controller consists of the shift register and the FlexRay protocol FSM.
This controller provides the following functions.

- Check and control of bit timings
- Reception and transmission of FlexRay frames and symbols
- Check of header CRC
- Generation and check of frame CRC
- Connection to the bus driver

This block connects to the following blocks.

- Physical layer (bus driver)
- Temporary storage buffer RAM
- Message handler
- Global time unit
- System universal control
- Frame and symbol processing
- Network management
- Interrupt control

■ Global time unit (GTU: Global Time Unit)

The global time unit provides the following functions.

- Microtick generation
- Macrotick generation
- Fault-tolerant clock synchronization by the FTM algorithm
  - Rate compensation
  - Offset compensation
- Cycle counter
- Static segment timing control
- Dynamic segment (mini slot) timing control
- Support of external clock compensation

■ System universal control (SUC: System Universal Control)

The system universal control provides the following functions.

- Configuration
- Wakeup
- Startup
- Normal operation
- Passive operation
- Monitor mode

■ Frame and symbol processing (FSP: Frame and Symbol Processing)

Frame and symbol processing controls the following functions.

- Correct timing check for frames and symbols
- Validity test for syntax and semantics of received frames
- Slot status flag setting

■ Network management (NEM: Network Management)

The network management provides the following functions:

- Handling of network management vector

■ Interrupt control (INT: Interrupt Control)

The interrupt control performs the following functions:

- Supply of error and interrupt flag
- Enable/disable control of interrupt causes
- Control of allocation of interrupt factors to two module interrupt lines
- Enable/disable of two module interrupt lines
- Management of two interrupt timers
- Stop of watch time capturing

37.3 Configuration of FlexRay

The FlexRay controller has 2K bytes of address space (0xD000 to 0xD7FF) and 32-bit registers. Host access to the message RAM (access from host CPU) is provided via the input and output buffers. These buffers buffer data transferred to/from the message RAM in order to avoid contention between host access and message transmission/reception.

The number of available message buffers N depends on the payload length of configured message buffers. The maximum number of message buffers is 128 pieces and the maximum payload length is 254 bytes.

Allocation of message buffers is shown in Figure 37.3-1. The message buffers are classified into three consecutive groups.

Static buffer	- Transmission/reception buffers allocated to static segments
Static and dynamic buffer	- Transmission/reception buffers allocated to static or dynamic segments
FIFO	- Reception FIFO

The allocation of message buffers can be changed by setting the message RAM setting register **MRC** in the DEFAULT_CONFIG or CONFIG state.

The first group works as static message buffers.

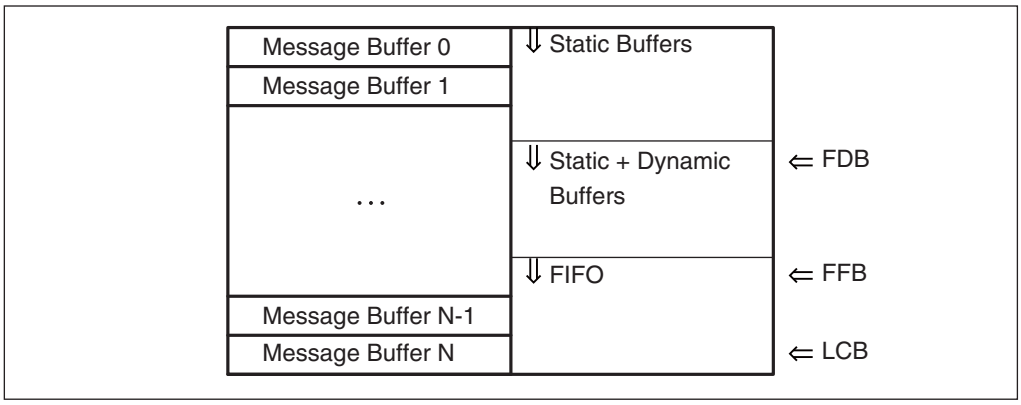
The second group works as static/dynamic message buffers. The message buffers belonging to this group may be reset from dynamic segment to static segments or from static segment to dynamic segments during operation according to the **MRC.SEC [1:0]** status.

The message buffers belonging to the third group are coupled to one reception FIFO.

The message buffer 0 is the static message buffer used to store and transmit startup/synchronization frame or single slot frame (frames transmitted in the single slot mode) according to the **SUCC1.TXST**, **SUCC1.TXSY**, and **SUCC1.TSM** settings. The message buffer 0 needs to incorporate key slot IDs and can be set (reset) only in the DEFAULT_CONFIG or CONFIG state.

The FDB, FFB, and LCB in Figure 37.3-1 indicate the first dynamic buffer number, **MRC.FDB [7:0]**; the first FIFO buffer number, **MRC.FFB [7:0]**; and the final message buffer number, **MRC.LCB [7:0]**, respectively.

Figure 37.3-1 Allocation of message buffers



(Note) All registers for the FlexRay controller are accessed in 32 bits.

Table 37.3-1 List of Registers of FlexRay (1 / 4)

Address	+0	+1	+2	+3
Customer Register				
0xD000	Version information register (CIFO)			
0xD004	Control register(CIF1)			
0xD008 - 0xD00C	Reserved			
Special Registers				
0xD010	Reserved (write prohibited)			
0xD014	Reserved (write prohibited)			
0xD018	Reserved			
0xD01C	Lock registers (LCK)			
Interrupt-Related Register				
0xD020	Error interrupt registers (EIR)			
0xD024	Status interrupt register (SIR)			
0xD028	Error interrupt pin selection register (EILS)			
0xD02C	Status interrupt pin selection register (SILS)			
0xD030	Error interrupt effective register (set) (EIES)			
0xD034	Error Interrupt effective Registers (reset) (EIER)			
0xD038	Status interrupt effective register (set) (SIES)			
0xD03C	Status interrupt effective register (reset) (SIER)			
0xD040	Interrupt pin effective register (ILE)			
0xD044	Timer 0 setting register 0 (T0C)			
0xD048	Timer 0 setting register 1 (T1C)			
0xD04C	Stopwatch register 1 (STPW1)			
0xD050	Stopwatch register 2 (STPW2)			

Table 37.3-1 List of Registers of FlexRay (2 / 4)

Address	+0	+1	+2	+3
0xD054 - 0xD07C	Reserved			
Communication Controller (CC) Control Registers				
0xD080	SUC setup register 1 (SUCC1)			
0xD084	SUC setup register 2 (SUCC2)			
0xD088	SUC setup register 3 (SUCC3)			
0xD08C	NEM setup register (NEMC)			
0xD090	PRT setup register 1 (PRTC1)			
0xD094	PRT setup register 2 (PRTC2)			
0xD098	MHD setup register (MHDC)			
0xD09C	Reserved			
0xD0A0	GTU setup register 1 (GTUC1)			
0xD0A4	GTU setup register 2 (GTUC2)			
0xD0A8	GTU setup register 3 (GTUC3)			
0xD0AC	GTU setup register 4 (GTUC4)			
0xD0B0	GTU Setting Register 5 (GTUC5)			
0xD0B4	GTU Setting Register 6 (GTUC6)			
0xD0B8	GTU Setting Register 7 (GTUC7)			
0xD0BC	GTU Setting Register 8 (GTUC8)			
0xD0C0	GTU Setting Register 9 (GTUC9)			
0xD0C4	GTU Setting Register 10 (GTUC10)			
0xD0C8	GTU Setting Register 11 (GTUC11)			
0xD0CC - 0xD0FC	Reserved			
Communication Controller (CC) Status Register				
0xD100	CC Status Vector Register (CCSV)			
0xD104	CC error vector register (CCEV)			
0xD108 -0xD10C	Reserved			
0xD110	Slot Counter Value Register (SCV)			
0xD114	Macrotick and Cycle Counter Value Register (MTCCV)			
0xD118	Rate compensation value register (RCV)			
0xD11C	Offset compensation value register (OCV)			
0xD120	Synchronous frame status register (SFS)			
0xD124	Symbol Window and NIT Status Register (SWNIT)			
0xD128	Collective channel status register (ACS)			
0xD12C	Reserved			
0xD130	Even-number cycle synchronous frame ID register 1 (ESID1)			
0xD134	Even-number cycle synchronous frame ID register 2 (ESID2)			
0xD138	Even-number cycle synchronous frame ID register 3 (ESID3)			
0xD13C	Even-number cycle synchronous frame ID register 4 (ESID4)			
0xD140	Even-number cycle synchronous frame ID register 5 (ESID5)			
0xD144	Even-number cycle synchronous frame ID register 6 (ESID6)			
0xD148	Even-number cycle synchronous frame ID register 7 (ESID7)			
0xD14C	Even-number cycle synchronous frame ID register 8 (ESID8)			
0xD150	Even-number cycle synchronous frame ID register 9 (ESID9)			
0xD154	Even-number cycle synchronous frame ID register 10 (ESID10)			
0xD158	Even-number cycle synchronous frame ID register 11 (ESID11)			
0xD15C	Even-number cycle synchronous frame ID register 12 (ESID12)			
0xD160	Even-number cycle synchronous frame ID register 13 (ESID13)			

Table 37.3-1 List of Registers of FlexRay (3 / 4)

Address	+0	+1	+2	+3
0xD164	Even-number cycle synchronous frame ID register 14 (ESID14)			
0xD168	Even-number cycle synchronous frame ID register 15 (ESID15)			
0xD16C	Reserved			
0xD170	Odd-number cycle synchronous frame ID register 1 (OSID1)			
0xD174	Odd-number cycle synchronous frame ID register 2 (OSID2)			
0xD178	Odd-number cycle synchronous frame ID register 3 (OSID3)			
0xD17C	Odd-number cycle synchronous frame ID register 4 (OSID4)			
0xD180	Odd-number cycle synchronous frame ID register 5 (OSID5)			
0xD184	Odd-number cycle synchronous frame ID register 6 (OSID6)			
0xD188	Odd-number cycle synchronous frame ID register 7 (OSID7)			
0xD18C	Odd-number cycle synchronous frame ID register 8 (OSID8)			
0xD190	Odd-number cycle synchronous frame ID register 9 (OSID9)			
0xD194	Odd-number cycle synchronous frame ID register 10 (OSID10)			
0xD198	Odd-number cycle synchronous frame ID register 11 (OSID11)			
0xD19C	Odd-number cycle synchronous frame ID register 12 (OSID12)			
0xD1A0	Odd-number cycle synchronous frame ID register 13 (OSID13)			
0xD1A4	Odd-number cycle synchronous frame ID register 14 (OSID14)			
0xD1A8	Odd-number cycle synchronous frame ID register 15 (OSID15)			
0xD1AC	Reserved			
0xD1B0	Network management register 1 (NMV1)			
0xD1B4	Network management register 2 (NMV2)			
0xD1B8	Network management register 3 (NMV3)			
0xD1BC - 0xD2FC	Reserved			
Message Buffer Control Registers				
0xD300	Message RAM setting register (MRC)			
0xD304	FIFO rejection filter register (FRF)			
0xD308	FIFO rejection filter mask register (FRFM)			
0xD30C	FIFO critical level register (FCL)			
Message Buffer Status Register				
0xD310	Message handler status register (MHDS)			
0xD314	Final dynamic transmission slot register (LDTS)			
0xD318	FIFO status register (FSR)			
0xD31C	Message handler limitation flag (MHDF)			
0xD320	Transmission request register 1 (TXRQ1)			
0xD324	Transmission request register 2 (TXRQ2)			
0xD328	Transmission request register 3 (TXRQ3)			
0xD32C	Transmission request register 4 (TXRQ4)			
0xD330	New data register 1 (NDAT1)			
0xD334	New data register 2 (NDAT2)			
0xD338	New data register 3 (NDAT3)			
0xD33C	New data register 4 (NDAT4)			
0xD340	Message buffer status update register 1 (MBSC1)			
0xD344	Message buffer status update register 2 (MBSC2)			
0xD348	Message buffer status update register 3 (MBSC3)			
0xD34C	Message buffer status update register 4 (MBSC4)			
0xD350 - 0xD3EC	Reserved			
Identification Register				
0xD3F0	Core release register (CREL)			

Table 37.3-1 List of Registers of FlexRay (4 / 4)

Address	+0	+1	+2	+3
0xD3F4	Endian register (ENDN)			
0xD3F8 - 0xD3FC	Reserved			
Input Buffer				
0xD400 - 0xD4FC	Write data section register [1-64] (WRDSn)			
0xD500	Write header section register 1 (WRHS1)			
0xD504	Write header section register 2 (WRHS2)			
0xD508	Write header section register 3 (WRHS3)			
0xD50C	Reserved			
0xD510	Input buffer command mask register (IBCM)			
0xD514	Input buffer command request register (IBCR)			
0xD518 - 0xD5FC	Reserved			
Output Buffer				
0xD600 - 0xD6FC	Read data section register [1-64] (RDDSn)			
0xD700	Read header section register 1 (RDHS1)			
0xD704	Read header section register 2 (RDHS2)			
0xD708	Read header section register 3 (RDHS3)			
0xD70C	Message buffer status register (MBS)			
0xD710	Output buffer command mask register (OBCR)			
0xD714	Output Buffer Command Request Register (OBCR)			
0xD718 - 0xD7FC	Reserved			

37.3.1 Customer register

The address space from 0xD000 to 0xD007 are customer registers. Version information, and FlexRay control (DMA support. interrupt register, FlexRay reset, and buffer data SWAP) are allocated.

37.3.1.1 Version information register (CIFO)

CIF0	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	Initial value
0xD000	VERSION																04FF_5BFF
	R,WX																

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	VERSION															
	R,WX															

No.	Function name	Function
bit31 to bit24	Manufacturer ID code	Manufacturer ID code is set. 0x04 is read.Data writing is ineffective.
bit23 to bit16	Version bits	These bits indicate LSI versions. 0xFF is read. Data writing is ineffective.
bit15 to bit8	LSI identification number bits	These bits indicate LSI identification numbers. 0x5B is read. Data writing is ineffective.
bit7 to bit0	FlexRay IP identification bits	These bits indicate FlexRay IP identification numbers. 0xFF is read.When these bits are 0xFF, IP information is stored in the CREL register. Read this information when the CREL register is necessary. Data writing is ineffective.

37.3.1.2 Control register(CIF1)

CIF1

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	Initial value
0xD004	DREQO	DLVLO	DMODO	DENBO	DREQI	DLVLI	DMODI	DENBI	Reserved			Reserved				MASK	0000_0000
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R0,WX			R/W	R/W	R/W	R/W	R/W	

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Reserved	RESET		SWAP	TREQ1	TENB1	TREQ0	TENB0	Reserved							
R0,WX	R0,W		R/W	R(RM1),W	R/W	R(RM1),W	R/W	R0,WX							

bit31 R(RM1),W

DREQO	DMA request flag of output buffer	
	Read	Write
0	No DMA request	DMA request is cleared.
1	DMA request exists	Disabled

Note: At read access of the read-modify-write instruction, "1" is always read.

bit30 R/W

DLVLO	DMA request of output buffer Level/Edge selection	
	During DMODO=0	During DMODO=1
0	DMA request level is output buffer busy	DMA request is output buffer busy's falling edge.
1	DMA request level is inversion of output buffer busy.	DMA request is output buffer busy's rising edge.

bit29 R/W

DMODO	DMA trigger of output buffer mode selection
0	Output buffer busy level
1	Output buffer busy edge

bit28 R/W

DENBO	Output buffer DMA request enable signal
0	Disabled
1	Enabled

bit27 R(RM1),W

DREQI	DMA request flag of input buffer host	
	Read	Write
0	No DMA request	DMA request is cleared.
1	DMA request exists	Disabled

Note: At read access of the read-modify-write instruction, "1" is always read.

bit26 R/W

DLVLI	DMA request of input buffer host Level/Edge selection	
	During DMODI=0	During DMODI=1
0	DMA request level is input buffer host busy.	DMA request is input buffer host busy's falling edge.
1	DMA request level is inversion of input buffer host busy.	DMA request is input buffer host busy's rising edge.

bit25 R/W

DMODI	DMA trigger mode selection of input buffer host
0	Input buffer host busy level
1	Input buffer host busy edge

bit24 R/W

DENBI	DMA request signal of input buffer host
0	Disabled
1	Enabled

bit23 to bit21 R0

Reserved bits
"0" should always be written. At reading, "0" is read.

bit20 to bit17 R/W

Reserved bits
"0" should always be written.

bit16 R/W

MASK	For DMA channel 0 DMA interrupt mask selection
0	Mask disabled
1	Mask enabled

bit15 R0

Reserved bits
"0" should always be written. At reading, "0" is read.

bit14, bit13 R0,W

RESET	FlexRay reset
00	Reset is performed only when writing is done in such sequence as 00 → 01 → 10 → 11.

bit12 R/W

SWAP	Buffer data SWAP enable
0	SWAP off
1	SWAP on

bit11 R(RM1),W

TREQ1	Timer 1 interrupt request	
	Read	Write
0	No timer 1 interrupt request	Timer 1 interrupt request clear
1	Timer 1 interrupt request exists	Disabled

Note: At read access of the read-modify-write instruction, "1" is always read.

bit10 R/W

TENB1	Timer 1 interrupt trigger display selection
0	Output mask (fixed at "0")
1	Select timer 1's rising edge

bit9 R(RM1),W

TREQ0	Timer 0 interrupt request	
	Read	Write
0	No timer 0 interrupt request	Timer 0 interrupt request clear
1	Timer 0 interrupt request exists	Disabled

Note: At read access of the read-modify-write instruction, "1" is always read.

bit8 R/W

TENB0	Timer 0 interrupt trigger display selection
0	Output mask (fixed at "0")
1	Select timer 0's rising edge

bit7 to bit0 R0

Reserved bits
"0" should always be written. At reading, "0" is read.

: Initial value

Bit name		Function
bit31	DREQO	DMA request flag of output buffer It displays existence or non-existence of DMA request at reading: During DMODO = 0: Output buffer busy level is displayed (reversed by DLVLO). During DMODO = 1: DMA request of output buffer busy, for which edge detection is performed, is displayed. (Level is output). During writes, DMA request by edge detection is cleared: When set to "0": DMA request is cleared. When set to "1": Disabled * At read access of the read-modify-write instruction, "1" is always read. * During writing, write "1" if DMA request by edge detection needs not be cleared. * Clearing DMA request by writing "0" into DREQO is allowed only for DMA request by edge detection. * To DMA request by edge detection, "0" clear is also applied if DMA transfer is generated. * DMA request by edge detection maintains "1" state until it is cleared. The clearing is required before edge detection. * DMA request by edge detection is maintained, and if this maintained content is set to DMODO="1", DMA request will be generated.When changing the setting from DMODO="0"to "1", an edge detection result which is detected during DMODO="0" is output as it is if DMODO is changed to "1". When edge detection is expected after changing the setting to DMODO="1", the clearing is required before changing the DMODO setting. * In the event that edge detection of DMA request and clearing of DREQO occur simultaneously, DMA request is prioritized and the clearing will be ignored.To perform the clearing, clearing operation for DREQO is required once again. * The read contents of DREQO and the contents of DMA request during DENBO="1" are identical with each other, except in the case of read access of the read-modify-write instruction. * See "CHAPTER 27 DMA CONTROLLER (DMAC)" as for the setting of the DMA transfer.
bit30	DLVLO	DMA level/edge selection of output buffer Indication contents of register DREQO during register DMODO=0: When set to "0": DREQO indicates output buffer busy. When set to "1": DREQO indicates inversion of output buffer busy. Edge detection method of output buffer busy during register DMODO=1: When set to "0": Output buffer busy's falling edge is detected. When set to "1": Output buffer busy's rising edge is detected. * Edge detection result ("1" if detected) of output buffer busy which corresponds to the DLVLO setting is maintained.The edge detection result cannot be referred to during DMODO="0".
bit29	DMODO	DMA trigger mode selection of output buffer Output information selection of register DREQO and DMA request: When set to "0": Output buffer busy is output. When set to "1": Output buffer busy's edge detection state is output. * DMODO is the selector of information to be output to register DREQO and DMA request.There is no function that disables edge detection. Edge detection is always enabled.
bit28	DENBO	Output buffer's DMA request output enable bit When set to "0": DMA request output is disabled. When set to "1": DMA request output is enabled.

Bit name		Function
bit27	DREQI	DMA request flag of input buffer host It displays existence or non-existence of DMA request at reading: During DMODI = 0: Input buffer host busy's level is displayed (reversed by DLVLI). During DMODI = 1: DMA request of input buffer host busy, for which edge detection is performed, is displayed. (Level is output). During writes, DMA request by edge detection is cleared: When set to "0": DMA request is cleared. When set to "1": Disabled * At read access of the read-modify-write instruction, "1" is always read. * During writing, write "1" if DMA request by edge detection needs not be cleared. * Clearing DMA request by writing "0" into DREQI is allowed only for DMA request by edge detection. * To DMA request by edge detection, "0" clear is also applied if DMA transfer is generated. * DMA request by edge detection maintains "1" state until it is cleared. The clearing is required before edge detection. * DMA request by edge detection is maintained, and if this maintained content is set to DMODI="1", DMA request will be generated.When changing the setting from DMODI="0"to "1", an edge detection result which is detected during DMODI="0" is output as it is if DMODI is changed to "1". When edge detection is expected after changing the setting to DMODI="1", the clearing is required before changing the DMODI setting. * In the event that edge detection of DMA request and clearing of DREQI occur simultaneously, DMA request is prioritized and the clearing will be ignored.To perform the clearing, clearing operation for DREQI is required once again. * The read contents of DREQI and the contents of DMA request during DENBI="1"are identical with each other, except in the case of read access of the read-modify-write instruction. * See "CHAPTER 27 DMA CONTROLLER (DMAC)" as for the setting of the DMA transfer.
bit26	DLVLI	DMA level/edge selection of input buffer host Indication contents of register DREQI during register DMODI=0: When set to "0": DREQI indicates input buffer host busy. When set to "1": DREQI indicates inversion of input buffer host busy. Edge detection method of input buffer host busy: When set to "0": Input buffer host busy's falling edge is detected. When set to "1": Input buffer host busy's rising edge is detected. * Edge detection result ("1" if detected) of input buffer host busy which corresponds to the DLVLI setting is maintained.The edge detection result cannot be referred to during DMODI="0".
bit25	DMODI	DMA trigger mode selection of input buffer host. Output information selection of register DREQI and DMA request: When set to "0": Input buffer host busy is output. When set to "1": Input buffer host busy's edge detection state is output. * DMODI is the selector of information to be output to register DREQI and DMA request.There is no function that disables edge detection. Edge detection is always enabled.
bit24	DENBI	Input buffer host's DMA request output enable bit When set to "0": DMA request output is disabled. When set to "1": DMA request output is enabled.
bit23 to bit21	Reserved	These bits are reserved. These bits are always "0" when read. Write "0" to these bits.
bit20 to bit17	Reserved	These bits are reserved. When writing to these bits, write "0".
bit16	MASK	Selection of DMA interrupt mask for DMA channel 0 When you set this bit to "0": Disabled When you set this bit to "1": Enabled. DMA channel 0's interrupt is masked in DMA transfer of DMA channel 0, which has been caused as a result of activation factor of output buffer busy while the output buffer busy remains valid. * In the event that this bit is valid and the output buffer busy remains valid even though the output buffer busy is not the activation factor, DMA channel 0's DMA interrupt is masked.
bit15	Reserved	This bit is reserved. This bit is always "0" when read. Write "0" to this bit.

Bit name		Function
bit14, bit13	RESET	FlexRay reset They support key code. If the writing sequence to the reset register is "00" → "01" → "10" → "11", reset will be output to FlexRay macro. * If writing to an address in which read access or reset bit is not contained is performed in the course of key code writing, it is considered that the key code writing is suspended.If you want to reset once again, write all over again from "00". * Because there is no bit mask function, care must be taken on writing value of bit, except reset.When the setting value needs to be maintained, be sure to write previous value to the required bit. However, it is advisable to write "1" if interrupt needs not be cleared, because TREQ1, TREQ0, DREQO, and DREQI clear interrupt with writing of "0". * The key code writing before completion of DMA transfer, which is for FlexRay's address space, will be suspended after DMA starts during the key code writing. * Wring "00"is always regarded as start of key code. Additionally, the suspension of key code writing by means of writing of "00" is regarded as a start of a new key code writing. Example: Start Suspension & Start Reset output ↓ ↓ ↓ "00" → "01" → "00" → "01" → "10" → "11"
bit12	SWAP	Buffer data SWAP enable Byte swap selection. The following settings are reflected in RAM area: When set to "0": Swap disabled ([31:24] [23:16] [15:8] [7:0]) When set to "1": Swap enabled ([7:0] [15:8] [23:16] [31:24]) Accessing register will not be affected.
bit11	TREQ1	Timer 1 interrupt request At reading, TREQ1's interrupt request is shown. During TENB1 = 0: Timer 1's level is shown. During TENB1 = 1: "1" is shown (level output) upon detection of timer 1's rising edge. At writing, TREQ1's interrupt request is cleared. When set to "0": Timer interrupt request is cleared. When set to "1": Disabled * At read access of the read-modify-write instruction, "1" is always read. * Interrupt request by edge detection is maintained, and if this maintained content is set to TENB1="1", the interrupt request will be generated.When changing the setting from TENB1="0" to "1", an edge detection result which is detected during TENB1="0" is output as it is if TENB1 is changed to "1". When edge detection is expected after changing the setting to TENB1="1", the clearing is required before changing the TENB1 setting. * In the event that TREQ1's interrupt request and clearing of TREQ1 occur simultaneously, the clearing will be ignored. To perform the clearing, clearing operation for TREQ1 is required once again.
bit10	TENB1	Timer 1 interrupt enable When set to "0": Interrupt prohibited When set to "1": Interrupt enabled
bit9	TREQ0	Timer 0 interrupt request At reading, TREQ0's interrupt request is shown. During TENB0 = 0: Timer 0's level is shown. During TENB0 = 1: "1" is shown (level output) upon detection of timer 1's rising edge. At reading, TREQ0's interrupt request is cleared. When set to "0": Timer interrupt request is cleared. When set to "1": Disabled * At read access of the read-modify-write instruction, "1" is always read. * Interrupt request by edge detection is maintained, and if this maintained content is set to TENB0="1", the interrupt request will be generated.When changing the setting from TENB0="0" to "1", an edge detection result which is detected during TENB0="0" is output as it is if TENB0 is changed to "1". When edge detection is expected after changing the setting to TENB0="1", the clearing is required before changing the TENB0 setting. * In the event that TREQ0's interrupt request and clearing of TREQ0 occur simultaneously, the clearing will be ignored. To perform the clearing, clearing operation for TREQ0 is required once again.

Bit name		Function
bit8	TENB0	Timer 0 interrupt enable When set to "0": Interrupt prohibited When set to "1": Interrupt enabled
bit7 to bit0	Reserved	These bits are reserved. These bits are always "0" when read. Write "0" to these bits.

37.3.2 Special Registers

37.3.2.1 Lock Registers (LCK: Lock Register)

The lock registers are only for writing. 0x0000 is returned when reading the registers.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCK 0xD01C	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Initial value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W									CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0
Initial value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit7] CLK [7:0] Setting lock key (Configuration Lock Key)

Perform two consecutive writing (unlock sequence) to **CLK [7:0]** before exiting from the CONFIG state by writing **SUCC1.CMD [3:0]** ="0010"(READY command). The writing procedure are shown below. However, if other write access is made during the writing access procedures, the following procedures have to be repeated because the CONFIG state remains active:

First writing: **LCK.CLK [7:0]** = "1100 1110" (0xCE)

Second writing: **LCK.CLK [7:0]** = "0011 0001" (0x31)

Third writing: **SUCC1.CMD [3:0]** = "0010" (CHI command READY)

(Note) The host uses 32-bit access for reading/writing of all bit fields. 16-bit access must be programmed to be accessed consecutively.

37.3.3 Interrupt-Related Register

37.3.3.1 Error Interrupt Register (EIR)

When one of the errors listed below is detected, the corresponding flag will be set to "1". This flag will be cleared to "0" by writing "1" to the corresponding bit and will retain the value until a write of "0" takes place. Writing "0" does not influence the operation. A hardware reset does clear this register to "0".

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIR 0xD020	R	0	0	0	0	0	TABB	LTVB	EDB	0	0	0	0	0	TABA	LTVA	EDA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] PEMC POC Error Mode Change Flag (POC Error Mode Changed)

This flag is set to "1" when the error mode specified with **CCEV.ERRM [1:0]** has been changed.

1 = Error mode changed

0 = Error mode not changed

[bit1] CNA Command disable notification flag (Command Not Accepted)

This flag indicates that **SUCC1:CMD [3.0]** has been reset to "0000" on account that the requested command could not be used in current POC state or CHI command has been locked (**CCL** ="1").

1 = CHI command not accepted

0 = CHI command accepted

[bit2] SFBM Synchronous fame count shortage flag (Sync Frames Below Minimum)

This flag is set to "1" when the synchronous frame reception count during the previous communication cycle is less than the minimum number of frames required by the FlexRay protocol. This flag should be cleared after a transition to NORMAL_ACTIVE state because there is a possibility that this flag be set during startup.

1 = The number of synchronous frames received are less than the minimum number of frames required

0 = Synchronous node: one or more synchronous frames received

Asynchronous node: two or more synchronous frames received

[bit3] SFO Synchronous frame count excess flag (Sync Frame Overflow)

This flag is set to "1" when the synchronous frame reception count during the previous communication cycle or a different synchronous frame ID count which was received during double cycle (even/odd) exceeds the maximum number of frames set with **GTUC2.SNM[3:0]**.

1 = The number of received synchronous frames exceeding the value set with **GTUC2.SNM [3:0]**

0 = The number of received synchronous frames not exceeding the value set with **GTUC2.SNM [3:0]**

[bit4] CCF Clock compensation failure flag (Clock Correction Failure)

If one of the error below occurs, "1" will be set at the end of the cycle.

- Disappearance of rate compensation
- Disappearance of offset compensation
- Excess of clock compensation limit

The clock compensation status can be monitored on **CCEV** register and **SFS** register. This flag should be cleared after a transition to NORMAL_ACTIVE state because there is a possibility that this flag be set during startup.

1 = Clock compensation error

0 = No clock compensation error

[bit5] CCL CHI command lock flag (CHI Command Locked)

This flag indicates that **SUCC1.CMD[3.0]** is reset to "0000 " for the reason that the previous CHI command has not been executed completely. In this case, the **CNA** bit is also set to "1".

1 = CHI command not accepted

0 = CHI command accepted

[bit6] PERR Parity error flag (Parity Error)

This flag notifies a parity error. This flag will be set to "1" when a parity error is detected while reading one of RAM blocks on FlexRay controller. This flag will be cleared to "0" when the parity error flag on MHDS register is cleared to "0". See Section "37.3.7.1 Message Handler Status Register (MHDS)".

1 = A parity error detected

0 = No parity error detected

[bit7] RFO Reception FIFO overrun flag (Receive FIFO Overrun)

This flag is set to "1" when a reception FIFO overrun is detected. This flag will be cleared when a reception FIFO is read.

1 = A reception FIFO overrun detected

0 = A reception FIFO overrun not detected

[bit8] EFA Empty FIFO access flag (Empty FIFO Access)

This flag is set to "1" when a host requests a message transfer from the reception FIFO through the output buffer while reception FIFO is in empty state.

1 = An occurrence of host access in the reception FIFO empty state

0 = No host access in the reception FIFO empty state

[bit9] IIBA Illegal input buffer access flag (Illegal Input Buffer Access)

This flag is set to "1" when a host requests a change to message buffer through input buffer and under the following conditions:

1) When a host has written to the input buffer command request register to make the following changes in the state except the DEFAULT_CONFIG or CONFIG state:
   - A change of the header section to the buffer when you configure the message buffer 0 to use for key slot transmission (startup frame/synchronous frame transmission, or frame transmission in single slot mode)
   - A change of header section to the static massage buffer with the buffer numbers smaller than **MRC.FDB [7:0]** while **MRC.SEC [1:0]** = "01"

- A change of header section to the static/dynamic message buffer while **MRC.SEC [1:0]** = "1x"
- A change of header section or data section to the message buffer which belongs to the reception FIFO

2) When a host has written to the register of input buffer while **IBCR.IBSYH** is set to "1":

1 = An illegal host access to the input buffer occurred

0 = No illegal host access to the input buffer occurred

[bit10] IOBA Illegal output buffer access flag (Illegal Output Buffer Access)

This flag is set to "1" when a host requests a transfer of message buffer from message RAM to the output buffer while **OBCR.OBSYS** is set to "1".

1 = An illegal host access to the output buffer occurred

0 = No illegal host access to the output buffer occurred

[bit11] MHF Message handler limitation flag (Message Handler Constraints Flag)

This flag indicates the limitation state of the message handler. This flag is set when one of the **MHDF.SNUA**, **MHDF.SNUB**, **MHDF.FNFA**, **MHDF.FNFB**, **MHDF.TBFA**, **MHDF.TBFB**, and **MHDF.WAHP** flags turned from "0" to "1".

1 = Message handler failure detected

0 = No message handler failure detected

[bit16] EDA Channel A error detect flag (Error Detected on Channel A)

This flag is set to "1" when one of **ACS.SEDA**, **ACS.CEDA**, **ACS.CIA**, and **ACS.SBVA** is changed from "0" to "1".

1 = An error detected on channel A

0 = No error detected on channel A

[bit17] LTVA Channel A transmission error detect flag (Latest Transmit Violation Channel A)

This flag indicates the latest detection of channel A transmission error.

1 = The latest transmission error detected on channel A

0 = A transmission error is not detected on channel A

[bit18] TABA Channel A slot border exceed transmission detect flag (Transmission Across Boundary Channel A)

This flag notifies the occurrence of slot border exceeded transmission on channel A.

1 = Slot border exceeded transmission is detected on channel A

0 = Slot border exceeded transmission is not detected on channel A

[bit24] EDB Channel B error detect flag (Error Detected on Channel B)

This flag is set to "1" when one of **ACS.SEDB**, **ACS.CEDB**, **ACS.CIB**, and **ACS.SBVB** is changed from "0" to "1".

1 = An error detected on channel B

0 = No error detected on channel B

[bit25] LTVB Channel B transmission error detect flag (Latest Transmit Violation Channel B)

This flag indicates the latest detection of channel B transmission error.

1 = The latest transmission error detected on channel B

0 = A transmission error is not detected on channel B

[bit26] TABB Channel B slot border exceed transmission detect flag (Transmission Across Boundary Channel B)

This flag notifies the occurrence of slot border exceeded transmission on channel B.

1 = Slot border exceeded transmission is detected on channel B

0 = Slot border exceeded transmission is not detected on channel B

37.3.3.2 Status interrupt register (SIR)

This register sets the corresponding flag to "1" when one of the events listed below is detected. This flag will be cleared by writing "1"to the corresponding bit and will retain the value until a write of "1" takes place. Writing "0" does not influence the operation. A hardware reset does clear this register.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIR 0xD024	R	0	0	0	0	0	0	MTSB	WUPB	0	0	0	0	0	0	MTSA	WUPA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R / W	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] WST Wake up status flag (Wakeup Status)

This flag is set to "1" when the **CCSV.WSV [2:0]** changes to other status except UNDEFINED.

1 = A wake up status transition detected

0 = No wake up status transition detected

[bit1] CAS Collision avoidance symbol flag (Collision Avoidance Symbol)

This flag is set to "1" when a CAS has received.

1 = A collision avoidance symbol received

0 = No collision avoidance symbol received

[bit2] CYCS Communication cycle start flag (Cycle Start Interrupt)

This flag is set to "1" when a communication cycle has started.

1 = A communication cycle started

0 = No communication cycle started

[bit3] TXI Transmission completion flag (Transmit Interrupt)

This flag is set to "1" after a frame transmission has succeeded if MBI bits of each message buffer are set to "1". (See Figure 37.4-16.)

1 = At least one frame transmission from the transmission buffer with its **MBI** bit set to "1" succeeded

0 = No frame transmission from the transmission buffer with its **MBI** bit set to "1"

[bit4] RXI Reception complete flag (Receive Interrupt)

This flag is set to "1" when the payload segment of an effective frame is stored in the reception buffer while the MBI bits of each message buffer are set to "1". (See Figure 37.4-16.)

1 = At least one data section of reception buffer with its **MBI** bit set to "1" updated

0 = No data section of reception buffer with its **MBI** bit set to "1" updated

[bit5] RFNE Reception FIFO flag (Receive FIFO Not Empty)

This flag is set to "1" when effective frames are stored in the reception FIFO.

1 = Reception FIFO not empty

0 = Reception FIFO empty

[bit6] RFCL Reception FIFO full flag (Receive FIFO Critical Level)

This flag is set to "1" when the reception FIFO level (**FSR.RFFL [7:0]**) is greater than or equal to limitation level (**FCL.CL [7:0]**).

1 = Reception FIFO level is greater than or equal to limitation level.

0 = Reception FIFO level is less than limitation level.

[bit7] NMVC Network management vector change flag (Network Management Vector Changed)

This flag indicates a change to the network management vector.

1 = Network management vector changed

0 = Network management vector not changed

[bit8] TI0 Timer 0 flag (Timer Interrupt 0)

This flag is set to "1" when the value of timer 0 and the value of **T0C** match.

1 = The value of timer 0 and the value of **T0C** matched

0 = The value of timer 0 and the value of **T0C** not matched

[bit9] TI1 Timer 1 flag (Timer Interrupt 1)

This flag is set to "1" when the value of timer 1 and the value of **T1C** match.

1 = The value of timer 1 and the value of **T1C** matched

0 = The value of timer 1 and the value of **T1C** not matched

[bit10] TIBC Input buffer transfer complete flag (Transfer Input Buffer Completed)

This flag is set to "1" when **IBCR.IBSYS** is reset after a transfer from the input buffer to message RAM has completed.

1 = A transfer between the input buffer and message RAM completed

0 = A transfer between the input buffer and message RAM not completed

[bit11] TOBC Output buffer transfer complete flag (Transfer Output Buffer Completed)

This flag is set to "1" when **OBCR.OBSYS** is reset after a transfer from message RAM to the output buffer has been completed.

1 = A transfer between message RAM and the output buffer completed

0 = A transfer between message RAM and the output buffer not completed

[bit12] SWE Stopwatch event flag (Stop Watch Event)

After the stopwatch activates, values of the cycle counter and the macrotick are stored in the stopwatch register. (See the STPW1 register.)

1 = Stopwatch event occurred

0 = No stopwatch event occurred

[bit13] SUCS Startup success flag (Startup Completed Successfully)

This flag is set to "1" when the startup has succeeded and the state is changed to NORMAL_ACTIVE.

1 = Startup successful

0 = Startup not successful

[bit14] MBSI Message buffer status update flag (Message Buffer Status Interrupt)

This flag is set to "1" when the message buffer status (**MBS**) has changed (see Figure 37.4-16) while **MBI** bit of the message buffer is set to "1".

1 = At least one of message buffer statuses of **MBI** bit set to "1" changed.

0 = No message buffer statuses of **MBI** bit set to "1" changed.

[bit15] SDS Dynamic segment start flag (Start of Dynamic Segment)

This flag is set to "1" when a dynamic segment has started.

1 = A dynamic segment started

0 = A dynamic segment not started

[bit16] WUPA Channel A wake up pattern reception flag (Wakeup Pattern Channel A)

This flag is set to "1" when a wake up pattern has been received on channel A while CC is in wake up, ready, startup state, or monitor mode.

1 = A wake up pattern received on channel A

0 = No wake up pattern received on channel A

[bit17] MTSA Channel A MTS reception flag (MTS Received on Channel A) (vSS!ValidMTSA)

This flag indicates that a media access test symbol (MTS) has been received on channel A while in the previous symbol window. This flag will be updated at the end of symbol window.

1 = MTS symbol received on channel A

0 = No MTS symbol received on channel A

[bit24] WUPB Channel B wake up pattern reception flag (Wakeup Pattern Channel B)

This flag is set to "1" when a wake up pattern has been received on channel B while CC is in wake up, ready, startup state, or monitor mode.

1 = A wake up pattern received on channel B

0 = No wake up pattern received on channel B

[bit25] MTSB Channel B MTS reception flag (MTS Received on Channel B) (vSS!ValidMTSB)

This flag indicates that a media access test symbol (MTS) has been received on channel B while in the previous symbol window. This flag will be updated at the end of symbol window.

1 = MTS symbol received on channel B

0 = No MTS symbol received on channel B

37.3.3.3 Error Interrupt Pin Selection Register (EILS: Error Interrupt Line Select)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EILS 0xD028	R	0	0	0	0	0	TABBL	LTVBL	EDBL	0	0	0	0	0	TABAL	LTVAL	EDAL
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register determines which pins to use for an interrupt generated by the error interrupt flag on **EIR** register.

1 = **INT1** pin was determined to be used for the interrupt

0 = **INT0** pin was determined to be used for the interrupt

[bit0] PEMCL: POC error mode change interrupt pin selection (POC Error Mode Changed Interrupt Line)

[bit1] CNAL: Command disable notification interrupt pin selection (Command Not Accepted Interrupt Line)

[bit2] SFBML: Synchronous frame count shortage interrupt pin selection (Sync Frames Below Minimum Interrupt Line)

[bit3] SFOL: Synchronous frame count excess interrupt pin selection (Sync Frame Overflow Interrupt Line)

[bit4] CCFL: Clock compensation failure interrupt pin selection (Clock Correction Failure Interrupt Line)

[bit5] CCLL: CHI command lock interrupt pin selection (CHI Command Locked Interrupt Line)

[bit6] PERRL: Parity error interrupt pin selection (Parity Error Interrupt Line)

[bit7] RFOL: Reception FIFO overrun interrupt pin selection (Receive FIFO Overrun Interrupt Line)

[bit8] EFAL: Empty FIFO access interrupt pin selection (Empty FIFO Access Interrupt Line)

[bit9] IIBAL: Illegal input buffer access interrupt pin selection (Illegal Input Buffer Access Interrupt Line)

[bit10] IOBAL: Illegal output buffer access interrupt pin selection (Illegal Output Buffer Access Interrupt Line)

[bit11] MHFL: Message handler limitation flag interrupt pin selection

[bit16] EDAL: Channel A error detection interrupt pin selection (Error Detected on Channel A Interrupt Line)

[bit17] LTVAL: Channel A transmission error detection interrupt pin selection (Latest Transmit Violation Channel A Interrupt Line)

[bit18] TABAL: Channel A slot border exceed transmission detection interrupt pin selection (Transmission Across Boundary Channel A Interrupt Line)

[bit24] EDBL: Channel B error detection interrupt pin selection (Error Detected on Channel B Interrupt Line)

[bit25] LTVBL: Channel B transmission error detection interrupt pin selection (Latest Transmit Violation Channel B Interrupt Line)

[bit26] TABBL: Channel B slot border exceed transmission detection interrupt pin selection (Transmission Across Boundary Channel B Interrupt Line)

37.3.3.4 Status interrupt pin selection register (SILS: Status Interrupt Line Select)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILS	R	0	0	0	0	0	0	MTSBL	WUPBL	0	0	0	0	0	0	MTSAL	WUPAL
0xD02C	W																
Reset		0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R/W	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Reset		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This register determines which pins to use for an interrupt generated by the status interrupt flag on **SIR** register.

1 = **INT1** pin was determined to be used for the interrupt

0 = **INT0** pin was determined to be used for the interrupt

[bit0] WSTL: Wakeup status interrupt pin selection (Wakeup Status Interrupt Line)

[bit1] CASL: Collision avoidance symbol interrupt pin selection (Collision Avoidance Symbol Interrupt Line)

[bit2] CYCSL: Communication cycle start interrupt pin selection (Cycle Start Interrupt Line)

[bit3] TXIL: Transmission complete interrupt pin selection (Transmit Interrupt Line)

[bit4] RXIL: Reception complete interrupt pin selection (Receive Interrupt Line)

[bit5] RFNEL: Reception FIFO interrupt pin selection (Receive FIFO Not Empty Interrupt Line)

[bit6] RFCLL: Reception FIFO critical level interrupt pin selection (Receive FIFO Critical Level Interrupt Line)

[bit7] NMVCL: Network management vector change interrupt pin selection (Network Management Vector Changed Interrupt Line)

[bit8] TI0L: Timer 0 interrupt pin selection (Timer Interrupt 0 Line)

[bit9] TI1L: Timer 1 interrupt pin selection (Timer Interrupt 1 Line)

[bit10] TIBCL: Input buffer transfer complete interrupt pin selection (Transfer Input Buffer Completed Interrupt Line)

[bit11] TOBCL: Output buffer transfer complete interrupt pin selection (Transfer Output Buffer Completed Interrupt Line)

[bit12] SWEL: Stopwatch event interrupt pin selection (Stop Watch Event Interrupt Line)

[bit13] SUCSL: Startup success interrupt pin selection (Startup Completed Successfully Interrupt Line)

[bit14] MBSIL: Message buffer status change interrupt pin selection (Message Buffer Status Interrupt Line)

[bit15] SDSL: Dynamic segment start interrupt pin selection (Start of Dynamic Segment Interrupt Line)

[bit16] WUPAL: Channel A wake up pattern reception interrupt pin selection (Wakeup Pattern Channel A Interrupt Line)

[bit17] MTSAL: Channel A MTS reception interrupt pin selection (Media Access Test Symbol Channel A Interrupt Line)

[bit24] WUPBL: Channel B wake up pattern reception interrupt pin selection (Wakeup Pattern Channel B Interrupt Line)

[bit25] MTSBL: Channel B MTS reception interrupt pin selection (Media Access Test Symbol Channel B Interrupt Line)

37.3.3.5 Error Interrupt Enable Registers (EIES, EIER: Error Interrupt Enable Set / Reset)

These registers configure which status changes in the error interrupt register (EIR) to generate the interrupt.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIES,R 0xD030 0xD034	R	0	0	0	0	0	TABBE	LTVBE	EDBE	0	0	0	0	0	TABAE	LTVAE	EDAE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interrupt enable flag will be enabled when "1" is written to the address 0xD030 and will be disabled when "1" is written to the address 0xD034. Writing "0" to either address does not change this enable flag.

The same value is read out of the both addresses.

1 = Interrupt enabled

0 = Interrupt disabled

[bit0] PEMCE: POC error mode change interrupt enable flag (POC Error Mode Changed Interrupt Enable)

[bit1] CNAE: Command disable notification interrupt enable flag (Command Not Accepted Interrupt Enable)

[bit2] SFBME: Synchronous frame count shortage interrupt enable flag (Sync Frames Below Minimum Interrupt Enable)

[bit3] SFOE: Synchronous frame count excess interrupt enable flag (Sync Frame Overflow Interrupt Enable)

[bit4] CCFE: Clock compensation failure interrupt enable flag (Clock Correction Failure Interrupt Enable)

[bit5] CCLE: CHI command lock interrupt enable flag (CHI Command Locked Interrupt Enable)

[bit6] PERRE: Parity error interrupt enable flag (Parity Error Interrupt Enable)

[bit7] RFOE: Reception FIFO overrun interrupt enable flag (Receive FIFO Overrun Interrupt Enable)

[bit8] EFAE: Empty FIFO access interrupt enable flag (Empty FIFO Access Interrupt Enable)

[bit9] IIBAE: Illegal input buffer access interrupt enable flag (Illegal Input Buffer Access Interrupt Enable)

[bit10] IOBAE: Illegal output buffer access interrupt enable flag (Illegal Output Buffer Access Interrupt Enable)

[bit11] MHFE: Message handler limitation flag interrupt enable flag (Message Handler Constraints Flag Interrupt Enable)

[bit16] EDAE: Channel A error detect interrupt enable flag (Error Detected on Channel A Interrupt Enable)

[bit17] LTVAE: Channel A transmission error detect interrupt enable flag (Latest Transmit Violation Channel A Interrupt Enable)

[bit18] TABAE: Channel A slot border exceed transmission detect interrupt enable flag (Transmission Across Boundary Channel A Interrupt Enable)

[bit24] EDBE: Channel B error detect interrupt enable flag (Error Detected on Channel B Interrupt Enable)

[bit25] LTVBE: Channel B transmission error detect interrupt enable flag (Latest Transmit Violation Channel B Interrupt Enable)

[bit26] TABBE: Channel B slot border exceed transmission detect interrupt enable flag (Transmission Across Boundary Channel B Interrupt Enable)

37.3.3.6 Status Interrupt Enable Registers (SIES, SIER; Status Interrupt Enable Set / Reset)

These registers configure which status changes in the status interrupt register (SIR) to generate the interrupt.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIES,R 0xD038 0xD03C	R	0	0	0	0	0	0	MTS BE	WUP BE	0	0	0	0	0	0	MTS AE	WUP AE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R / W	SDSE	MBSTE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMV CE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interrupt enable flag will be enabled when "1" is written to the address 0xD038 and will be disabled when "1" is written to the address 0xD03C. Writing "0" to either address does not change this enable flag.

The same value is read out of the both addresses.

1 = Interrupt enabled

0 = Interrupt disabled

[bit0] WSTE: Wakeup status interrupt enable flag (Wakeup Status Interrupt Enable)

[bit1] CASE: Collision avoidance symbol interrupt enable flag (Collision Avoidance Symbol Interrupt Enable)

[bit2] CYCSE: Communication cycle start interrupt enable flag (Cycle Start Interrupt Enable)

[bit3] TXIE: Transmission complete interrupt enable flag (Transmit Interrupt Enable)

[bit4] RXIE: Reception complete interrupt enable flag (Receive Interrupt Enable)

[bit5] RFNEE: Reception FIFO interrupt enable flag (Receive FIFO Not Empty Interrupt Enable)

[bit6] RFCLE: Reception FIFO critical level interrupt enable flag (Receive FIFO Critical Level Interrupt Enable)

[bit7] NMVCE: Network management vector change interrupt enable flag (Network Management Vector Changed Interrupt Enable)

[bit8] TI0E: Timer 0 interrupt enable flag (Timer Interrupt 0 Enable)

[bit9] TI1E: Timer 1 interrupt enable flag (Timer Interrupt 1 Enable)

[bit10] TIBCE: Input buffer transfer complete interrupt enable flag (Transfer Input Buffer Completed Interrupt Enable)

[bit11] TOBCE: Output buffer transfer complete interrupt enable flag (Transfer Output Buffer Completed Interrupt Enable)

[bit12] SWEE: Stopwatch event interrupt enable flag (Stop Watch Event Interrupt Enable)

[bit13] SUCSE: Startup success interrupt enable flag (Startup Completed Successfully Interrupt Enable)

[bit14] MBSIE: Message buffer status change interrupt enable flag (Message Buffer Status Interrupt Enable)

[bit15] SDSE: Dynamic segment start interrupt enable flag (Start of Dynamic Segment Interrupt Enable)

[bit16] WUPAE: Channel A wakeup pattern reception interrupt enable flag (Wakeup Pattern Channel A Interrupt Enable)

[bit17] MTSAE: Channel A MTS reception interrupt enable flag (MTS Received on Channel A Interrupt Enable)

[bit24] WUPBE Channel B wakeup pattern reception interrupt enable flag (Wakeup Pattern Channel B Interrupt Enable)

[bit25] MTSBE Channel B MTS reception interrupt enable flag (MTS Received on Channel B Interrupt Enable)

37.3.3.7 Interrupt Pin Enable Register (ILE: Interrupt Line Enable)

By setting "1"to EINT0 bit and EINT1 bit of two interrupt pins (INT0 and INT1) respectively, you can enable/disable the interrupts separately.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILE	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD040	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EINT1	EINT0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] EINT0: Interrupt pin INT0 enable flag (Enable Interrupt Line 0)

1 = Interrupt pin (INT0) is enabled

0 = Interrupt pin (INT0) is disabled

[bit1] EINT1: Interrupt pin INT1 enable flag (Enable Interrupt Line 1)

1 = Interrupt pin (INT1) is enabled

0 = Interrupt pin (INT1) is disabled

37.3.3.8 Timer 0 setting register (T0C:Timer 0 Configuration)

Specifies the time when the timer 0 interrupt occurs in cycle counts or macroticks. When the timer 0 interrupt occurs, the interrupt output INT2 will be set to "1" during one macrotick, and SIR:TI0 will be set to "1".

Timer 0 can operate while POC is in NORMAL_ACTIVE state or NORMAL_PASSIVE state. In other states, timer 0 will stop operation.

When you reconfigure timer 0, be sure to stop the timer by writing "0" to the T0RC bit.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T0C	R	0	0	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO
0xD044	W			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	T0CC	T0CC	T0CC	T0CC	T0CC	T0CC	T0CC	0	0	0	0	0	0	T0MS	T0RC
	W		6	5	4	3	2	1	0								
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] T0RC: Timer 0 operation control (Timer 0 Run Control)

1 = Timer 0 remains run

0 = Timer 0 stopped

[bit1] T0MS: Timer 0 operation mode selection (Timer 0 Mode Select)

1 = Continuous mode

0 = Single shot mode

[bit8 to bit14] T0CC [6:0] Timer 0 cycle code setting (Timer 0 Cycle Code)

It is the code provided to set the cycle set to use for timer 0 interrupt generation. See Section "37.4.7.2 Cycle Counter Filtering" for details of the configuration of cycle code.

[bit16 to bit29] T0MO [13:0] Timer 0 macrotick offset setting (Timer 0 Macrotick Offset)

Sets the offset time of occurrence of the timer 0 interrupt in macroticks from the beginning of each cycle which was set with the cycle set. The cycle set will be set with T0CC.

(Notes) Interrupt output INT2 immediately outputs "L" when the state transition occurs from NORMAL_ACTIVE or NORMAL_PASSIVE to another state or when the timer 0 stops by clearing **T0RC** to "0".
The timer 0 can be obtained in exchange for a macrotick counter value. There is no counter for exclusive use for the timer 0.

37.3.3.9 Timer 1 setting register (T1C:Timer 1 Configuration)

The timer 1 interrupt occurs when the timer 1 reaches the macrotick counts specified. The interrupt output (INT3) will be set to "1" during one macrotick, and SIR:TI1 will be set to "1".

Timer 1 can operate while POC is in NORMAL_ACTIVE state or NORMAL_PASSIVE state. In other states, timer 1 will stop operation.

When you reconfigure timer 1, be sure to stop the timer by writing "0" to the T1RC bit.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T1C	R	0	0	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC
0xD048	W			13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T1MS	T1RC
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] T1RC: Timer 1 operation control (Timer 1 Run Control)

1 = Timer 1 remains run

0 = Timer 1 stopped

[bit1] T1MS: Timer 1 operation mode selection (Timer 1 Mode Select)

1 = Continuous mode

0 = Single shot mode

[bit16 to bit29] TIMC [13:0] Timer 1 macrotick count (Timer 1 Macrotick Count)

A timer 1 interrupt occurs when the timer 1 matches the macrotick counts specified.

Effective values: 2 to 16383 MT (continue mode)
1 to 16383 MT (single shot mode)

(Notes) Interrupt output (INT3) immediately outputs "L" when the state transition occurs from NORMAL_ACTIVE or NORMAL_PASSIVE to another state or when the timer 0 stops by clearing T1RC to "0".

37.3.3.10 Stopwatch register 1 (STPW1:Stop Watch Register 1)

The stopwatch is activated at the time of STOPWT pin's falling edge or when its falling edge is input, when the interrupt (0, 1) occurs, or when the host writes "1" to the SSWT bit. Increments of the macrotick counter will start after the stopwatch is activated. Actual values of the cycle counter and the macrotick are stored in this stopwatch register (stopwatch event). Also, these values can be read from the host.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STPW1 0xD04C	R	0	0	SMTV 13	SMTV 12	SMTV 11	SMTV 10	SMTV 9	SMTV 8	SMTV 7	SMTV 6	SMTV 5	SMTV 4	SMTV 3	SMTV 2	SMTV 1	SMTV 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	SCCV5	SCCV4	SCCV3	SCCV2	SCCV1	SCCV0	0	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0] ESWT: Stopwatch trigger enable (Enable Stop Watch Trigger)

If the stopwatch trigger is enabled, the stopwatch is activated by the edge signal of the input pin (**STOPWT**) or interrupt 0 or 1 signal (a rising edge of **INT0** or **INT1** signal). In the single shot mode, after the actual cycle counter value and macrotick value have been stored in the stopwatch register, this bit is reset to "0".

1 = Enabled
0 = Disabled

[bit1] SWMS: Stopwatch mode select (Stop Watch Mode Select)

1 = Continuous mode
0 = Single mode

[bit2] EDGE: Stopwatch trigger edge select (Stop Watch Trigger Edge Select)

1 = Rising edge
0 = Falling edge

[bit3] SSWT: Software stopwatch trigger (Software Stop Watch Trigger)

When the host sets this bit to "1", the stopwatch activates. After the actual cycle counter value and macrotick value have been stored in the stopwatch register, this bit is cleared to "0".

This bit is write-enabled during **ESWT**="0".

1 = Activates the stopwatch by software triggering

0 = Clears the software trigger.

[bit4] EETP: External trigger pin enable (Enable External Trigger Pin)

When **ESWT**="1", the edge signal of input pin **STOPWT** will be the stopwatch trigger.

1 = Enabled

0 = Disabled

[bit5] EINT0: Interrupt 0 trigger enable (Enable Interrupt 0 Trigger)

When **ESWT**="1", the interrupt 0 event will be the stopwatch trigger.

1 = Enabled

0 = Disabled

[bit6] EINT1: Interrupt 1 trigger enable (Enable Interrupt 1 Trigger)

When **ESWT**="1", the interrupt 1 event will be the stopwatch trigger.

1 = Enabled

0 = Disabled

[bit8 to bit13] SCCV [5:0] Stopwatch event generation cycle counter value (Stopped Cycle Counter Value)

It is the cycle counter value when a stopwatch event occurs. The valid values are 0 to 63.

[bit16 to bit29] SMTV [13:0] Stopwatch event generation macrotick value (Stopped Macrotick Value)

The macrotick counter value when a stopwatch event occurs. The valid values are 0 to 15999.

(Notes) The ESWT bit and SSWT bit should not be set to "1" simultaneously. If done, the write access is ignored. Also, both bits hold the previous values. Either of an external stopwatch trigger or a software stopwatch trigger can be used.

37.3.3.11 Stopwatch register 2 (STPW2:Stop Watch Register 2)

The stopwatch counter value of channel A and channel B can be read from the host.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STPW2 0xD050	R	0	0	0	0	0	SSCVB 10	SSCVB 9	SSCVB 8	SSCVB 7	SSCVB 6	SSCVB 5	SSCVB 4	SSCVB 3	SSCVB 2	SSCVB 1	SSCVB 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SSCVA 10	SSCVA 9	SSCVA 8	SSCVA 7	SSCVA 6	SSCVA 5	SSCVA 4	SSCVA 3	SSCVA 2	SSCVA 1	SSCVA 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit10] SSCVA [10:0] Stopwatch counter value of channel A (Stop Watch Captured Slot Counter Value Channel A)

The stopwatch counter value (0 to 2047) of channel A during event occurrence

[bit16 to bit26] SSCVB [10:0] Stopwatch counter value of channel B (Stop Watch Captured Slot Counter Value Channel B)

The stopwatch counter value (0 to 2047) of channel B during event occurrence

37.3.4 Communication controller (CC) control registers

This section explains the registers used for control of FlexRay's communication controller (CC). In the FlexRay protocol specifications, the application setup data must be set in the CONFIG state. Note that data writing in the configuration register is not locked in the DEFAULT_CONFIG state.

When a hardware reset is entered, the system transits to the DEFAULT_CONFIG state and each register is initialized. To transit the protocol operation controller (POC) from the DEFAULT_CONFIG state to the CONFIG state, set **CMD[3:0]** = "0001" (CHI command CONFIG). Also, to transit from the CONFIG state to the READY state, follow the procedure given in Section "37.3.2.1 Lock Registers (LCK: Lock Register)".

Any bit identified by an asterisk (*) can be updated in the DEFAULT_CONFIG state or CONFIG state only.

37.3.4.1 SUC Configuration Register 1 (SUCC1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC1 0xD080	R	0	0	0	0	CCHB*	CCHA*	MTSB*	MTSA*	HCSE*	TSM*	WUCS*	PTA4*	PTA3*	PTA2*	PTA1*	PTA0*
	W																
Reset		0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	CSA4*	CSA3*	CSA2*	CSA1*	CSA0*	0	TXSY*	TXST*	PBSY	0	0	0	CMD3	CMD2	CMD1	CMD0
	W																
Reset		0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

[bit0 to bit3] CMD [3:0] CHI command vector

This CHI command vector is always write-enabled, but certain commands are valid in the certain POC state only. If a command is invalid, it is not executed. In such case, the CHI command vector **CMD [3:0]** is reset to "0000" (command_not_accepted) and **EIR.CNA** is set to "1". If both **EIR.CCL** and **EIR.CNA** are set to "1" for the reason that the previous CHI command has not been executed completely, the CHI command must be repeated. If a command to change to a POC state is issued in the POC state except in the HALT state, this command is ignored, and the **EIR.CNA** will not be set.

0000 = command_not_accepted
0001 = CONFIG
0010 = READY
0011 = WAKEUP
0100 = RUN
0101 = ALL_SLOTS
0110 = HALT
0111 = FREEZE
1000 = SEND_MTS
1001 = ALLOW_COLDSTART

1010 = RESET_STATUS_INDICATORS
1011 = MONITOR_MODE
1100 = CLEAR_RAMS
1101 = reserved
1110 = reserved
1111 = reserved

If **CMD [3:0]** is read, the recently accepted CHI command is shown. The actual POC state is monitored with **CCSV.POCS [5:0]**. The "reserved" CHI command belongs to the hardware test function.

Generally, before setting the CHI command, the host has to check SUCC1.PBSY.

command_not_accepted

CMD [3:0] ="0000"writing is reset to **CMD [3:0]** ="0000", if one of the followings occurred.

- An invalid command was set.
- A command was set during change of internal POC state.
- A new command was set during CHI command execution.
- **command_not_accepted** was set.

If a command is invalid, it is not executed. In such case, the CHI command vector **CMD [3:0]** is reset to "0000" (command_not_accepted) and **EIR.CNA** is set to "1". An interrupt occurs if it is enabled.

CONFIG

If **CMD[3:0]** ="0001" is set in the DEFAULT_CONFIG state, READY state, or MONITOR_MODE state, a transition to the CONFIG state occurs. If **CMD[3:0]** ="0001" is set in the HALT state, a transition to the DEFAULT_CONFIG state occurs. If **CMD [3:0]** ="0001" is set in another state, **CMD [3:0]** = "0000" (command_not_accepted) is reset.

READY

If **CMD [3:0]** ="0010" is set in the CONFIG state, NORMAL_ACTIVE state, NORMAL_PASSIVE state, STARTUP state or WAKEUP state, a transition to the READY state occurs. If **CMD[3:0]** ="0010" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

WAKEUP

If **CMD [3:0]** ="0011" is set in the READY state, a transition to the WAKEUP state occurs. If **CMD[3:0]** ="0011" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

RUN

If **CMD [3:0]** ="0100" is set in the READY state, a transition to the STARTUP state occurs. If **CMD[3:0]** ="0100" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

ALL_SLOTS

If **CMD [3:0]** ="0101" is set in the NORMAL_ACTIVE state or NORMAL_PASSIVE state, a transition from the SINGLE slot mode to the ALL slot mode occurs after successful startup or integration at the next end of this cycle. If **CMD[3:0]** ="0101" is set in another state, **CMD[3:0]** ="0000" (command_not_accepted) is reset.

HALT

If **CMD [3:0]** ="0110" is set in the NORMAL_ACTIVE state or NORMAL_PASSIVE state, the **CCSV.HRQ** stop request bit is set to "1" and a transition to the HALT state occurs at the next end of this cycle. If **CMD[3:0]** ="0110" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

FREEZE

If **CMD [3:0]** ="0111" is set, the **CCSV.FSI** freeze status indicator is set to "1" and a transition to the HALT state occurs immediately. This can be set in any state.

SEND_MTS

If ALL slot mode (**CCSV.SLM [1:0]** = "11") is set first, and then **CMD [3:0]** ="1000" is set in the NORMAL_ACTIVE state, a single MTS symbol is sent on the channel set by the **MTSA** and **MTSB** in the next symbol window. If **CMD[3:0]** ="1000" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

ALLOW_COLDSTART

If **CMD [3:0]** ="1001" is set in a state other than DEFAULT_CONFIG, CONFIG or HALT state, the **CCSV.CSI** bit must be cleared to "0" so that the cold start of node is enabled. If **CMD[3:0]** ="1001" is set in the DEFAULT_CONFIG state, CONFIG state, HALT state, or MONITOR_MODE state, **CMD[3:0]** ="0000"(command_not_accepted) is reset. Also, both **TXST** and **TXSY** bits must be set to enable the cold start.

RESET_STATUS_INDICATORS

If **CMD [3:0]** ="1010" is set, the **CCSV.CSNI**, **CCSV.CSAI** and **CCSV.WSV [2:0]** status flags are reset. This is executed in the READY state and STARTUP state, **CMD [3:0]** ="0000"(command_not_accepted) is reset in other states.

CLEAR_RAMS

If **CMD [3:0]** ="1100" is set in the DEFAULT_CONFIG state or CONFIG state, **MHDS.CRAM** is set to "1". If **CMD[3:0]** ="1100" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset. The **MHDS.CRAM** bit is set to "1" even after the hardware reset. **If MHDS.CRAM** is set to "1", all of internal RAM blocks are initialized to zero. When the RAM is being initialized, the PBSY flag shows the POC busy state. When CHI command CLEAR_RAMS(CMD [3:0] ="1100") is being executed, an access to the configuration register and status register is allowed.

The 2048HCLK cycles are required to initialize internal RAM blocks of the FlexRay controller. After the hardware reset or after setting of **CMD[3:0]** = "1100" (CHI command CLEAR_RAMS), the IBF and OBF bits must not be accessed when the internal RAM block is being initialized.

Before setting of **CMD [3:0]** ="1100", make sure that no signal is transferred between the message RAM and the IBF/OBF.

This setting causes the message buffer status registers (**MHDS**, **TXRQ1/2/3/4**, **NDAT1/2/3/4**, and **MBSC1/2/3/4**) to reset.

(Notes) Those commands that are accepted, excluding the CLEAR_RAM and SEND_MTS commands, change the POC state of SCLK domain within 8 cycles of HCLK or SCLK, whichever is slower, at the CHI falling edge, provided that the POC is not busy and remains unchanged in the applicable frame. The reading of register CCSV is delayed in synchronization from SCLK to HCLK domain and in CPU interface. The maximum additional delay is 12 cycles of slower clock of HCLK and SCLK.

MONITOR_MODE

If **CMD[3:0]** ="1011" is set in the CONFIG state, a status transition to the MONITOR_MODE occurs. In this mode, the FlexRay frame and wake up pattern can be received. Also, a coding error can be detected. However, the integrity of the received frame time is not checked. This mode can be used for debugging. For example, if the FlexRay network has failed to start up, this mode is used to analyze the failure cause. If **CMD[3:0]** ="1011" is set in another state, **CMD[3:0]** ="0000"(command_not_accepted) is reset.

[bit7] PBSY: POC Busy

It indicates that the POC is busy and no command can be received. **If PBSY**="1", **CMD [3:0]** is locked from an access for data writing. After the hardware reset, this bit is set to "1" when the internal RAM is initialized.

1 = The POC is busy. **CMD [3:0]** is locked.

0 = The POC is idling. **CMD [3:0]** is write-enabled.

[bit8] TXST: Startup frame key slot transmit setting

(Transmit Startup Frame in Key Slot) (pKeySlotUsedForStartup)

This bit specifies whether to use the key slot for startup frame transmission or not. This bit can be changed in the DEFAULT_CONFIG state or CONFIG state only.

1 = Uses the key slot to transmit the startup frame. Cold start is set.

0 = Does not use the key slot to transmit the startup frame. No cold start is set.

[bit9] TXSY: Sync frame key slot transmit setting (Transmit Sync Frame in Key Slot) (pKeySlotUsedForSync)

This bit specifies whether to use the key slot for Sync frame transmission or not. This bit can be changed in the DEFAULT_CONFIG state or CONFIG state only.

1 = Uses the key slot to transmit the Sync frame. Sync is set.

0 = Does not use the key slot to transmit the Sync frame. Neither Sync nor cold start is set.

(Note) To transmit the startup frame, set both **TXST** and **TXSY** bits to "1".

[bit11 to bit15] CSA [4:0] Cold start retry count (Cold Start Attempts) (gColdStartAttempts)

When the network of a cold start node starts up and if a valid response is not received by another node, these bits define the maximum allowable retry count over which the network repeats to start up. This value can be changed in the DEFAULT_CONFIG state or CONFIG state only. Be sure to set the same value for all nodes of the cluster. The valid values are 2 to 31.

[bit16 to bit20] PTA [4:0] Passive-to-active state transition required cycle pair count (Passive to Active) (pAllowPassiveToActive)

These bits define the effective clock correction time that is required for transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state, as the continuous even/odd cycle pair count. If set to "00000", the transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state is not allowed. These bits can be changed in the DEFAULT_CONFIG state or CONFIG state only. The valid values are 0 to 31 even/odd cycle pair counts.

[bit21] WUCS: Wakeup pattern transmit channel select (Wakeup Channel Select) (pWakeupChannel)

This bit selects a channel for wakeup pattern transmission. This bit change is ignored in a state which is neither DEFAULT_CONFIG state nor CONFIG state.

1 = Sends a wakeup pattern on channel B.

0 = Sends a wakeup pattern on channel A.

[bit22] TSM: Transmit slot mode select (Transmission Slot Mode) (pSingleSlotEnabled)

This bit selects the initial transmit slot mode. In the SINGLE slot mode, the preset key slot is only used for data transmission. The key slot ID consists of the header sections of each message buffer 0 and 1 which comply with the MRC.SPLM. bit as well as of message buffer 0. If TSM = "1", each message buffer 0 and 1 and the message buffer 0 can consist only of DEFAULT_CONFIG or CONFIG.In the ALL slot mode, all slots can be used for data transmission.

This bit can be changed in the DEFAULT_CONFIG state or CONFIG state only. However, if **CMD [3:0]** is set to "0101" in the NORMAL_ACTIVE state or NORMAL_PASSIVE state and if the ALL_SLOT command is applied, the status can be transited to the ALL slot mode. **The TSM** is a write-only bit. The current slot mode is monitored by the **CCSV.SLM [1:0]**.

1 = SINGLE slot mode

0 = ALL slot mode

[bit23] HCSE: Halt setting due to clock sync error (Halt due to Clock Sync Error) (pAllowHaltDueToClock)

This bit controls the status transition to the HALT state due to a clock sync error. This bit can be changed in the DEFAULT_CONFIG state or CONFIG state only.

1 = Transits to the HALT state if a clock sync error occurs.

0 = Maintains the NORMAL_PASSIVE state even if a clock sync error has occurred.

[bit24] MTSA: MTS symbol transmission setting on channel A (Select Channel A for MTS Transmission)

This bit uses channel A whether to send MTS symbols or not. This bit is cleared to "0" by default, and it can be changed in the DEFAULT_CONFIG state or CONFIG state only.

1 = Uses channel A to send MTS symbols.

0 = Does not use channel A to send MTS symbols.

[bit25] MTSB: MTS symbol transmission setting on channel B (Select Channel B for MTS Transmission)

This bit uses channel B whether to send MTS symbols or not. This bit is cleared to "0" by default, and it can be changed in the DEFAULT_CONFIG state or CONFIG state only.

1 = Uses channel B to send MTS symbols.

0 = Does not use channel B to send MTS symbols.

(Notes) MTSA and B will also change from other states except the DEFAULT_CONFIG or CONFIG state, if writing to SUCC1 register directly is performed in the 37.3.2.1 Lock Register (LCK)'s unlock sequence. This is combined with the CHI command SEND_MTS. If both MTSA and MTSB are set simultaneously and if **CMD [3:0]** is set to "1000", the MTS symbols are sent to both channels.

[bit26] CCHA: Channel A connection setting (Connected to Channel A) (pChannels)

This bit sets whether to connect the node to channel A or not.

1 = Connects the node to channel A.

0 = Does not connect the node to channel A.

[bit27] CCHB: Channel B connection setting (Connected to Channel B) (pChannels)

This bit sets whether to connect the node to channel B or not.

1 = Connects the node to channel B.

0 = Does not connect the node to channel B.

Table 37.3-2 Correspondence between CHI command and CMD [3:0] in FlexRay protocol specifications (Overview) (1 / 2)

CHI command	Where processed (POC States)	CHI Command Vector CMD [3:0]
ALL_SLOTS	POC:normal active, POC:normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC:default config, POC:config, POC:halt	ALLOW_COLDSTART
CONFIG	POC:default config, POC:ready	CONFIG
CONFIG_COMPLETE	POC:config	Unlock sequence & READY
DEFAULT_CONFIG	POC:halt	CONFIG
FREEZE	All	FREEZE

Table 37.3-2 Correspondence between CHI command and CMD [3:0] in FlexRay protocol specifications (Overview) (2 / 2)

CHI command	Where processed (POC States)	CHI Command Vector CMD [3:0]
HALT	POC:normal active, POC:normal passive	HALT
READY	All except POC:default config, POC:config, POC:ready, POC:halt	READY
RUN	POC:ready	RUN
WAKEUP	POC:ready	WAKEUP

37.3.4.2 SUC setup register 2 (SUCC2: SUC Configuration Register 2)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC2 0xD084	R	0	0	0	0	LTN3*	LTN2*	LTN1*	LTN0*	0	0	0	LT20*	LT19*	LT18*	LT17*	LT16*
	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	LT 15*	LT 14*	LT 13*	LT 12*	LT 11*	LT 10*	LT 9*	LT 8*	LT 7*	LT 6*	LT 5*	LT 4*	LT 3*	LT 2*	LT 1*	LT 0*
	W																
Reset		0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

[bit0 to bit20]LT [20:0] Listen timeout value (Listen Timeout) (pdListenTimeout)

These bits set a listen timeout for startup and wakeup in μT units,
The value range which can be set by "pdListenTimeout" is from 1284 to 1283846 μT.

[bit24 to bit27] LTN [3:0] Listen timeout noise value (Listen Timeout Noise) (gListenNoise-1)

These bits set a listen timeout upper limit for startup and wakeup, as a multiple of pdListenTimeout value, in a noise environment. The value range which can be set by "gListenNoise" is from 2 to 16. **LTN [3:0]** has to be the same value for all nodes of the cluster.

(Notes) The noise timeout time for wakeup and startup is calculated as follows.
pdListenTimeout • gListenNoise = **LT [20:0]** • (**LTN [3:0] +1**)

37.3.4.3 SUC setup register 3 (SUCC3: SUC Configuration Register 3)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC3	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD088	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	WCF3*	WCF2*	WCF1*	WCF0*	WCP3*	WCP2*	WCP1*	WCP0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

[bit0 to bit3] WCP [3:0] Maximum time of PASSIVE state transition clock correction loss (Maximum Without Clock Correction Passive) (gMaxWithoutClockCorrectionPassive)

These bits define a clock correction loss time that causes a transition from the NORMAL_ACTIVE state to the NORMAL_PASSIVE state, as the consecutive even/odd cycle pair count. Set the same value for all nodes of the cluster. The valid values are 1 to 15 cycle pair counts.

[bit4 to bit7] WCF [3:0] Maximum time of HALT state transition clock correction loss (Maximum Without Clock Correction Fatal) (gMaxWithoutClockCorrectionFatal)

These bits define a clock correction loss time that causes a transition from NORMAL_ACTIVE or NORMAL_PASSIVE state to HALT state, as the consecutive even/odd cycle pair count. Set the same value for all nodes of the cluster. The valid values are 1 to 15 cycle pair counts.

(Notes) No transition to the HALT state will take place if SUCC1.HCSE is not set.

37.3.4.4 NEM Configuration Register (NEMC)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMC	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD08C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	NML3*	NML2*	NML1*	NML0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit3] NML [3:0] Network management vector length

(gNetworkManagementVectorLength)

These bits set the length of the network management vector. The configured length must be the same at all nodes of the cluster. The valid values are 0 to 12 bytes.

37.3.4.5 PRT Configuration Register 1 (PRTC1)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTC1 0xD090	R	RWP5*	RWP4*	RWP3*	RWP2*	RWP1*	RWP0*	0	RXW8*	RXW7*	RXW6*	RXW5*	RXW4*	RXW3*	RXW2*	RXW1*	RXW0*
	W																
Reset		0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	BRP1*	BRP0*	SPP1*	SPP0*	0	CASM6	CASM5 *	CASM4 *	CASM3 *	CASM2 *	CASM1 *	CASM0	TSST3*	TSST2*	TSST1*	TSST0*
	W																
Reset		0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1

[bit0 to bit3] TSST [3:0] Transmission start sequence transmitter (gdTSSTransmitter)

These bits define a transmit start sequence (TSS) time in units of bit time (1 bit time = 4 μT = 100 ns at 10 Mbps). Set the same value for all nodes of the cluster. The valid values are 3 to 15 bit time.

[bit5 to bit10] CASM [6:0] Collision avoidance symbol max (gdCASRxLowMax)

These bits define the upper limit length of the acceptance window to be used for collision avoidance symbols (CAS).**CASM** bit6 is fixed to 1. The valid values are 67 to 99 bit time.

[bit12, bit13] SPP [1:0] Strobe point position

These bits define a sample count. The sampling is repeated for the number of times specified by the SPP [1:0] bits, and the bit values (High/Low) are determined by the majority of the sampled values (High/Low).

00, 11= Sample 5
01 = Sample 4
10 = Sample 6

(Note) SPP [1:0] = "00" holds true for FlexRay Protocol V2.1. The alternating strobe point positions were able to be used to compensate asymmetry on the physical layer.

[bit14, bit15] BRP [1:0] Baud rate prescaler (gdSampleClockPeriod, pSamplePerMicrotick)

These bits set a baud rate on the FlexRay bus. One bit time always consists of 8 samples (gdSampleClockPeriod × 8). As for the setting of the system clock SCLK, see "CHAPTER 7 FLEXRAY DEDICATED CLOCK".

00:
gdSampleClockPeriod = 1/f_sclk(s)
pSamplesPerMicrotick = 2

01:
gdSampleClockPeriod = 2/f_sclk(s)
pSamplesPerMicrotick = 1

10, 11:
gdSampleClockPeriod = 4/f_sclk(s)
pSamplesPerMicrotick = 1

[bit16 to bit24] RXW [8:0] Wakeup symbol receive window length (gdWakeupSymbolRxWindow)

These bits set a window length of the wakeup pattern to be received by the node, using the bit time count. Set the same value for all nodes of the cluster. The valid values are 76 to 301 bit time.

[bit26 to bit31] RWP [5:0] Repetitions of Tx wakeup pattern (pWakeupPattern)

These bits set the number of times to transmit a wakeup symbol. The valid values are 2 to 63.

37.3.4.6 PRT Configuration Register 2 (PRTC2)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTC2 0xD094	R	0	0	TXL5*	TXL4*	TXL3*	TXL2*	TXL1:	TXL0*	TXI7*	TXI6*	TXI5*	TXI4*	TXI3*	TXI2*	TXI1*	TXI0*
	W																
Reset		0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	RXL5*	RXL4*	RXL3*	RXl2*	RXL1*	RXL0*	0	0	RXI5*	RXi4*	RXI3*	RXI2*	RXI1*	RXI0*
	W																
Reset		0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0

[bit0 to bit5] RXI [5:0] Wakeup symbol receive idle phase time

(gdWakeupSymbolRxIdle)

These bits set the idling phase time of the wakeup symbol to be received by the node, as the bit time count. Set the same value for all nodes of the cluster. The valid values are 14 to 59 bit time.

[bit8 to bit13] RXL [5:0] Wakeup symbol receive low time (gdWakeupSymbolRxLow)

These bits set the low time of the wakeup symbol to be received by the node, as the bit time count. Set the same value for all nodes of the cluster. The valid values are 10 to 55 bit time.

[bit16 to bit23] TXI [7:0] Wakeup symbol transmit idle phase time (gdWakeupSymbolTxIdle)

These bits set the idling phase time of the wakeup symbol to be sent by the node, as the bit time count. Set the same value for all nodes of the cluster. The valid values are 45 to 180 bit time.

[bit24 to bit29] TXL [5:0] Wakeup symbol transmit low time

(gdWakeupSymbolTxLow)

These bits set the low time of the wakeup symbol to be transmitted by the node, as the bit time count. Set the same value for all nodes of the cluster. The valid values are 15 to 60 bit time.

37.3.4.7 MHD Configuration Register (MHDC)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDC 0xD098	R	0	0	0	SLT12*	SLT11*	SLT10*	SLT9*	SLT8*	SLT7*	SLT6*	SLT5*	SLT4*	SLT3*	SLT2*	SLT1*	SLT0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	SFDL6*	SFDL5*	SFDL4*	SFDL3*	SFDL2*	SFDL1*	SFDL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit6] SFDL [6:0] Static frame data length (gPayloadLengthStatic)

These bits set the payload length of the entire cluster for all frames to be sent by the static segment. The actual payload length (in bytes) is a double of the value set by these bits. The payload length must be the same at all nodes of the cluster. The valid values are 0 to 127.

[bit16 to bit28] SLT [12:0] Start of latest transmit end mini-slot value (pLatestTx)

These bits set the maximum minislot value immediately before the frame transmission is disabled by the dynamic segment.**If SLT [12:0]** are set to "0", no data is sent to the dynamic segment. The valid values are 0 to 7981 minislots.

37.3.4.8 GTU Configuration Register 1 (GTUC1)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC1 0xD0A0	R	0	0	0	0	0	0	0	0	0	0	0	0	UT19*	UT18*	UT17*	UT16*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	UT15*	UT14*	UT13*	UT12*	UT11*	UT10*	UT9*	UT8*	UT7*	UT6*	UT5*	UT4*	UT3*	UT2*	UT1*	UT0*
	W																
Reset		0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

[bit0 to bit19] UT [19:0] Microtick per cycle (pMicroPerCycle)

These bits set the microtick of the communication cycle. The valid values are 640 to 640000 μT.

37.3.4.9 GTU Configuration Register 2 (GTUC2)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC2 0xD0A4	R	0	0	0	0	0	0	0	0	0	0	0	0	SNM3*	SNM2*	SNM1*	SNM0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MPC13*	MPC12*	MPC11*	MPC10*	MPC9*	MPC8*	MPC7*	MPC6*	MPC5*	MPC4*	MPC3*	MPC2*	MPC1*	MPC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

[bit0 to bit13] MPC [13:0] Macrotick per cycle (gMacroPerCycle)

These bits set the macrotick of the communication cycle. Set the same cycle length for all nodes of the cluster. The valid values are 10 to 16000 MT.

[bit16 to bit19] SNM [3:0] Sync node max (gSyncNodeMax)

These bits set the maximum number of nodes that transmit a synchronous frame (the frame having the **SYN** synchronous frame indicator bit of "1"). Set the same value for all nodes of the cluster. The valid values are 2 to 15.

37.3.4.10 GTU Configuration Register 3 (GTUC3)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC3 0xD0A8	R	0	MIOB6*	MIOB5*	MIOB4*	MIOB3*	MIOB2*	MIOB1*	MIOB0*	0	MIOA6*	MIOA5*	MIOA4*	MIOA3*	MIOA2*	MIOA1*	MIOA0*
	W																
Reset		0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	UIOB7*	UIOB6*	UIOB5*	UIOB4*	UIOB3*	UIOB2*	UIOB1*	UIOB0*	UIOA7*	UIOA6*	UIOA5*	UIOA4*	UIOA3*	UIOA2*	UIOA1*	UIOA0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit7] UIOA [7:0] Microtick initial offset channel A

(pMicroInitialOffset [A])

These bits set the microtick count between the macrotick boundary after the channel A secondary time reference point and the real-time reference point. The parameter depends on pDelayCompensation [A], and it must be set for each channel separately. The valid values are 0 to 240 μT.

[bit8 to bit15] UIOB [7:0] Microtick initial offset channel B

(pMicroInitialOffset [B])

These bits set the microtick count between the macrotick boundary after the channel B secondary time reference point and the real-time reference point. The parameter depends on pDelayCompensation [B], and it must be set for each channel separately. The valid values are 0 to 240 μT.

[bit16 to bit22] MIOA [6:0] Macrotick initial offset channel A (pMacroInitialOffset [A])

These bits set the macrotick count between the macrotick boundary after the channel A secondary time reference point and the static slot boundary. This value is based on the nominal macrotick time. Set the same value for all nodes of the cluster. The valid values are 2 to 72 MT.

[bit24 to bit30] MIOB [6:0] Macrotick initial offset channel B (pMacroInitialOffset [B])

These bits set the macrotick count between the macrotick boundary after the channel B secondary time reference point and the static slot boundary. This value is based on the nominal macrotick time. Set the same value for all nodes of the cluster. The valid values are 2 to 72 MT.

37.3.4.11 GTU Configuration Register 4 (GTUC4)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only. For details on the **NIT [13:0]** and **OCS [13:0]** settings, see "■ Configuration of NIT Start and Offset Correction Start".

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC4 0xD0AC	R	0	0	OCS13*	OCS12*	OCS11*	OCS10*	OCS9*	OCS8*	OCS7*	OCS6*	OCS5*	OCS4*	OCS3*	OCS2*	OCS1*	OCS0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	NIT13*	NIT12*	NIT11*	NIT10*	NIT9*	NIT8*	NIT7*	NIT6*	NIT5*	NIT4*	NIT3*	NIT2*	NIT1*	NIT0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

[bit0 to bit13] NIT [13:0] Network idle time start (gMacroPerCycle - gdNIT - 1)

These bits set a start point of network idling time (NIT) at the end of communication cycle indicated by the macrotick count.

Macrotick = gMacroPerCycle - gdNIT - 1

is satisfied, the NIT is started. Set the same value for all nodes of the cluster. The valid values are 7 to 15997 MT.

[bit16 to bit29] OCS [13:0] Offset correction start (gOffsetCorrectionStart - 1)

These bits determine the offset correction start position of the NIT phase. This is calculated from the cycle start position. Set the same value for all nodes of the cluster. The valid values are 8 to 15998 MT.

37.3.4.12 GTU Configuration Register 5 (GTUC5)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC5 0xD0B0	R	DEC7*	DEC6*	DEC5*	DEC4*	DEC3*	DEC2*	DEC1*	DEC0*	0	0	0	CDD4*	CDD3*	CDD2*	CDD1*	CDD0*
	W																
Reset		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	DCB7*	DCB6*	DCB5*	DCB4*	DCB3*	DCB2*	DCB1*	DCB0*	DCA7*	DCA6*	DCA5*	DCA4*	DCA3*	DCA2*	DCA1*	DCA0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit7] DCA[7:0] Reception delay compensation Channel A (pDelayCompensation [A])

These bits are used to compensate the reception delay in channel A. This value guarantees the propagation delay assumed in the range of 0.0125 to 0.05 μs, up to cPropagationDelayMax set in units of microticks. Actually, use the minimum propagation delay time of all nodes that are synchronous with each other. The valid values are 0 to 200 μT.

[bit8 to bit15] DCB[7:0] Reception delay compensation channel B (pDelayCompensation [B])

These bits are used to compensate the reception delay in channel B. This value guarantees the propagation delay assumed in the range of 0.0125 to 0.05 μs, up to cPropagationDelayMax set in units of microticks. Actually, use the minimum propagation delay time of all nodes that are synchronous with each other. The valid values are 0 to 200 μT.

[bit16 to bit20] CDD [4:0] Cluster drift damping (pClusterDriftDamping)

To minimize the rounding accumulation, the cluster drift damping is set which is used for clock synchronous. The valid values are 0 to 20 μT.

[bit24 to bit31] DEC [7:0] Decoding correction value (pDecodingCorrection)

The decoding compensation value is set which is used to determine a primary time reference point. The valid values are 14 to 143 μT.

37.3.4.13 GTU Configuration Register 6 (GTUC6)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC6 0xD0B4	R	0	0	0	0	0	MOD10*	MOD9*	MOD8*	MOD7*	MOD6*	MOD5*	MOD4*	MOD3*	MOD2*	MOD1*	MOD0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ASR10*	ASR9*	ASR8*	ASR7*	ASR6*	ASR5*	ASR4*	ASR3*	ASR2*	ASR1*	ASR0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0 to bit10] ASR [10:0] Accepted startup range (pdAcceptedStartupRange)

These bits set the extension range of a measurement error for the startup frame, using the microtick count. The valid values are 0 to 1875 μT.

[bit16 to bit26] MOD [10:0] Maximum oscillator drift (pdMaxDrift)

These bits set the maximum drift offset, in units of μT, between two asynchronous nodes in one communication cycle. The valid values are 2 to 1923 μT.

37.3.4.14 GTU Configuration Register 7 (GTUC7)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC7 0xD0B8	R	0	0	0	0	0	0	NSS9*	NSS8*	NSS7*	NSS6*	NSS5*	NSS4*	NSS3*	NSS2*	NSS1*	NSS0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	SSL9*	SSL8*	SSL7*	SSL6*	SSL5*	SSL4*	SSL3*	SSL2*	SSL1*	SSL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

[bit0 to bit9] SSL [9:0] Static slot length (gdStaticSlot)

These bits macrotically set the static slot period in macrotick. Set the same static slot length for all nodes of the cluster. The valid values are 4 to 659 MT.

[bit16 to bit25] NSS [9:0] Number of static slots (gNumberOfStaticSlots)

These bits set the number of static slots in cycles. At least two cold start nodes must be configured to start up the FlexRay network. Set the same number of static slots for all nodes of the cluster. The valid values are 2 to 1023.

37.3.4.15 GTU Configuration Register 8 (GTUC8)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC8 0xD0BC	R	0	0	0	NMS12*	NMS11*	NMS10*	NMS9*	NMS8*	NMS7*	NMS6*	NMS5*	NMS4*	NMS3*	NMS2*	NMS1*	NMS0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	MSL5*	MSL4*	MSl3*	MSl2*	MSl1*	MSl0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

[bit0 to bit5] MSL [5:0] Minislot length (gdMinislot)

These bits macrotically set the minislot period in macrotick. Set the same mini slot length for all nodes of the cluster. The valid values are 2 to 63 MT.

[bit16 to bit28] NMS [12:0] Number of minislots (gNumberOfMinislots)

These bits set the number of minislots in one-cycle dynamic segment. Set the same number of minislots for all nodes of the cluster. The valid values are 0 to 7986.

37.3.4.16 GTU Configuration Register 9 (GTUC9)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC9 0xD0C0	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSI1*	DSI0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MAPO4*	MAPO3*	MAPO2*	MAPO1*	MAPO0*	0	0	APO5*	APO4*	APO3*	APO2*	APO1*	APO0*
	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

[bit0 to bit5] APO [5:0] Action point offset (gdActionPointOffset)

These bits macrotically set the action point offset of the static slot and of the symbol window in macrotick. Set the same value for all nodes of the cluster. The valid values are 1 to 63 MT.

[bit8 to bit12] MAPO [4:0] Minislot action point offset (gdMinislotActionPointOffset)

These bits macrotically set the action point offset of the dynamic segment minislot in macrotick. Set the same value for all nodes of the cluster. The valid values are 1 to 31 MT.

[bit16, bit17] DSI [1:0] Dynamic slot idle phase

(gdDynamicSlotIdlePhase)

These bits set the idle phase time in the dynamic slot. The time must be equal to or greater than the idle detection time. Set the same value for all nodes of the cluster. The valid values are 0 to 2 minislots.

37.3.4.17 GTU Configuration Register 10 (GTUC10)

This register setting can be changed in the DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC10 0xD0C4	R	0	0	0	0	0	MRC10*	MRC9*	MRC8*	MRC7*	MRC6*	MRC5*	MRC4*	MRC3*	MRC2*	MRC1*	MRC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MOC13*	MOC12*	MOC11*	MOC10*	MOC9*	MOC8*	MOC7*	MOC6*	MOC5*	MOC4*	MOC3*	MOC2*	MOC1*	MOC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

[bit0 to bit13] MOC [13:0] Maximum offset correction value (pOffsetCorrectionOut)

These bits set the allowable maximum offset correction value used for the internal clock synchronous algorithm (absolute value). The total of the internal offset compensation and external offset compensation is collated with this value. The valid values are 5 to 15266 μT.

[bit16 to bit26] MRC [10:0] Maximum rate correction value (pRateCorrectionOut)

These bits set the allowable maximum rate correction value used for the internal clock synchronous algorithm. The total of the internal rate compensation and external rate compensation (absolute value) is collated with this value. The valid values are 2 to 1923 μT.

37.3.4.18 GTU Configuration Register 11 (GTUC11)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC11 0xD0C8	R	0	0	0	0	0	ERC2*	ERC1*	ERC0*	0	0	0	0	0	EOC2*	EOC1*	EOC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	ERCC1	ERCC0	0	0	0	0	0	0	EOCC1	EOCC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit0, bit1] EOCC [1:0] External offset correction control (vExternOffsetControl)

Writing one of the following setting values to **EOCC [1:0]** enables the external offset correction. Change the values outside of NIT.

00, 01 = no external offset correction value
10 = subtract the external offset correction value from the calculated offset correction value
11 = add the external offset correction value to the calculated offset correction value

[bit8, bit9] ERCC [1:0] External rate correction control (vExternRateControl)

Writing one of the following setting values to **ERCC [1:0]** enables the external rate correction. Change the values outside of NIT.

00, 01 = no external rate correction value

10 = subtract the external rate correction value from the calculated rate correction value

11 = add the external rate correction value to the calculated rate correction value

[bit16 to bit19] EOC [2:0] External offset correction (pExternOffsetCorrection)

These bits set, in microticks, an external offset correction value that is used for the internal clock synchronization algorithm. This value is used for addition to a calculated offset correction value or for subtraction from an offset correction value. It is applied during NIT. This register setting can be changed in the DEFAULT_CONFIG state or CONFIG state only. The valid values are 0 to 7 μT.

[bit24 to bit26] ERC [2:0] External rate correction (pExternRateCorrection)

These bits set, in microticks, an external rate correction value that is used for the internal clock synchronization algorithm. This value is used for addition to a calculated rate correction value or for subtraction from a rate correction value. It is applied during NIT. This register setting can be changed in the DEFAULT_CONFIG state or CONFIG state only. The valid values are 0 to 7 μT.

37.3.5 Communication Controller (CC) Status Register

The HCLK frequency allows the contents of the status vector to change every time faster than the host polls the status vector.

37.3.5.1 CC Status Vector (CCSV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCSV	R	0	0	PSL5	PSL4	PSL3	PSL2	PSL1	PSL0	RCA4	RCA3	RCA2	RCA1	RCA0	WSV2	WSV1	WSV0
0xD100	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	CSI	CSAI	CSNI	0	0	SLM1	SLM0	HRQ	FSI	POCS5	POCS4	POCS3	POCS2	POCS1	POCS0
	W																
Reset		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

POCS [5:0] Protocol operation control status

These bits display the execution state of the current POC.

00 0000 = DEFAULT_CONFIG state
00 0001 = READY state
00 0010 = NORMAL_ACTIVE state
00 0011 = NORMAL_PASSIVE state
00 0100 = HALT state
00 0101 = MONITOR_MODE state
00 0110...00 1110 = reserved
00 1111 = CONFIG state

These bits display the current POC state in the wake up procedure.

01 0000 = WAKEUP_STANDBY state
01 0001 = WAKEUP_LISTEN state
01 0010 = WAKEUP_SEND state
01 0011 = WAKEUP_DETECT state
01 0100...01 1111 = reserved

These bits display the current POC state in the startup procedure.

10 0000 = STARTUP_PREPARE state
10 0001 = COLDSTART_LISTEN state
10 0010 = COLDSTART_COLLISION_RESOLUTION state
10 0011 = COLDSTART_CONSISTENCY_CHECK state
10 0100 = COLDSTART_GAP state
10 0101 = COLDSTART_JOIN state
10 0110 = INTEGRATION_COLDSTART_CHECK state
10 0111 = INTEGRATION_LISTEN state
10 1000 = INTEGRATION_CONSISTENCY_CHECK state
10 1001 = INITIALIZE_SCHEDULE state
10 1010 = ABORT_STARTUP state
10 1011 = START UP_SUCCESS state
10 1011...11 1111 = reserved

FSI Freeze status indicator (vPOC!Freeze)

This bit indicates that **CMD [3:0]** ="0111" (CHI command FREEZE) have been set or that the HALT state was entered because such an error occurred that required a quick state transition to HALT. This bit is reset when the HALT state has transited to the DEFAULT_CONFIG state.

HRQ Halt request (vPOC!CHIHaltRequest)

This bit indicates that the host requested that the HALT state be entered at the end of the communication cycle. This bit is reset when the HALT state has transited to the DEFAULT_CONFIG state or the READY state has been entered.

SLM [1:0] Slot mode (vPOC!SlotMode)

These bits indicate the slot mode of the current POC in the READY, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE modes. These bits are set to the SINGLE slot mode by default. The default is changed to ALL with **SUCC1.TSM.** The SINGLE slot mode changes to the ALL slot mode via ALL_PENDING when the CHI command **CMD [3:0]** ="0101"(ALL_SLOTS) is set in the NORMAL_ACTIVE or NORMAL_PASSIVE state. These bits are set to the SINGLE slot mode in all the other states.

00 = SINGLE slot mode

01 = reserved

10 = ALL_PENDING

11 = ALL slot mode

CSNI Coldstart noise indicator (vColdStartNoise)

This bit indicates that the cold start procedure has been executed under the condition in which there are many noises. This bit is reset by the CHI reset command, transition from RESET_STATUS_INDICATORS or HALT to DEFAULT_CONFIG, or transition from the READY to STARTUP state.

CSAI Coldstart abort indicator

This bit indicates that the cold start was stopped. This bit is reset by the CHI reset command, transition from RESET_STATUS_INDICATORS or HALT to DEFAULT_CONFIG, or transition from the READY to STARTUP state.

CSI Cold start inhibit (vColdStartInhibit)

This bit indicates that the node is invalidated for the cold start. This flag is always set to "1" in the READY state that is entered through the CHI command. This flag is reset by setting **SUCC1.CMD [3:0]** ="1001" (CHI command ALLOW_COLDSTART).

1 = The cold start of node disabled

0 = The cold start of node enabled

WSV [2:0] Wakeup status (vPOC!WakeupStatus)

These bits indicate the current wakeup status (see Section "37.4.5.6 WAKEUP State"). They are reset by the CHI reset command, or transition from RESET_STATUS_INDICATORS or DEFAULT_CONFIG to CONFIG.

000 = UNDEFINED:
The wakeup is not started.

001 = RECEIVED_HEADER:
In the WAKEUP_LISTEN state, if a frame header having no error is received on any one of the channels, this status is set when the wakeup ends.

010 = RECEIVED_WUP:
In the WAKEUP_LISTEN state, if a valid wakeup pattern is received on the specified wakeup channel, this status is set when the wakeup ends.

011 = COLLISION_HEADER:
If a collision is detected when a valid header is received on any one of the channels during wakeup pattern transmission, this status is set when the wakeup stops.

100 = COLLISION_WUP:
If a collision is detected when a valid wakeup pattern is received on the specified wakeup channel during wakeup pattern transmission, this status is set when the wakeup stops.

101 = COLLISION_UNKNOWN:
If the wakeup timer passes the specified time while a valid wakeup pattern or valid frame header is not received, control exits from the WAKEUP_DETECT state and this status is set when the wakeup stops.

110 = TRANSMITTED:
This status is set when the wakeup pattern transmission is completed normally.

111 = reserved

RCA [4:0] Remaining coldstart attempts
(vRemainingColdstartAttempts)

These bits indicate the number of remaining cold start attempts. The run command resets this counter to the maximum number of cold start attempts set by **SUCC1.CSA [4:0]**. The initial values in the CONFIG and DEFAULT_CONFIG state are also equal to the value of **SUCC1.CSA [4:0]** 0.

PSL [5:0] POC status log

These bits indicate POCS [5:0] statuses before the HALT state. These bits are set in the HALT state. HALT has been set by the FREEZE command during the HALT state, but FSI has not yet been set, i.e. the HALT state has not been reached by the FREEZE command. These bits are reset to "000000" when the state is no more in HALT.

37.3.5.2 CC Error Vector Register (CCEV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCEV 0xD104	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	PTAC4	PTAC3	PTAC2	PTAC1	PTAC0	ERRM1	ERRM0	0	0	CCFC3	CCFC2	CCFC1	CCFC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This bit is reset when the HALT state has transited to the DEFAULT_CONFIG state or the READY state has been entered.

CCFC [3:0] Clock correction failed counter
(vClockCorrectionFailed)

If an offset or rate correction disappearance error is occurring, one is incremented at the end of the odd-number communication cycle. If neither an offset correction disappearance error nor a rate correction disappearance error is occurring, the clock correction fail counter is reset to "0" at the end of the odd-number communication cycle. This counter stops at 15.

ERRM [1:0] Error mode (vPOC!ErrorMode)

These bits are used to indicate the error mode of the current POC.

00 = ACTIVE
01 = PASSIVE
10 = COMM_HALT
11 = reserved

PTAC [4:0] Passive-to-active state transition cycle pair count counter (vAllowPassiveToActive)

These bits are used to indicate the number of continuous even/odd cycle pairs that are passed since the rate correction time and offset correction time are valid while the node is waiting for the transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. The state transition occurs when **PTAC [4:0]** is equal to **SUCC1.PTA [4:0] -1**.

37.3.5.3 Slot Counter Value Register (SCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCV 0xD110	R	0	0	0	0	0	SCCB 10	SCCB9	SCCB8	SCCB7	SCCB6	SCCB5	SCCB4	SCCB3	SCCB2	SCCB1	SCCB0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SCCA 10	SCCA9	SCCA8	SCCA7	SCCA6	SCCA5	SCCA4	SCCA3	SCCA2	SCCA1	SCCA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

SCCA [10:0] Slot counter channel A (vSlotCounter [A])

These bits indicate the current slot counter value for channel A. This value is "1" at the start of the communication cycle and is incremented at the end of each of the static slot until the cycle ends. The valid values are 0 to 2047.

SCCB [10:0] Slot counter channel B (vSlotCounter [B])

These bits indicate the current slot counter value for channel B. This value is "1" at the start of the communication cycle and is incremented at the end of each of the static slot until the cycle ends. The valid values are 0 to 2047.

37.3.5.4 Macrotick and Cycle Counter Value Register (MTCCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MTCCV 0xD114	R	0	0	0	0	0	0	0	0	0	0	CCV5	CCV4	CCV3	CCV2	CCV1	CCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MTV13	MTV12	MTV11	MTV10	MTV9	MTV8	MTV7	MTV6	MTV5	MTV4	MTV3	MTV2	MTV1	MTV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

MTV [13:0] Macrotick value (vMacrotick)

These bits are used to indicate the current macrotick value. This value is "0" at the start of the communication cycle and is incremented until the end of the cycle. The valid values are 0 to 15999.

CCV [5:0] Cycle counter value (vCycleCounter)

These bits are used to indicate the current cycle counter value. This value is incremented from when the communication cycle starts. The valid values are 0 to 63.

37.3.5.5 Rate Correction Value Register (RCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCV 0xD118	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

RCV [11:0] Rate correction value (vRateCorrection)

These bits are used to indicate the rate correction value (two's complement). This value is calculated within the controller before being restricted with maximum rate correction value **GTUC10.MRC [10:0]**. If the value exceeds the maximum rate correction value, the SFS.RCLR flag is set to "1".

37.3.5.6 Offset correction value register (OCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCV 0xD11C	R	0	0	0	0	0	0	0	0	0	0	0	0	0	OCV18	OCV17	OCV16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OCV15	OCV14	OCV13	OCV12	OCV11	OCV10	OCV9	OCV8	OCV7	OCV6	OCV5	OCV4	OCV3	OCV2	OCV1	OCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

OCV [18:0] Offset correction value (vOffsetCorrection)

These bits are used to indicate the offset correction value (two's complement). This value is internally calculated before being restricted with maximum offset correction value **GTUC10.MOC [10:0]**. If the value exceeds the maximum offset correction value, the **SFS.OCLR** flag is set to "1".

(Notes) The external rate/offset correction values are added to rate/offset correction values restricted with the maximum rate/offset correction values.

37.3.5.7 Sync Frame Status Register (SFS)

The maximum value of the valid synchronous frame is 15 in one communication cycle.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFS 0xD120	R	0	0	0	0	0	0	0	0	0	0	0	0	RCLR	MRCS	OCLR	MOCS
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	VSBO3	VSBO2	VSBO1	VSBO0	VSBE3	VSBE2	VSBE1	VSBE0	VSAO3	VSAO2	VSAO1	VSAO0	VSAE3	VSAE2	VSAE1	VSAE0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

VSAE [3:0] Valid sync frames channel A, **even** communication cycle

These bits are used to display the number of valid synchronous frames received in the even-number communication cycle on Channel A. The value increments by "1" if the synchronous frame transmission is permitted by SUCC1.TXSY. The register value is updated in NIT of each even-number communication cycle.

VSAO [3:0] Valid sync frames channel A, **odd** communication cycle

These bits are used to display the number of valid synchronous frames received in the odd-number communication cycle on Channel A. The value increments by "1" if the synchronous frame transmission is permitted by SUCC1.TXSY. The register value is updated in NIT of each odd-number communication cycle.

VSBE [3:0] Valid sync frames channel B, **even** communication cycle

These bits are used to display the number of valid synchronous frames received in the even-number communication cycle on Channel B. The value increments by "1" if the synchronous frame transmission is permitted by SUCC1.TXSY. The register value is updated in NIT of each even-number communication cycle.

VSBO [3:0] Valid sync frames channel B, **odd** communication cycle

These bits are used to display the number of valid synchronous frames received in the odd-number communication cycle on Channel B. The value increments by "1" if the synchronous frame transmission is permitted by SUCC1.TXSY. The register value is updated in NIT of each odd-number communication cycle.

(Notes) The above bit field is only valid when each channel is assigned with **SUCC1.CCHA** or **SUCC1.CCHB**.

MOCS Missing offset correction signal

This bit is used to indicate that the offset correction has not been calculated because no synchronous frame was received. This flag is updated at the start of the offset correction phase.

1 = Offset correction signal is lost.

0 = Offset correction signal is valid.

OCLR Offset correction limit reached

This bit is used to indicate that the offset correction value has exceeded the limit value defined with **GTUC10.MOC [13:0]**. This flag is updated at the start of the offset correction phase.

1 = The offset correction value has exceeded the limit value.

0 = The offset correction value has not exceeded the limit value.

MRCS Missing rate correction signal

This bit is used to indicate that the rate correction has not been calculated because no even/odd synchronous frame pair was received. This flag is updated at the start of the offset correction phase.

1 = Rate correction signal is lost.

0 = Rate correction signal is valid.

RCLR Rate correction limit reached

This bit is used to indicate that the rate correction value has exceeded the limit value defined with **GTUC10.MRC [10:0]**. This flag is updated at the start of the offset correction phase.

1 = The rate correction value has exceeded the limit value.

0 = The rate correction value has not exceeded the limit value.

37.3.5.8 Symbol Window and NIT Status Register (SWNIT)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWNIT	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD124	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The following shows the bits having the status information related to the symbol window. These bits are updated at the end of the symbol window of each channel. They are not updated during startup.

This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

SESA Syntax error in symbol window channel A (vSS!SyntaxErrorA)

1 = A syntax error has been detected in the symbol window of Channel A.

0 = A syntax error has not been detected in the symbol window of Channel A.

SBSA Slot boundary violation in symbol window channel A (vSS!BViolationA)

1 = A slot boundary error has been detected in the symbol window of Channel A.

0 = A slot boundary error has not been detected in the symbol window of Channel A.

TCSA Transmission conflict in symbol window channel A (vSS!TxConflictA)

1 = A transmission collision has been detected in the symbol window of Channel A.

0 = A transmission collision has not been detected in the symbol window of Channel A.

SESB Syntax error in symbol window channel B (vSS!SyntaxErrorB)

1 = A syntax error has been detected in the symbol window of Channel B.

0 = A syntax error has not been detected in the symbol window of Channel B.

SBSB Slot boundary violation in symbol window channel B (vSS!BViolationB)

1 = A slot boundary error has been detected in the symbol window of Channel B.

0 = A slot boundary error has not been detected in the symbol window of Channel B.

TCSB Transmission conflict in symbol window channel B (vSS!TxConflictB)

1 = A transmission collision has been detected in the symbol window of Channel B.

0 = A transmission collision has not been detected in the symbol window of Channel B.

MTSA MTS Received on channel A
(vSS!ValidMTSA)

This flag indicates that a media access test symbol has been received on Channel A while in the latest symbol window.

These bits are updated by CC at the end of the symbol window of each channel. The interrupt flag SIR.MTSA will be set to "1" when this bit is set to "1".

1 = MTS symbol has been detected in Channel A.

0 = MTS symbol has not been detected in Channel A.

MTSB MTS Received on channel B
(vSS!ValidMTSB)

This flag indicates that a media access test symbol has been received on Channel B while in the latest symbol window.

These bits are updated by CC at the end of the symbol window of each channel. The interrupt flag SIR.MTSA will be set to "1" when this bit is set to "1".

1 = MTS symbol has been detected in Channel B.

0 = MTS symbol has not been detected in Channel B.

The following shows the bits having the status information related to NIT. These bits are updated at the end of NIT of each channel.

SENA Syntax error during NIT channel A (vSS!SyntaxErrorA)

1 = A syntax error has been detected between NITs on Channel A.

0 = A syntax error has not been detected between NITs on Channel A.

SBNA Slot boundary violation during NIT channel A
(vSS!BViolationA)

1 = A slot boundary error has been detected between NITs on Channel A.

0 = A slot boundary error has not been detected between NITs on Channel A.

SENB Syntax Error during NIT Channel B (vSS!SyntaxErrorB)

1 = A syntax error has been detected between NITs on Channel B.

0 = A syntax error has not been detected between NITs on Channel B.

SBNB Slot Boundary Violation during NIT Channel B
(vSS!BViolationB)

1 = A slot boundary error has been detected between NITs on Channel B.

0 = A slot boundary error has not been detected between NITs on Channel B.

37.3.5.9 Aggregated Channel Status Register (ACS)

This register provides the status generated during the communication slot channel operation regardless of the allocation of transmission and reception for all communication slots. In addition, this register includes the status data from a symbol window and NIT. The status data is updated after each slot (updated to the latest at the end of the next slot).

Each flag of this register is cleared by setting "1" to the corresponding bit position. Writing "0" to this bit has no effect on the flag. This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACS 0xD128	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	SBVB	CIB	CEDB	SEDB	VFRB	0	0	0	SBVA	CIA	CEDA	SEDA	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VFRA Valid frame received on channel A (vSS!ValidFrameA)

This bit is used to indicate that one or more valid frames have been received by a static slot or dynamic slot on Channel A.

1 = A valid frame has been received on Channel A.

0 = A valid frame has not been received on Channel A.

SEDA Syntax Error Detected on Channel A (vSS!SyntaxErrorA)

This bit is used to indicate that one or more syntax errors have been detected in the static slot, dynamic slot, symbol window or NIT on Channel A.

1 = A syntax error has been detected on Channel A.

0 = A syntax error has not been detected on Channel A.

CEDA Content error detected on channel A (vSS!ContentErrorA)

This bit is used to indicate that one or more valid frames with content errors were received by a static slot or dynamic slot on Channel A.

1 = A frame with content errors has been received on Channel A.

0 = A frame with content errors has not been received on Channel A.

CIA Add-on communication Indicator Channel A

This bit is used to indicate that one or more valid frames have been received by the slot that included add-on communication of some sort on Channel A. In short, one or more slot received valid frames and one of the following combinations occurred: syntax error, content error and slot boundary error.

1 = A frame including add-on communication of some sort has been received on Channel A.

0 = A frame including add-on communication of some sort has not been received on Channel A.

SBVA Slot boundary violation on channel A (vSS!BViolationA)

This bit is used to indicate that one or more slot boundary errors were detected in the static slot, dynamic slot, symbol window or NIT on Channel A.

1 = A slot boundary error has been detected on Channel A.

0 = A slot boundary error has not been detected on Channel A.

VFRB Valid frame received on channel B (vSS!ValidFrameB)

This bit is used to indicate that one or more valid frames were received by a static slot or dynamic slot on Channel B.

1 = A valid frame has been received on Channel B.

0 = A valid frame has not been received on Channel B.

SEDB Syntax error detected on channel B (vSS!SyntaxErrorB)

This bit is used to indicate that one or more syntax errors have been detected in the static slot, dynamic slot, symbol window or NIT on Channel B.

1 = An syntax error has been detected on Channel B.

0 = An syntax error has not been detected on Channel B.

CEDB Content error detected on channel B (vSS!ContentErrorB)

This bit is used to indicate that one or more valid frames with content errors were received by a static slot or dynamic slot on Channel B.

1 = A frame with content errors has been received on Channel B.

0 = A frame with content errors has not been received on Channel B.

CIB Add-on communication indicator channel B

This bit is used to indicate that one or more valid frames have been received by the slot that included add-on communication of some sort on Channel B. In short, one or more slot received valid frames and one of the following combinations occurred: syntax error, content error and slot boundary error.

1 = A frame including add-on communication of some sort has been received on Channel B.

0 = A frame including add-on communication of some sort has not been received on Channel B.

SBVB Slot boundary violation on channel B (vSS!BViolationB)

This bit is used to indicate that one or more slot boundary errors have been detected in the static slot, dynamic slot, symbol window or NIT on Channel B.

1 = A slot boundary error has been detected on Channel B.

0 = A slot boundary error has not been detected on Channel B.

(Notes) If a slot contains only one frame and the slot boundary at the end of the slot turns into the idle phase, the conditions to set flags CIA and CIB are also fulfilled.

The interrupt flag EIR.EDB will be set to "1" when one of the flags **SEDB**, **CIB**, **CEDB**, and **SBVB** turned from "0" to "1". The interrupt flag EIR.EDA will be set to "1" when one of the flags **SEDA**, **CEDA**, **CIA**, and **SBVA** flags turned from "0" to "1".

37.3.5.10 Even-number Cycle Synchronous Frame ID Register (ESID [1:15])

The 15 registers from **ESID1** to **ESID15** store synchronous frame IDs received in an even-number communication cycle in ascending order and are used for synchronizing up to the limit given by gSyncNodeMax. For this reason, the smallest of reception synchronous frame IDs is stored in the register **ESID1**. The register **ESID1** stores a transmission synchronous frame ID, and **RXEA** and **RXEB** are set when a node transmits a synchronous frame in an even-number communication cycle. The register value is updated in NIT of each even-number communication cycle. This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESIDn 0xD130-0xD168	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXEB	RXEA	0	0	0	0	EID9	EID8	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EID [9:0] Even Sync ID (vsSyncIDListA,B even)

These bits are used to indicate synchronous frame IDs of the even-number communication cycle.

RXEA Even-number cycle synchronous frame reception on channel A

This bit is used to indicate that a synchronous frame corresponding to a synchronous ID was received in an even-number cycle on Channel A. Alternatively the node is configured at the sync nodes where key slot = **EID [9:0]** (ESID1 only).

1 = Sync frame received on Channel A / Transmission sync frame

0 = Sync frame not received on Channel A / No transmission sync frame

RXEB Even-number cycle synchronous frame reception on channel B

This bit is used to indicate that a synchronous frame corresponding to a synchronous ID was received in an even-number cycle on Channel B. Alternatively the node is configured at the sync nodes where key slot = **EID [9:0]** (ESID1 only).

1 = Sync frame received on Channel B / Transmission sync frame

0 = Sync frame not received on Channel B / No transmission sync frame

37.3.5.11 Odd-number Cycle Synchronous Frame ID Register (OSID [1:15])

The 15 registers from **OSID1**to **OSID15** store synchronous frame IDs received in an odd- or even-number communication cycle in ascending order and are used for synchronizing up to the limit given by gSyncNodeMax. For this reason, the smallest of reception synchronous frame IDs is stored in the register **OSID1**. The register **OSID1** stores a transmission synchronous frame ID, and **RXOA** and **RXOB** are set when a node transmits a synchronous frame in an odd-number communication cycle. The register value is updated in NIT of each odd-number communication cycle. This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSIDn 0xD170-0xD1A8	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RXOB	RXOA	0	0	0	0	OID9	OID8	OID7	OID6	OID5	OID4	OID3	OID2	OID1	OID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OID [9:0] Odd-number cycle synchronous frame ID (vsSyncIDListA,B odd)

These bits are used to indicate synchronous frame IDs of the odd-number communication cycle.

RXOA Odd-number cycle synchronous frame reception on Channel A

This bit is used to indicate that a synchronous frame corresponding to a synchronous ID was received in an odd-number cycle on Channel A. Alternatively the node is configured at the sync nodes where key slot = **OID [9:0]** (OSID1 only).

1 = Sync frame received on Channel A / Transmission sync frame

0 = Sync frame not received on Channel A / No transmission sync frame

RXOB Odd-number cycle synchronous frame reception on Channel B

This bit is used to indicate that a synchronous frame corresponding to a synchronous ID was received in an odd-number cycle on Channel B. Alternatively the node is configured at the sync nodes where key slot = **OID [9:0]** (OSID1 only).

1 = Sync frame received on Channel B / Transmission sync frame

0 = Sync frame not received on Channel B / No transmission sync frame

37.3.5.12 Network Management Register (NMV [1...3])

Three network management registers store generated NM vector (configurable 0 to 12 bytes). The NM vector is generated with bit-by-bit OR operation on each NM vector (valid static frame where PPI=1) received on each channel. In NORMAL_ACTIVE or NORMAL_PASSIVE state, the NM vector is updated at the end of each communication cycle. This register is reset when exiting from the CONFIG state or entering into the STARTUP state.

The NMVn register longer than the set NM vector is not valid.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NMVn 0xD1B0-0xD1B8	R	NM 31	NM 30	NM 29	NM 28	NM 27	NM 26	NM 25	NM 24	NM 23	NM 22	NM 21	NM 20	NM 19	NM 18	NM 17	NM 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	NM 15	NM 14	NM 13	NM 12	NM 11	NM 10	NM 9	NM8	NM7	NM6	NM5	NM4	NM3	NM2	NM1	NM0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 37.3-2 below shows the allocation of byte data in a network management vector.

Figure 37.3-2 Allocation of Byte Data in Network Management Vector

Bit / Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NMV1	Data3								Data2								Data1								Data0							
NMV2	Data7								Data6								Data5								Data4							
NMV3	Data11								Data10								Data9								Data8							

37.3.6 Message Buffer Control Registers

37.3.6.1 Message RAM Configuration Register (MRC)

The message RAM configuration register defines message buffers allocated to static segment, dynamic segment and FIFO. This register can be written only in the DEFAULT_CONFIG or CONFIG state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRC 0xD300	R	0	0	0	0	0	SPLM*	SEC1*	SEC0*	LCB7*	LCB6*	LCB5*	LCB4*	LCB3*	LCB2*	LCB1*	LCB0*
	W																
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*	FDB0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FDB [7:0] First dynamic buffer number

0 = There is no buffer group exclusively set to static segments.

1-127 = Message buffers from 0 to FDB-1 are allocated to static segments.

≥ 128 = There is no buffer set to dynamic segments.

FFB [7:0] First buffer number of FIFO

0 = All message buffers are allocated for the FIFO area.

1-127 = Message buffers from FFB to LCB are allocated for the FIFO area.

≥ 128 = There is no message buffer allocated for the FIFO area.

LCB [7:0] Last configured message buffer number

0...127 = The number of message buffers is (LCB+1).

≥ 128 = There is no set message buffer.

SEC [1:0] Secure buffer

These bits are invalid in the DEFAULT_CONFIG or CONFIG state.

For the temporary unlocking, see 37.4.12.4.

00 = Resetting the message buffers having numbers equal to or less than FFB is possible.

Exceptions: In the sync frame transmission or the single slot mode, the operating message buffer 0 (and the message buffer 1 also if SPLM is "1") is always locked.

01 = Reset for the message buffers having numbers smaller than FDB are locked.
In addition, the message buffers set to the static segments having buffer numbers equal to or bigger than FDB cannot be transmitted.

10 = Reset is locked for all message buffers.

11 = Reset is locked for all message buffers.
In addition, the message buffers set to the static segments having buffer numbers equal to or bigger than FDB cannot be transmitted.

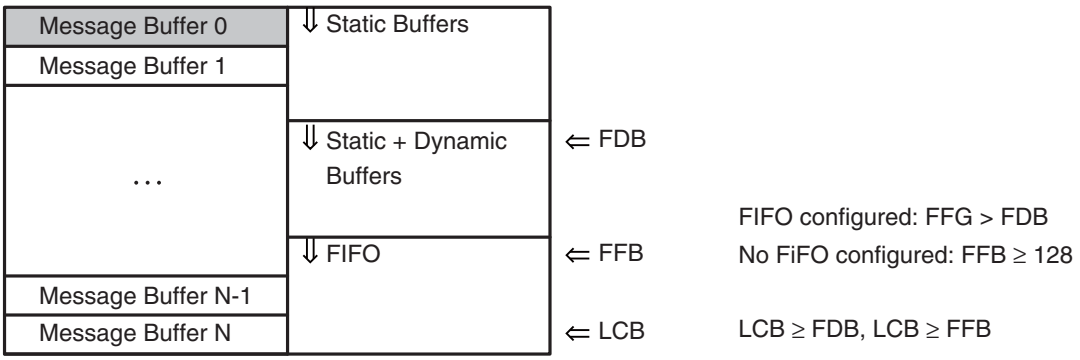
SPLM Sync frame payload multiplex

This bit is valid when the node is configured as a sync node (**SUCC1.TXSY**="1") or in the single slot mode (**SUCC1.TSM**="1"). When this bit is set to "1", message buffer 0 and 1 are dedicated to transmitting sync frames with different payload data on Channels A and B. When this bit is set to "0", the sync frames contain the same payload data on both channels, and are transmitted from message buffer 0. Note that message buffer 1 must be selected according to the channel filter settings for message buffer 0.

0 = Reset is locked for both message buffers 0 and 1.

1 = Reset is locked for message buffer 0.

(Note) If the node is configured as a sync node (SUCC1.TXSY="1") or in the single slot mode (**SUCC1.TSM**="1"), each message buffer 0 and 1 must be prepared as a sync frame or a single slot frame and set with the key slot ID of the node specifications. If the node is not configured as a sync node (SUCC1.TXSY="0") or the single slot message buffer 0 and 1, each message buffer 0 and 1 is handled in the same manner as other message buffers are.



Verify that **FDB [7:0], FFB [7:0],** and **LCB [7:0]** are correctly configured. If not, operation is not ensured. CC does not check wrong settings.

(Note) The maximum number of header sections is 128. This means up to 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The data section length can be configured differently for each message buffer. See Section "37.4.12 Message RAM" for details.
Should two or more message buffers be allocated to the slot 1 through the cycle filtering, they should be arranged at the beginning of the "static buffer" or the "static + dynamic buffer".
The FlexRay protocol specifications require each node to send frames to its key slot. For this reason, the message buffer 0 is reserved for transmission of the key slot. Thus, up to 127 message buffers can be allocated to FIFO. Nevertheless, the non-protocol communication without transmission slots in the static segment will continue to work.
Set the data section lengths and payloads of all of the message buffers that belong to the FIFO using WRHS2.PLC [6:0] and WRHS3.DP [10:0] to DP0 to the same values.
The reset of message buffers belonging to FIFO is locked when CC is not in the DEFAULT_CONFIG or CONFIG state.

37.3.6.2 FIFO Rejection Filter Register (FRF)

A bit string to be compared with the reception frame channel, frame ID and cycle count is set for the FIFO rejection filter register. This register determines whether or not FIFO rejects the message by combining with the FIFO rejection filter mask register. This register can be written only in the DEFAULT_CONFIG or CONFIG state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRF 0xD304	R	0	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
	W																
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	FID10*	FID 9*	FID 8*	FID 7*	FID 6*	FID 5*	FID 4*	FID 3*	FID 2*	FID 1*	FID 0*	CH1*	CH0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [1:0] Channel filter

11 = Reception disabled

10 = Reception only on Channel A

01 = Reception only on Channel B

00 = Reception on both channels

(Note) When the reception on both channels is set, both frames (from Channels A and B) in a static segment are stored in FIFO even though they are the identical frame.

FID [10:0] Frame ID filter

These bits indicate that a frame ID is rejected in FIFO. In the additional configuration of the register FRFM, the corresponding frame ID filter bit that creates further reject frame IDs is ignored.

If the frame ID "0" is set to this filter value while **FRFM.MFID10:0** is "0", FIFO receives all frame IDs.

0...2047 = Frame ID filter value

CYF [6:0] Cycle code filter

The 7-bit cycle counter filter specifies a cycle set and determines the communication cycle applied to the frame ID and channel filters. If the frame ID filter and channel filter are not applied to a cycle by the cycle set being specified by this register, no frame is received in this cycle. See Section "37.4.7.2 Cycle Counter Filtering" for details of the settings of the cycle counter filter.

RSS Rejection of messages in a static segment

When this bit is set to "1", FIFO receives messages in a dynamic segment only.

1 = Messages in a static segment are not received.

0 = Messages in a static and dynamic segments are received.

RNF Reject null frames

When this bit is set to "1", received null frames will not be stored in FIFO.

1 = All null frames are not stored in FIFO.

0 = Null frames are stored in FIFO.

37.3.6.3 FIFO Rejection Filter Mask Register (FRFM)

The FIFO rejection filter mask register specifies bits to be compared with FRF.FID to perform rejection filtering. When a bit in this register is set to "1", the bits are not compared with corresponding FRF.FID bits. This register can be written only in the DEFAULT_CONFIG or CONFIG state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD308	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MFID	MFID9	MFID8	MFID7	MFID6	MFID5	MFID4	MFID3	MFID2	MFID1	MFID0	0	0
	W				10*	*	*	*	*	*	*	*	*	*	*		
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MFID [10:0] Mask Frame ID Filter

1 = Ignores corresponding frame ID filter bits.

0 = Corresponding frame ID filter bits are used for rejection filtering.

37.3.6.4 FIFO Critical Level Register (FCL)

This register can be written only in the DEFAULT_CONFIG or CONFIG state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCL	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD30C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	CL7*	CL6*	CL5*	CL4*	CL3*	CL2*	CL1*	CL0*
	W																
Reset		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

CL [7:0] Critical level

When the values for the reception FIFO fill level FSR.RFFL [7:0] are higher than this register value, the critical level flag FSR.RFCL is set. When a value equal to or higher than 128 is set, the critical level flag FSR.RFCL is not set. The SIR.RFCL signal is set, and an interrupt signal is generated when interrupts are enabled.

37.3.7 Message Buffer Status Register

37.3.7.1 Message Handler Status Register (MHDS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDS	R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
0xD310	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM	MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Writing "1" to a writable bit of this register will clear the register. Writing "0" does not influence the bit. A hardware reset does clear this register.

PIBF Detection of parity error, input buffer RAM 1, 2

1 = A parity error occurred when the input buffer RAM1, 2 is read.

0 = No parity error occurred.

POBF Detection of parity error, output buffer RAM 1, 2

1 = A parity error occurred when the output buffer RAM1, 2 is read.

0 = No parity error occurred.

PMR Detection of parity error Message RAM

1 = A parity error occurred when the message RAM is read.

0 = No parity error occurred.

PTBF1 Detection of parity error, transient buffer RAM A

1 = A parity error occurred when the temporary storage buffer RAM A is read.

0 = No parity error occurred.

PTBF2 Detection of parity error, transient buffer RAM B

1 =A parity error occurred when the temporary storage buffer RAM B is read.

0 = No parity error occurred.

(Notes) **EIR.PERR** is set to "1" when one of the **PIBF**, **POBF**, **PMR**, **PTBF1**, and **PTBF2** turned from "0" to "1".

FMBD Faulty message buffer detected

1 = The message buffer referenced by **FMB [6:0]** includes errors due to parity error.

0 = There are no other failed message buffers.

MFMB Multiple faulty message buffers detected

1 = Another failed message buffer was detected while the flag **FMBD** is set.

0 = There are no other failed message buffers.

CRAM Clear all internal RAMs

This bit is used to indicate whether or not the CHI command CLEAR_RAMS (**CMD [3:0]** ="1100") is in process (writing "0" to all bits of all internal RAM blocks). This bit is set to "1" by the hardware reset or with the CHI command CLEAR_RAMS.

1 = CHI command CLEAR_RAM is in process.

0 = CHI command CLEAR_RAM is not in process.

FMB [6:0] Faulty message buffer

These bits are used to indicate the message buffer number when a parity error occurred in the following cases.

- When the message buffer is read
- When the data is transferred from the input buffer or the temporary storage buffers 1 and 2 to the message buffer

This value is valid only when one of the flags **PIBF**, **PMR**, **PTBF1**, **PTBF2**, and **FMBD** is set. This flag is updated after the flag FMBD is reset. It is not updated while the flag **FMBD** is set.

MBT [6:0] Message buffer number transmitted

These bits are used to indicate the recently and successfully transmitted message buffer number. When the message buffer is set to the single shot mode, the respective **TXR** flags of the registers **TXRQ1/2/3/4** are reset.

MBU [6:0] Message buffer updated

These bits are used to indicate the recently updated message buffer number. The flags **ND** and **MBS** in the registers **NDAT1/2/3/4** and **MBSC1/2/3/4** respectively that correspond to this message buffer will also be updated.

(Notes) **MBT [6:0]** and **MBU [6:0]** will be reset when exiting from the CONFIG state or entering into the STARTUP state.

37.3.7.2 Last Dynamic Transmit Slot (LDTS)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LDTS 0xD314	R	0	0	0	0	0	LDTB 10	LDTB9	LDTB8	LDTB7	LDTB6	LDTB5	LDTB4	LDTB3	LDTB2	LDTB1	LDTB0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	LDTA 10	LDTA9	LDTA8	LDTA7	LDTA6	LDTA5	LDTA4	LDTA3	LDTA2	LDTA1	LDTA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state, or by the CHI command CLEAR_RAMS (**CMD [3:0]** ="1100").

LDTA [10:0] Last dynamic transmission channel A

These bits are used to indicate the vSlotCounter [A] value in a dynamic segment when the last frame is sent on Channel A. This value is updated at the end of a dynamic segment and "0" is set when no frame is sent in the dynamic segment.

LDTB [10:0] Last dynamic transmission channel B

These bits are used to indicate the vSlotCounter [B] value in a dynamic segment when the last frame is sent on Channel B. This value is updated at the end of a dynamic segment and "0" is set when no frame is sent in the dynamic segment.

37.3.7.3 FIFO Status Register (FSR)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FSR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD318	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	RFFL7	RFFL6	RFFL5	RFFL4	RFFL3	RFFL2	RFFL1	RFFL0	0	0	0	0	0	RFO	RFCL	RFNE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is reset when exiting from the CONFIG state or entering into the STARTUP state, or by the CHI command CLEAR_RAMS (CMD [3:0] ="1100").

RFNE Receive FIFO not empty

This bit is set when a valid frame (a null frame that depends on data or rejection mask) is received and stored in FIFO. In addition, an interrupt flag **SIR.RFNE** is set. After the host reads all messages from FIFO, this bit is reset.

1 = A reception FIFO is not empty.

0 = A reception FIFO is empty.

RFCL Receive FIFO critical level

This flag is set when the reception FIFO fill level **RFFL [7:0]** is equal to or greater than the set critical level **FCL.CL [7:0]** . In addition, this flag is cleared right after the level goes below the value.**When RFCL** is set from "0" to "1", **SIR.RFCL** is set to "1", and an interrupt is generated when the setting is valid.

1 = A reception FIFO satisfies the critical level.

0 = A reception FIFO is lower than the critical level.

RFO Receive FIFO overrun

This bit is set when the reception FIFO overrun is detected. The oldest message is overwritten when an overrun occurred. In addition, an interrupt flag **EIR.RFO** is set. This flag is cleared by FIFO read access.

1 = A reception FIFO overrun occurred.

0 = No reception FIFO overrun occurred.

RFFL [7:0] Receive FIFO fill level

These bits are used to indicate the number of FIFO buffers that have not been read by the host. The maximum value is 128.

37.3.7.4 Message Handler Constraints Flags (MHDF)

Message handlers have some constrains regarding the HCLK frequency, message RAM setting and FlexRay bus traffic. These constrains are indicated by setting the MHDF flag.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDF	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD31C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Writing "1" to a writable bit of this register will clear the register. Writing "0" does not influence the bit. A hardware reset does clear this register.

This register is reset when exiting from the CONFIG state or entering into the STARTUP state, or by the CHI command CLEAR_RAMS (**CMD [3:0]** ="1100").

SNUA Status not updated channel A

This bit is set when the status MBS of the message buffer cannot be updated on Channel A due to the overload of the message handler.

1 = No MBS on Channel A updated.

0 = No overload at MBS update on Channel A

SNUB Status not updated channel B

This bit is set when the status MBS of the message buffer cannot be updated on Channel B due to the overload of the message handler.

1 = No MBS on Channel B updated.

0 = No overload at MBS update on Channel B

FNFA Find sequence not finished channel A

This bit is set when the find sequence (the message RAM is scanned to match with the message buffer) cannot be completed on Channel A due to the overload of the message handler.

1 = Channel A sequence incompletion detected.

0 = No Channel A sequence incompletion detected.

FNFB Find sequence not finished channel B

This bit is set when the find sequence (the message RAM is scanned to match with the message buffer) cannot be completed on Channel B due to the overload of the message handler.

1 = Channel B sequence incompletion detected.

0 = No Channel B sequence incompletion detected.

TBFA Transient buffer access failure A

This bit is set when reading/writing access from/to TBF A requested by PRT A cannot be completed within the available time.

1 = TBF A access failed

0 = TBF A access succeeded

TBFB Transient buffer access failure B

This bit is set when reading/writing access from/to TBF B requested by PRT B cannot be completed within the available time.

1 = TBF B access failed

0 = TBF B access succeeded

TNSA Transmission not started channel A

This bit is set when the transmission scheduled for Channel A at the action point of the slot where the message handler is configured is not prepared to start.

1 = Transmission of Channel A not started

0 = Transmission of Channel A started

TNSB Transmission not started channel B

This bit is set when the transmission scheduled for Channel B at the action point of the slot where the message handler is configured is not prepared to start.

1 = Transmission of Channel B not started

0 = Transmission of Channel B started

WAHP Write attempt to header partition

This bit is set when the message handler attempts to write message data to a split header of the message RAM due to incomplete allocation of the message buffer, except for in DEFAULT_CONFIG and CONFIG states. Writing will not be executed to protect header splitting from unintentional write access.

1 = Header partition written

0 = Header partition not written

(Note) When **SNUA**, **SNUB**, **FNFA**, **FNFB**, **TBFA**, **TBFB**, **TNSA**, **TNSB**, or **WAHP** signal turns from "0" to "1", the **EIR.MHF** interrupt flag is set to "1".

37.3.7.5 Transmission Request 1/2/3/4 (TXRQ1/2/3/4)

These four registers reflect the status of the TXR flags for all of the configured message buffers. If the number of message buffers configured is less than 128, the remaining TXR flags have no effect on operation.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXRQ4 0xD32C	R	TXR 127	TXR 126	TXR 125	TXR 124	TXR 123	TXR 122	TXR 121	TXR 120	TXR 119	TXR 118	TXR 117	TXR 116	TXR 115	TXR 114	TXR 113	TXR 112
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR 111	TXR 110	TXR 109	TXR 108	TXR 107	TXR 106	TXR 105	TXR 104	TXR 103	TXR 102	TXR 101	TXR 100	TXR 99	TXR 98	TXR 97	TXR 96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXRQ3 0xD328	R	TXR 95	TXR 94	TXR 93	TXR 92	TXR 91	TXR 90	TXR 89	TXR 88	TXR 87	TXR 86	TXR 85	TXR 84	TXR 83	TXR 82	TXR 81	TXR 80
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR 79	TXR 78	TXR 77	TXR 76	TXR 75	TXR 74	TXR 73	TXR 72	TXR 71	TXR 70	TXR 69	TXR 68	TXR 67	TXR 66	TXR 65	TXR 64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXRQ2 0xD324	R	TXR 63	TXR 62	TXR 61	TXR 60	TXR 59	TXR 58	TXR 57	TXR 56	TXR 55	TXR 54	TXR 53	TXR 52	TXR 51	TXR 50	TXR 49	TXR 48
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR 47	TXR 46	TXR 45	TXR 44	TXR 43	TXR 42	TXR 41	TXR 40	TXR 39	TXR 38	TXR 37	TXR 36	TXR 35	TXR 34	TXR 33	TXR 32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXRQ1 0xD320	R	TXR 31	TXR 30	TXR 29	TXR 28	TXR 27	TXR 26	TXR 25	TXR 24	TXR 23	TXR 22	TXR 21	TXR 20	TXR 19	TXR 18	TXR 17	TXR 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR 15	TXR 14	TXR 13	TXR 12	TXR 11	TXR 10	TXR 9	TXR 8	TXR 7	TXR 6	TXR 5	TXR 4	TXR 3	TXR 2	TXR 1	TXR 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXR [127:0] Transmission request

When the flags are set to "1", the relevant message buffer is set as a transmission buffer and the transmission of the buffer is in process. In a single shot mode, the flags are reset after the transmission completes.

37.3.7.6 New Data Register 1/2/3/4 (NDAT1/2/3/4)

These four registers reflect the status of the **ND** flags for all of the configured message buffers. When the message buffer is set as a transmission buffer, the **ND** flag for the message buffer does not influence the operation. When the number of set message buffers is below 128, the remaining **ND** flags do not influence the operation.

The registers are reset when exiting from the CONFIG state or entering into the STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT4 0xD33C	R	ND 127	ND 126	ND 125	ND 124	ND 123	ND 122	ND 121	ND 120	ND 119	ND 118	ND 117	ND 116	ND 115	ND 114	ND 113	ND 112
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND 111	ND 110	ND 109	ND 108	ND 107	ND 106	ND 105	ND 104	ND 103	ND 102	ND 101	ND 100	ND 99	ND 98	ND 97	ND 96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT3 0xD338	R	ND 95	ND 94	ND 93	ND 92	ND 91	ND 90	ND 89	ND 88	ND 87	ND 86	ND 85	ND 84	ND 83	ND 82	ND 81	ND 80
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND 79	ND 78	ND 77	ND 76	ND 75	ND 74	ND 73	ND 72	ND 71	ND 70	ND 69	ND 68	ND 67	ND 66	ND 65	ND 64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT2 0xD334	R	ND 63	ND 62	ND 61	ND 60	ND 59	ND 58	ND 57	ND 56	ND 55	ND 54	ND 53	ND 52	ND 51	ND 50	ND 49	ND 48
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND 47	ND 46	ND 45	ND 44	ND 43	ND 42	ND 41	ND 40	ND 39	ND 38	ND 37	ND 36	ND 35	ND 34	ND 33	ND 32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT1 0xD330	R	ND 31	ND 30	ND 29	ND 28	ND 27	ND 26	ND 25	ND 24	ND 23	ND 22	ND 21	ND 20	ND 19	ND 18	ND 17	ND 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND 15	ND 14	ND 13	ND 12	ND 11	ND 10	ND 9	ND 8	ND 7	ND 6	ND 5	ND 4	ND 3	ND 2	ND 1	ND 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ND [127:0] New data

These flags are set to "1" when each message buffer data section is updated by valid reception frames that passed the set message buffer filter. These flags are not set for invalid frame reception excepting the

message buffer to the reception FIFO. The flags are cleared to "0" when the header section of the corresponding message buffer is reconfigured or the data section is transferred to the output buffer.

37.3.7.7 Message buffer status update register 1/2/3/4 (MBSC1/2/3/4: Message Buffer Status Changed 1/2/3/4)

These four registers reflect the status of the **MBC** flag of set all message buffers. If the number of set message buffers is less than 128, the remaining **MBC** flag does not make sense.

The registers are reset when turning from the CONFIG state or turning to the STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBSC4 0xD34C	R	MBC 127	MBC 126	MBC 125	MBC 124	MBC 123	MBC 122	MBC 121	MBC 120	MBC 119	MBC 118	MBC 117	MBC 116	MBC 115	MBC 114	MBC 113	MBC 112
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC 111	MBC 110	MBC 109	MBC 108	MBC 107	MBC 106	MBC 105	MBC 104	MBC 103	MBC 102	MBC 101	MBC 100	MBC 99	MBC 98	MBC 97	MBC 96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBSC3 0xD348	R	MBC 95	MBC 94	MBC 93	MBC 92	MBC 91	MBC 90	MBC 89	MBC 88	MBC 87	MBC 86	MBC 85	MBC 84	MBC 83	MBC 82	MBC 81	MBC 80
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC 79	MBC 78	MBC 77	MBC 76	MBC 75	MBC 74	MBC 73	MBC 72	MBC 71	MBC 70	MBC 69	MBC 68	MBC 67	MBC 66	MBC 65	MBC 64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBSC2 0xD344	R	MBC 63	MBC 62	MBC 61	MBC 60	MBC 59	MBC 58	MBC 57	MBC 56	MBC 55	MBC 54	MBC 53	MBC 52	MBC 51	MBC 50	MBC 49	MBC 48
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC 47	MBC 46	MBC 45	MBC 44	MBC 43	MBC 42	MBC 41	MBC 40	MBC 39	MBC 38	MBC 37	MBC 36	MBC 35	MBC 34	MBC 33	MBC 32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBSC1 0xD340	R	MBC 31	MBC 30	MBC 29	MBC 28	MBC 27	MBC 26	MBC 25	MBC 24	MBC 23	MBC 22	MBC 21	MBC 20	MBC 19	MBC 18	MBC 17	MBC 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC 15	MBC 14	MBC 13	MBC 12	MBC 11	MBC 10	MBC 9	MBC 8	MBC 7	MBC 6	MBC 5	MBC 4	MBC 3	MBC 2	MBC 1	MBC 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MBC [127:0] Message Buffer Status Changed

This flag is set to "1" when the status flag (**VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB**) of each message buffer is changed. The flags are cleared to "0" when the header section of the corresponding message buffer is reconfigured or the data section is transferred to the output buffer.

37.3.8 Identification Register

37.3.8.1 Core Release Register (CREL)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREL 0xD3F0	R	REL3	REL2	REL1	REL0	STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	STEP0	YEAR3	YEAR2	YEAR1	YEAR0
	W																
Reset		0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MON7	MON6	MON5	MON4	MON3	MON2	MON1	MON0	DAY7	DAY6	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
	W																
Reset		0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0

DAY [7:0] Date (Design Time Stamp, Day)
2 digits (BCD)

MON [7:0] Month (Design Time Stamp, Month)
2 digits (BCD)

YEAR [3:0] Year (Design Time Stamp, Year)
1 digit (BCD)

STEP [7:0] Release step (Step of Core Release)
2 digits (BCD)

REL [3:0] Release (Core Release)
1 digit (BCD)

37.3.8.2 Endian Register (ENDN)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENDN 0xD3F4	R	ETV 31	ETV 30	ETV 29	ETV 28	ETV 27	ETV 26	ETV 25	ETV 24	ETV 23	ETV 22	ETV 21	ETV 20	ETV 19	ETV 18	ETV 17	ETV 16
	W																
Reset		1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ETV 15	ETV 14	ETV 13	ETV 12	ETV 11	ETV 10	ETV9	ETV8	ETV7	ETV6	ETV5	ETV4	ETV3	ETV2	ETV1	ETV0
	W																
Reset		0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1

ETV [31:0] Endianness Test Value
Test value 0x87654321

37.3.9 Input buffer

The input buffer consists of double buffers: the input buffer host and input buffer shadow. Data is transferred from the input buffer shadow to the message RAM while the host can write to the input buffer host. The input buffer stores the header and data sections to be transferred to the selected message buffer. In addition, this buffer is used to configure the message buffer of the message RAM and update the data section of the transmission buffer.

When the input buffer updates the header section of the message buffer in the message RAM, the message buffer status is automatically reset to zero as shown in Section "37.3.10.5 Message Buffer Status Register (MBS: Message Buffer Status)".

Update the header section of the message buffer belonging to the reception FIFO only in the DEFAULT_CONFIG or CONFIG state.

For details about the data transfer between the input buffer (IBF) and the message RAM, see "• Data transfer from the input buffer to the message RAM" in Section "37.4.11.2 Host access to message RAM".

37.3.9.1 Write data section register (WRDSn: Write Data Section [1...64])

This register sets the data to be transferred to the data section of message buffer. Those data (DWn) will be written to the message RAM in the order when they are transmitted from DW1 (byte0, byte1) to DW_{PL} (the data counts in two bytes defined with PL = payload length).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRDSn 0xD400-0xD4Fc	R	MD 31	MD 30	MD 29	MD 28	MD 27	MD 26	MD 25	MD 24	MD 23	MD 22	MD 21	MD 20	MD 19	MD 18	MD 17	MD 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD 15	MD 14	Md 13	MD 12	MD 11	MD 10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD [31:0] Message Data

MD [7:0] = DW_{2n-1}, $byte_{4n-4}$

MD [15:8] = DW_{2n-1}, $byte_{4n-3}$

MD [23:16] = DW_{2n}, $byte_{4n-2}$

MD [31:24] = DW_{2n}, $byte_{4n-1}$

(Note) **DW127** is placed to **WRDS64.MD [15:0]**. In this case, **WRDS64.MD [31:16]** will not be used (indefinite data). Input buffer RAM will be initialized to 0 at the end of a hardware reset or by a CHI command CLEAR_RAMS (**CMD [3:0]** ="1100").
FlexRay bus is transferred from each msb bit of WRDSn [7:0], WRDSn [15:8], WRDSn [23:16] and WRDSn [31:24].
Read the register ENDN to check how to adjust the data to the endian of the host CPU.

37.3.9.2 Write header section register 1 (WRHS1: Write Header Section 1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRHS1 0xD500	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC 6	CYC 5	CYC 4	CYC 3	CYC 2	CYC 1	CYC 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID 10	FID 9	FID 8	FID 7	FID 6	FID 5	FID 4	FID 3	FID 2	FID 1	FID 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FID [10:0] Frame ID

These bits indicate the frame ID of selected message buffer. The frame ID defines the slot number of each message transmission/reception. The message buffer with frame ID = 0 is invalid.

CYC [6:0] Cycle code

These 7-bit code determines the cycle set used for cycle counter filtering. See Section "37.4.7.2 Cycle Counter Filtering" for details of the configuration of cycle code.

CHA, **CHB** Channel Filter Control

This 2-bit channel filtering field related to each buffer has the roles for a filter of the reception buffer and for the control field of the transmission buffer.

CHA	CHB	Transmit Buffer transmit frame on	Receive Buffer store frame received from
1	1	both channels (static segment only)	channel A or B (store first semantically valid frame, static segment only)
1	0	channel A	channel A
0	1	channel B	channel B
0	0	no transmission	ignore frame

(Note) The message buffer is set for the dynamic segment. The frame is not sent and the received frame is ignored when both bits of the channel filter control (**CHA**, **CHB**) are set to "1".
(Same function as **CHA** = **CHB** = "0")

CFG Message Buffer Configuration Bit

This bit is used for setting each buffer as a transmission buffer or a reception buffer. The message buffer which belongs to reception FIFO will be invalid.

1 = The corresponding buffer is configured as a **transmission buffer**.

0 = The corresponding buffer is configured as a **reception buffer**.

PPIT Payload Preamble Indicator Transmit

This bit is used for controlling the states of the payload preamble indicator on transmission frames. If this bit is set to the static message buffer, each message buffer will hold the network management information. If this bit is set to the dynamic message buffer, the first 2 bytes of the payload segment will be used for message ID filtering. Message ID filtering for reception frame is not supported in FlexRay controller.

1 = Set a payload preamble indicator

0 = No set payload preamble indicator

TXM Transmission Mode

This bit selects transmission modes (See Section "37.4.8.3 Transmission Buffer").

1 = Single shot mode

0 = Continuous mode

MBI Message Buffer Interrupt

This bit enables transmission/reception interrupts for each message buffer. **SIR.RXI** or **SIR.MBSI** is set to "1" after a message has received in the reception buffer. The **SIR.TXI** flag is set to "1" after a message has been sent from the transmission buffer successfully.

1 = Transmission/reception interrupt of the corresponding message buffer is effective

0 = Transmission/reception interrupt of the corresponding message buffer is ineffective

37.3.9.3 Write header section register 2 (WRHS2: Write Header Section 2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRHS2	R	0	0	0	0	0	0	0	0	0	PLC	PLC	PLC	PLC	PLC	PLC	PLC
0xD504	W										6	5	4	3	2	1	0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC	CRC	CRC	CRC	CRC	CRC	CRC	CRC	CRC	CRC	CRC
	W						10	9	8	7	6	5	4	3	2	1	0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC [10:0] Header CRC (vRF!Header!HeaderCRC)

Reception buffer: Needs no configuration

Transmission buffer: Header CRC will be computed and configured by the host

For header CRC calculation, the payload length of frame must be reported to the host. The payload length of all frames in the static segment is configured with MHDC.SFDL [6:0].

PLC [6:0] Payload Length Configured

These bits indicate the data section length (in two bytes) set by the host. The static frame payload length set with **MHDC.SFDL [6:0]** in the static segment defines the payload length of whole static frames.If the payload length set by **PLC [6:0]** is shorter than that set by **MHDC.SFDL [6:0]**, padding bytes are inserted to ensure the payload length of the static frame. The padding bytes are expressed as "0" (See Section "37.4.8.3 Transmission Buffer").

37.3.9.4 Write header section register 3 (WRHS3: Write Header Section 3)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRHS3 0xD508	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DP [10:0] Data Pointer

These bits indicate the pointer to the first 32-bit data of the data section of message buffer.

37.3.9.5 Input Buffer Command Mask Register (IBCM)

This register sets how to update the message buffer selected by **IBCR** register. When the IBF host and the IBF shadow are switched, mask bits **LHSH**, **LDSH**, **STXRH** and mask bits **LHSS**, **LDSS**, **STXRS** will also be switched.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCM 0xD510	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRS	LDSS	LHSS
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRH	LDSH	LHSH
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LHSH Load Header Section Host

1 = A header section being transferred from the input buffer to the message RAM

0 = No header section being transferred

LDSH Load Data Section Host

1 = A data section being transferred from the input buffer to the message RAM

0 = No data section being transferred

STXRH Set Transmission Request Flag Host

If this bit is set to "1", the **TXR** flag in the message buffer selected will be set to "1" in the **TXRQ1/2/3/4** registers and the messages in their transmission buffers will be released. In a single shot mode, these flags will be cleared after the completion of the transmission.

1 = TXR flag being set and message in the transmission buffer being released

0 = TXR flag being reset

LHSS Load Header Section Shadow

1 = A header section being transferred from the input buffer to the message RAM or those operations completed

0 =No header section being transferred from the input buffer to the message RAM

LDSS Load Data Section Shadow

1 = A data section being transferred from the input buffer to the message RAM or those operations completed

0 =No data section being transferred from the input buffer to the message RAM

STXRS Set Transmission Request Flag Shadow

1 = TXR flag being set and message in the transmission buffer being released or those operations completed

0 = TXR flag being reset

37.3.9.6 Input buffer command request register (IBCR: Input Buffer Command Request)

If you write the target message buffer number in the message RAM to **IBRH [6:0]**, the IBF host and the IBF shadow will be switched. Further, the message buffer numbers stored in **IBRH [6:0]** and **IBRS [6:0]** will also be switched. (See "• Data transfer from the input buffer to the message RAM" in Section "37.4.11.2 Host access to message RAM".)

This write operation will set "1" to **IBSYS** bit. Then, the message handler starts to transfer the contents of the IBF shadow to the message buffer in the message RAM selected by **IBRS [6:0]**.

While transferring data from the IBF shadow to the message buffer in the message RAM, you can write the next transmission message to the IBF host. IBSYS will be cleared to "0" after the transfer between IBF shadow and message RAM completed. When you write the target message buffer number of the next transmission message to **IBRH [6:0],** the next transfer to the message RAM will start.

When a write access to **IBRH [6:0]** occurs while **IBSYS** = "1", **IBSYH** will be set to "1". After the current data transfer from the IBF shadow to the message RAM has completed, the message buffer numbers being stored in **IBRH [6:0]** and **IBRS [6:0]** will be switched as well as the IBF host and the IBF shadow will be switched. Then, **IBSYH** will be reset to "0". If "1" remains in **IBSYS**, the next transfer to the message RAM will start.

While "1" is set in both **IBSYS** and **IBSYH**, if a write to this input buffer register takes place, the error flag **EIR.IIBA** will be set to "1". In this case, the input buffer will not be changed.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCR	R	IBSYS	0	0	0	0	0	0	0	0	IBRS6	IBRS5	IBRS4	IBRS3	IBRS2	IBRS1	IBRS0
0xD514	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	IBSYH	0	0	0	0	0	0	0	0	IBRH6	IBRH5	IBRH4	IBRH3	IBRH2	IBRH1	IBRH0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IBRH [6:0] Input Buffer Host Transfer Request

These flags select the target message buffer number in the message RAM for a data transfer from input buffer. The valid values are 0x00 to 0x7F (0 to 127).

IBSYH Input Buffer Busy Host

This flag will be set to "1" after a write to **IBRH [6:0]** while **IBSYS** is "1". IBSYH will be cleared to "0" after the current data transfer between IBF shadow and message RAM completed.

1 = A message transfer suspended

0 = No message transfer suspended

IBRS [6:0] Input Buffer Request Shadow

These flags indicate the currently updated or the last updated target message buffer number. The valid values are 0x00 to 0x7F (0 to 127).

IBSYS Input Buffer Busy Shadow

This flag will be set to "1" after a write to **IBRH [6:0]** takes place. **IBSYS** will be cleared to "0" after the transfer between IBF shadow and message RAM has been completed.

1 = A transfer between IBF shadow and message RAM in progress

0 = A transfer between IBF shadow and message RAM completed

37.3.10 Output buffer

Output buffer has double buffering structure comprising an output buffer host and an output buffer shadow and is used to read a message buffer from the message RAM. Selected message buffers will be transferred from the message RAM to the output buffer shadow while the host can read an output buffer host. For details about the data transfer between the output buffer (OBF) and the message RAM, see "• Data transfer from the input buffer to the message RAM" in Section "37.4.11.2 Host access to message RAM".

37.3.10.1 Read Data Section Register (RDDSn: Read Data Section [~164])

This register sets the data read from the data section of the message buffer. Those data (DWn) will be read from the message RAM in the order when they were received, from DW1(byte 0, byte 1) to DW_{PL} (the data counts in two bytes defined with PL = payload length).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDDSn 0xD600-0xD6FC	R	MD 31	MD 30	MD 29	MD 28	MD 27	MD 26	MD 25	MD 24	MD 23	MD 22	MD 21	MD 20	MD 19	MD 18	MD 17	MD 16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD 15	MD 14	MD 13	MD 12	MD 11	MD 10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD [31:0] Message Data

MD [7:0] = $DW2_{n-1}$, $byte4_{n-4}$

MD [15:8] = $DW2_{n-1}$, $byte4_{n-3}$

MD [23:16] = $DW2_{n}$, $byte4_{n-2}$

MD [31:24] = $DW2_{n}$, $byte4_{n-1}$

(Note) **DW127** will be placed in **RDDS64.MD [15:0]**. In this case, **RDDS64.MD [31:16]** will not be used (indefinite data).
Output buffer RAM will be initialized to 0 at the end of a hardware reset or by a CHI command CLEAR_RAMS (**CMD [3:0]** ="1100").

37.3.10.2 Read header section register 1 (RDHS1: Read Header Section 1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDHS1 0xD700	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Followings are the values to set by using **WRHS1**.

FID [10:0] Frame ID
CYC [6:0] Cycle code
CHA, CHB Channel Filter Control
CFG Message Buffer Configuration Bit
PPIT Payload Preamble Indicator Transmit
TXM Transmission Mode
MBI Message Buffer Interrupt

If the message buffer read from the message RAM belongs to the reception FIFO, the **FID [10:0]** will hold received frame IDs while **CYC [6:0], CHA**, **CHB**, **CFG**, **PPIT**, **TXM**, and **MBI** are reset to "0".

37.3.10.3 Read header section register 2 (RDHS2: Read Header Section 2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDHS2 0xD704	R	0	PLR 6	PLR 5	PLR 4	PLR 3	PLR 2	PLR 1	PLR 0	0	PLC 6	PLC 5	PLC 4	PLC 3	PLC 2	PLC 1	PLC 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC 10	CRC 9	CRC 8	CRC 7	CRC 6	CRC 5	CRC 4	CRC 3	CRC 2	CRC 1	CRC 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC [10:0] Header CRC (vRF!Header!HeaderCRC)

Reception buffer: Header CRC is updated by reception frames.

Transmission buffer: The header CRC configured by the message transfer from the input buffer will be displayed.

PLC [6:0] Payload Length Configured

These bits indicate the data section length (in 2 bytes) set by the host.

PLR [6:0] Payload Length Received (vRF!Header!Length)

These bits indicate the payload length updated on reception frame.

The operation, regarding to the received payload length and the set payload length, to store the message into the message buffer is as follows.

PLR [6:0] > PLC [6:0] : The payload data stored in the message buffer will be truncated to the payload length of the configured **PLC [6:0]** or **PLC [6:0]** +1.

PLR [6:0] > PLC [6:0] : Received payload data will be stored in the data section of the message buffer. The remaining data bytes in the data section set with **PLC [6:0]** will be filled with indefinite data.

PLR [6:0] =0: The data section in the message buffer is filled with indefinite data.

PLC [6:0] =0: No data section in the message buffer. No data will be stored in the data section of the message buffer.

(Note) Message RAM is composed of 4 bytes. When you store reception data in the data section of the message buffer, the 2-byte data word written to the message buffer will be **PLC [6:0]** rounded up to the next even number. Set the same value to **PLC [6:0]** for all message buffers in the reception FIFO. Header 2 will be updated only by data frames.

37.3.10.4 Read header section register 3 (RDHS3: Read Header Section 3)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDHS3 0xD708	R	0	0	RES	PPI	NFI	SYN	SFI	RCI	0	0	RCC 5	RCC 4	RCC 3	RCC 2	RCC 1	RCC 0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DP [10:0] Data Pointer

These bits indicate the pointer to the first 32-bit data of the data section of message buffer.

RCC [5:0] Receive Cycle Count (vRF!Header!CycleCount)

These bits indicate the cycle counter value to be updated by reception frame.

RCI Received on Channel Indicator (vSS!Channel)

This bit indicates which channel received the reception frames to update each reception buffer.

1 = Frames received on channel A

0 = Frames received on channel B

SFI Startup Frame Indicator (vRF!Header!SuFIndicator)

This bit indicates that the received frame is a startup frame.

1 = Reception frame is a startup frame

0 = Reception frame is not a startup frame

SYN Sync Frame Indicator (vRF!Header!SyFIndicator)

This bit indicates that received frame is a synchronous frame.

1 = Reception frame is a synchronous frame

0 = Reception frame is not a synchronous frame

NFI Null Frame Indicator (vRF!Header!NFIndicator)

If this bit is "0", there is no valid data in the payload segment of the received frame.

1 = Reception frame is not a null frame

0 = Reception frame is a null frame

PPI Payload Preamble Indicator (vRF!Header!PPIndicator)

This bit indicates whether a network management vector or a message ID is included in the payload segment of the received frame or not.

1 = Static segment: Network management vector included at the beginning of the payload
Dynamic segment: Message ID included at the beginning of the payload

0= Neither network management vector nor message ID is included in the payload segment of the reception frame

RES Reserved Bit (vRF!Header!Reserved)

Reflects the state of reserved bit received.

(Note) Header 3 will be updated only by data frames.

37.3.10.5 Message Buffer Status Register (MBS: Message Buffer Status)

Message buffer status will be updated by channel allocated at the end of the next slot to a slot allocated to the message buffer. If only one (A or B) channel is allocated to a message buffer, the status flag of the other channel will be cleared to "0". If both channels are allocated to a message buffer, the status flags of those channels will be updated.

The message buffer status always shows the state of the last slot allocated to the message buffer. If the host updates a message buffer through the input buffer, all the MBS flags will be reset even if their IBCM bit is set. See Sections "37.4.7 Filtering and Masking", "37.4.8 Transmission Procedure" and "37.4.9 Reception Procedure" for the details of transmission/reception filtering. If the message handler changes one of the flags: **VFRA**, **VFRA**, **SEOA**, **SEOB**, **CEOA**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB**, the MBC flag of each message buffer of MBSC1/2/3/4 register will be set.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBS	R	0	0	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	0	0	CCS5	CCS4	CCS3	CCS2	CCS1	CCS0
0xD70C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FTB	FTA	0	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **Status flags for reception and transmission buffers**

VFRA Valid Frame Received on Channel A (vSS!ValidFrameA)

This bit is set to "1" when a valid frame is received on ch.A.

1 = A valid frame received on Channel A

0 = No valid frame received on Channel A

VFRB Valid Frame Received on Channel B (vSS!ValidFrameB)

This bit is set to "1" when a valid frame is received on ch.B.

1 = A valid frame received on ch.B

0 = No valid frame received on ch.B

SEOA Syntax Error Observed on Channel A (vSS!SyntaxErrorA)

This bit indicates that a syntax error has been detected on the slot assigned to ch.A.

1 = A syntax error detected on Channel A

0 = No syntax error detected on Channel A

SEOB Syntax Error Observed on Channel B (vSS!SyntaxErrorB)

This bit indicates that a syntax error has been detected on the slot assigned to ch.B.

1 = A syntax error detected on Channel B

0 = No syntax error detected on Channel B

CEOA Content Error Observed on Channel A (vSS!ContentErrorA)

This bit indicates that a content error has been detected on the slot assigned to ch.A.

1 = A content error detected on Channel A

0 = No content error detected on Channel A

CEOB Content Error Observed on Channel B (vSS!ContentErrorB)

This bit indicates that a content error has been detected on the slot assigned to ch.B.

1 = A content error detected on Channel B

0 = No content error detected on Channel B

SVOA Slot Boundary Violation Observed on Channel A (vSS!BViolationA)

This bit indicates that a slot boundary error has been detected on the slot assigned to ch.A. That is, this means that the channel is active at the beginning or at the end of the slot configured.

1 = A slot boundary error has been detected on Channel A

0 = No slot boundary error has been detected on Channel A

SVOB Slot Boundary Violation Observed on Channel B (vSS!BViolationB)

This bit indicates that a slot boundary error has been detected on the slot assigned to ch.B. That is, this means that the channel is active at the beginning or at the end of the slot configured.

1 = A slot boundary error has been detected on Channel B

0 = No slot boundary error has been detected on Channel B

- **Status flags for transmission buffers**

TCIA Transmission Conflict Indication Channel A
(vSS!TxConflictA)

This bit will be set to "1" when a transmission collision is detected on ch.A.

1 = A transmission collision detected on Channel A

0 = No transmission collision detected on Channel A

TCIB Transmission Conflict Indication Channel B
(vSS!TxConflictB)

This bit will be set to "1" when a transmission collision is detected on ch.B.

1 = A transmission collision detected on Channel B

0 = No transmission collision detected on Channel B

- **Status flags for reception buffers**

ESA Empty Slot Channel A

An empty slot means that the bus is in idle state, that is, a frame transmission has not been detected. This state will be checked on static slots and dynamic slots.

1 = Bus is in idle state on the slot assigned by Channel A

0 = Bus is not in idle state on the slot assigned by Channel A

ESB Empty Slot Channel B

An empty slot means that the bus is in idle state, that is, a frame transmission has not been detected. This state will be checked on static slots and dynamic slots.

1 = Bus is in idle state on the slot assigned by Channel B

0 = Bus is not in idle state on the slot assigned by Channel B

MLST Message Lost

This bit is set when a message has not been read before the message buffer is overwritten with a new message. A reception of null frame does not affect except for the message buffer being received on the reception FIFO. The flag is reset when a new message is saved in the message buffer after the ND flag of the message buffer is reset by writing the message buffer to IBF or by reading the message buffer from OBF.

1 = A message that has not been read was overwritten

0 = No message lost

FTA Frame Transmitted on Channel A

This bit indicates that a data frame has been sent to Channel A.

1 = A data frame sent to Channel A

0 = No data frame sent to Channel A

FTB Frame Transmitted on Channel B

This bit indicates that a data frame has been sent to Channel B.

1 = A data frame sent to Channel B

0 = No data frame sent to Channel B

(Note) Only the host can reset **FTA** and **FTB**. Therefore, the cycle count status **CCS [5:0]** is valid when the bit is set to "1".

CCS [5:0] Cycle Count Status

The cycle count is incremented when the status is updated.

Next status bit will be updated from both valid and invalid frame data in the reception buffer (CFG="0"). If a valid frame has not been received, the previous value will be kept.

It is meaningless in a transmission buffer and is ignored.

RCIS Received on Channel Indicator Status (vSS!Cahnnel)

1 = Frames received on channel A

0 = Frames received on channel B

SFIS Startup Frame Indicator Status (vRF!Header!SuFIndicator)

1 = Startup frame received

0 = No startup frame received

SYNS Sync Frame Indicator Status (vRF!Header!SyFIndicator)

1 = Sync frame received

0 = No sync frame received

NFIS Null Frame Indicator Status (vRF!Header!NFIndicator)

1 = Received frame is not a null frame

0 = Received frame is a null frame

PPIS Payload Preamble Indicater Status (vRF!Header!PPIndicator)

Payload preamble indicator defines whether a network management vector or a message ID is included in the payload segment of the received frame or not.

1 = Static segment: Network management included
Dynamic segment: Message ID included

0 = Not included

RESS Received Bit Status (vRF!Header!Reserved)

This bit indicates the states of reception reserved bits. The reserved bit will be sent as "0".

37.3.10.6 Output buffer command mask register (OBCM: Output Buffer Command Mask)

This register configures how to update the output buffer by the message buffer selected by **OBCR** register. When the message buffer transfer is requested by **OBCR.REQ**, mask bits **RDSS** and **RHSS** are copied to the internal register. When the OBF host and the OBF shadow are switched, the mask bits **RDSH**:**RHSH** and mask bits **RDSS**:**RHSS** will also be switched. Data transfer between the output buffer (OBF) and the message RAM is described in "• Data transfer from the message RAM to the output buffer" of Section "37.4.11.2 Host access to message RAM" in detail.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSH	RHSH
0xD710	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSS	RHSS
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RHSS Read Header Section Shadow

1 = A header section being transferred from the message RAM to the output buffer

0 = No header section read out

RDSS Read Data Section Shadow

1 = A data section being transferred from the message RAM to the output buffer

0 = No data section read out

RHSH Read Header Section Host

1 = A header section being transferred from the message RAM to the output buffer

0 = No header section read out

RDSH Read Data Section Host

1 = A data section being transferred from the message RAM to the output buffer

0 = No data section read out

(Note) After completing the header section transfer from the message RAM to the OBF shadow, the message buffer status update flag **MBS** in the selected message buffer of **MBSC1/2/3/4** will be cleared to "0".
After completing the data section transfer from the message RAM to the OBF shadow, the new data flag **ND** in the selected message buffer of **NDAT1/2/3/4** register will be cleared to "0".

37.3.10.7 Output Buffer Command Request Register (OBCR: Output Buffer Command Request)

When **OBSYS** is "0" and **REQ** is set to "1", **OBSYS** is set to "1". **OBRS [6:0]** is copied to the internal register and mask bits **OBCM.RDSS** and **OBCM.RHSS** are copied to the internal register OBCM. Then, the message buffer selected by **OBRS [6:0]** starts to be transferred from the message buffer to the OBF shadow. **OBSYS** bit will be cleared to "0" after the transfer between the message buffer and the OBF shadow.

Setting "1" to **VIEW** while **OBSYS** is "0" switches between the OBF host and the OBF shadow. Further, mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped by the internal register OBCM and are corresponding to each output buffer transfer. **OBRH [6:0]** indicates the number of message buffers accessible by CPU.

OBSYS will be set to "1" when **REQ** and **VIEW** are set to "1" at the same time while **OBSYS** is "0". Then, the OBF host and the OBF shadow are switched. Further, mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped by the internal register and are corresponding to each output buffer transfer. Then, **OBRS [6:0]** will be copied to the internal register. Then, it starts transferring the message RAM of the selected message buffer to the OBF shadow. CPU can read the message buffer transferred from the OBF host in process of transfer. The **OBSYS** bit will be cleared to "0" after the transfer between the message RAM and the OBF shadow has been completed.

If you write this output buffer register while **OBSYS** is "1", the error flag **EIR.IOBA** will be set to "1". In this case, the output buffer will not be changed.

Data transfer between the output buffer and the message RAM is described in "• Data transfer from the input buffer to the message RAM" in Section "37.4.11.2 Host access to message RAM".

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBCR 0xD714	R	0	0	0	0	0	0	0	0	0	OBRH6	OBRH5	OBRH4	OBRH3	OBRH2	OBRH1	OBRH0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OBSYS	0	0	0	0	0	REQ	VIEW	0	OBRS6	OBRS5	OBRS4	OBRS3	OBRS2	PBRS1	OBRS0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OBRS [6:0] Output Buffer Request Shadow

These flags indicate the message buffer number to transfer from the message RAM to the OBF shadow. The valid values are 0x00 to 0x7F (0 to 127). If the first message buffer number in the reception FIFO has been written to this register, the message buffer specified with GET Index (GIDX) (See Section "37.4.10 FIFO Function") will be transferred to the OBF shadow.

VIEW View Shadow Buffer

This flag switches between OBF shadow and OBF host. Write operations are possible only when **OBSYS** is "0".

1 = Switches between OBF shadow and OBF host.

0 = Does not switch between OBF shadow and OBF host.

REQ Request Message RAM Transfer

The message buffer specified by **OBRS [6:0]** starts to be transferred from the message RAM to the OBF shadow.Write operations are possible only when **OBSYS** is "0".

1 = A transfer from message RAM to OBF shadow requested

0 = No transfer from message RAM to OBF shadow requested

OBSYS Output Buffer Shadow Busy

This flag will be set to "1" after **REQ** bit has been set to "1". **OBSYS** will be cleared to "0" after the transfer between the message RAM and the OBF shadow has been completed.

1 = A transfer between massage RAM and OBF shadow in progress

0 = No transfer between massage RAM and OBF shadow in progress

OBRH [6:0] Output Buffer Request Host

These flags indicate the message buffer numbers currently accessible via **RDHS [1...3], MBS** and **RDDS [1...64]**. Writing "1" to **VIEW** launches a switch between the OBF shadow and the OBF host, which enables the transferred message buffer to become accessible. The valid values are 0x00 to 0x7F (0 to 127).

37.4 Operation of FlexRay

This section describes the protocol function of FlexRay. For more information about FlexRay protocol, see FlexRay Protocol Specification V2.1.

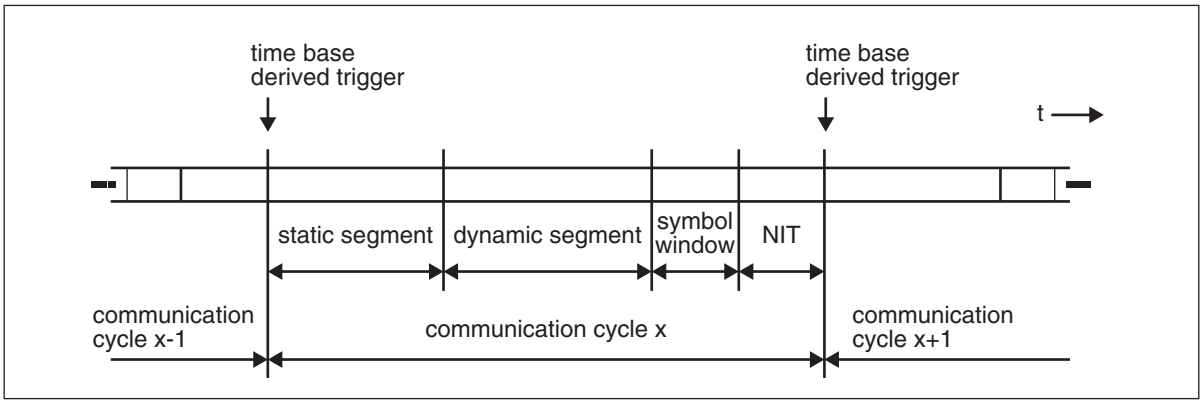
37.4.1 Communication cycle

The FlexRay communication cycle consists of the following elements:

- Static segment
- Dynamic segment (option)
- Symbol window (option)
- Network idle time (NIT)

Network communication time (NCT) consists of the static segment, dynamic segment, and symbol window. For each communication channel, the slot counter starts from "1" and is incremented up to the end of the dynamic segment. In addition, both channels use the same synchronized macrotick.

Figure 37.4-1 Structure of Communication Cycle



■ Static segment

The static segment is characterized by the following points:

- The slot is protected by the bus guardian (if available).
- Frame transmission is initiated at the action point of each static slot.
- Both channels have the same payload length in all frames.

Parameter: Number of static slots **GTUC7.NSS [9:0]**
Static slot length **GTUC7.SSL [9:0]**
Static frame data length **MHDC.SFDL [6:0]**
Action point offset **GTUC9.APO [5:0]**

■ Dynamic segment

The dynamic segment is characterized by the following points:

- The bus guardian is disabled (even if it is available) and all controllers have bus access.
- As the slot lengths are variable, both channels will be different.
- The transmission begins at the minislot action point.

Parameter: Number of minislots **GTUC8.NMS [12:0]**
Minislot length **GTUC8.MSL [5:0]**
Minislot action point offset **GTUC9.MAPO [4:0]**
End of transmission minislot value **MHDC.SLT [12:0]**

■ Symbol window

FlexRay Protocol Specification V2.1 defines three symbols.

- Wake up symbols (WUS): will be sent only in the WAKEUP state.
- Collision avoid symbol (CAS): will be sent only in the STARTUP state.
- Media Access Test Symbol (MTS): will be sent in the NORMAL_ACTIVE state in order to test the bus guardian.

In a symbol window interval, 1MTS symbol will be sent for each channel.
The symbol window is characterized by the following points:

- Transmit one symbol.
- Start MTS symbol transmission at the symbol window action point.

Parameter: Action point offset **GTUC9.APO [5:0]**
Network idle time start **GTUC4.NIT [13:0]**

■ Network idle time (NIT: Network Idle Time)

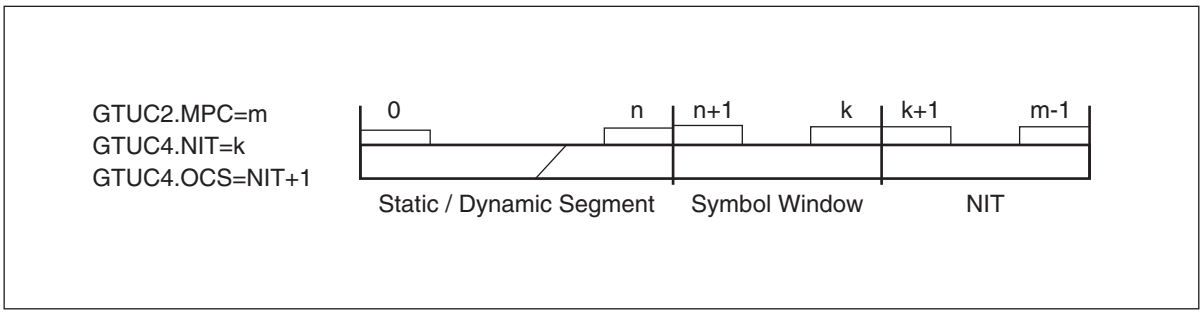
During the network idle time (NIT), the FlexRay controller performs the following tasks:

- Calculate clock correction time (offset and rate).
- After starting offset correction, implement offset correction for each macrotick.
- Perform cluster cycle related tasks.

Parameter: Network idle time start **GTUC4.NIT [13:0],**
Offset correction start **GTUC4.OCS [13:0]**

■ Configuration of NIT Start and Offset Correction Start

Figure 37.4-2 Configuration of NIT Start and Offset Correction Start



Assuming that the number of microtick gMacroPerCycle per cycle is m, set the value as **GTUC2.MPC** = m.
In addition, assume that the static/dynamic segment starts at macrotick 0 and ends at macrotick n.

n = Static segment length + Dynamic segment offset + Dynamic segment length - 1MT
= gNumberOfStaticSlots • gdStaticSlot + dynamic segment offset
+ gNumberOfMinislots • gdMinislot - 1MT

Set the static segment length in **GTUC7.SSL** and **GTUC7.NSS**.
Set the dynamic segment length in **GTUC8.MSL** and **GTUC8.NMS**.

Dynamic segment offset is calculated as follows:

if gdActionPointOffset ≤ gdMinislotActionPointOffset :
Dynamic segment offset = 0 MT

else if gdActionPointOffset > gdMinislotActionPointOffset :
Dynamic segment offset = gdActionPointOffset - gdMinislotActionPointOffset

If NIT starts at macrotick k+1 and ends at the last macrotick with m-1 cycles, make settings as follows:
GTUC4.NIT = k

In addition, set offset correction start in order to satisfy the following conditions:
GTUC4.OCS ≥ **GTUC4.NIT** + 1 = k+1

The symbol window length between static/dynamic segment end and NIT start is calculated as k-n.

37.4.2 Communication mode

FlexRay Protocol v2.1 supports the time trigger distribution mode (TT-D). This section explains the communication mode with the time trigger distribution (TT-D) mode.

■ Time trigger distribution mode (TT-D: Time-triggered Distributed)

The following communication modes are allowed in the TT-D mode:

- Pure static :Minimum 2 static slots + symbol window (option)
- Mixed static/dynamic:Minimum 2 static slots + dynamic segment + symbol window (option)

To enable operation in the time trigger distribution mode, at least 2 cold start nodes are required. In addition, for cluster startup, 2 faultless cold start nodes are required. Each startup frame must be a synchronization frame. All cold start nodes are synchronous nodes.

37.4.3 Clock synchronization

Distributed clock synchronization is used in the TT-D mode. Each node is synchronized with clusters by measuring the synchronous frame reception timing of the other nodes.

37.4.3.1 Global time

Each node has its individual clock and operates based on the global time concept. Global time consists of the vector with 2 values, i.e. cycle (cycle counter) and cycle time (macrotick counter).

- Macrotick (MT) = Basic measurement unit of FlexRay network time
  (Macrotick is equal to the integral multiple of microtick [μT])
- Cycle = Unit of 1 communication cycle period
  (Cycle is represented as macrotick [MT])

37.4.3.2 Local time

Inside the node, node operation time is defined based on the accuracy in the units of microtick. Microtick is the time unit specific to each controller obtained through system clocks of individual nodes. Therefore, if controllers are different in the same node, they may have different times. The accuracy of local time error measurement is in the units of (μT) microtick.

- Macrotick generation order: System clock → prescaler → microtick (μT)
- μT = Measuring unit of FlexRay controller time (clock correction is performed in the units of μT).
- Cycle counter + macrotick counter = Node global time local view

37.4.3.3 Synchronization process

The synchronous frame is used to perform clock synchronization. Only predefined synchronous nodes can transmit synchronous frames. The synchronous node should transmit the synchronous frame to both channels in the 2-channel cluster.

The following restrictions apply to FlexRay synchronization:

- 1 node has a maximum of 1 synchronous frame in each communication cycle.
- 1 cluster has a maximum of 15 synchronous frames in each communication cycle.
- For clock synchronization, all nodes must use predefined number of synchronous frames (**GTUC2.SNM [3:0]**.
- A minimum of 2 synchronous nodes are required for clock synchronization and startup.

To perform clock synchronization, time deviations between expected reception time and actually observed reception time for each synchronous frame received are measured during the static segment period. Correction time is calculated through the use of the FTM algorithm during the NIT (offset: all cycles, rate: odd number cycle). For details, see Chapter 8 of "FlexRay Protocol Specification V2.1".

- **Offset (phase) correction**

- Time deviations of the current cycle are measured.
- In the node with 2 channels, a smaller value measured in each channel is adopted as a calculation value.
- Calculation is performed during the NIT period of all communication cycles.
- The offset correction value calculated in the even number cycle is used only for error check.

- Error check is done by collating with the limit value.
- A correction value is an integer with a code of μT.
- Correction is performed in the odd number cycle. This is distributed over each macrotick ranging from offset correction start to cycle end (end of NIT) and lengthens and shortens the number of current cycles by several MTs by shifting the start position of the next cycle each node.

- **Rate (frequency) correction**

- Differences of each time deviation (differential deviation time) is measured for the even number cycle and odd number cycle.
- In the node with 2 channels, the average of differential deviation time measured in each channel is adopted as a calculation value.
- Calculation is performed during the NIT period of odd-number communication cycles.
- Cluster drift damping is performed using a global damping value.
- Error check is done by collating with the limit value.
- A correction value is an integer with a code of μT.
- Correction is performed in the next even-number/odd-number cycle pair. This is distributed in each microtick comprising 1 cycle and lengthens and shortens the number of current cycles by several μTs by shifting the start position of the next cycle each node.

- **Sync frame transmission**

Sync frames can be transmitted only from buffer 0 and buffer 1. Message buffer 1 is used to transmit a sync frame when the sync frame has different payloads on 2 channels. In this case, you must set "1" to the **MRC.SPLM** bit.

The message buffer to be used for sync frame transmission must be configured based on the key slot ID. It is possible to make settings only in the DEFAULT_CONFIG or CONFIG state.

For the node which transmits a sync frame, set **SUCC1.TXSY** to "1".

37.4.3.4 External clock synchronization

During normal operation, drift may occur excessively in independent clusters. If synchronization operation is necessary in independent clusters, external clock synchronization is required even if nodes in each cluster are synchronized. This can be achieved by the host reasoning the offset/rate correction time for the cluster.

- The external offset/rate correction value is an integer with a code.
- The external offset/rate correction value is added to the calculated offset/rate correction value.
- The total offset/rate correction time (external + internal) is checked with the configured limit value.

37.4.4 Error handling

Error handling implemented in FlexRay assumes that non-affected inter-node communication is ensured while low-order layer protocol error occurs in the node. In some cases, the operation to restart the FlexRay controller normal operation is required to be implemented in the application program. Transition of the error handling state sets **EIR.PEMC** to "1". An interrupt occurs if it is enabled. The actual error mode is displayed by **CCEV.ERRM [1:0]**.

Table 4-3 POC Error Mode

Error Mode	Activity
ACTIVE (green)	**Full operation** State: NORMAL_ACTIVE FlexRay controller is fully synchronized to support clock synchronization in the entire cluster. It is possible to obtain error state and status change information by reading the error interrupt flag and state interrupt flag from the **EIR** register and the **SIR** register. An interrupt occurs if it is enabled.
PASSIVE (yellow)	**Restricted operation** State: NORMAL_PASSIVE, FlexRay controller self-restoration enabled FlexRay controller stops transmitting frames and symbols. However, it is possible to process received frames. Clock synchronization is continuously performed based on the reception frame. No active clock synchronization in the entire cluster will be performed. It is possible to obtain error state and status change information by reading the error interrupt flag and state interrupt flag from the **EIR** register and the **SIR** register. An interrupt occurs if it is enabled.
COMM_HALT (red)	**Operation stop** State: HALT, FlexRay controller self-restoration disabled FlexRay controller stops frame and symbol processing, clock synchronization, and macrotick generation. It is possible to obtain error state and status change information by reading the error interrupt flag and state interrupt flag from the **EIR** register and the **SIR** register. The bus driver will be deactivated.

37.4.4.1 Clock correction failure counter

When the clock correction failure counter reaches the maximum time for PASSIVE state transition clock correction loss **SUCC3.WCP [3:0]**, a transition from the NORMAL_ACTIVE state to the NORMAL_PASSIVE state occurs. Also, when this counter reaches the maximum time for HALT state transition clock correction loss **SUCC3.WCF [3:0]**, a transition from the NORMAL_ACTIVE or NORMAL_PASSIVE state to the HALT state occurs.

The clock correction failure counter **CCEV.CCFC [3:0]** can be used for monitoring the period that cannot calculate the clock correction time of the node after the startup phase has passed. If either the offset correction signal loss bit **SFS.MOCS** or the rate correction signal loss bit **SFS.MRCS** is set to "1", the clock correction failure counter is incremented at the end of odd-numbered communication cycle.

If either the offset correction signal loss bit **SFS.MOCS** or the rate correction signal loss bit **SFS.MRCS** is not set to "1", the clock correction failure counter is set to "0" at the end of odd-numbered communication cycle.

The clock correction failure counter stops incrementing when it reaches the maximum time for HALT transition clock correction loss **SUCC3.WCF [3:0]** (that is, the maximum value of the counter is not returned to "0" when incremented). The clock correction failure counter is set to "0" when a transition from CONFIG state to READY state has occurred or when a transition to the NORMAL_ACTIVE state has occurred.

(Note) A transition to the HALT state does not occur when SUCC1.HCSE is not set.

37.4.4.2 Passive-to-active state transition required cycle pair counters

Passive-to-active state transition required cycle pair counters **SUCC1.PTAC [4:0]**
controls a POC transition from the NORMAL_PASSIVE sate to the NORMAL_ACTIVE state. **SUCC1.PTA [4:0]** defines a number of even- or odd-numbered cycle pairs for the valid clock correction time before transition from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state. If **SUCC1.PTA [4:0]** is set to "0", no transition can occur from the NORMAL_PASSIVE state to the NORMAL_ACTIVE state.

37.4.4.3 HALT command

When the host system detects an error, a transition to the HALT state can occur if **SUCC1.CMD [3:0]** = "0110" (CHI command HALT) is set.

If the command is executed in the NORMAL_ACTIVE state or NORMAL_PASSIVE state, the POC transits to the HALT state at the end of the current cycle. If the command is executed in another state, **SUCC1.CMD [3:0]** is set to "0000"= command_not_accepted and **EIR.CNA** bit is set to "1". An interrupt occurs if it is enabled.

37.4.4.4 FREEZE command

When the host system detects a serious error, a transition to the HALT state can occur if **SUCC1.CMD [3:0]** = "0111" (CHI command FREEZE) is set. This command causes a transition to the HALT state regardless of the current POC state.

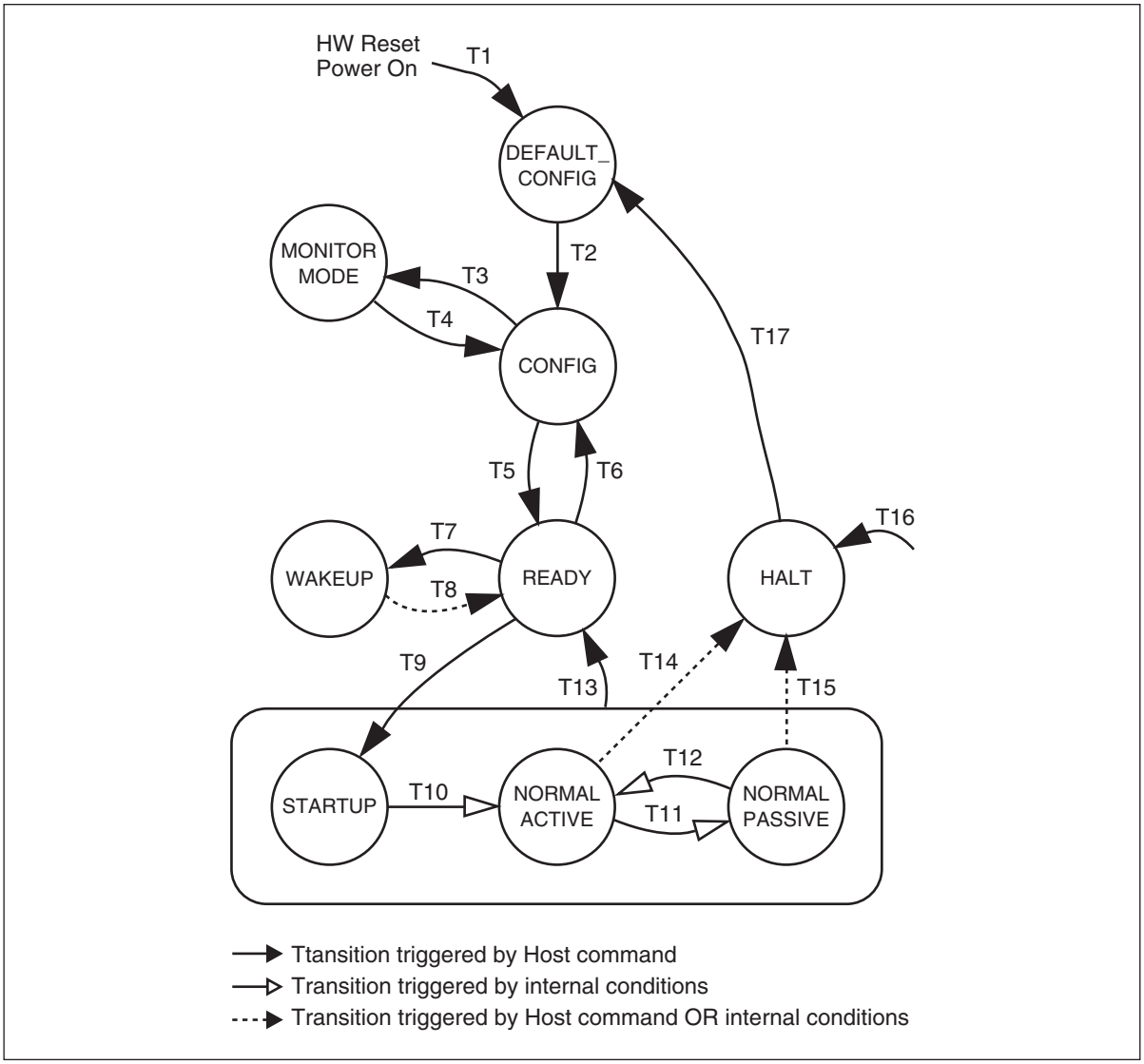
The transition to the HALT state can be read from **CCSV.PSL [5:0]** .

(Note) A transition to the HALT state does not occur when SUCC1.HCSE is not set.

37.4.5 Communication Controller State

37.4.5.1 A diagram of communication controller state

Figure 37.4-4 State Diagram of Communication Controller (CC)



The status transition is controlled by software reset by the host, the external pins RXDA/RXDB, POC state machine, and CHI command vector **SUCC1.CMD [3:0]**.

If **SUCC1.CMD [3:0]** = "0111"(CHI command FREEZE) is set, any state transits to the HALT state.

Table 37.4-1 State Transition Table of FlexRay Controller

Tn	State	From	To
1	• Hardware reset	All States	DEFAULT_CONFIG
2	• Configuration command Sets **SUCC1.CMD [3:0]** = "0001"(CHI command CONFIG)	DEFAULT_CONFIG	CONFIG
3	• Unlock sequence (by the monitor mode command) Sets **SUCC1.CMD [3:0]** = "1011"(CHI command MONITOR_MODE)	CONFIG	MONITOR_MODE
4	• Configuration command Sets **SUCC1.CMD [3:0]** = "0001"(CHI command CONFIG)	MONITOR_MODE	CONFIG
5	• Unlock sequence (by the ready command) Sets **SUCC1.CMD [3:0]** = "0010"(CHI command READY)	CONFIG	READY
6	• Configuration command Sets **SUCC1.CMD [3:0]** = "0001"(CHI command CONFIG)	READY	CONFIG
7	• Wakeup command Sets **SUCC1.CMD [3:0]** = "0011"(CHI command WAKEUP)	READY	WAKEUP
8	• Transmission of normal wakeup pattern • WUP reception • Frame header reception • Occurrence of wakeup collision • Ready command Sets **SUCC1.CMD [3:0]** = "0010"(CHI command READY)	WAKEUP	READY
9	• Run command Sets **SUCC1.CMD [3:0]** = "0100"(CHI command RUN)	READY	STARTUP
10	• Successful startup	STARTUP	NORMAL_ACTIVE
11	• The clock correction failure counter has reached the **SUCC3.WCP [3:0]** set value.	NORMAL_ACTIVE	NORMAL_PASSIVE
12	• The clock correction time valid cycle pair count has reached the **UCC1.PTA [4:0]** set value.	NORMAL_PASSIVE	NORMAL_ACTIVE
13	• Ready command Sets **SUCC1.CMD [3:0]** = "0010"(CHI command READY)	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
14	• The clock correction failure counter has reached the **SUCC3.WCF [3:0]** set value and **SUCC1.HCSE** bit is set to "1". • Halt command Sets **SUCC1.CMD [3:0]** = "0110"(Command HALT).	NORMAL_ACTIVE	HALT
15	• The clock correction failure counter has reached the **SUCC3.WCF [3:0]** set vale and the **SUCC1.HCSE** bit is set to "1". • Halt command Sets **SUCC1.CMD [3:0]** = "0110"(Command HALT).	NORMAL_PASSIVE	HALT
16	• Freeze command Sets **SUCC1.CMD [3:0]** = "0111"(CHI command FREEZE)	All States	HALT
17	• Configuration command Sets **SUCC1.CMD [3:0]** = "0001"(CHI command CONFIG)	HALT	DEFAULT_CONFIG

37.4.5.2 DEFAULT_CONFIG state

In the DEFAULT_CONFIG state, the FlexRay controller stops from operating. All configuration registers are access-enabled, and pins RXDA/RXDB/TXDA/TXDB/TXEN are inactive.

A transition to this state occurs if:

- A hardware reset is executed.
- A transition from the HALT state occurs.

To make a transition from the DEFAULT_CONFIG state to the CONFIG state, write **SUCC1.CMD**="0001".

37.4.5.3 CONFIG State

In the CONFIG state, the FlexRay controller stops from operating. All configuration registers are access-enabled, and pins RXDA/RXDB/TXDA/TXDB/TXEN are inactive. This state is used to initialize the FlexRay controller settings.

A transition to this state occurs if:

- A transition from the DEFAULT_CONFIG state has occurred.
- A transition from the MONITOR_MODE state or READY state has occurred.

If a transition to this state via the HALT state or DEFAULT_CONFIG state has occurred, the status information and settings can be analyzed. Before making a transition from the CONFIG state, check to see that the required settings have been made.

To make a transition from the CONFIG state, you must execute the unlock sequence described in Section "37.3.2.1 Lock Registers (LCK: Lock Register)". After unlocking the CONFIG state, you must set **SUCC1.CMD** to make a transition to the next state.

(Note) A transition from the CONFIG state to the READY state does not affect on both the message buffer status registers (**MHDS**, **TXRQ1/2/3/4**, **NDAT1/2/3/4**, **MBSC1/2/3/4**) and the status data stored in the message RAM.

37.4.5.4 MONITOR_MODE

After CONFIG state has been unlocked and **SUCC1.CMD**= "1011" have been written in, a transition to the MONITOR_MODE occurs. In this mode, the FlexRay frames and WAKEUP patterns can be received. The integrity of the received frame time is not checked. The matching of temporarily received frames is not checked. Therefore, the cycle counter filtering is not supported. This mode can be used, for example, for debugging if the startup of FlexRay network has failed. After writing of **SUCC1.CMD** = "0001", a transition to the CONFIG state occurs.

In the MONITOR_MODE, the first operation is made invalid. This means that a reception message buffer may be created only to receive data on a single channel. The received frame is stored in the message buffer indicated by the frame ID, and receives the channel. An invalid frame is treated as a data frame. After the frame reception, state bits MBS.VFRA, MBS.VFRB, MBS.MLST, MBS.RCIS, MBS.SFIS, MBS.SYNS, MBS.NFIS, MBS.PPIS and MBS.RESS are made as valid values. In the MONITOR_MODE, the reception FIFO is not available.

In the MONITOR_MODE, you cannot tell CAS from MTS symbol. If one of these symbols is detected on the channel, either SIR.MTSA or SIR.MTSB is set. SIR.CAS does not function in the MONITOR_MODE.

37.4.5.5 READY state

After CONFIG state has been unlocked and **SUCC1.CMD**= "1010" have been written in, a transition to the READY state occurs. A transition from READY state to WAKEUP state and a cluster wakeup are allowed; a transition from READY to STARTUP state and a cold start are allowed; and a transition from READY state to the integration in the currently executing cluster is also allowed.

If **SUCC1.CMD** = "0010" (CHI command READY) is written, a transition from the following state to the READY state occurs.

- CONFIG State
- WAKEUP state
- STARTUP state
- NORMAL_ACTIVE state
- NORMAL_PASSIVE state

If the following is written, a transition from the READY state to the respective state occurs.

- If **SUCC1.CMD** = "0001" (CHI command CONFIG) is set: ⇒ CONFIG state
- If **SUCC1.CMD** = "0011" (CHI command WAKEUP) is set: ⇒ WAKEUP state
- If **SUCC1.CMD** = "0100" (CHI command RUN) is set: ⇒ STARTUP state

(Note) A change of POC from READY state to STARTUP state does not affect the status bits (**MHDS [14:0]**), registers (**TXRQ1/2/3/4**), and status data of message RAM.

37.4.5.6 WAKEUP State

The following explains the wakeup setup for the FlexRay controller.

A transition from the READY state to the WAKEUP state occurs.

- When **SUCC1.CMD [3:0]** = "0011"(CHI command WAKEUP) is written.

A transition from the WAKEUP state to the READY state occurs.

- After the normal wakeup pattern has been sent.
- After the WUP has been received.
- After a WUP collision has been detected.
- After a frame header has been received.
- When **SUCC1.CMD [3:0]** = "0010"(CHI command READY) is written.

The wakeup of cluster must be made before startup of communication. When receiving a wakeup pattern of the channel, the bus driver wakes up other components than the node. In the cluster, at least one node generates a wakeup pattern.

The host system controls the entire wakeup procedure. First, the bus driver and FlexRay controller refer to the cluster state, set the FlexRay controller (and the bus guardian, if available), and wake up the cluster. This FlexRay controller setup allows you to send the special wakeup pattern separately to each available channel. The FlexRay controller must recognize the wakeup pattern only when it is in the wakeup state.

A wakeup can be used only for a single channel at a time. In the CONFIG state, the **SUCC1.WUCS** must be written to set the wakeup channel. It is guaranteed that the active communication is not affected on this channel. However, the normal operation from the startup phase cannot be checked for all nodes. Therefore, all nodes connected to the wakeup channel cannot guarantee if they have been woken up by wakeup pattern transmission. Also, if the dual-channel system is used, a wakeup pattern can be sent on a single channel only. If the cold start node requires the system startup, it wakes up the remaining channel before the communication startup.

In this wakeup procedure, if nodes connected to a single channel send wakeup patterns simultaneously, a node is sent. As the wakeup pattern is recovered quickly after the signal collision, another node can be woken up even if two nodes send wakeup patterns simultaneously and cause the collision.

After the wakeup, the FlexRay controller transits to the READY state and reports the wakeup status change by setting the **SIR.WST** flag to "1". The wakeup status vector can be read from **CCSV.WSV [2:0]**. When a valid wakeup pattern is received, either the **SIR.WUPA** or **SIR.WUPB** flag is set to "1".

Figure 37.4-5 POC Configuration in WAKEUP State

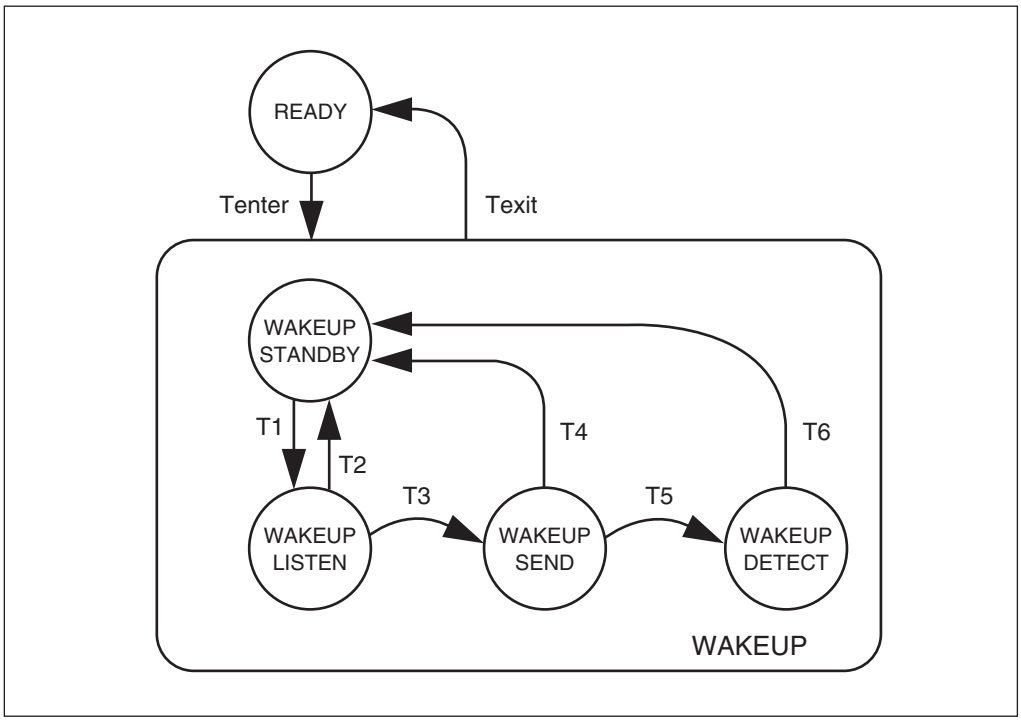


Table 37.4-2 WAKEUP State Transition

Tn	Condition	From	To
Enter	• Wakeup command Sets **SUCC1.CMD [3:0]** = "0011"(CHI command WAKEUP)	READY	WAKEUP
1	• Wakeup command Sets **SUCC1.CMD [3:0]** = "0011"(CHI command WAKEUP)	WAKEUP_STANDBY	WAKEUP_LISTEN
2	• A WUP was received on the wakeup channel selected by the **SUCC1.WUCS**. • A frame header was received on either of valid channels.	WAKEUP_LISTEN	WAKEUP_STANDBY
3	• A timer event occurred	WAKEUP_LISTEN	WAKEUP_SEND
4	• The normal wakeup pattern has been sent.	WAKEUP_SEND	WAKEUP_STANDBY
5	• A collision was detected.	WAKEUP_SEND	WAKEUP_DETECT
6	• The wakeup timer timed out. • A WUP was received on the wakeup channel selected by the **SUCC1.WUCS**. • A frame header was received on either of valid channels.	WAKEUP_DETECT	WAKEUP_STANDBY
Exit	• The wakeup has completed (after T2,T4 and T6). • Ready command Sets **SUCC1.CMD [3:0]** = "0010"(CHI command READY) (This CHI command resets to the WAKEUP_STANDBY state at the same time.)	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. These two timers are controlled by the two parameters: the listen timeout value **SUCC2.LT [20:0]** and the listen timeout noise value **SUCC2.LTN [3:0]** . The listen timeout is effective for the fast cluster wakeup in an environment without having noise, while the listen timeout noise is effective for the wakeup in an environment having a large noise interference.

In the WAKEUP_SEND state, a wakeup pattern is sent onto the preset channel to check the collision. After transition from the WAKEUP state to the READY state, a transition to the STARTUP state must be made by setting the **CMD [3:0]** ="0100"(CHI command RUN).

In the WAKEUP_DETECT state, the cause of wakeup collision detected in the WAKEUP_SEND state can be identified. The identification is canceled when the listen timeout set by **SUCC2.LT [20:0]** has expired. When a wakeup pattern is detected by another node or when a frame header is received, the direct transition to the READY state occurs. If no detection or reception is made, a transition from the WAKEUP_DETECT state occurs when the listen timeout has expired. In such case, the cause of wakeup collision is unknown.

The host must be conscious of a possible trouble during wakeup and take an action against it. If a node causes a wakeup during its startup, we recommend that the start must be delayed for the minimum time until another cold start node wakes up and is initialized.

"FlexRay Protocol Specification V2.1" recommends to wake up two channels using two different FlexRay controllers.

• Host system operations

The host system must adjust the wakeup of two channels and determine a specific channel to wake up. The host system starts to send a wakeup pattern. The destination bus driver detects the wakeup pattern, and posts it to the local host.

The following wakeup procedure is controlled by the host system (in the single-channel, wakeup procedure).

- Set the FlexRay controller in the CONFIG state.
  - Select a wakeup channel by setting the **SUCC1.WUCS** bits.
- Check the bus driver to determine if a WUP has been received or not.
- Start the bus driver of the selected wakeup channel.
- To make a transition to the READY state, write **SUCC1.CMD [3:0]** = "0010".
- Start waking up the specified channel by writing **SUCC1.CMD [3:0]** ="0011".
  - The FlexRay controller transits to the WAKEUP state.
  - After the wakeup, the FlexRay controller transits to the READY state and displays the wakeup status (**CCSV.POCS [5:0]**).
- Wait for the predefined time so that the other nodes can wake up and be ready for setting.
- The following procedure is used for the cold start node.
  - Wait until another channel enters the WUP state in the two-channel cluster configuration.
  - Reset the cold start disable flag **CCSV.CSI** by writing **SUCC1.CMD [3:0]** = "1001" (CHI command ALLOW_COLDSTART).

- Transit to the STARTUP state by writing **SUCC1.CMD [3:0]** = "0100" (CHI command RUN).

The following wake up procedure is initiated by the bus driver.

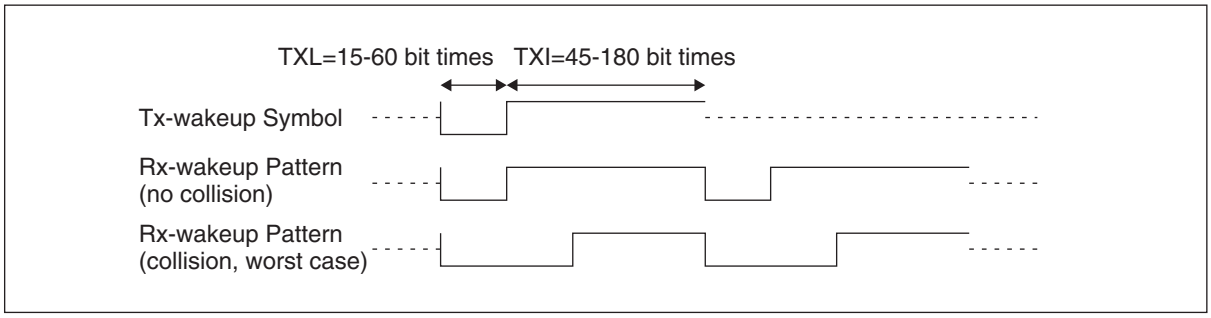
- The wake up is identified by the bus driver.
- The bus driver notifies the wave up event to the host.
- The host sets the FlexRay controller.
- If required, the host performs the following:
  - A wake up command to the second channel
  - A preset wait that allows other nodes to wake up and set up themselves
- Transit to the STARTUP state by writing **SUCC1.CMD [3:0]** = "0100" (CHI command RUN).

- **Wake up pattern (WUP: Wakeup pattern)**

A wake up pattern (WUP) consists of at least two wake up symbols (WUS). A wake up symbol and a wake up pattern are set by the **PRTC1** and **PRTC2** registers.

- A single-channel wake up and wake up symbol cannot be transmitted to both channels at the same time.
- In an environment where there are nodes that transmit at least two wake up patterns, return from a collision of signals is quick for wake up symbols.
  (two overlapping wake up symbols can be identified at any time).
- The same wake up symbols must be used on all the nodes of the cluster.
- The Low time of a wake up symbol is set by **PRTC2.TXL [5:0]**.
- The wake up symbol idle time used to listen to activities on the bus is set by **PRTC2.TXI [7:0]**.
- A wake up pattern consists of at least two transmission wake up symbols required for the wake up.
- The repetition count (2 to 63 repetitions) can be set by **PRTC1.RWP [5:0]**.
- The wake up symbol reception window length is set by **PRTC1.RXW [8:0]**.
- The Low time of wake up reception is set by **PRTC2.RXL [5:0]**.
- The wake up reception idle phase time is set by **PRTC2.RXI [5:0]**.

Figure 37.4-6 Timing of Wake Up Patterns



37.4.5.7 STARTUP state

On a node to be cold started, you should check that both the channels initially connected have been woken up using the STARTUP state.

The time required for all the nodes and stars to complete wake up and setting cannot be assumed. Since at least two nodes are required for the startup of cluster communications, it is recommended to postpone the startup of the node on which a wake up is to be generated for the duration of the minimum time required for another cold started node to complete the wake up, initial setting, and startup.

The delay time required for all the nodes and stars to complete wake up and setting is about several hundreds of milliseconds as a guideline (which, however, depends on the hardware to be used).

Startup is performed simultaneously on all channels. During startup, a node transmits only startup frames.

A fault-tolerant procedure for distributed startup is provided in advance for the initial synchronization of all the nodes. Generally, a node transits to the NORMAL_ACTIVE state in the following procedure (See Figure 37.4-7).

- Coldstart procedure for starting scheduled synchronization (leading coldstart node)
- Coldstart procedure for participating in other coldstart node (following coldstart node)
- Integration procedure for integrating into the existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only the coldstart node that has transmitted a CAS can transmit frames in the first four cycles after the CAS. Then, it is joined by other coldstart nodes and next by all other nodes.

A coldstart node transmits a synchronization frame to the key slot when **SUCC1.TXST** and **SUCC1.TXSY** are set to "1". The message buffer 0 has the key slot ID that defines the slot number to which startup frames are transmitted. In the frame header of a startup frame, the startup frame indicator is set to "1".

After a startup frame is transmitted, the transmission request flag **TXRQ1.TXR0** in the message buffer 0 is reset to "0". To transmit a data frame from the message buffer 0, you must set **TXRQ1.TXR0** to "1" via the **IBCR** register after transition to the NORMAL_ACTIVE state. If this setting is not made, a null frame is sent into the slot corresponding to the frame ID stored in the message buffer 0.

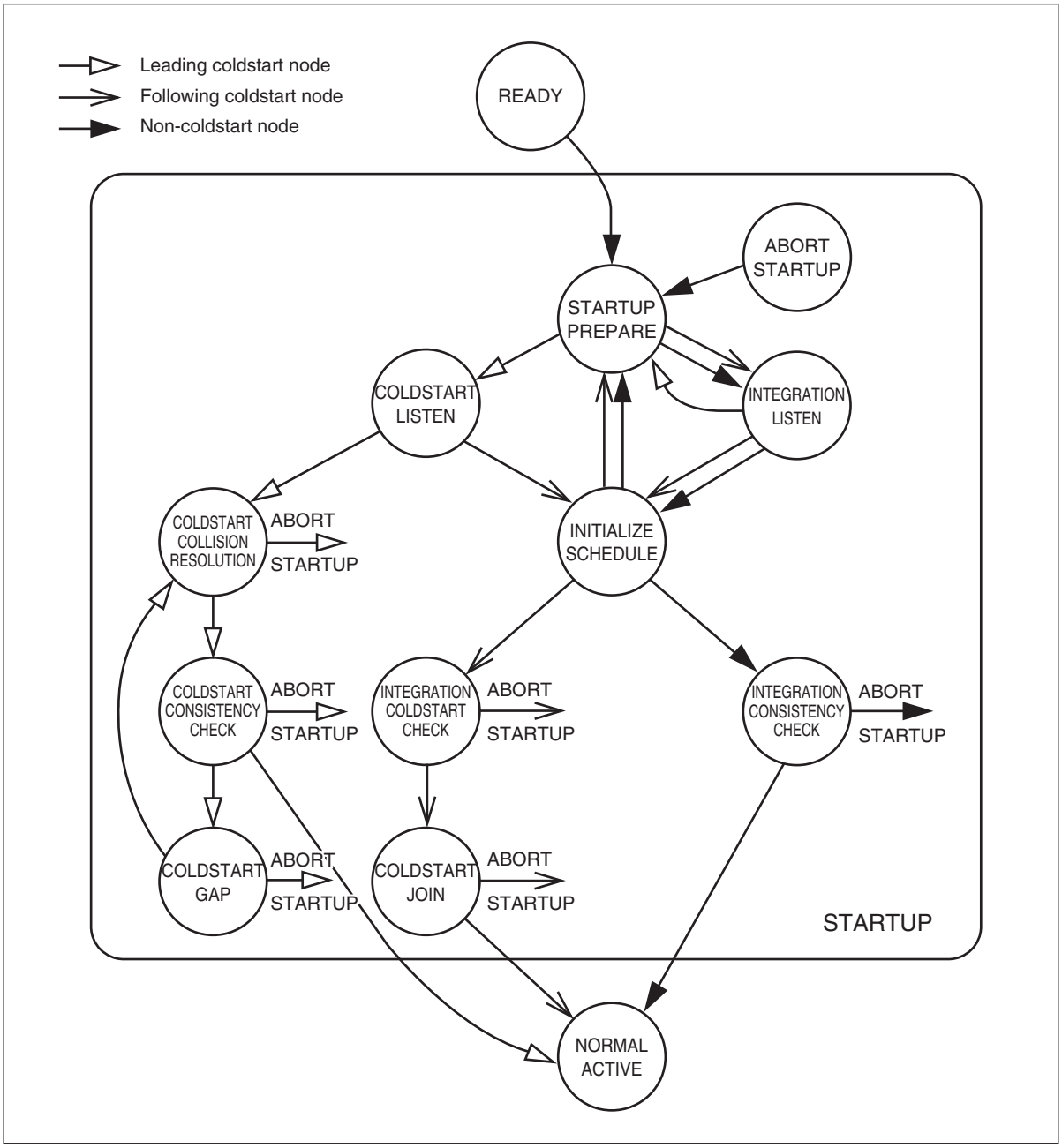
In a cluster consisting of 3 or more nodes, at least 3 nodes must be set as coldstart nodes. In a cluster consisting of 2 nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to start up.

Each startup frame must also be a synchronization frame. Therefore, all coldstart nodes are also synchronous nodes. The number of coldstart attempts is set by **SUCC1.CSA [4:0]**.

Non-coldstart nodes require at least two startup frames from other nodes for integration into a cluster. Non-coldstart nodes may start integration before coldstart nodes complete startup. However, non-coldstart nodes do not complete startup until at least two other coldstart nodes complete startup.

After both the non-coldstart and coldstart nodes acquire the Time Division Multiple Access (TDMA) schedule information and then a synchronization frame is received, passive integration is immediately started under the integration procedure. During the integration, the nodes conform their clock to the global clock (rate and offset) and their cycle time to the global cycle of the network. Afterwards, these settings are checked by all the available network nodes for consistency. The nodes can participate actively in the communications only after passing these checks.

Figure 37.4-7 State Diagram of Time Trigger Startup



- **Coldstart disable mode**

In coldstart disable mode (CCSV:CSI= "1"), nodes cannot initialize cluster communications. This means that startup cannot be initiated under the coldstart procedure. Such nodes can either be integrated to a running cluster or transmit a startup frame after other coldstart node starts initializing cluster communications.

The coldstart disable bit CCSV:CSI can be set while POC is in the READY state. Clear this bit by setting **SUCC1:CMD [3:0]** = "1001" (CHI command ALLOW_COLDSTART).

- **Startup timeout**

The FlexRay controller provides two different μT timers that support two timeout values (startup timeout and startup noise timeout). These two timers start upon transition to the COLDSTART_LISTEN state. If one of the two timers completes counting, the node terminates the other-node detection phase (transits from the COLDSTART_LISTEN state to other state) to start communications.

(Note) Since the startup timer and the startup noise timer are identical to the wake up timer and the wake up noise timer respectively, **SUCC2:LT[20:0]** and **SUCC2:LTN [3:0]** are used.

Startup timeout

The startup timeout sets a limit on the listen time to be used by the node to determine whether communications are already established between other nodes or at least one coldstart node is requesting the integration of other nodes. The setting for the startup timer is made in **SUCC2.LT [20.0]** (pdListenTimeout).

The setting for the startup timer is made in pdListenTimeout = **SUCC2.LT [20.0]** .

The startup timer is reactivated in the following events:

- Transition to the COLDSTART_LISTEN state
- Reaching of the idle state of both channels in the COLDSTART_LISTEN state

The startup timer is stopped in the following events:

- Detection of communications on one of the specified channels while in the COLDSTART_LISTEN state
- Transition from the COLDSTART_LISTEN state to other state

Once the startup timer exceeds the limit time, no timer overflow or periodical restart does not occur. The timer state is retained for later processing.

Startup noise timeout

The startup timer and the startup noise timer start upon transition from the STARTUP_PREPARE state to the COLDSTART_LISTEN state. A startup noise timeout is used to improve the reliability of the startup procedure under an environment where noise is generated. Startup noise timeout is determined by the settings of **SUCC2.LTN [3:0]** .

The setting for the startup noise timer is made in

pdListenTimeout • gListenNoize = **SUCC2.LT [20.0]** • **(SUCC2.LTN [3:0] +1)**.

The startup noise timeout can be calculated as follows:

SUCC2.LT [20:0] + (**SUCC2.LTN [3:0]** × **SUCC2.LT [20:0]**)

The startup noise timer is reactivated in the following events:

- Transition to the COLDSTART_LISTEN state
- Reception of a normally decoded header or CAS symbol in the COLDSTART_LISTEN state

The startup noise timer stops upon transition from the COLDSTART_LISTEN state to another state.

Once the startup timer exceeds the limit time, timer overflow or periodical restart does not occur. The timer state is retained for later processing. The startup noise timer is not reactivated if communications are detected in any channel. This timeout is defined as a final solution for nodes to ensure starting communication cluster even in noisy environment.

- **Activation process of a leading coldstart node (initiating a coldstart)**

A coldstart node in the COLDSTART_LISTEN state monitors the state of a channel to which it is connected.

If no communication is detected, the node transits to the COLDSTART_COLLISION_RESOLUTION state to initiate a coldstart. The first CAS symbol is transmitted in the first normal cycle. This cycle is called cycle 0.

The node starts to transmit its startup frame in cycle 0. Until the coldstart nodes are able to initiate a coldstart, several nodes may transmit a CAS symbol at the same time to initiate a startup under the coldstart procedure. This state is resolved in the first four cycles after CAS transmission.

If a node initiating a coldstart receives a CAS symbol or frame header in these four cycles, it transits to the COLDSTART_LISTEN state again. As a result, only one node in the cluster continues this coldstart procedure. In cycle 4, other coldstart nodes start transmitting their startup frames.

A node that initiates a coldstart after four cycles in the COLDSTART_COLLISION_RESOLUTION state transits to the COLDSTART_CONSISTENCY_CHECK state. This node collects all the startup frames in cycles 4 and 5 to conduct clock compensation. If clock compensation is completed without errors and at least one pair of valid startup frames is received, the node transits from the COLDSTART_CONSISTENCY_CHECK state to the NORMAL_ACTIVE state.

The number of coldstart attempts is set by **SUCC1:CSA [4.0]** . The number of remaining coldstart attempts can be read from **CCSV.RCA [4.0]** . The number of remaining attempts is decremented every time a coldstart attempt is made. If the number of remaining attempts is greater than 1, the node can transit to the COLDSTART_LISTEN state and, if it is greater than "0", the node can transit to COLDSTART_COLLISION_RESOLUTION state. If the number of coldstart attempts is "1", the node can integrate into a cluster but a coldstart is disabled.

- **Activation process of a following coldstart node (reacting to a leading coldstart node)**

A coldstart node, when transiting to the COLDSTART_LISTEN state, attempts to receive a valid pair of startup frames to obtain a cycle schedule and clock compensation from the leading coldstart node.

When the first valid startup frame is received, the node immediately transits to the INITIALIZE_SCHEDULE state. When the second valid startup frame is received and a cycle schedule is obtained, the node transits to the INTEGRATION_COLDSTART_CHECK state.

In the INTEGRATION_COLDSTART_CHECK state, accurate execution of clock compensation is guaranteed, and availability of a leading coldstart node (to which a following coldstart node initializes the schedule) is guaranteed. The following coldstart node collects all the synchronization frames and executes clock compensation in a succeeding cycle pair. If clock compensation does not return an error, and this node keeps receiving sufficient frames from the same node, it transits to the COLDSTART_JOIN state.

In the COLDSTART_JOIN state, the following coldstart node starts transmitting its own startup frames and continues transmitting such frames in the next cycle. They can be used by the leading coldstart node and the nodes that participate in it to check whether their cycle schedules are in synchronization with each other. If an error is detected in clock compensation, the participating nodes stop cluster integration. If a node in this state receives at least one valid startup frame in an even-number cycle, and receives at least one pair of valid startup frames in all the cycle pairs, the node transits from the COLDSTART_JOIN state to the NORMAL_ACTIVE state. Therefore, a following coldstart node transits from the STARTUP state to the NORMAL_ACTIVE state at least one cycle behind the leading coldstart node.

- **Activation process of a non-coldstart node**

A non-coldstart node in the INTEGRATION_LISTEN state monitors the state of a channel to which it is connected.

When the first valid startup frame is received, the node immediately transits to the INITIALIZE_SCHEDULE state. When the second valid startup frame is received and a cycle schedule is obtained, the node transits to the INTEGRATION_CONSISTENCY_CHECK state.

In the INTEGRATION_CONSISTENCY_CHECK state, the non-coldstart node verifies whether clock compensation is normally working and whether a sufficient number of coldstart nodes (at least two) are transmitting startup frames that are in synchronization with the cycle schedule. If an error is detected while clock compensation is working, the integration is discontinued.

In the first even-number cycle in this state, the node should receive either two valid startup frames or a valid startup frame from another node to which this non-coldstart node has integrated. Otherwise, the node discontinues integration.

In the first cycle pair in this state, the node should receive either two pairs of valid startup frames or a valid startup frame from another node to which this non-coldstart node has integrated. Otherwise, the node discontinues integration.

If, after the first cycle pair, the node does not receive two or more valid startup frames within the even-number cycle or two or more pairs of valid startup frames within one cycle pair, startup is discontinued.

For a node in this state to transit from the STARTUP state to the NORMAL_ACTIVE state, it should receive two pairs of valid startup frames in two cycle pairs respectively. As a result, the node transits from the STARTUP state to the NORMAL_ACTIVE state at the completion of at least one cycle pair after a node that initiated a coldstart and one cycle with an odd-number cycle number.

37.4.5.8 NORMAL_ACTIVE state

When the node that transmits the first CAS symbol and an additional node transit to the NORMAL_ACTIVE state, the startup phase of the entire cluster immediately ends. In the NORMAL_ACTIVE state, the transmission timing can be scheduled for all the transmission messages. As for synchronization frames, this includes all the data frames. The measurement of rates and offsets is started in all the even-number cycles (pairs of even-number and odd-number cycles are required).

The FlexRay controller supports normal communication functions in the NORMAL_ACTIVE state.

- Transmission and reception on the FlexRay bus is conducted according to the settings.
- Clock synchronization is active.

The FlexRay controller transits from the NORMAL_ACTIVE state to either of the following states.

- By **SUCC1:CMD [3:0]** = "0110" (CHI command HALT) being set, transits to the HALT state after completion of the current cycle.
- By **SUCC1:CMD [3:0]** = "0111" (CHI command FREEZE) being set, transits to the HALT state immediately.
- Transits to the HALT state with the error state changing from ACTIVE to COMM_HALT.
- Transits to the NORMAL_PASSIVE state with the error state changing from ACTIVE to PASSIVE.
- By **SUCC1:CMD [3:0]** = "0110" (CHI command READY) being set, transits to the READY state .

37.4.5.9 NORMAL_PASSIVE state

When the error state changes from ACTIVE to PASSIVE, the node transits from the NORMAL_ACTIVE state to the NORMAL_PASSIVE state.

In the NORMAL_PASSIVE state, the node can receive all the frames (The node is completely synchronized and can execute clock synchronization). Unlike in the NORMAL_ACTIVE state, however, the node does not actively participate in communications. This means that it does not transmit any symbol or frame.

In the NORMAL_PASSIVE state, the following operations are performed:

- Frames are received on the FlexRay bus.
- No frame or symbol is transmitted on the FlexRay bus.
- Clock synchronization is active.

The FlexRay controller transits from the NORMAL_PASSIVE state to either of the following states.

- By **SUCC1:CMD [3:0]** = "0110" (CHI command HALT) being set, transits to the HALT state after completion of the current cycle.
- By **SUCC1:CMD [3:0]** = "0111" (CHI command FREEZE) being sat, transits to the HALT state immediately.
- Transits to the HALT state with the error state changing from PASSIVE to COMM_HALT.
- Transits to the NORMAL_ACTIVE state with the error state changing from PASSIVE to ACTIVE. (This change in the error state occurs when **CCEV:PTAC [4:0]** are equal to **SUCC1:PTA [4.0]-1**.)
- By **SUCC1:CMD [3:0]** = "0110" (CHI command READY) being set, transits to the READY state .

37.4.5.10 HALT state

This state stops any communication (transmission or reception).

The FlexRay controller transits to the HALT state as follows.

- When **SUCC1.CMD [3:0]** = "0110" (CHI command HALT) is set, transits from NORMAL_ACTIVE state or NORMAL_PASSIVE state.
- When **SUCC1:CMD [3:0]** = "0111" (CHI command FREEZE) is set, transits from every possible states.
- When the clock compensation fatal counter has reached the maximum time of HALT state transition clock compensation loss, **WCF [3:0],** a transition from the NORMAL_ACTIVE state to the HALT state occurs. Set **SUCC1.HCSE** to "1".
- When the clock compensation fatal counter has reached the maximum time of HALT state transition clock compensation loss, **WCF [3:0],** a transition from the NORMAL_PASSIVE state to the HALT state occurs. Set **SUCC1.HCSE** to "1".

In the following cases, the FlexRay controller transits from the HALT state to the DEFAULT_CONFIG state.

- When **SUCC1:CMD [3:0]** = "0001" (CHI command CONFIG) is set

If **SUCC1:CMD [3:0]** = "0110" (CHI command HALT) is set, the **CCSV:HRQ** bit is set to "1" and a transition to the HALT state occurs at the end of the next cycle.

When **SUCC1:CMD [3:0]** = "0111" (CHI command FREEZE) is set, transits to HALT state and the **CCSV.FSI** bit is set to "1".

A transition to the HALT state can be read from **CCSV:PSL [5.0]** .

37.4.6 Network Management

The generated network management (NM) vector can be read from the registers **NMV1 to NMV3**. The FlexRay controller executes the bit OR operation for all NM vectors of the reception valid NM frames where the payload preamble indicator (PPI) has been set. Only a static frame can be set as an NM frame. The NM vector is updated after the end of each cycle.

The NM vector length can be set to 0 to 12 bytes, using the **NEMC.NML [3:0]** . The same NM vector length must be set at all nodes of the cluster.

To set the transmission buffer for the frames where the PPIT bit is set, the **PPIT** bit of the header section of each transmission buffer should be set via the **WRHS1.PPIT**. Also, the NM information should be set in the data section of each transmission buffer.

The mechanisms to evaluate the NM vector should be implemented with applications.

(Notes) If the message buffer is set to transmit or receive a network management frame, the payload length set in header 2 of each message buffer must be equal to each other or greater than the NM vector length set in **NEMC.NML [3:0]** .
Since passing through the HALT status does not increase the cycle count, it does not update the NM vector. In this case, HMV1 ...3 keep the value of former cycles.

37.4.7 Filtering and Masking

The current slot, cycle counter value, channel ID (channels A and B), and message buffer settings are compared and used for filtering. If certain information matches during their comparison, the message buffers are updated or transmitted.

The filtering is made by the followings:

- Slot counter
- Cycle counter
- Channel ID

The following combinations of filters can be used for filtering during transmission and reception.

- Slot counter + Channel ID
- Slot counter + Channel ID + Cycle counter

All of the preset filters must match the reception message information so that the reception messages are stored in the message buffers.

(Note) The FIFO acceptance filter can be set by the FIFO rejection filter and the FIFO rejection filter mask.

The massage is sent to a time slot having the corresponding frame ID set to the preset channel. If cycle counter filtering is valid, the preset cycle filter value must also match.

37.4.7.1 Slot counter filtering

All transmission/reception buffers have a frame ID in the header section. Because the frame ID is assigned to the slot that corresponds to the transmission/reception buffer, it is compared with the current slot counter value.

If two or more buffers are set for one frame ID and one channel ID simultaneously, and if those buffers have cycle counter filter values that matches the one slot, the message buffer having the smallest buffer number is used.

37.4.7.2 Cycle Counter Filtering

The cycle counter filtering is based on the concept of cycle set. For filtering, a filter matching detection means the filter coincides with one of the cycle set elements. The cycle set is defined by the cycle code field of header section 1 in each message buffer.

If message buffer 0 has been set by the **SUCC1.TXST**, **SUCC.TXSY** and **SUCC1.TSM** bits so that the startup/synchronous frame or the single slot frame is stored, the cycle counter filtering of message buffer 0 must be disabled.

(Note) Sharing of the static time slot is not allowed by cycle counter filtering between different nodes of the FlexRay network.

The cycle count belonging to the cycle set is shown in Table 37.4-3.

Table 37.4-3 Cycle Set Definition

Cycle code	Matching cycle counter values		
0b000000x	all Cycles		
0b000001c	every second Cycle	at (Cycle Count) mod2	= c
0b00001cc	every fourth Cycle	at (Cycle Count) mod4	= cc
0b0001ccc	every eighth Cycle	at (Cycle Count) mod8	= ccc
0b001cccc	every sixteenth Cycle	at (Cycle Count) mod16	= cccc
0b01ccccc	every thirty-second Cycle	at (Cycle Count) mod32	= ccccc
0b1cccccc	every sixty-fourth Cycle	at (Cycle Count)mod64	= cccccc

Table 37.4-4 gives some examples of valid cycle set that can be used for cycle counter filtering.

Table 37.4-4 Examples of Valid Cycle Set

Cycle code	Matching cycle counter values
0b0000011	1-3-5-7-....-63↵
0b0000100	0-4-8-12-....-60↵
0b0001110	6-14-22-30-....-62↵
0b0011000	8-24-40-56↵
0b0100011	3-35↵
0b1001001	9↵

When a message is received, it is stored if the cycle counter value between the message receiving cycles matches the cycle set element of the reception buffer. The other filtering standards must also be satisfied.

The transmission buffer contents are sent to the specified channel when the cycle setting element matches the current cycle counter value. The other filtering standards must also be satisfied.

37.4.7.3 Channel ID filtering

The header section of each message buffer in the message RAM has the 2-bit channel filtering field (CHA, CHB). It functions as a filter of the reception buffer and a control field for the transmission buffer (see Table 37.4-5).

Table 37.4-5 Settings of Channel Filtering

CHA	CHB	Transmit Buffer transmit frame	Receive Buffer store valid receive frame
1	1	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)
1	0	on channel A	received on channel A
0	1	on channel B	received on channel B
0	0	no transmission	ignore frame

If the slot counter filtering standard and the cycle counter filtering standard are satisfied, the transmission buffer contents are sent to the channel that has been specified by the channel filtering field. However, the transmission setting to both channels (setting **CHA** and **CHB**) is allowed in the static segment only.

If the slot counter filtering standard and the cycle counter filtering standard are satisfied, the valid reception frames are stored when they are received on the channel that has been specified by the channel filtering field. However, the frame reception setting on both channels (setting **CHA** and **CHB**) is allowed in the static segment only.

(Note) If the message buffer is set in the dynamic segment and if the bits of both channel filtering fields (**CHA**, **CHB**) are set to "1", no frame is sent and the reception frames are ignored (this is the same as for the function if **CHA** = **CHB** = "0" is set).

37.4.7.4 FIFO filtering

A single reject filter and a single reject filter mask are provided for FIFO filtering. The reject filter consists of the channel filter **FRF.CH [1:0],** frame ID filter **FRF. FID [10:0],** and cycle counter filter**FRF.CYF [6:0]** . The FRF register and FRFM register can be set in the DEFAULT_CONFIG state or CONFIG state only. The filter setting in the header section of a message buffer belonging to the FIFO group is ignored.

The 7-bit cycle counter filter **FRF.CYF [6:0]** specifies a cycle set and determines the communication cycle applied to the frame ID and channel filters. If the frame ID filter and channel filter are not applied to a cycle by the cycle set being specified by this register, no frame is received in this cycle.

If the channel ID, frame ID, and cycle counter are not rejected by filtering with the preset rejection filter and rejection filter mask, and if the frame does not match the dedicated reception buffer, the valid reception frame is stored in the FIFO memory.

37.4.8 Transmission Procedure

37.4.8.1 Static segment

If some transmission messages are held for a static segment, messages having the frame ID that corresponds to the next transmission slot are selected as the next transmission message.

The data section of the transmission buffer assigned to the static segment can be updated before the previous time slot ends. This means that the message transfer from the input buffer must be started when a message is written in the input buffer command request register at the end of this time slot.

37.4.8.2 Dynamic segment

If a transmission message is held in a dynamic segment, a message having the highest priority (or the smallest frame ID) is selected as the next transmission message. On the other hand, different slot counter strings may be generated between channel A and channel B in the dynamic segment (if messages having different frame IDs are transmitted on both channels simultaneously).

The data section of the transmission buffer assigned to the dynamic segment can be updated before the previous slot ends. This means that the message transfer from the input buffer must be started when a message is written in the input buffer command request register at the end of this time slot.

The end of transmission minislot value **MHDC.SLT [12:0]** defines the maximum mini-slot value that can be transmitted before a frame transmission is disabled in the dynamic segment of the current cycle.

37.4.8.3 Transmission Buffer

The message buffer can be used as the transmission buffer if the **CFG** bit of the header section of each message buffer is set to "1" thorough setting of **WRHS1**.

The transmission buffer can be assigned to a channel of the FlexRay controller as follows.

- Static segment: Channel A or Channel B, Channel A and Channel B
- Dynamic segment: Channel A or Channel B

Message buffer 0 is set by **SUCC1.TXST**, **SUCC1.TXSY** and **SUCC1.TSM** bits, and it is used as the dedicated buffer to store the startup frame and synchronous frames, or as the buffer dedicated to the specified single slot frames only. In this case, message buffer 0 can be set again in the DEFAULT_CONFIG state or CONFIG state only. This ensures that every node transmits one or more startup frame/synchronous frame for one communication cycles. No startup frame or synchronous frame can be sent from another message buffer.

All message buffers, except for buffer 0 being set for static or dynamic segment transmission, can be set again during its execution, using the **MRC.SEC [1:0]** settings. (See Section "37.4.11.1 Reset of Message Buffer".) However, for the data pointer in the header partition refers the data partition in the message RAM, the user may configure the message buffer incorrectly when changing the setting of payload length and data pointer in the header section of the message buffer.

If the message buffer is set again during its execution (or if the header section is updated), this message buffer may not be sent in each communication cycle.

Since the FlexRay controller does not have the header CRC calculation function, the user must assign the header CRC for all transmission buffers. If the network management is required, the host system must set the **PPIT** bit of the header section of each message buffer to "1" and set the network management information in the data section of the message buffer (See Section "37.4.6 Network Management").

The payload length, in 2 bytes, is stored in the payload length field. If the payload length of the preset static transmission buffer is shorter than the value set with **MHDC.SFDL [6:0],** some padding bytes will be inserted so that the payload length of static frame is guaranteed. The padding bytes are expressed as "0".

(Note) If the payload length is an odd number (PLC = 1, 3, 5...), a zero in 16-bit must be written at the end of massage buffer.

Each transmission buffer can set the transmission mode using the transmission mode flag **TXM**. If this bit is set to "1", the transmission message is sent in the single shot mode. If this bit is set to "0", the transmission message is sent in the continuous mode.

In the single shot mode, the **TXR** flags of each message buffer are cleared to "0" when the transmission ends. Then, the transmission buffer can be overwritten by the next transmission message.

In the continuous mode, the **TXR** flags of each message buffer are not cleared to "0" when the transmission ends. In this case, frames are sent each time the filtering standard matches. The **TXR** flags can be cleared to "0" if the respective message buffer number is written in the **IBCR** register before the **IBCM.STXRH** bit is set to "0".

If two or more transmission buffers satisfy the filtering standard, the one with the smallest buffer number is used to transmit frames of each slot.

37.4.8.4 Frame transmission

To assign the transmission message buffers, use the following procedure.

- Set the transmission buffer in the message RAM using the **WRHS1**, **WRHS2** and **WRHS3** settings.
- Write data in the data section of the transmission buffer via **WRDSn** setting.
- Write the target buffer number in the **IBCR** register to allow the setting from the input buffer to the message RAM and to allow message data transfer.
- If the **IBCM** register is set to transmit messages, the transmission request flag **TXR** of each message buffer is set to "1" immediately after message transfer from the input buffer. The message buffer is waited from transmitting.
- The message buffer can be checked whether its message transmission has completed or not if the **TXR** bit of **TRXQ1/TRXQ2/TRXQ3/TRXQ4** register is checked (**TXR**=0) (in the single shot mode only).

When the message transmission is completed, the **TXR** flags of **TXRQ1/TXRQ2/TXRQ3/TXRQ4** register are cleared to "0" (in the single shot mode). If the **MBI** bit of the header section of this message buffer is set to "1", the **SIR.TXI** bit is set to "1". An interrupt occurs if it is enabled.

37.4.8.5 Null frame transmission

When the transmission request flag is not set to "1" before the transmission time in a static segment and there are no other transmission buffers that match the filtering standard, the FlexRay controller transmits a null frame whose indicator and payload data are cleared to "0".

A null frame is transmitted in the following cases.

- When the transmission request flag is not set (TXR=0) for the message buffer that matches the filtering standard and has the smallest buffer number
- When all the transmission buffers have cycle counter filters that do not match the current cycle. In this case, the message buffer status MBS will not be updated.

A null frame is not transmitted in a dynamic segment.

37.4.9 Reception Procedure

37.4.9.1 Reception buffer

The message buffer can be used as the dedicated reception buffer if the **CFG** bit of the header section of each message buffer is set to "0" thorough setting of **WRHS1**.

The reception buffer can be assigned to a channel of the FlexRay controller as follows.

- Static Segment: Channel A or Channel B
  Channel A and Channel B
- Dynamic Segment: Channel A or Channel B

The FlexRay controller stores all frame elements of the reception buffer (excluding frame CRC) that matches with the filtering standard.

All reception message buffers set for static or dynamic segment can be reset using the **MRC.SEC [1:0]** settings (see Section "37.4.11.1 Reset of Message Buffer"). However, the reception messages of each communication cycle may be lost when the header section of the message buffer is reset during execution.

When two or more buffers match the filtering standard, the reception buffer with the smallest message buffer number is updated by the reception message.

37.4.9.2 Frame reception

To assign the reception message buffers, use the following procedure.

- Set the reception buffer in the message RAM using the **WRHS1**, **WRHS2** and **WRHS3** settings.
- Write the target message buffer number in the **IBCR** register to allow the transfer of settings from the input buffer to the message RAM.

When this step is executed, the message buffer will function as an active reception buffer and perform acceptance filtering every time a message is received. The first reception buffer that matches the filtering standard is updated by the reception message.

If a valid payload segment is stored in the data section of the message buffer, the **ND** flags of the **NDAT1/NDAT2/NDAT3/NDAT4** registers will be set to "1". In addition, if the **MBI** bit of the header section of this message buffer is set to "1", the **SIR.RXI** flag will set to "1". An interrupt occurs if it is enabled.

If the message buffer is updated and the **ND** bit is already set to "1", **MBS.MLST** of the reception message buffer will be set and unprocessed message data will be lost.

If there is no frame in a slot or either null frame or destruction frame is received in the slot, the data section of the message buffer set for this slot will not be updated. In this case, only the status flag of each message buffer will be updated.

When the status flag of the header section of the message buffer is updated, the **MBS** flags of the **MBSC1/2/3/4** registers are set to "1". If the **MBI** bit of the header section of this message buffer is set to "1", the **SIR.MBSI** bit is set to "1". An interrupt occurs if it is enabled.

When the payload length of the reception frame **PLR [6:0]** is longer than the value set to **PLC [6:0]** of the header section of each message buffer, the data field stored in the message buffer will be truncated the extra length to set the specified length.

For data transfer between the output buffer (OBF) and the message RAM, follow the procedure described in "• Data transfer from the message RAM to the output buffer" of Section "37.4.11.2 Host access to message RAM".

(Note) The **ND** and **MBS** flags will be cleared to "0" when the payload data and header of the received message are transferred to the output buffer.

37.4.9.3 Null frame reception

The payload segment of a received null frame is not reflected in the reception buffer. When a null frame is received, the header section of the reception buffer will be updated using the reception null frame. The null frame indicator **NFI** of the header section of the received message buffer is cleared to "0".

When the status flag of the header section of the message buffer is updated, the **MBS** flags of the **MBSC1/2/3/4** registers are set to "1". If the **MBI** bit of the header section of this message buffer is set to "1", the **SIR.MBSI** bit is set to "1". An interrupt occurs if it is enabled.

37.4.10 FIFO Function

37.4.10.1 Details

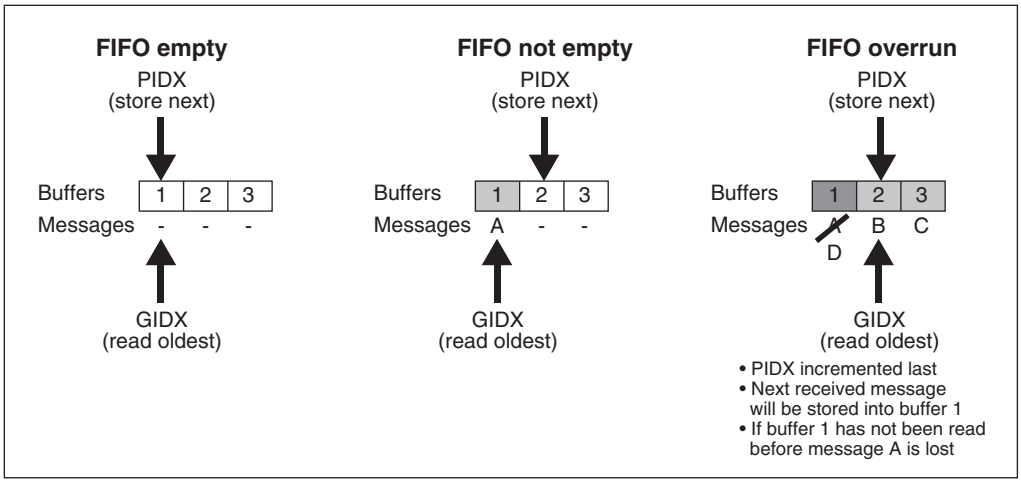
A group of message buffers can be set as an FIFO buffer. The message buffers belonging to the FIFO message group, which are from the one referred by **MRC.FFB [7:0]** to the one referred by **MRC.LCB [7:0],** are adjacent each other in the register map. Up to 127 message buffers can be assigned to the FIFO.

FIFO stores all valid reception messages that do not match with the filtering condition of the dedicated reception buffer and match with the set FIFO filtering condition. At this time, the frame ID, payload length, reception cycle count and status bit of the specified FIFO message buffer are overwritten by the reception frame. The **SIR.RFNE** bit indicates that FIFO is not empty, the **SIR.RFF** bit is set to "1" when the reception FIFO is full, and the **EIR.RFO** bit indicates that FIFO overrun is detected. An interrupt occurs if it is enabled.

The FIFO has two index registers, PUT Index (PIDX) and GET Index (GIDX) registers, connected to it. The PIDX register is the place where the next message is stored in FIFO. When a new message is received, the PIDX register writes the message to the specified message buffer. At this time, the PIDX register value is incremented and the message buffer to store the next message is indicated. When the PIDX register value goes beyond the highest message buffer number in FIFO, the lowest message buffer number is assigned to the PIDX register. The GIDX register is used to specify the next message buffer to be read in FIFO. The GIDX register value is incremented after the content of the message buffer that belongs to the FIFO message group is transferred to the output buffer, The PIDX and GIDX registers cannot be accessed.

When the PIDX register value reaches the GIDX register value, FIFO is full. If the next message is written before the oldest message in FIFO is read, both the PIDX and GIDX registers will be incremented and the oldest message will be overwritten by the new message. This will set the EIR:RFO flag to "1".

Figure 37.4-8 FIFO Status: Empty, Non-empty, Overrun



If the PIDX register and GIDX register values are different, FIFI non-empty status will be detected, and the **SIR.RFNE** flag will be set to "1". This means that there is at least one received message in FIFO. Figure 37.4-8 explains three statuses, FIFO empty, FIFO non-empty and FIFO overrun, when there are three message buffer in FIFO.

The FIFO rejection filter (FRF) defines the filtering pattern to reject messages. The filter consists of the channel filter, frame ID and cycle counter filter. When the **FRF.RSS** bit is set to "1", all messages received by the static segment are rejected by the FIFO filter. When the **FRF.RNF** bit is set to "1", received null frames are not stored in FIFO.

The FIFO rejection filter mask (FRFM) specifies the bit not to be used for reject filtering in the frame ID filtering of the FIFO rejection filter register.

37.4.10.2 FIFO setting

Set the same payload length of **PLC [6:0]** via **WRHS2** for all message buffers belonging to FIFO. In addition, set the data pointer to the first 32-bit word of the data section to each message buffer of the massage RAM thorough setting of **WRHS3**.

Every information requested by the acceptance filters is set by FIFO rejection filters and FIFO rejection filter masks, so the filtering condition need not to be set for the header section of message buffers belonging to FIFO.

(Note) It is recommended for WRHS1.MBI to be set to "0" and for MBI bits to be programmed in message buffers as FIFO to avoid the occurrence of RX interruption.
When the payload length of the reception frame is longer than the value set to **PLC [6:0]** of the header section of each message buffer, the data field stored in the message buffer in FIFO is truncated to the length of **PLC [6:0]**.

37.4.10.3 Access to FIFO

To read from FIFO, transfer from the message RAM to the output buffer by writing the first message buffer number of FIFO (referenced by **MRC.FFB [7:0]**) to the **OBCR** register. Thus, the message buffer specified by the GIDX register is transferred to the output buffer. Then, the GIDX register value is incremented.

37.4.11 Message handling

The message handler controls data transfer between the input/output buffers and the message RAM and between the message RAM and the temporary storage buffer RAM. The internal RAM is accessed in 32 + 1 bits. Additional bits are used for parity check.

The message handler state machine controls access to the message buffer stored in the message RAM. This avoids collision of access between hosts to two FlexRay channel protocol controllers and the message RAM.

The message buffer frame IDs assigned to the static segment should be between 1 and **GTUC7.NSS [9:0]** . The message buffer frame IDs assigned to the dynamic segment should be between **GTUC7.NSS [9:0]** +1 and 2047.

The reception message which does not match with the filtering condition of dedicated reception buffer (static or dynamic segment) is stored in the reception FIFO when it matches with the filtering condition of the FIFO rejection filter.

37.4.11.1 Reset of Message Buffer

When the application requires more than 128 message buffers, static message buffer and dynamic message buffer may be reset during FlexRay controller operation. This reset will be executed if the header sections for each message buffer are updated via the input buffer registers **WRHS1...3**.

The reset shall be valid for the **MRC.SEC [1:0]** control bits in the RAM setting register.

When the message buffer is not updated by the reception frame or the transmission message of the message buffer is not transmitted before the reset starts, the message will be lost.

The timing, which the reset message buffer gets ready for transmission and reception after the frame ID reset, depends on the current slot counter status at completion of the header section update. Thus, depending on the cycle of resetting of message buffers, the reception frame may fail to update a massage buffer, or a message in the message buffer may not be transmitted.

The reset completes by the NIT start even through the message RAM scan does not complete. The message RAM scan for slot 2 to 15 starts from the beginning of the slot 1 of the real cycle. The message RAM scan for slot 1 is performed before the cycle if it is scanned in parallel with each message RAM scan even when there is a message buffer set for slot 1 of the next cycle.

The first dynamic message buffer number is set to MRC.FDB [7:0]. If the message RAM scan starts while CC is in the dynamic segment, the scan starts from the message buffer number set to MRC.FDB [7:0] .

To reset the message buffer to use it for slot 1 of the next cycle, you must follow the procedures below.

- When the message buffer reset to slot 1 is "static buffer", reset this message buffer before the last message RAM scan evaluates it in the static segment of the real cycle.
- When the message buffer reset to slot 1 is "static and dynamic buffer", reset this message buffer before the last message RAM scan evaluates it in the real cycle.
- When NIT starts, the message RAM scan will complete. If the message RAM scan cannot evaluate the reset message buffer, the message buffer will not be taken into account during the next cycle.

(Note) Be careful in resetting the message buffer for certain possibilities of a message loss. The resetting of a message buffer in a continuous cycle, the reception frame may update no massage buffer, or no message in the message buffer may be transmitted.

37.4.11.2 Host access to message RAM

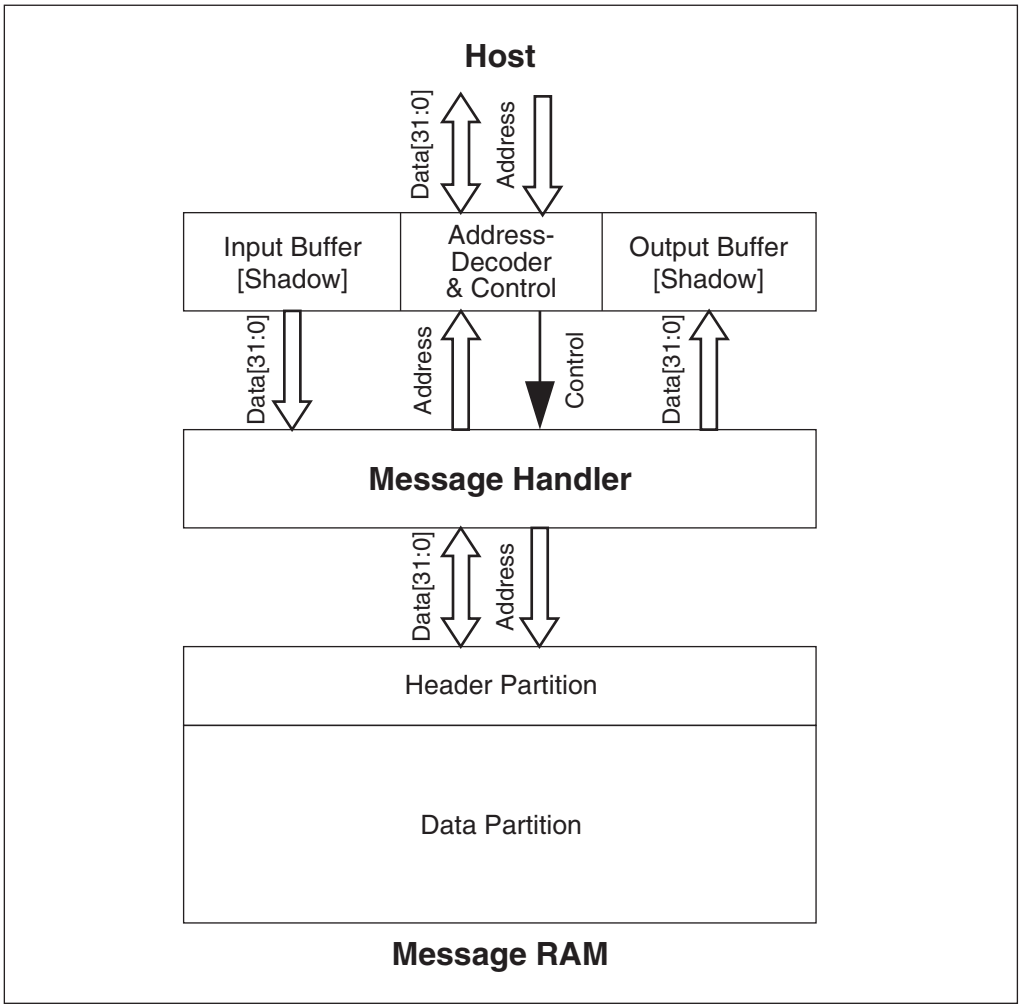
The message transfer between the input buffer and the message RAM and between the message RAM and the output buffer will be started if the message buffer numbers to be transferred are written to the **IBCR** and **OBCR** registers respectively.

The **IBCM** and **OBCM** registers can be used separately for reading/writing the header section and data section of a selected message buffer.

When the **IBCM.STXRH** bit is set to "1", the **IBCM.STXRS** bit will be set to "1" and the transmission request flag **TXR** of the message buffer will be automatically set to "1" after the selected message buffer is updated. When the **IBCM.STXRH** bit is set to "0", the **IBCM.STXRS** bit will be cleared to "0" and the transmission request flag **TXR** of the selected message buffer will be cleared to "0". This clearing operation can be used for stopping the transmission from the message buffer operating in the continue mode.

The input buffer (IBF) and output buffer (OBF) have the double buffering structure. The IBF host/OBF host of the double buffering structure are accessed from the host and the other IBF shadow/OBF shadow are accessed from the message handler for data transfer between IBF/OBF and message RAM.

Figure 37.4-9 Host Access to Message RAM

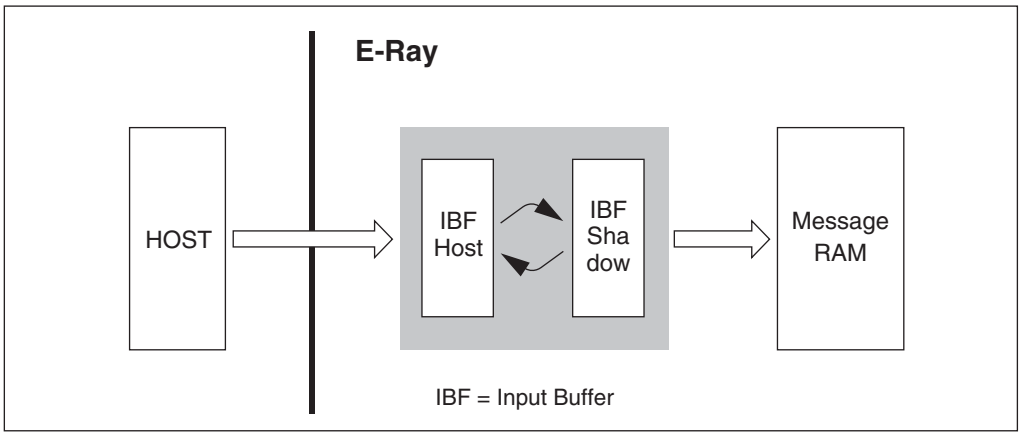


- **Data transfer from the input buffer to the message RAM**

In order to configure and update the message buffer of the message RAM, data must be written to the **WRDSn** and the header to the **WRHS1...3**. Configuring the **IBCM** will select specific operations.

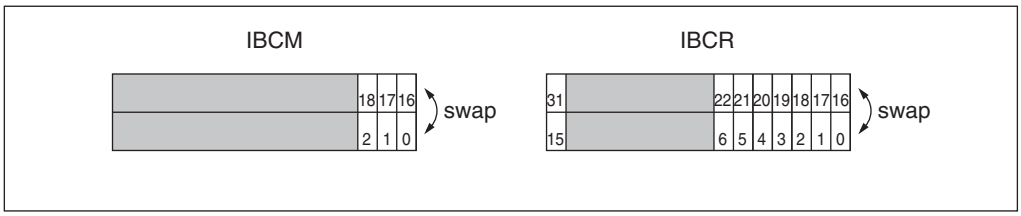
Writing the target message buffer number of the message RAM to the **IBCR.IBRH [6:0]** will switch the IBF host and the IBF shadow (see Figure 37.4-10).

Figure 37.4-10 Double Buffer Structure of the Input Buffer



Moreover, **IBCM** register bits and **IBCR**R register bits will be exchanged to maintain associations with each section of the IBF (see Figure 37.4-11).

Figure 37.4-11 Bit Exchange between the IBCM Register and IBCR Register



This write operation will set the **IBCR.IBSYS** to "1". The message handler starts to send the contents of the IBF shadow to the message buffer in the message RAM selected by **IBCR.IBRS [6:0]** .

It is possible to write the next message to the IBF host while transferring data from the IBF shadow to the target message buffer of the message RAM. After the transfer is completed between the IBF shadow and message RAM, the **IBCR.IBSYS** is cleared to "0". Then, when writing the next target message buffer number to the **IBCR.IBRH [6:0],** the next transfer to the message RAM will be initiated.

When a write operation to **IBCR.IBRH [6:0]** occurs while **IBCR.IBSYS** = "1", **IBCR.IBSYH** will be set to "1". When data transfer is completed from the IBF shadow to the message RAM, **IBCR.IBSYH** is cleared to "0". **IBCR.IBSYS** will retain "1" and the next transfer to the message RAM will be initiated. Furthermore, the message buffer number and command mask flag stored in the **IBCR.IBRH [6:0]** and **IBCR.IBRS [6:0]** will be switched simultaneously.

Example of input buffer configuration procedure:

Configure/update the first message buffer through the IBF.

Write to **WRDSn**, for the data section

Write to **WRHS 1...3**, for the header section

Writing the command mask: Writing to IBCM.LHSH, IBCM.LDSH and IBCM.STXRH

Data transfer request to the target message buffer: Writing to the **IBCR.IBRH [6:0]** .

Configure/update the second message buffer through the IBF.

Write to **WRDSn**, for the data section

Write to **WRHS 1...3**, for the header section

Writing the command mask: Writing to IBCM.LHSH, IBCM.LDSH and IBCM.STXRH

Data transfer request to the target message buffer: Writing to the **IBCR.IBRH [6:0]** after the **IBCR.IBSYH** is cleared to "0".

Configure/update the third message buffer through the IBF.

... (hereinafter, the second message buffer configuration/update procedure will be repeated)

(Note) When the **IBCR.IBSYH** is "1", access to the input buffer sets the error flag **EIR.IIBA** to "1". In this case, access is invalid.

Table 37.4-6 Assignment of the Input Buffer Command Mask Bit

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 37.4-7 Assignment of the Input Buffer Request Mask Bit

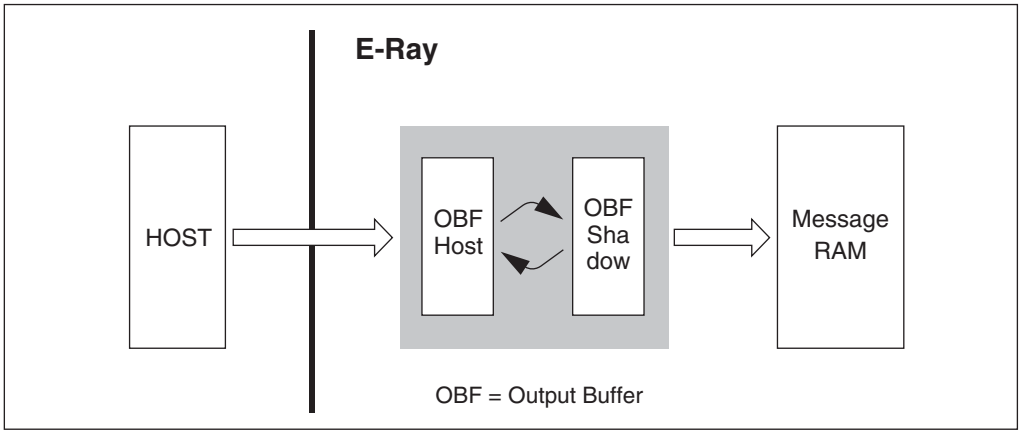
Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22...16	r	IBRS [6:0]	IBF Request Shadow, number of message buffer currently / lately updated
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH [6:0]
6...0	r/w	IBRH [6:0]	IBF Request Host, number of message buffer to be updated next

- **Data transfer from the message RAM to the output buffer**

Any data must be written to the **OBCR** register to read the message buffer from the message RAM because this will trigger data transfer configured in the **OBCM**. After transfer is completed, the data received from the **RDDSn**, **RDHS1...3** and **MBS** can be read.

Set the buffer number of the transfer source message buffer in the message RAM using **OBCR.OBRS [6:0]** .

Figure 37.4-12 Double Buffer Structure of the Output Buffer

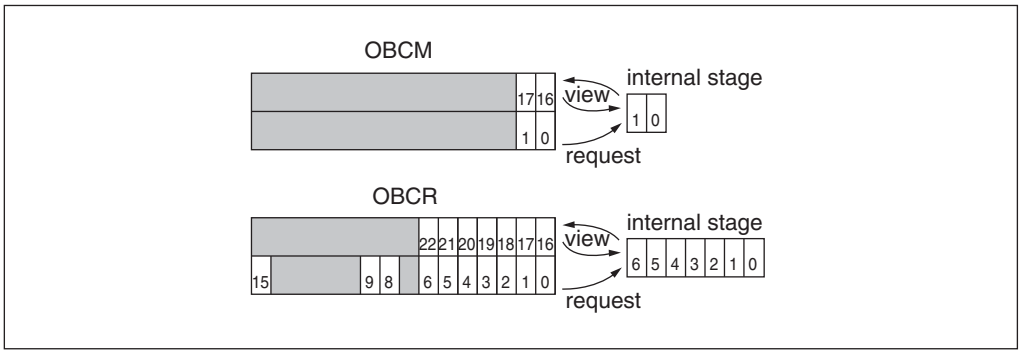


Similar to the **OBCM.RHSS**, **OBCM.RDSS**, **OBCM.RHSH** and **OBCM.RDSH** bits, as well as the **OBCR.OBRS [6:0]** and **OBCR.OBRH [6:0]** bits, the OBF host and OBF shadow will be exchanged via the settings of the **OBCR.VIEW** and **OBCR.REQ** bit.

Setting the **OBCR.REQ** bit to "1" will copy each bit of the **OBCM.RHSS** and **OBCM.RDSS** as well as each bit of the **OBCR.OBRS [6:0]** to the internal register (see Figure 37.4-13).

After setting the **OBCR.REQ** to "1", **OBCR.OBSYS** is set to "1". Then, transfer will be initiated from the message RAM of the message buffer selected by the **OBCR.OBRS [6:0]** to the OBF shadow. **OBCR.OBSYS** bit will be cleared to "0" after the transfer between message RAM and OBF shadow. While the **OBCR.OBSYS** is "0", it is possible to set the **OBCR.REQ** and **OBCR.VIEW** to "1".

Figure 37.4-13 Bit Exchange between the OBCM Register and OBCR Register



While the bit **OBCR.OBSYS** is "0", it is possible to exchange the OBF host and OBF shadow by setting the **OBCR.VIEW** to "1" (see Figure 37.4-12). Moreover, the bit **OBCR.OBRH [6:0], OBCM.RHSH**, and **OBCM.RDSH** are exchanged with the internal register. Storing the copied data from the **OBCR.OBRS [6:0]** and **OBCM.RHSS** as well as **OBCM.RDSS** before **OBCR.VIEW** is set to "1", the exchange ensures that the message buffer number read from **OBCR.OBRH [6:0]** and the mask setting read from the **OBCM.RHSH** and the **OBCM.RDSH** match the transfer data read from the OBF host (see Figure 37.4-13).

After this exchange, it is possible to read transferred message buffers from the OBF host while the message handler is ready to transfer the next message from the message RAM to the OBF shadow.

OBSYS will be set to "1" when **REQ** and **VIEW** are set to "1" at the same time while **OBSYS** is "0". Then, the OBF host and the OBF shadow are switched. Further, mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped by the internal register and are corresponding to each output buffer transfer. Then, **OBRS [6:0]** will be copied to the internal register and be transferred from the message RAM of the specified message buffer to OBF shadow. CPU can read the message buffer transferred from the OBF host in process of transfer. The **OBSYS** bit will be cleared to "0" after the transfer between the message RAM and the OBF shadow has been completed.

Example of output buffer configuration procedure:

Request of transfer to the first message buffer for the OBF shadow

Writing the command mask: Writing to the **OBCM.RHSS** and **OBCM.RDSS**

First message buffer transfer request: Writing to the **OBCR.OBRS [6:0]** and **OBCR.REQ**

Wait until the **OBCR.OBSYS** is cleared to "0".

Request of transfer to the second message buffer for the OBF shadow, reading the first message buffer from the OBF host

Writing the command mask: Writing to the **OBCM.RHSS**, **OBCM.RDSS**

OBF host-shadow exchange for the first message, transfer request for the second message: Writing to the **OBCR.VIEW**, **OBCR.REQ** and **OBCR.OBRS [6:0]**

First message read

Wait until the **OBCR.OBSYS** is cleared to "0".

Request of transfer to the third message buffer for the OBF shadow, reading the second message buffer from the OBF host

Writing the command mask: Writing the **OBCM.RHSS**, **OBCM.RDSS**

OBF host-shadow exchange for the second message, transfer request for the third message: Writing to the **OBCR.VIEW**, **OBCR.REQ** and **OBCR.OBRS [6:0]**

Second message read

Wait until the **OBCR.OBSYS** is cleared to "0".

...(repeat the same procedure)

Read the nth message buffer from the OBF host (no more message buffer transfer shall be requested)

OBF host-shadow exchange for the nth message: Writing to the **OBCR.VIEW** (not to **OBCR.OBRS [6:0]**)

The nth message read

Table 37.4-8 Assignment of the Output Buffer Command Mask Bit

Pos.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 37.4-9 Assignment of the Output Buffer Request Mask Bit

Pos.	Access	Bit	Function
22...16	r	OBRH [6:0]	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6...0	r/w	OBRS [6:0]	OBF Request Shadow, number of message buffer for next request

37.4.11.3 FlexRay Protocol Controller Access for Message RAM

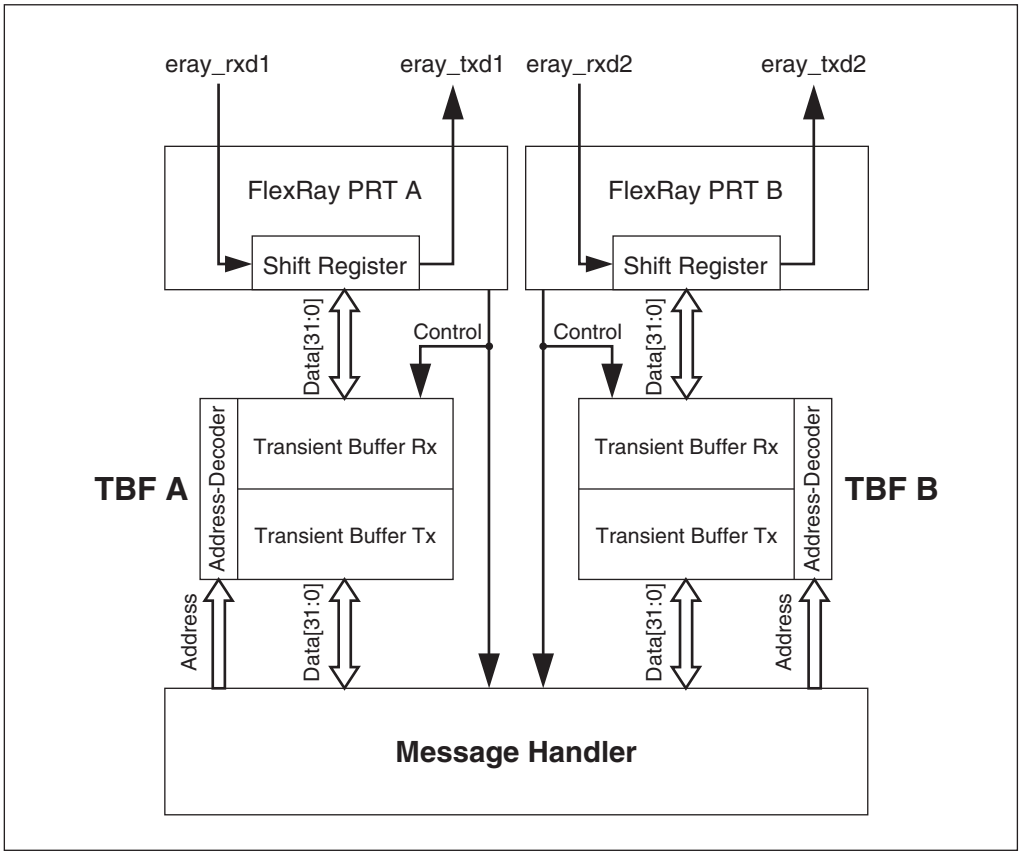
Two temporary storage buffer RAM (TBF A, B) are used to buffer data when transferring data between two FlexRay channel protocol controllers and message RAM.

Two temporary storage buffer RAM are configured as double buffers, and two complete FlexRay messages can be stored. While one buffer can be accessed by a message handler, the other buffer is assigned to the corresponding channel protocol controller.

For example, in the case that the message handler writes the transmission message to the temporary storage buffer Tx, the FlexRay channel protocol controller can store the message that is currently being received in the temporary storage buffer Rx. While the message that was stored to the temporary storage buffer Tx is being transmitted, the message handler (in the case that it has passed the acceptance filter) will update the message buffer by transferring the stored message that was most recently received in the temporary storage buffer Rx to the message RAM.

The data transfer between the temporary storage buffer RAM and the FlexRay channel protocol controller shift register will occur in 32-bit word units. This uses the independent 32-bit shift register of the FlexRay message most effectively.

Figure 37.4-14 Access to the Temporary Storage Buffer RAM



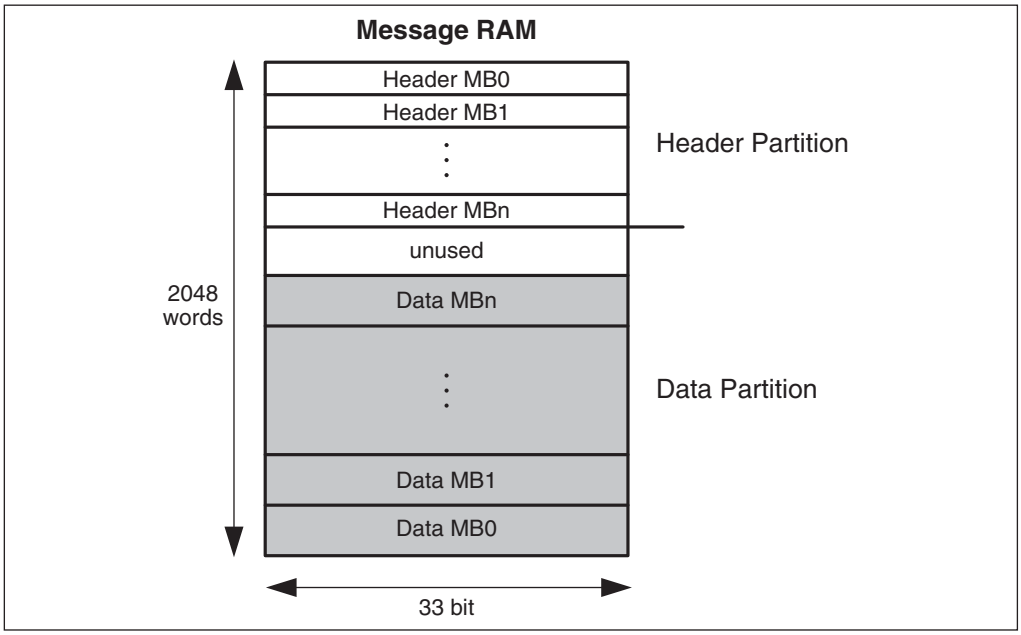
37.4.12 Message RAM

In order to avoid collision between host access to message RAM and FlexRay message transmission/reception, the message buffer of the message RAM cannot be directly accessed. Access is processed through the input buffer and output buffer. Message RAM can store up to 128 message buffers.

Message RAM is composed of 2048 bytes × 33 bits = 67,584 bits. Each 32-bit data is protected by a parity bit. Message RAM is structured as indicated in Figure 37.4-15 so that the number of variable length data bytes (from 0 to 254) can be contained for each FlexRay frame.

Data partition begins from (**MRC.LCB**+1) × 4 words in message RAM (1 word = 32 + 1 bits).

Figure 37.4-15 An Example of Setting of the Message Buffer in the Message RAM



Header Partition

This stores the header section of the configured message buffer.

This supports up to 128 message buffers.

Each message buffer has header section with 4 words (1 word = 32 + 1 bits).

Each message buffer header 3 has an 11-bit data pointer for each data section in the data partition.

Data partition

This is a variable length storage area that can store data sections with different data length. The number of the largest message buffers for various data length is indicated below.

Each data section is 254 byte length and 30 message buffers.

Each data section is 128 byte length and 56 message buffers.

Each data section is 48 byte length and 128 message buffers.

(Note) Please configure the allocated portion of header partition + data partition to 2048 words (1 word = 33 bits) or less.

37.4.12.1 Header Partition

Message buffer status and message buffer setting elements are stored in header partition of the message RAM as indicated in Figure 37.4-16. Header section setting of the message buffer is completed through **IBF** (**WRHS1...3**). Reading from the header section is performed through **OBF** (**RDHS1...3 + MBS**). In order to define the start point of the data section of the each message buffer, please configure the data pointer in the header section. In addition, please do not modify the data pointer during execution. Please configure (reconfigure) the message buffer that belongs to FIFO message group in DEFAULT_CONFIG state or CONFIG state.

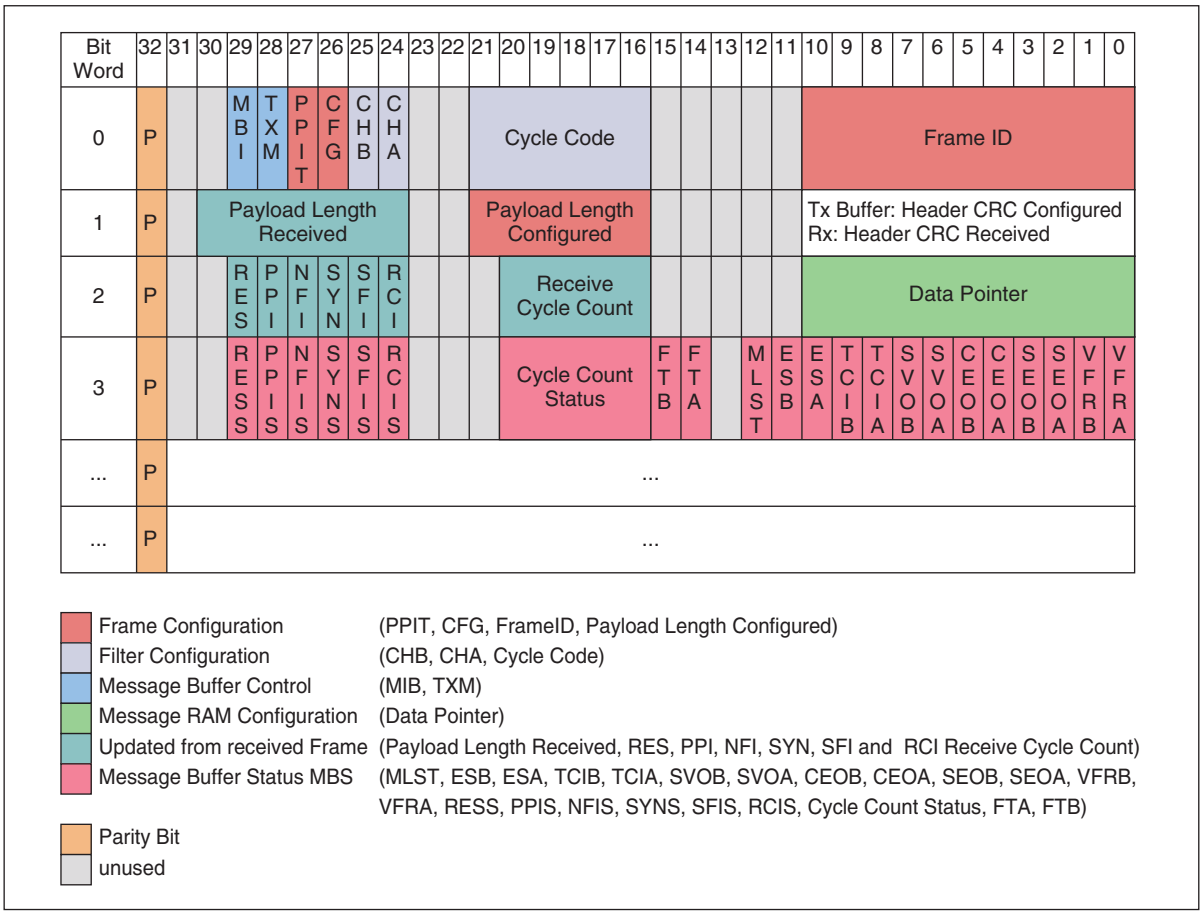
Header section of each message buffer occupies 4 words (1 word = 32 + 1 bits) in header partition of the message RAM. Header section of the message buffer 0 begins from the top of the message RAM.

Please calculate the transmission buffer header CRC.

Received payload length **PLR [6:0],** reception cycle counter **RCC [5:0],** reception channel indicator **RCI**, startup frame indicator **SFI**, synchronous frame indicator **SYN**, null frame indicator **NFI**, payload preamble indicator **PPI**, and reserved bit **RES** are updated by effective reception frame (including effective null frame).

Header 4 word area of each configured message buffer contains each message buffer status **MBS**.

Figure 37.4-16 Header Section of the Message Buffer in the Message RAM



Header 1

The following parameters are written via **WRHS1** and read via **RDHS1**.

Frame ID	- Slot counter filtering setting
Cycle code	- Cycle counter filtering setting
CHA, CHB	- Channel filtering setting
CFG	- Message buffer setting: reception/transmission
PPIT	- Payload preamble indicator transmission
TXM	- Transmission mode setting: single shot/continuous
MBI	- Message buffer transmission/reception interrupt enable flag

Header 2

The following parameters are written via **WRHS2** and read via **RDHS2**.

Header CRC	- Transmission buffer: computes them based on frame header segment
	- Reception buffer: updated by reception frame
Set payload length	- Set data section length (in 2 bytes)
Received payload length	- Payload segment length stored in the reception frame (in 2 bytes)

Header 3

The following parameters are written via **WRHS3** and read via **RDHS3**.

Data pointer - The pointer to the start point of the data section in the data partition

The following parameters are read via **RDHS3**.

This is valid only for the reception buffer and updated by reception frame.

Reception cycle count	- Cycle count values stored from reception frames
RCI	- Reception channel indicator
SFI	- Startup frame indicator
SYN	- Synchronous frame indicator
NFI	- Null frame indicator
PPI	- Payload preamble indicator
RES	- Reserved bit

Header 4

This header is read via **MBS** and will be updated at the end of the configured slot.

The following parameters are valid for transmission buffer and reception buffer.

VFRA	- Channel A reception valid frame
VFRB	- Channel B reception valid frame
SEOA	- Channel A syntax error
SEOB	- Channel B syntax error
CEOA	- Channel A content error
CEOB	- Channel B content error
SVOA	- Channel A slot boundary error
SVOB	- Channel B slot boundary error

The following parameters are valid only for transmission buffer.

TCIA	- Channel A transmission collision indicator
TCIB	- Channel B transmission collision indicator

The following parameters are valid only for reception buffer.

ESA	- Channel A empty slot
ESB	- Channel B empty slot
MLST	- Message lost
FTA	- Channel A frame transmission
FTB	- Channel B frame transmission
Cycle Count Status	- Real cycle count at a status update
RCIS	- Channel indicator reception
SFIS	- Startup frame indicator status
SYNS	- Sync frame indicator status
NFIS	- Null frame indicator status
PPIS	- Payload preamble indicator status
RESS	- Reserved bit status

37.4.12.2 Data partition

Data partitions in the message RAM store data sections of the message buffer set for reception/transmission as defined in the header partition. 0 to 254 data bytes can be set for each message buffer. To optimize data transfers between host interface and message RAM, and between shift registers of two FlexRay channel protocol controllers and message RAMs, the bit width of message RAMs will be set with 32 bits + 1 parity bit.

The data partition starts immediately after the end of the header partition. When you set the message buffer in the message RAM, configure it so that the data pointer points to the address within the data partition. Following Figure 37.4-17 shows an example of how to store the message RAM to the data partitions for the data section of the message buffer configured.

The start point and the end point of the data section of message buffer are determined by the data pointer and the payload length configured in the header section of the message buffer. This will enable you to use the RAM space of the message buffer flexibly with different data lengths.

If the size of the data section is odd number in 2 bytes, the remaining 16 bits of the last 32-bit word will not be used (See Figure 37.4-17).

Figure 37.4-17 Structure Example of Data Section in Message RAM

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	P	unused								unused								unused								unused							
...	P	unused								unused								unused								unused							
...	P	MBn Data3								MBn Data2								MBn Data1								MBn Data0							
...	P	...								...								...								...							
...	P	...								...								...								...							
...	P	MBn Data(m)								MBn Data(m-1)								MBn Data(m-2)								MBn Data(m-3)							
...	P	...								...								...								...							
...	P	...								...								...								...							
...	P	...								...								...								...							
...	P	MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
...	P	...								...								...								...							
...	P	MBn Data(k)								MBn Data(k-1)								MBn Data(k-2)								MBn Data(k-3)							
2046	P	MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
2047	P	unused								unused								MB0 Data5								MB0 Data4							

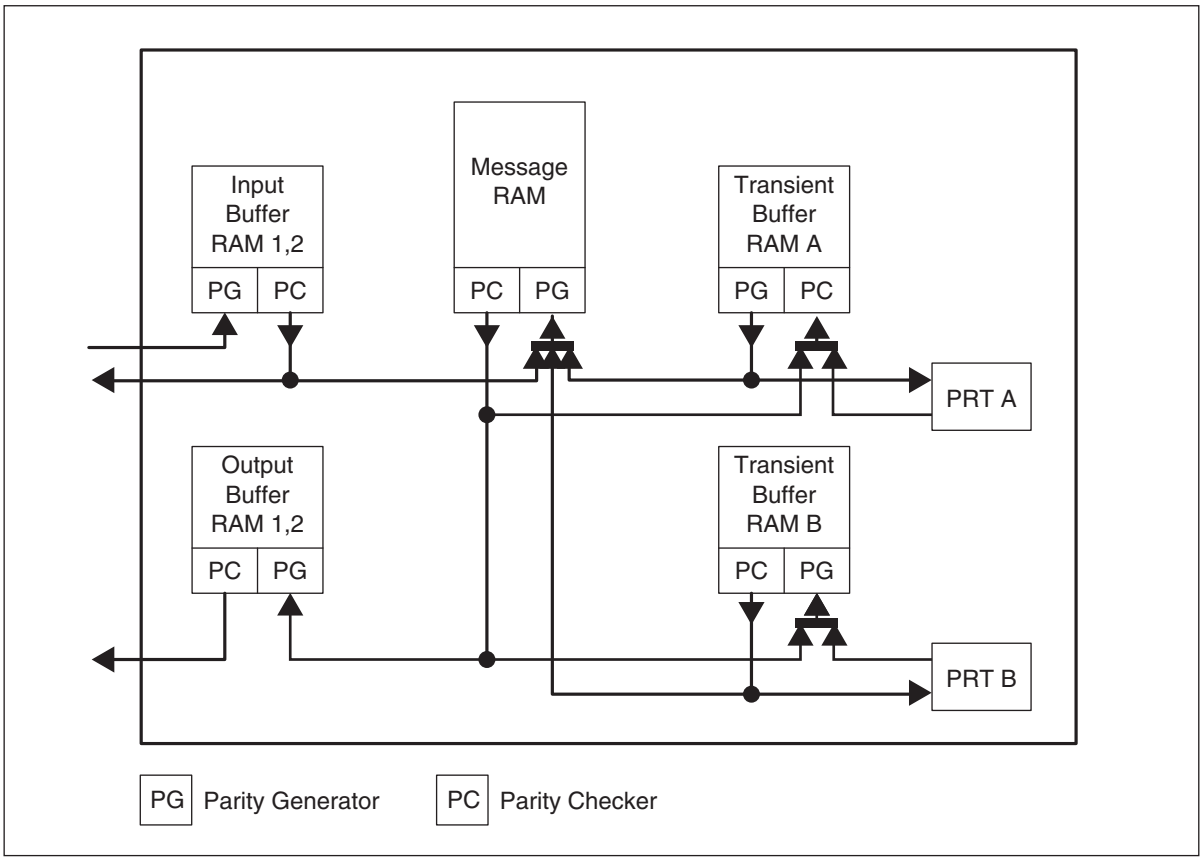
37.4.12.3 parity check

FlexRay controller implements a parity check mechanism to guarantee the consistency of the data stored in 7 RAM blocks. The RAM block has a parity generator/checker connected to it as shown in Figure 37.4-18 and the parity generator will generate a parity bit when data is being written to the RAM block. FlexRay controller uses even parity (the number of "1"s in a 32-bit word being an even number generates a "0" parity bit). Then, the parity bit will be stored with each data. The parity is also checked whenever the data is read from the RAM block. Internal data bus in the FlexRay controller has 32-bit width.

When a parity error is detected, its error flag will be set to "1". The parity error flag (**MHDS.PIBF**, **MHDS.POBF**, **MHDS.PMR**, **MHDS.PTBF1** and **MHDS.PTBF2**) and the error message buffer indicator (**MHDS.FMBD**, **MHDS.MFMB** and **MHDS.FMB [6:0]**) are in the message handler status register. Those error flags control the error interrupt flag EIR.PERR.

Figure 37.4-18 shows the data paths between RAM blocks and between the parity generator/checker.

Figure 37.4-18 Parity Generation and Check



(Note) Parity generator and parity checker are the blocks independent from the RAM blocks.

When a parity error is detected, the procedure below will be executed.

In all cases:

- Each parity error flag in the **MHDS** register will be set.
- The parity error flag **EIR.PERR** is set, if an interrupt is enable, it will occur.

Special cases:

1)A parity error during a data transfer from input buffer RAM1, 2 to message RAM

a) Header and data transfer, or data transfer:

- **MHDS.PIBF** bit will be set.
- **MHDS.FMBD** bit will be set to indicate that **MHDS.FMB [6:0]** have been updated.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.
- For the transmission buffer where a parity error occurred, the transmission request bit will not be set.

b) Data Transfer

The parity errors in reading header parts of each message in the message RAM

- **MHDS.PMR** bit will be set.
- **MHDS.FMBD** bit will be set to indicate that **MHDS.FMB [6:0]** have been updated.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.
- Data of the message buffer is not updated.
- For the transmission buffer where a parity error occurred, the transmission request bit will not be set.

2) A parity error during a data transfer from input buffer RAM1, 2 to the host
- **MHDS.PIBF** bit will be set.

3) A parity error while scanning the header section of message RAM
- **MHDS.PMR** bit will be set.
- **MHDS.FMBD** bit will be set to indicate that **MHDS.FMB [6:0]** have been read.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.
- The message buffer with a parity error will be ignored.

4) A parity error during a data transfer from message RAM to temporary storage buffer RAM1, 2
- **MHDS.PMR** will be set
- **MHDS.FMBD** will be set to indicate that **MHDS.FMB [6:0]** have been read.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.
- A frame transmission from failed message buffer will be stopped.

5) A parity error during a data transfer from temporary storage buffer RAM1, 2 to the channel protocol controller 1, 2
- **MHDS.PIBF1**, **2** will be set.
- A frame transmission from failed temporary storage buffer will be stopped.

6) A parity error during a data transfer from temporary storage buffer RAM1, 2 to message RAM
- **MHDS.PIBF1**, **2** will be set.
- **MHDS.FMBD** will be set to indicate that **MHDS.FMB [6:0]** have been updated.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.

7) A parity error during a data transfer from message RAM to output buffer RAM
- **MHDS.PMR** will be set.
- **MHDS.FMBD** will be set to indicate that **MHDS.FMB [6:0]** have been read.
- **MHDS.FMB [6:0]** display the message buffer number where an error occurs.

8) A parity error during a data transfer from output buffer RAM to the host
- **MHDS.POBF** will be set.

9) A parity error during a data read of temporary storage buffer RAM1, 2

If a parity error occurred for the message handler reading a frame with network management information (PPI=1) from the temporary storage buffer RAM1 and 2, the network management vector NMV1 to NMV3 corresponding the frame will not be updated.

37.4.12.4 Handling parity errors

Restoration of a parity error with transfer.

● Self-restoration
- Input buffer RAM 1, 2
- Output buffer RAM 1, 2
- Data of Message RAM

- Temporary buffer RAM A
- Temporary buffer RAM B

The parity errors mentioned above could be restored through overwriting by accessing CPU or FlexRay communication.

● Clear RAM command

In the states of DEFAULT_CONFIG or CONFIG, the RAM in the entire module can be initialized to "0".

● Temporary unlocking of header part

A parity error in the header part of a locked message buffer will be restored by transferring the portion from input buffer to a locked header of the buffer.

For this transfer, the write access to IBCR (the number of massage buffers has been specified) should be done prior to unlock CONFIG state (see Section "37.3.2.1 Lock Registers (LCK: Lock Register)").

The data, regardless of whether it or the lock via MRC.SEC [1:0] belongs to FIFO, will be updated by unlocked messages/buffer headers in a single transfer.

37.4.12.5 Temporary unlock of header section

Parity errors in a header in fixed message buffers can be fixed by transferring it from the input buffer (IBF) to the fixed buffer header section. For this transfer, the unlock series normally used to maintain the current CONFIG status should be done prior to the writing to IBCR (the number of massage buffers has been specified) (see Section "37.3.2.1 Lock Registers (LCK: Lock Register)").

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by MRC.SEC[1:0], and will be updated with new data.

37.4.13 Interrupts

Interrupt pins to generate interrupts without delay are provided when one of the following events occurs: an error generated, a status change detected, a frame transmission/reception, or a timer event occurred. This enable you to quickly take an action toward the error states, status changes, and timer events. However the operation rate required for the application could not be met if too many interrupts occurred. To cope with this, FlexRay controller supports the feature to enable/disable an interrupt separately.

An interrupt occurs in the following cases.

- An error is detected
- A status flag is set
- Timer reaches the set value
- A message transfer from input buffer to message buffer or the message RAM to output buffer are completed
- A stopwatch event occurs

The features of event displays and interrupt generations when a status changes or an error occurs are implemented independently with two tasks. Every event will be displayed whether their interrupts are valid or not. You can obtain current error information and status information by reading EIR register and SIR register.

Table 37.4-10 List of Module Interrupt Flag and Interrupt Line Valid Flag (1 / 2)

Register	Bit	Function
EIR	PEMC	Protocol Error Mode Changed Flag
	CNA	Command Not Valid
	SFBM	Sync Frames Below Minimum
	SFO	Sync Frame Overflow
	CCF	Clock Correction Failure
	CCL	CHI Command Locked
	PERR	Parity Error
	RFO	Receive FIFO Overrun
	EFA	Empty FIFO Access
	IIBA	Illegal Input Buffer Access
	IOBA	Illegal Output Buffer Access
	MHF	Message Handler Constraints Flag
	EDA	Error Detected on Channel A
	LTVA	Latest Transmit Violation Channel A
	TABA	Transmission Across Boundary Channel A
	EDB	Error Detected on Channel B
	LTVB	Latest Transmit Violation Channel B
	TABB	Transmission Across Boundary Channel B

Table 37.4-10 List of Module Interrupt Flag and Interrupt Line Valid Flag (2 / 2)

Register	Bit	Function
SIR	WST	Wakeup Status
	CAS	Collision Avoidance Symbol
	CYCS	Cycle Start Interrupt
	TXI	Transmit Interrupt
	RXI	Receive Interrupt
	RFNE	Receive FIFO not Empty
	RFF	Receive FIFO Full
	NMVC	Network Management Vector Changed
	TI0	Timer Interrupt 0
	TI1	Timer Interrupt 1
	TIBC	Transfer Input Buffer Completed
	TOBC	Transfer Output Buffer Completed
	SWE	Stop Watch Event
	SUCS	Startup Completed Successfully
	MBSI	Message Buffer Status Interrupt
	SDS	Start of Dynamic Segment
	WUPA	Wakeup Pattern Received on Channel A
	MTSA	MTS Received on Channel A
	WUPB	Wakeup Pattern received on Channel B
	MTSB	MTS Received on Channel B
ILE	EINT0	Enable Interrupt Line 0
	EINT1	Enable Interrupt Line 1

Interrupt pins, INT0 and INT1 are controlled by valid interrupts. Furthermore, each interrupt pin, INT0 and INT1 can be set to **ILE.EINT0** or **ILE.EINT1** respectively, which allows you to select them to enable/disable separately.

The two timer interrupts generated by interrupt timer 0 and interrupt timer 1 can be used on pin INT2 in 16-bit non-multiplex bus mode and on pins INT2 and INT3 in 16-bit multiplex bus mode. Those can be set via **T0C** register and **T1C** register.

The stopwatch event will be generated by input pin **STOPWT**.

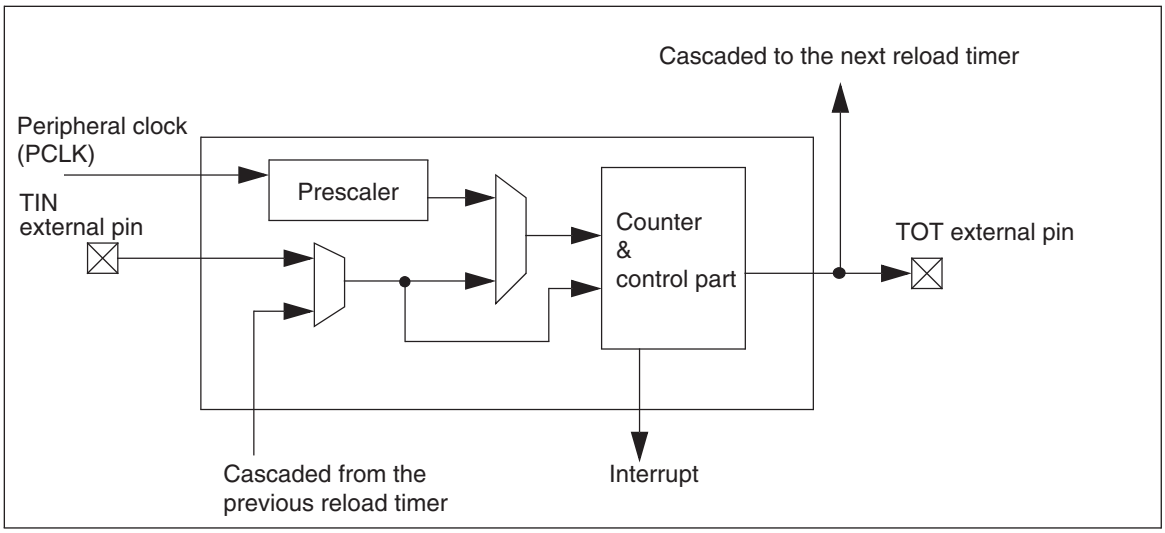
SIR.TIBC bit and **SIR.TOBC** bit will be set to "1" separately after a data transfer between IBF/OBF and message RAM is completed.

CHAPTER 38 RELOAD TIMER

38.1 Overview

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 38.1-1 Block Diagram of Reload Timer (1 Channel, Overview)



38.2 Features

A 4-channel reload timer is installed in this product. Each channel is configured as follows:

- 16-bit down counter × 1
- 16-bit reload register × 1
- 16-bit reload/compare/capture register × 1
- Buffer described above × 1
- 6-bit prescaler for internal count clock creation × 1
- External trigger/event input (TIN) × 1
- External toggle output (TOT) × 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)

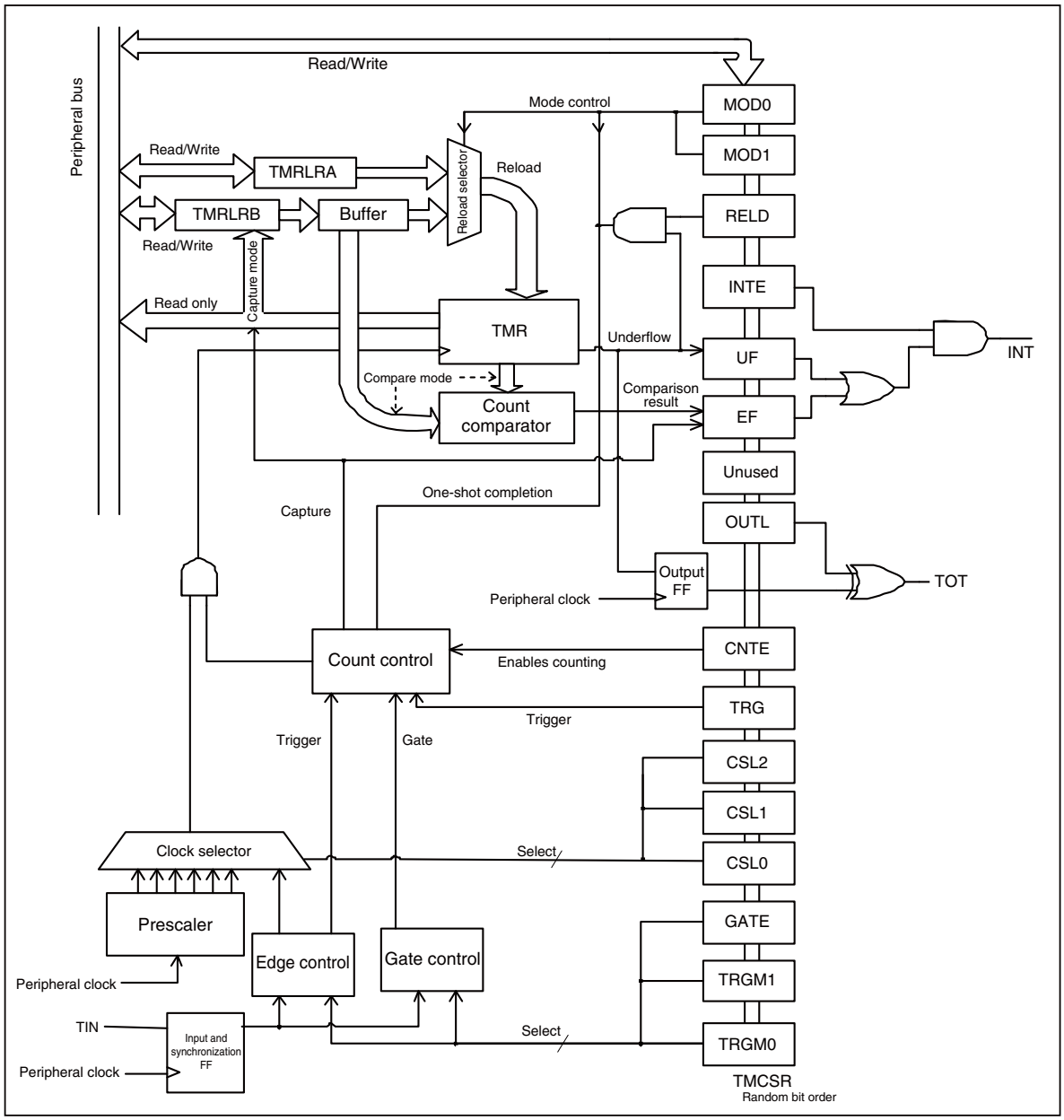
Code: FR81S09_RLT-1v0-91580MS-1-E

  - 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.)

38.3 Configuration

Figure 38.3-1 Block Diagram of Reload Timer (1 Channel, Detailed)



38.4 Registers

■ Table of Base Addresses (Base_addr) and External Pins

Table 38.4-1 Table of Base Addresses (Base_addr) and External Pins

Channels	Base_addr	External pin	
		TOT	TIN
0	0x0060	TOT0	TIN0
1	0x0100	TOT1	TIN1
2	0x0108	TOT2	TIN2
3	0x0110	TOT3	TIN3

■ Registers Map

Table 38.4-2 Registers Map

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

38.4.1 TMCSR (TiMer Control and Status Registers) Control Status Registers

This register controls the operating mode and interrupt.

It is not possible to rewrite any data other than bit7 and bit3 to 0 when bit1:CNTE=1.

It is possible to rewrite bit15-8 and bit6-4 and write counter operation enabling by writing CNTE=1 simultaneously. It is also possible to rewrite bit15-8, bit6-4 and write operation deactivation by writing CNTE=0 simultaneously.

- **TMCSR0 to TMCSR3: Address 0066_H, 0106_H, $001E_H$, 0116_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
MOD[1:0]		TRGM[1:0]		CSL[2:0]			GATE	
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
EF	-	OUTL	RELD	INTE	UF	CNTE	TRG	
0	0	0	0	0	0	0	0	Initial value
R(RM1), W	R/W0	R,W	R,W	R/W	R (RM1), W	R/W	R0,W	Attribute

[bit15, bit14] MOD (MODe) Mode selection bits

MOD1, MOD0	Operation mode
00	Single mode (initial value)
01	Dual mode
10	Compare mode
11	Capture mode

[bit13, bit12] TRGM1, TRGM0 (TRiGger input Mode select) TIN Input mode selection bits

These bits control input pin functions. The functions of the interval timer mode differ from those of the event counter mode.

[Interval timer mode, trigger input (bit8:GATE bit="0")]

Select an effective external edge which can be a reload trigger through TIN input in the following manner:

TRGM1, TRGM0	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:

TRGM1, TRGM0	TIN effective level
x0	TIN pin "L" counted only during the input period (initial value)
x1	TIN pin "H" counted only during the input period

[Effective edge setting at the event counter mode]

In the event counter mode, select an edge for external event detection in the following manner: Every time an external event is detected, the counter value is decremented. When an external event is selected, the setting of the bit8:GATE bit becomes invalid.

TRGM1, TRGM0	Count target edge
00	Not used
01	Rising edge
10	Falling edge
11	Both edges

[bit11 to bit9] CSL2 to CSL0 (Count source SeLect) Count source selection bits

These are count source selection bits. Select a count source from the internal clock (peripheral clock (PCLK)) and the external event (TIN input) in the following manner: When the event counter mode is set, set the count effective edge using bits 13, 12:TRGM[1:0].

CSL2 to CSL0	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:TOT0, ch.2:TOT1, ch.3:TOT2)	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, 10, 9:CSL[2:0]=000 - 101) at the interval timer mode in the following manner.

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 and clear operation are performed in the compare mode. The values read with read-modify-write instructions will always be "1".

[bit6] Reserved

Always write "0" to this bit.

[bit5] OUTL (OUTput Level) Output polarity selection bit

This bit controls output polarity of the timer output pin (TOT).

OUTL	TOT initial value	TOT 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 in the following manner:

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 interrupt request in case of underflow/compare match (event counter mode)/capture in the following manner:

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 decremented 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 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
"0" write	Do nothing.
"1" write	A software trigger is generated.

When "0" is written into this bit, the timer takes no action. 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.

38.4.2 TMR (16bit TiMer Register) 16-bit Timer Register

This register can read the timer count value.

Always perform 16-bit access to this register.

- **TMR0 to TMR3: Address 0062_H, 0102_H, $010A_H$, 0112_H (Access:Half-word)**

15	14	• • •	2	1	0	bit
TMR[15:0]						
X	X	• • •	X	X	X	Initial value
R,WX	R,WX	• • •	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TMR (TiMeR) 16-bit timer

This register can read the counter value of the 16-bit timer. The initial value is indefinite.

38.4.3 TMRLRA, TMRLRB(16bit TiMer ReLoad Register A/B) 16-bit Timer Reload Register A, 16-bit Timer Reload Register B

This register sets the count initial value and other items.

Always perform 16-bit access to this register.

- **TMRLRA0 to TMRLRA3: Address 0060_H, 0100_H, 0108_H, 0110_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
TMRLRA[15:0]						
X	X	• • •	X	X	X	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

- **TMRLRB0 to TMRLRB3: Address 0064_H, 0104_H, $010C_H$, 0114_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
TMRLRB[15:0]						
X	X	• • •	X	X	X	Initial value
R,W	R,W	• • •	R,W	R,W	R,W	Attribute

[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 is where the count initial value is stored. The TMRLRA can be used in any mode with no regard to the bit15, 14:MOD[1:0] setting in the TMCSR register.

The TMRLRB is to be used in any mode with regard to the bit15, 14:MOD[1:0] setting in the TMCSR register in the following manner:

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 (TOT) 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 (TOT) to be outputted is shown in the table below.

MOD[1:0]	Mode	RELD	OUTL	TOT 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 TOT output time (TOUT) when the register is used as the single mode and dual mode in the interval time mode:

TOUT =(Setting value of this register + 1) × count source cycle

38.5 Operation

38.5.1 Configuration

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}).

■ Count Source

Select decrement conditions of the down counter in the TMCSR.CSL[2:0].

Table 38.5-1 Count Source List

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:TOT0, ch.2:TOT1, ch.3:TOT2)	Event counter mode
111	External event (TIN input)	

■ Timer Underflow Cycle

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:

$$\text{TUF} = \text{Peripheral clock (PCLK) cycle} \times \text{prescaler division value (2-64)} \times (\text{Reload register value(TMRLRA/B)} + 1)$$

■ Trigger

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, software trigger is always used. In the interval timer mode, settings are made in the TMCSR register.

■ Gate

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 38.5-2 TIN Effective Level

TRGM[0]	TIN effective level
0	TIN pin "L" counted only during the input period (initial value)
1	TIN pin "H" counted only during the input period

■ Counter Operation Selection

Select the operation in case of counter underflow in the mode select bits (bit15, 14: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 38.5-3 Counter Operation List

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 the section of dual one-shot)	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

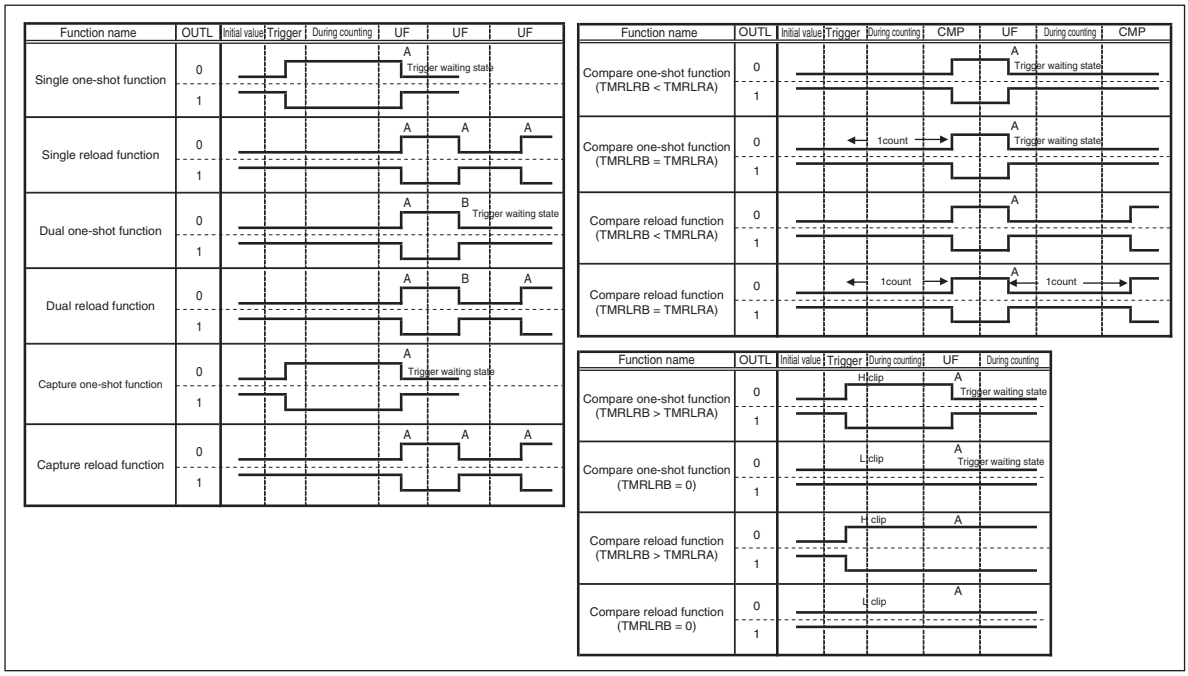
■ TOT Pin Level Setting

Set pin output polarity using bit5:OUTL bit in the TMCSR register.

The relationships between events and the TOT in each function are as follows:

A/B of the section of the UF (underflow) below indicates whether down counting underflow has occurred with a value when loading TMRLRA data or TMRLRB data. CMP (compare-match) shows the timing of down counting from TMRLRB = TMR.

Figure 38.5-1 TOT Output Change in Each Event



38.5.2 Operation Procedures

■ Activation

Writing "1" into the bit1:CNTE bit of the TMCSR register changes the counter state to activation trigger waiting.

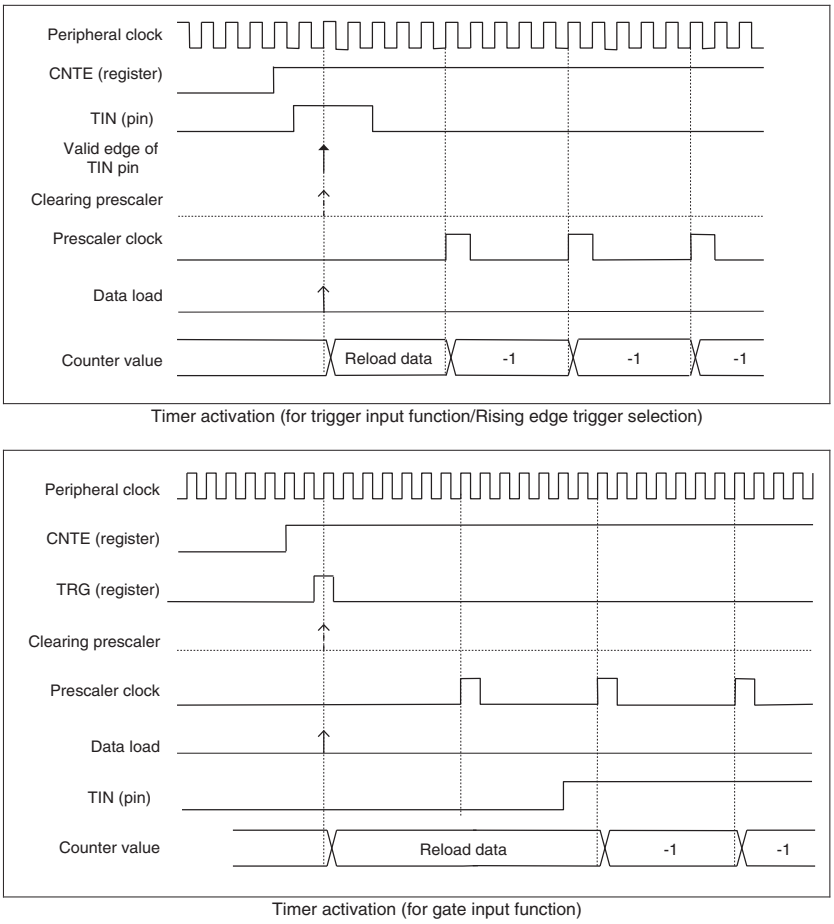
- TIN input during trigger input functioning

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 38.5-2 Timer Activation



■ Retrigger

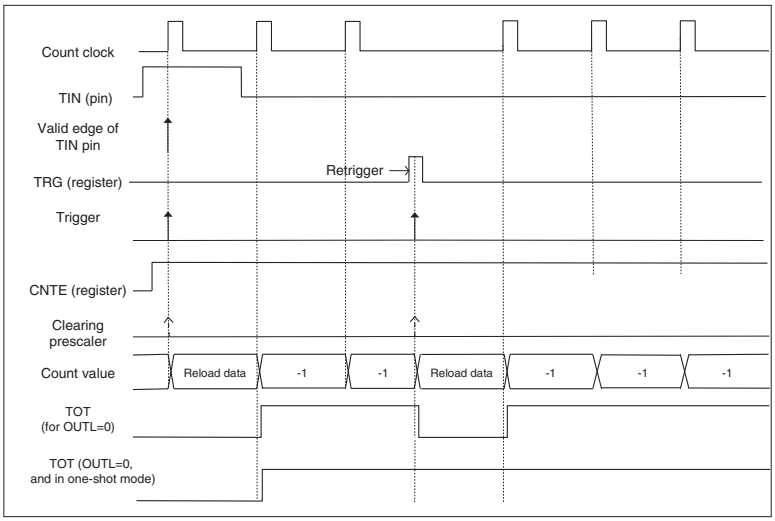
Trigger which is generated during timer counting is called "retrigger". In this case, the following actions are taken:

1. Initialize TOT
2. Load the reload register value to the counter
3. Clear the 6-bit prescaler
4. Continue counting

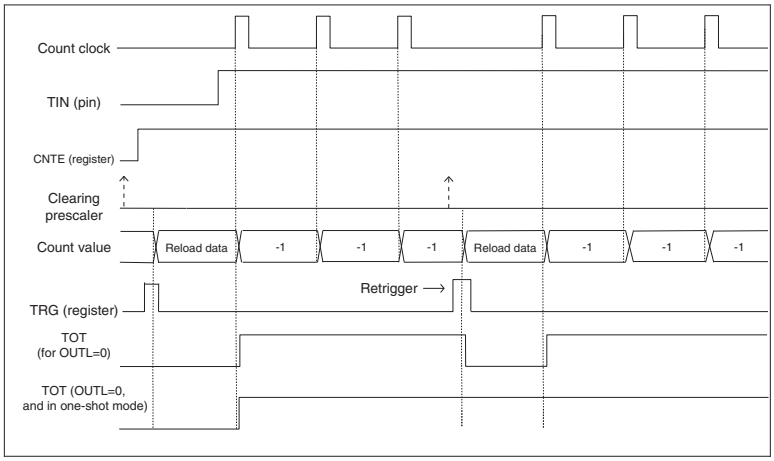
Only in the capture mode, retrigger generation transfers a value being counted to the TMRLRB to set the EF bit of the TMSCR register.

(Note) In one-shot mode, retrigger does not initialize TOT.

Figure 38.5-3 Retrigger Operation



Operation at retrigger (TIN is trigger input, at rising edge trigger and one-shot output)



Operation at retrigger (TIN is gate input, count at H level and one-shot output)

■ Underflow/Reload

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.

■ Generation of Interrupt Requests

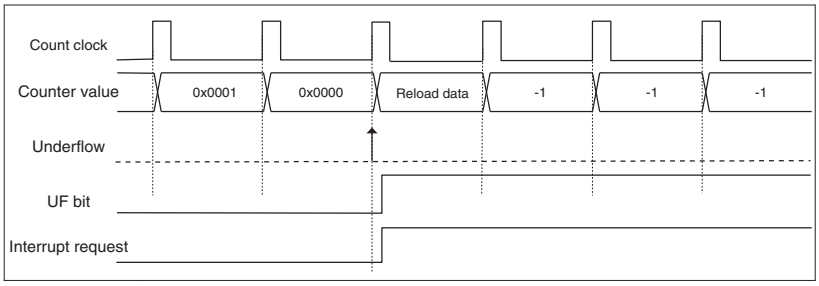
When bit3:INTE bit of TMCSR register is "1", if bit2:UF bit/bit7:EF bit are set, an interrupt request is generated. In interval timer mode, UF bit and EF bit will be set under the following conditions.

- 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 TMCSR register and a clear of UF bit by writing "0" occurred concurrently, writing "0" to UF bit will be invalid and UF bit will be set. When a set of bit7:EF bit and a clear of EF bit by writing "0" occurred concurrently, writing "0" to EF bit will be invalid and EF bit will be set.

The following is the example of generation of interrupt requests.

Figure 38.5-4 Example of UF Interrupt Request Output Operation



UF interrupt request output operation (bit4 : RELD= 1 of TMCSR register)

■ Concurrent Operation of a Registers Write and a Timer Activation

The following table shows the operation when a register write by a user and the timer operation occurred simultaneously.

Table 38.5-4 Concurrent Operation

Writing to register	Operation of timer	Operation to execute
A clear of UF bit by writing "0"	Setting to UF bit	Setting to UF bit (Writing "0" will be disabled.)
A clear of EF bit by writing "0"	Setting of EF bit	Setting of EF bit (Writing "0" will be disabled.)
Writing to the reload register	Loading of timer by retrigger	Reloading old data (The written value will be loaded next time)

38.5.3 Operation of Counters

■ Single One-shot Operation

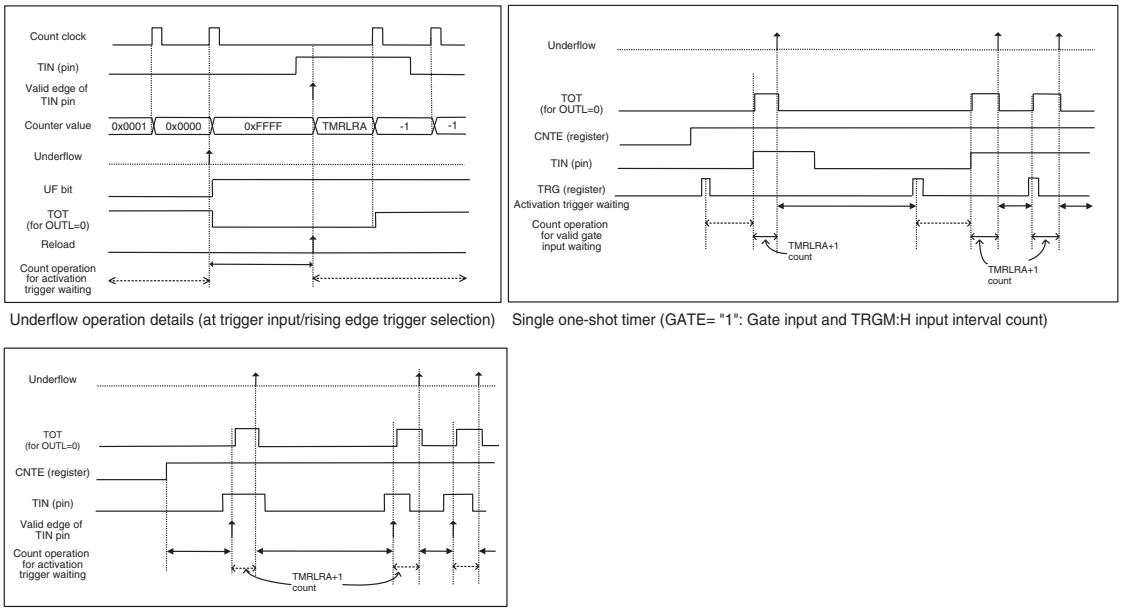
When bit15, bit14:MOD[1:0]="00" and bit4:RELD="0" of the TMCSR register, the single one-shot operation will be performed in which the timer stops with 0xFFFF by an occurrence of an underflow.

In the single one-shot configuration, if an underflow occurs, the following operation will be performed.

- Sets 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 TOT output.
- Timer is waiting for a trigger.

For single one-shot timer, TMRLRA turns to the initial value of the counter when a reload took place. TMRLRB is not used.

Figure 38.5-5 Single One-shot Operation



Underflow operation details (at trigger input/rising edge trigger selection)

Single one-shot timer (GATE= "1": Gate input and TRGM:H input interval count)

Single one-shot timer (GATE= "0": For trigger input/rising edge trigger selection)

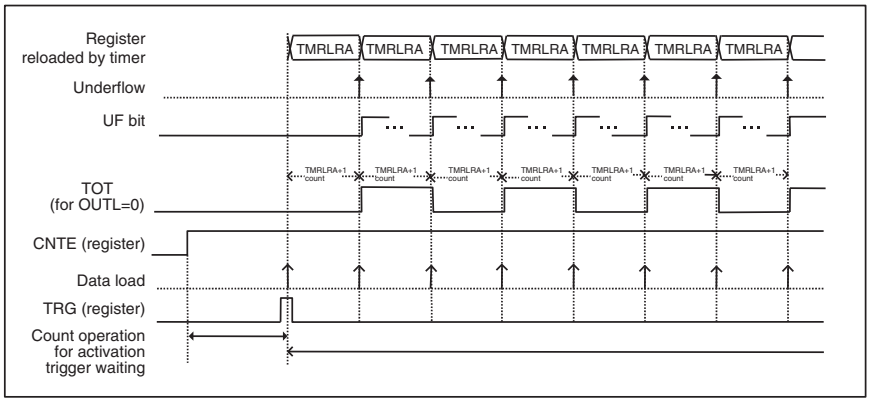
■ Single Reload Operation

When bit15, bit14:MOD[1:0]=00, and bit4:RELD=1 of TMCSR register, the single reload operation will be performed.

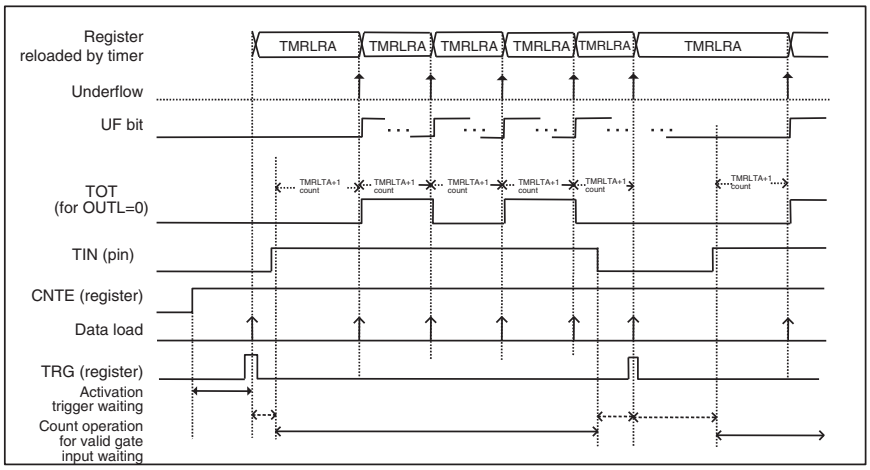
In single reload operation, a value will be loaded from TMRLRA to the timer by trigger input, and a down count (decrementing the count) will start. When an underflow occurs, the value is reloaded from TMRLRA again and the down count operation continues. The value of TMRLRA represents the time the timer will reload. 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 TOT output.
- Continues decrementing count.

Figure 38.5-6 Single Reload Operation



Single reload function (GATE = "0": trigger input)



Single reload function (GATE= "0": Gate input and TRGM:H input interval count)

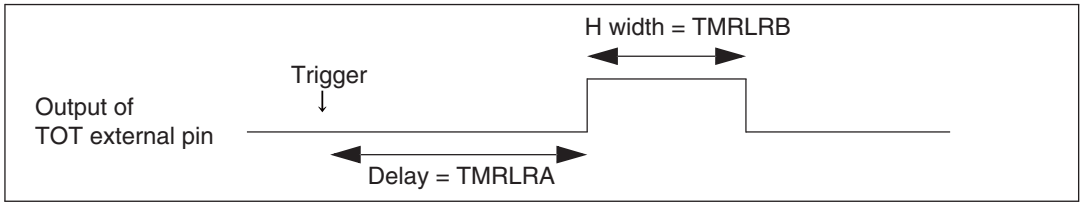
■ Dual One-shot Operation

When bit15, bit14:MOD[1:0]=01, and bit4:RELD=0 of TMCSR register, the timer performs the dual one-shot operation. This can be used as a one-shot PPG.

In dual one-shot operation, values are loaded into the counter one by one in the order of TMRLRA → TMRLRB and decrements the counter for each load. The counter will stop by the second underflow.

When bit5:OUTL="0" of the TMCSR register, the value of TMRLRA represents the time interval between a timer activation (TOT output is in L level) and a toggling of the TOT output to "H", and the value of TMRLRB represents the time interval of H width of the TOT output.

Figure 38.5-7 TOT Pulse Width



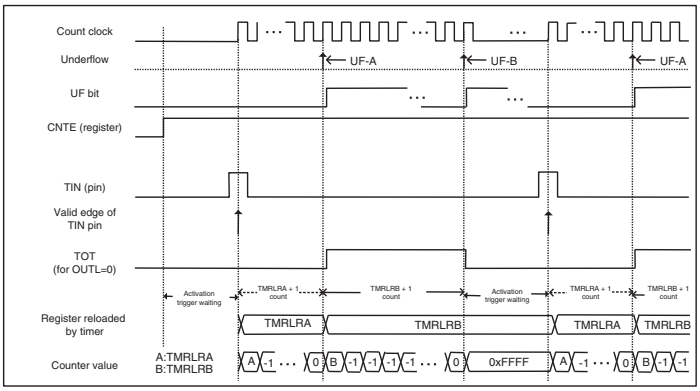
When the first underflow occurs (UF-A), the following operation will take place.

- 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 TOT output.
- Starts a downcount from TMRLRB.

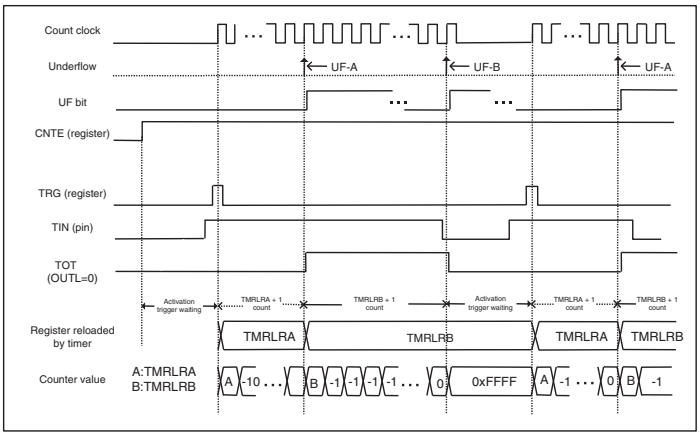
When the second underflow (UF-B) occurs, the following operation will take place.

- 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 TOT output.
- Timer is waiting for an activation trigger.

Figure 38.5-8 Dual One-shot Operation



Dual one-shot operation (trigger input/rising edge trigger selection)



Dual one-shot operation (gate input)

■ Dual Reload Operation

When bit15, bit14:MOD[1:0]=01, and bit4:RELD=1 of TMCSR register, the timer performs the dual reload operation.

In dual reload operation, loads TMRLRA and TMRLRB alternatively and decreases 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 decreases the counter, and if another underflow occurs, loads TMRLRA onto the counter and decrements the counter, and so on.

When bit5:OUTL="0" of the TMCSR register, the value of TMRLRA represents the time interval between a timer activation (TOT output is in L level) and a toggling of the TOT output to "H", and the value of TMRLRB represents the time interval of H width of the TOT output.

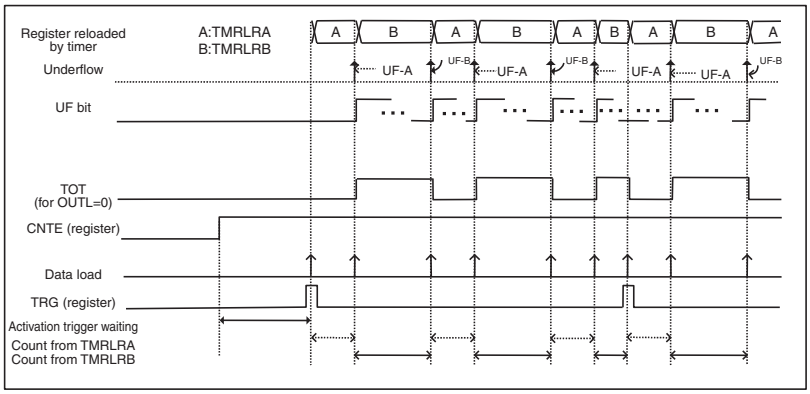
If an underflow (UF-A) occurs at the downcount after loading a value from the TMRLRA, the following operation will be performed.

- 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 TOT output.
- Starts a downcount from TMRLRB.

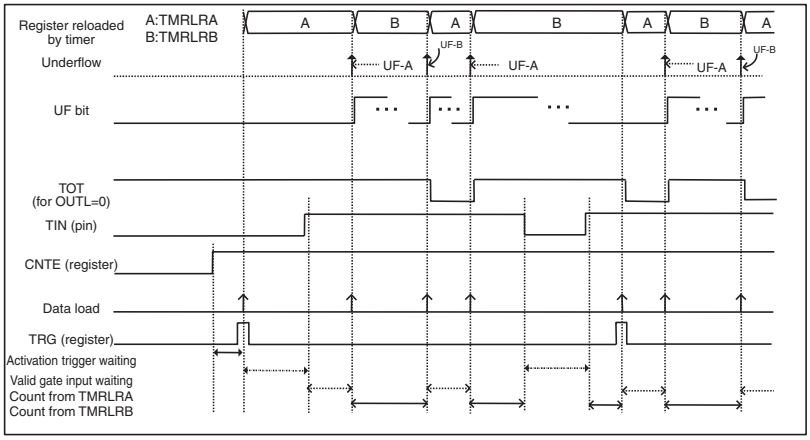
If an underflow (UF-B) occurs at the downcount after loading a value from the TMRLRB, the following operation will be performed.

- 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 TOT output.
- Starts a downcount from TMRLRA.

Figure 38.5-9 Dual Reload Operation



Dual reload function (GATE= "0": Trigger input)



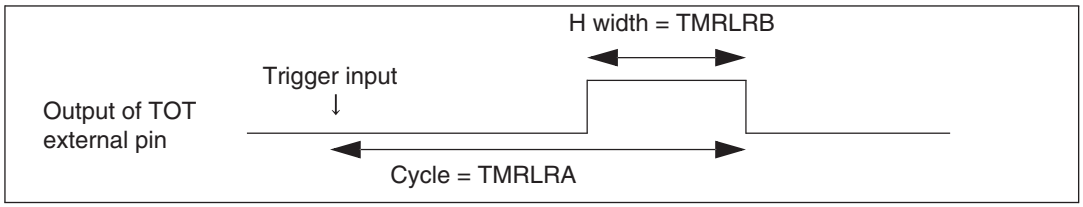
Dual reload function (GATE= "1": gate input, H input interval count)

■ Compare One-shot Operation

When bit15, bit14:MOD[1:0]=10, and bit4:RELD=0 of TMCSR register, the compare one-shot operation in which the counter value (TMR) and the value of TMRLRB register are compared for each downcount will be performed. After accepting a trigger, the value of TMRLRA register is loaded and the downcount starts. When decrementing the count from the value of a compare matched (TMR = TMRLRB), the TOT output will be inverted. When an underflow occurs, count operation stopped, TOT output initialized, and goes into the activation trigger wait state.

The value of TMRLRA indicates the time interval between the activation of timer and the end of it and the value of TMRLRB indicates the counter value when an output of the H width of TOT output starts. When OUTL="0" and TMR < TMRLRB, the TOT output will become "H level".

Figure 38.5-10 TOT Interval, Pulse Width



From the start of a downcount to TMR = TMRLRB (while TMR is greater than or equal to TMRLRB), the following operation will be performed.

- TOT output continues to hold the initial value.
- The timer continues to count.

If a downcount from TMR = TMRLRB occurs, the following operation will be performed.

- Inverts TOT output.
- The timer continues to count.

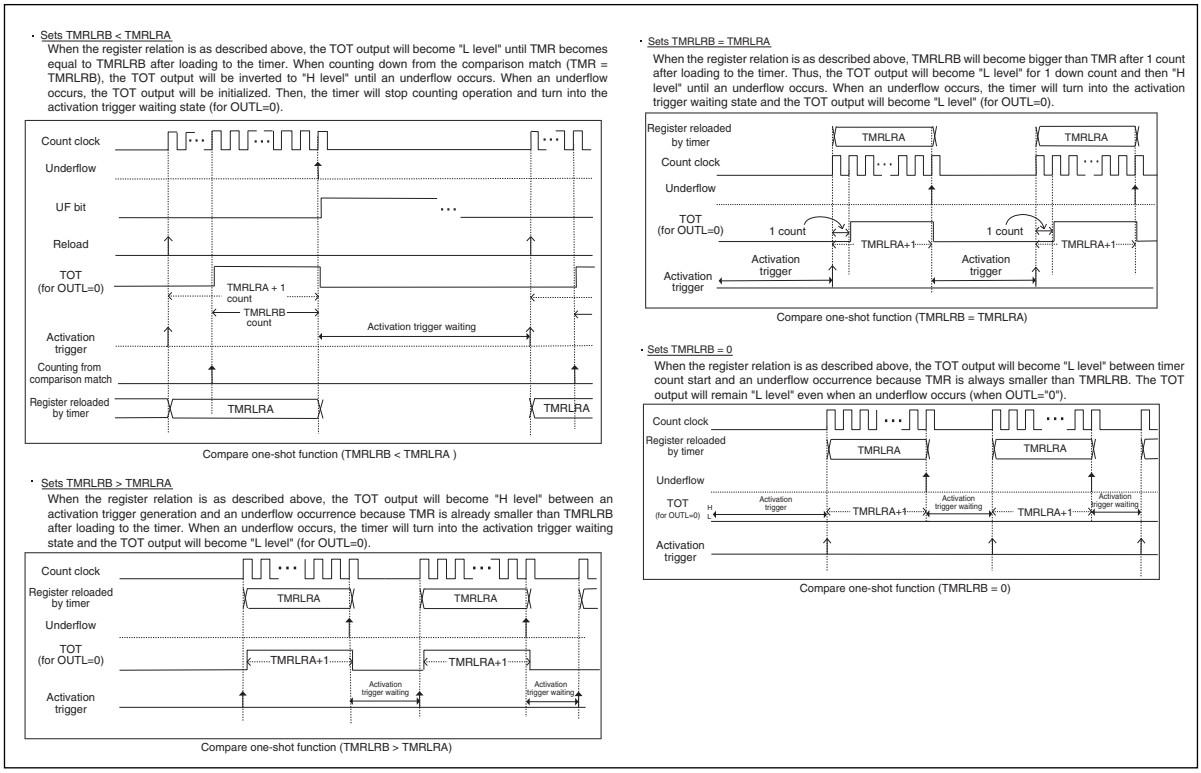
(For the compare operation in interval timer mode, bit7:EF bit of TMCSR register will not be set.))

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 TOT output.
- The timer stops with 0xFFFF.
- Timer is waiting for an activation trigger.

The operation of compare feature depends on the relationship between TMRLRA and TMRLRB.

Figure 38.5-11 Compare One-shot Operation



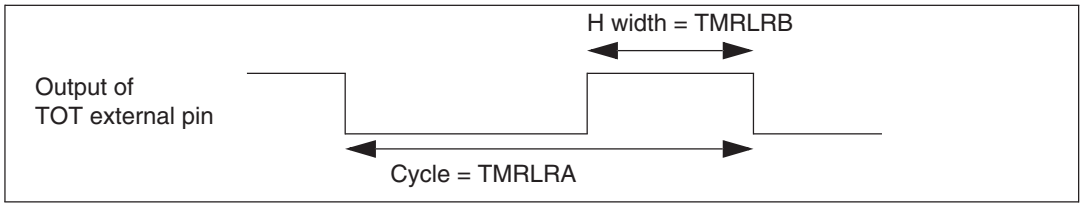
■ Compare Reload Operation

When bit15, bit14:MOD[1:0]=10, and bit4:RELD=1 of TMCSR register, the timer compares a counter value (TMR) to the value of TMRLRB for each downcount and if a compare matched (TMR = TMRLRB) is detected, a downcount starts and the TOT 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 downcount 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 TOT output inverted from "L level output" to "H level output".

When TMR + 1 = TMRLRB, TOT output will invert to "H level" (when OUTL=0).

Figure 38.5-12 TOT Interval, Pulse Width



From the start of a downcount to TMR = TMRLRB (while TMR is greater than or equal to TMRLRB), the following operation will be performed.

- TOT output continues to hold the initial value.
- Count continues.

When a downcount starts from TMR = TMRLRB, the following operation will be performed.

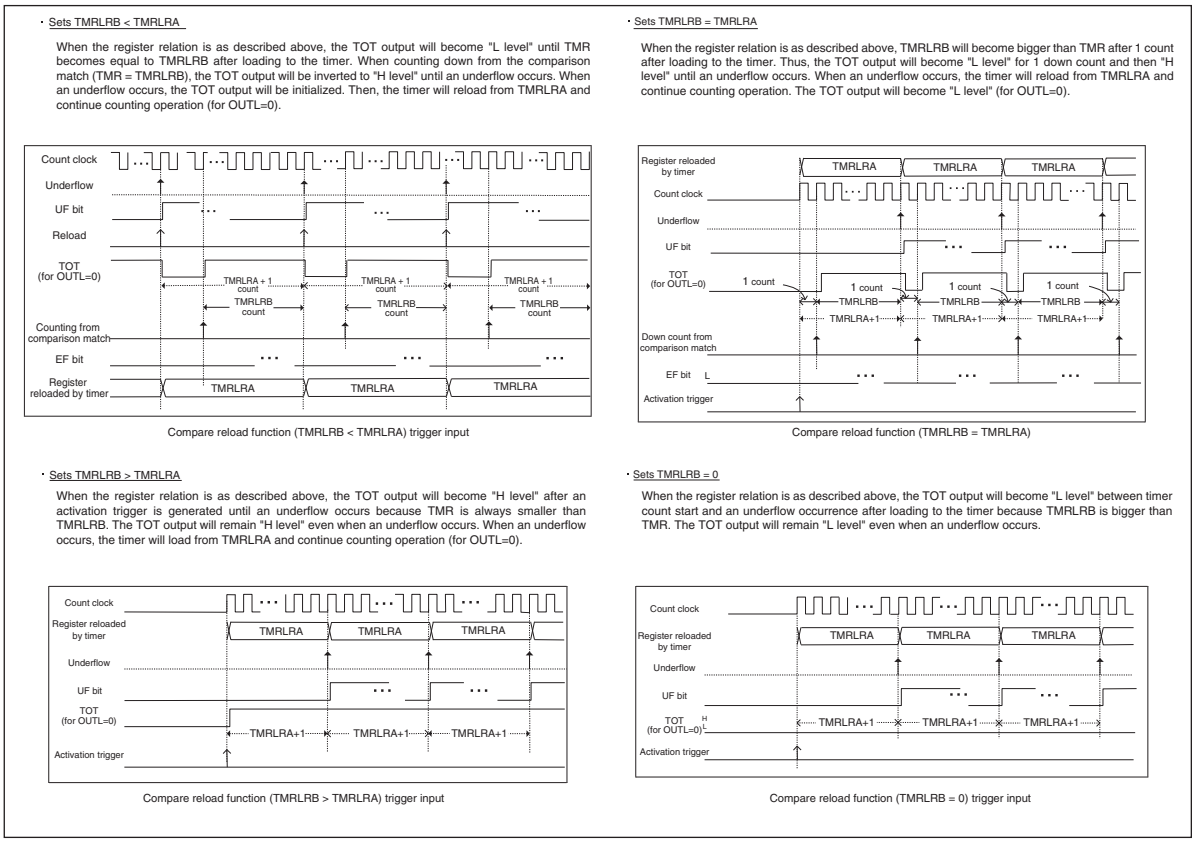
- Inverts TOT output.
- Count continues.
  (For the compare operation in interval timer mode, bit7:EF bit of TMCSR register will not be set.)

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 TOT output.
- Reloads a value from TMRLRA.
- The timer continues to count.

The operation of compare feature depends on the relationship between TMRLRA and TMRLRB.

Figure 38.5-13 Compare Reload Operation



■ Capture Mode

When bit15, bit14:MOD[1:0]=11 of TMCSR register, the timer will perform capture operation. When a retrigger occurs, captures the TMR value onto the TMRLRB register 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), it is necessary to 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. EF interrupt will also not be generated.

The timer operation and the TOT output will be the same for the single one-shot feature and the single reload feature.

(Note) In one-shot mode, retrigger does not initialize TOT.

Figure 38.5-14 Operation of Capture

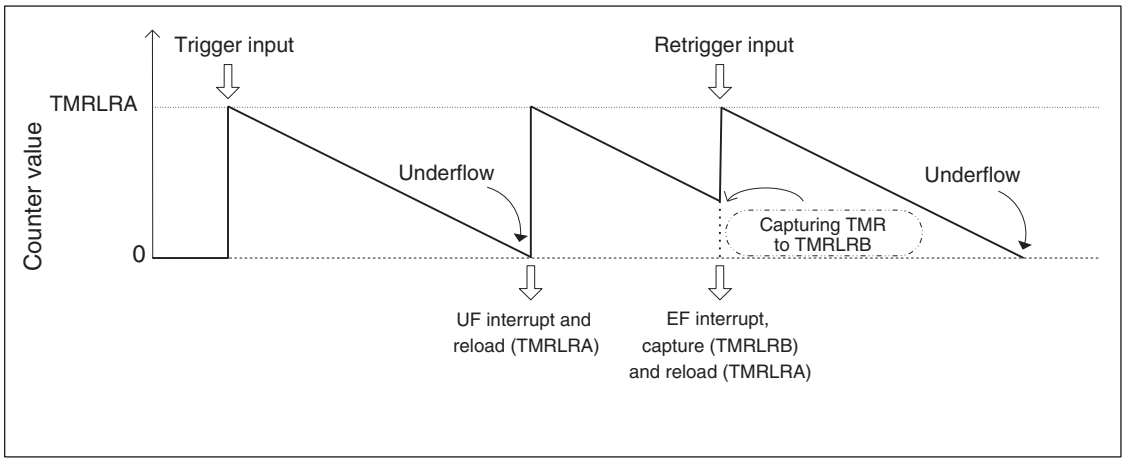


Figure 38.5-15 Flowchart of Trigger Input Features in Interval Timer Mode

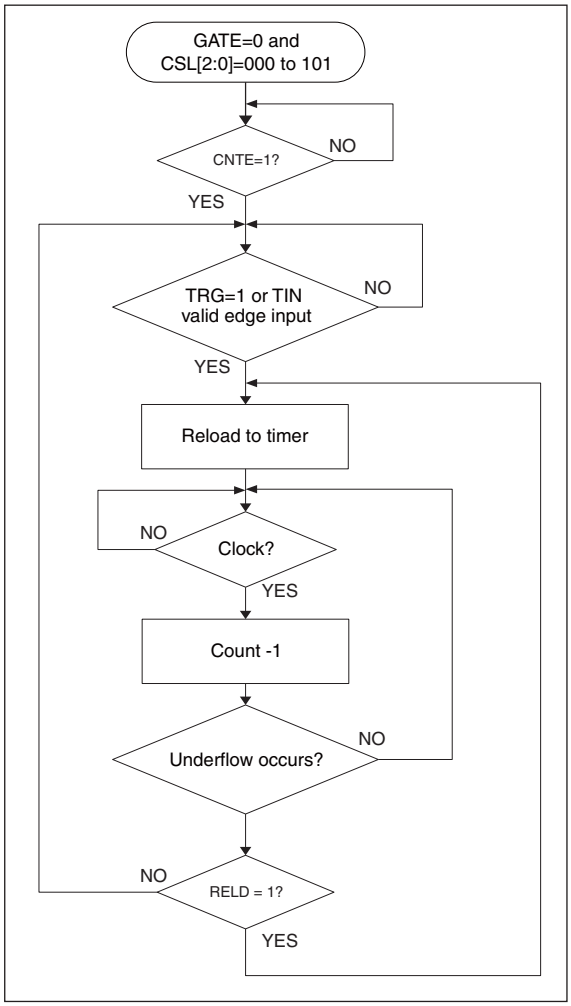
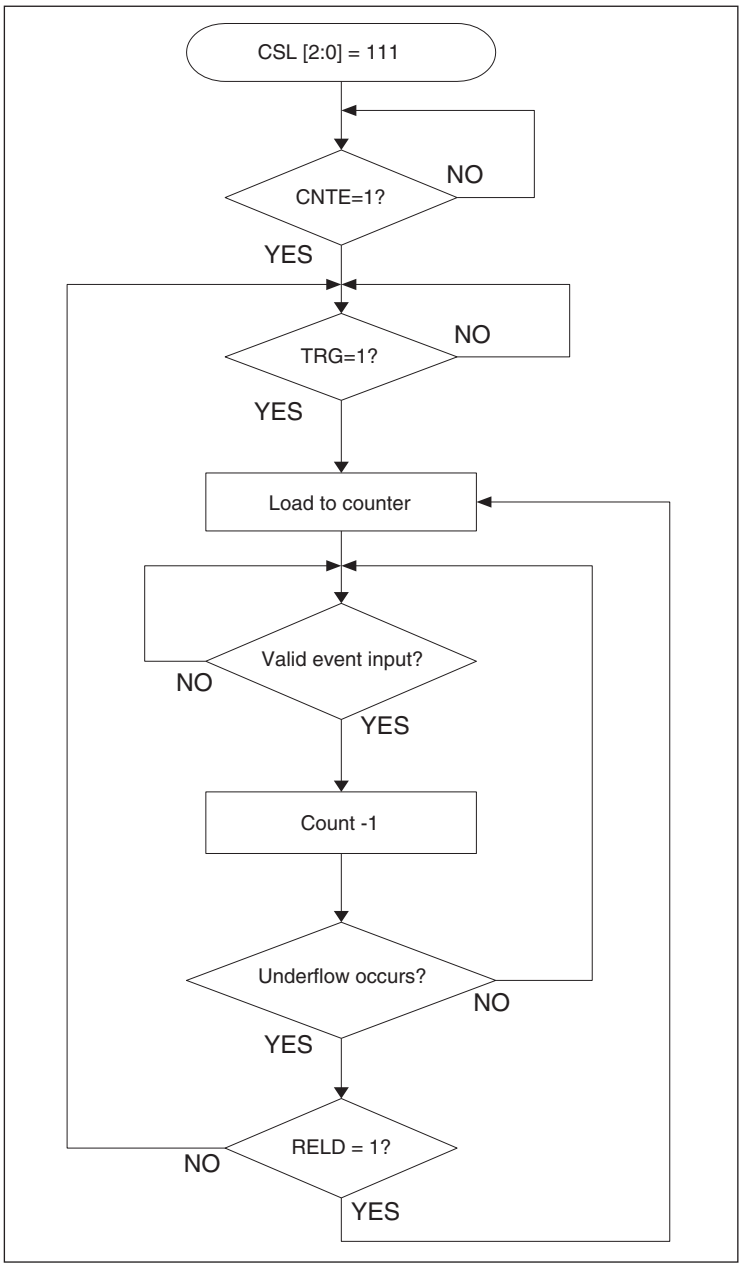


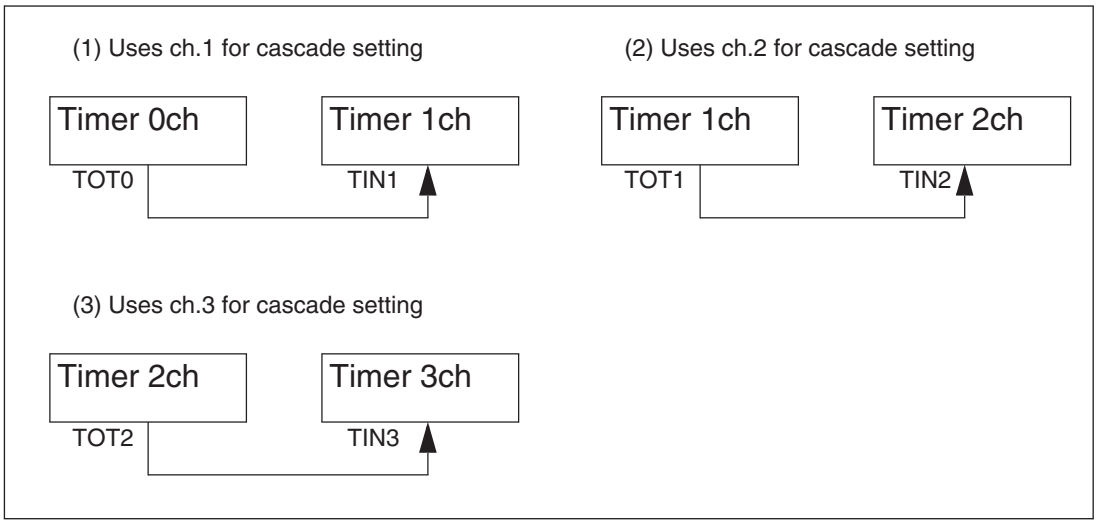
Figure 38.5-16 Flowchart in Event Counter Mode



38.5.4 Cascade Input

When you select cascade input (bit11 to 9:CSL[2:0]=110 of TMCSR register), you can use the timer's channel 0 output (TOT0) for the input for channel 1 (TIN1), channel 1 output (TOT1) for the input for channel 2 (TIN2), and channel 2 output (TOT2) for the input for channel 3 (TIN3).

Figure 38.5-17 Timer Input/Output in Cascade Input Configuration



38.5.5 Priority of Concurrent Operations

When two events to decide the timer operation occur simultaneously, the priority of deciding the operating state is shown below.

1. Writing to register
2. Trigger input
3. Underflow
4. Clock input

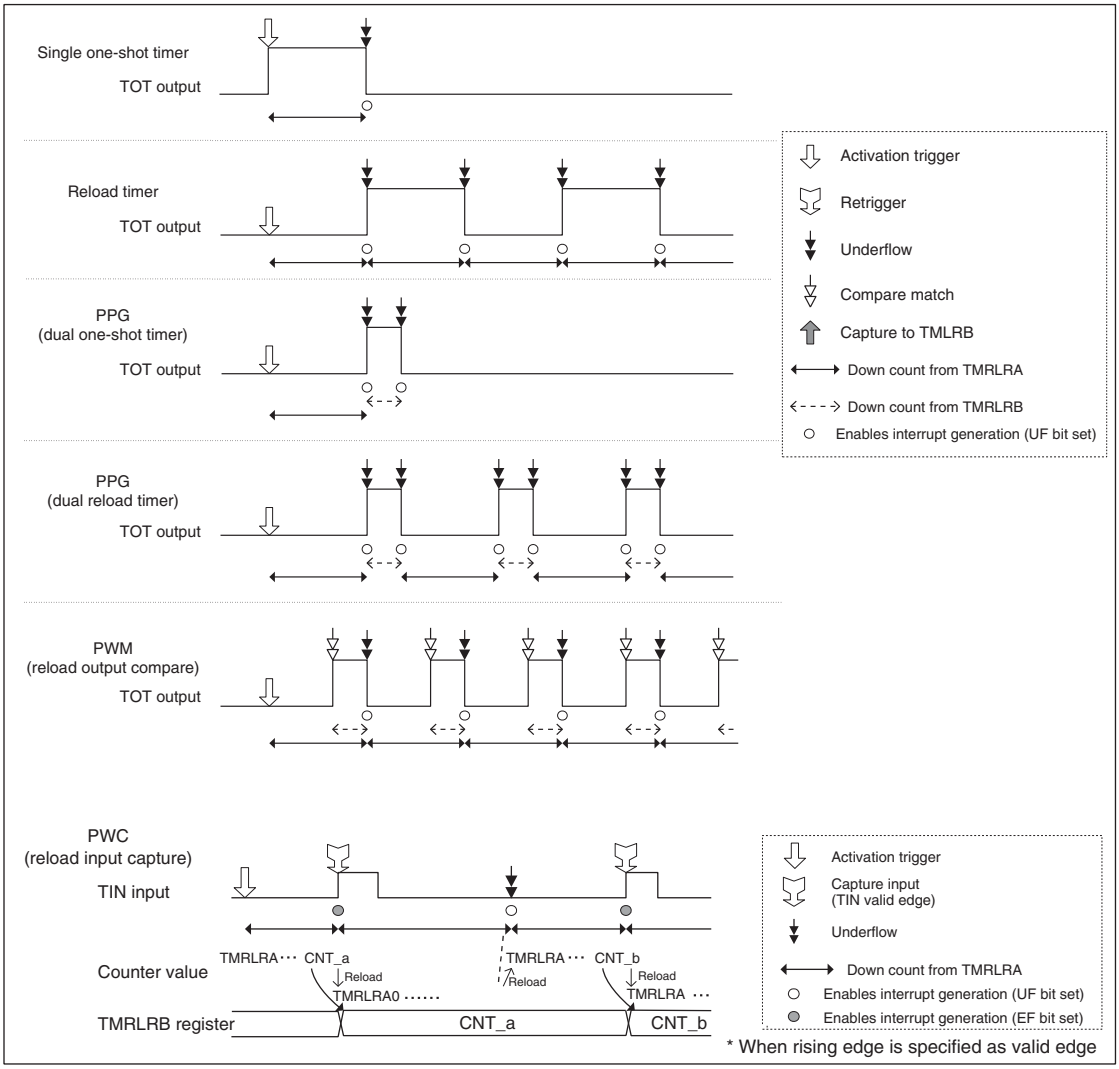
When a set of each flag by the timer operation and a clear of a flag by register write occur concurrently, the priority of deciding the operation is shown below.

1. Setting flag by the timer operation
2. Writing to a register for a clear of flag to the UF bit/EF bit

38.6 Application Note

This section shows the typical functions which can be realized with this timer.

Figure 38.6-1 Example



The following are some configurations for use of example figure above.

Table 38.6-1 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	–––

38.6.1 Single One-shot Timer

Single one-shot timer loads a value from TMRLRA register onto the counter and starts to decrement the counter (downcount operation) when trigger is input. When an underflow occurs, the counting stops.

TOT pin outputs "H level" in counting and when an underflow occurs it will output "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
- : No effect on operation
*1: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*2: TOT 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
- : No effect on 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*3: TOT 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

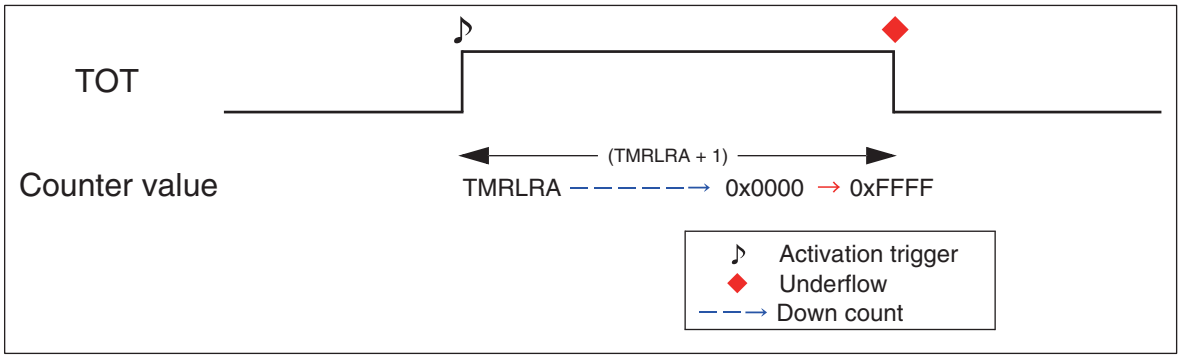
(3) When using TIN input as a trigger input

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
- : No effect on 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
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 TRG bit or a input of effective external edge from TIN pin)
• Input an effective level when you use TIN pin input as the gate input

Figure 38.6-2 Operation Example (OUTL=0)



38.6.2 Reload Timer

The reload timer loads from TMRLRA register onto the counter and repeats the downcount operation each time underflow occurs. The TOT outputs "L level" while the count is going on from the activation trigger to the occurrence of the first underflow, then the output will be inverted to "H level" at the timing of the occurrence of the first underflow, inverting the outputs whenever an underflow occurs. When a retrigger occurs, TOT 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
- : No effect on operation
*1: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*2: TOT 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
- : No effect on 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
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

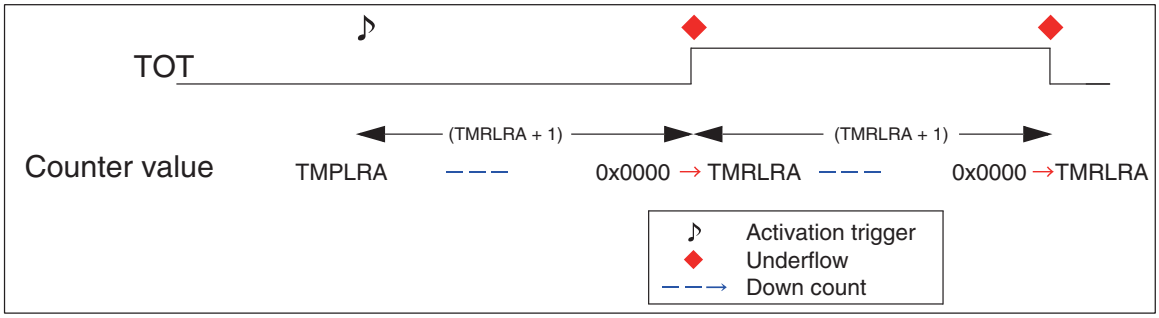
(3) When using TIN input as a trigger input

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
- : No effect on operation
*1: TIN effective edge setting TRGM[1:0]=00 ... Do 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
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 TRG bit or a input of effective external edge from TIN pin)
• Input an effective level when you use TIN pin input as the gate input

Figure 38.6-3 Operation Example (OUTL=0)



38.6.3 PPG

PPG is the feature which generates 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 downcount when an underflow occurs.

When RELD="0", Activation trigger → TMRLRA load → Downcount → underflow → TMRLRB load → Downcount → underflow stops the downcount.

When RELD="1", loads TMRLRA/TMRLRB alternatively and executes downcount whenever an underflow occurs, such as Activation trigger → TMRLRA load → Downcount → underflow → TMRLRB load → Downcount → underflow → TMRLRA load → Downcount → underflow → TMRLRB load and so on.

The TOT outputs "L level" while counting until the occurrence of an underflow caused by the downcount from TMRLRA, and outputs "H level" while counting until the occurrence of an underflow caused by the downcount from TMRLRB. When a retrigger occurs, TOT output returns to its initial value.

(Note) In one-shot mode, retrigger does not initialize TOT.

[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
- : No effect on operation
*1: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
* 2: TOT 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

- : No effect on 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
*: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
* 3: TOT 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

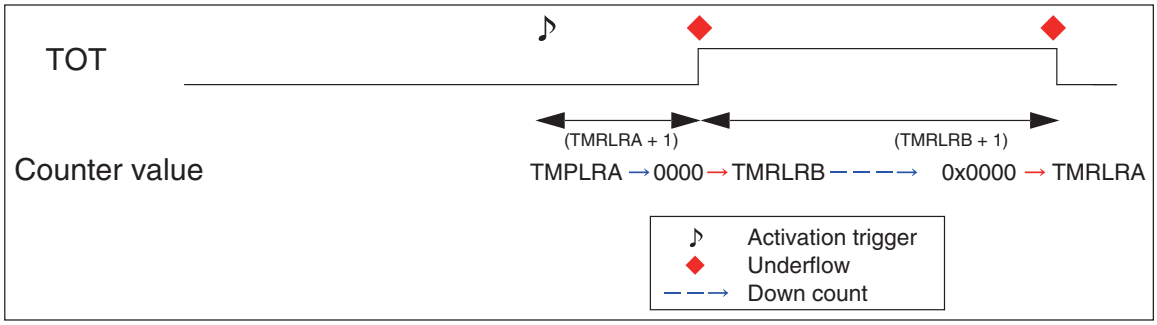
TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	*1	*2	0	-	*3	*4	*5	-	1	S		

(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
- : No effect on operation
*1: TIN effective edge setting TRGM[1:0]=00 ... Do 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*3: TOT 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)
- Input an effective level when you use TIN pin input as the gate input

Figure 38.6-4 Operation Example (OUTL=0)



38.6.4 PWM

PWM is the feature which generates output pulse by configuring the pulse interval and H width.

An activation trigger launches a load from TMRLRA to the counter and executes a downcount.

TOT outputs "L level" after an activation trigger and then outputs "H level" when the counter value becomes smaller than the TMRLRB value. When an underflow occurs, TOT output returns to its initial value. (When OUTL="0")

When RELD="0", Activation trigger → TMRLRA load → Downcount → "underflow" then the downcount stops.

When RELD="1", TMRLRA is loaded and the counters are decreased for each load whenever an underflow occurs, such as Activation trigger → TMRLRA load → Downcount → underflow → TMRLRA load → Downcount, 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
- : No effect on operation
*1: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*2: TOT 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 TOT output with L clip output, set to TMRLRB = "0".
To use TOT 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
- : No effect on 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*3: TOT 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 TOT output with L clip output, set to TMRLRB = "0".
To use TOT output with H clip output, set to TMRLRB = "TMRLRA + 1".

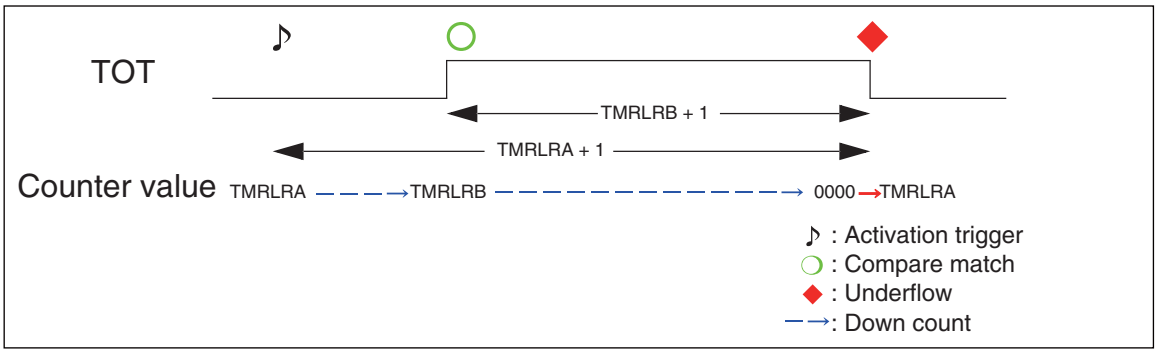
(3) When using TIN input as a trigger input

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
- : No effect on operation
*1: TIN effective edge setting TRGM[1:0]=00 ... Do 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
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*3: TOT 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 TOT output with L clip output, set to TMRLRB = "0".
To use TOT 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 38.6-5 Operation Example (OUTL=0)



38.6.5 PWC

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 downcount operation. A trigger input during a count enables the counter value at that time to be captured onto TMRLRB, which allows measuring the time interval between triggers to input.

[Configuration] To use the timer as PWC, configure as follows.

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
11	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)
S: Use at timer activation
- : No effect on operation
*1: TIN effective edge setting TRGM[1:0]=00 ... Do not detect external trigger edge
TRGM[1:0]=01 ... Rising edge
TRGM[1:0]=11 ... Both edges
*2: Count clock division setting CSL[2:0]= 000 ... Division of peripheral clock (PCLK) by 2
010 ... Division of peripheral clock (PCLK) by 8
011 ... Division of peripheral clock (PCLK) by 16
100 ... Division of peripheral clock (PCLK) by 32
101 ... Division of peripheral clock (PCLK) by 64
*3: TOT 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

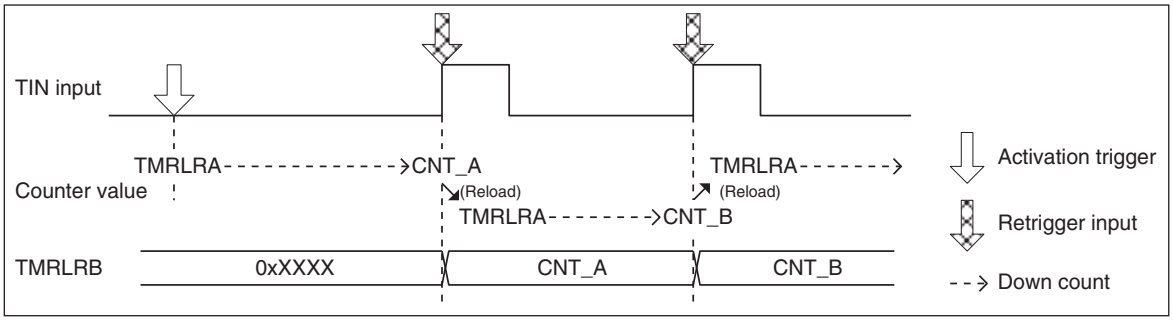
[Timer activation] Follow the steps below to activate the timer.

- Input an activation trigger (a write of "1" to TRG bit or an input of effective external edge from TIN pin)

  While downcounting, 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 38.6-6 Operation Example (TRGM=01)



CHAPTER 39 UP/DOWN COUNTER

39.1 Overview

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 "$FFFF_H$".

The MB91580M/S series incorporate 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.

39.2 Features

- 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.

Code: FG20-1v2-91580MS-1-E

Table 39.2-1 lists the up/down counter operating modes.

Table 39.2-1 Up/Down Counter Operating Modes

Operating mode	Count timing	Counting direction
Timer mode	Internal clock	Counting down
Up/down count mode	External clock	Counting up/counting down
Phase difference count mode (multiply-by-two/ multiply-by-four)	Phase of the input signal from an external signal input pin	Counting up/counting down

- Reload compare function: You can select one of the following three:
  - **Compare function**

    The compare function clears the counter value 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.

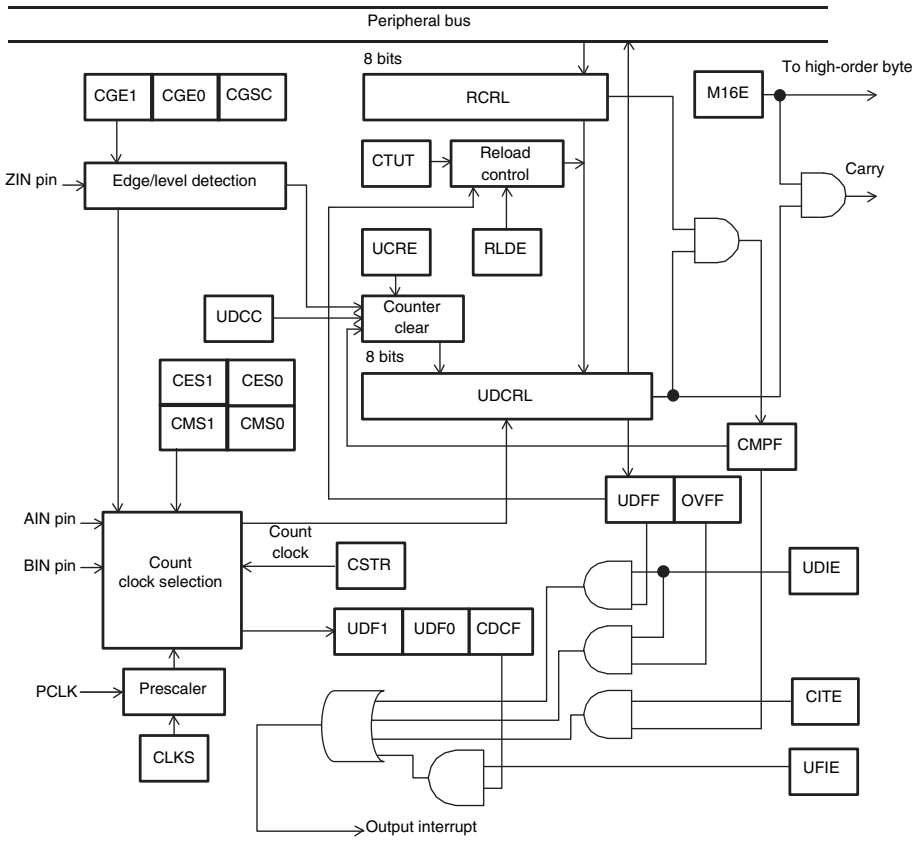
39.3 Configuration

This section shows the up/down counter configuration.

■ Block Diagram of the Up/Down Counter

Figure 39.3-1 shows the up/down counter block diagram using ch.0 as an example.

Figure 39.3-1 Block Diagram of the Up/Down Counter



RCRL : Reload compare register low-order (RCRL0, RCRL1)
UDCRL: Up/down count register low-order (UDCRL0, UDCRL1)

- Reload compare register (RCR)
  This register sets reload and compare values of the up/down counter.
  As shown below, this counter consists of high-order 8 bits and low-order 8 bits.
  To use the register in 8-bit mode, use the low-order side.
  - Reload compare register high-order (RCRH)
  - Reload compare register low-order (RCRL)
- Up/down count register (UDCR)
  This register operates as the counter for the up/down counter.
  As shown below, this counter consists of high-order 8 bits and low-order 8 bits.
  To use the register in 8-bit mode, use the low-order side.
  - Up/down count register high-order (UDCRH)
  - Up/down count register low-order (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 39.3-1 lists the clocks used for the up/down counter.

Table 39.3-1 Clocks Used for the Up/Down Counter

Clock name	Description	Remarks
Operating 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

39.4 Registers

This section explains the configuration and functions of the registers used for the up/down counter.

■ Correspondence between Pins and Channels

The following table shows the correspondence between channels and pins.

Channel number	External signal input pin		
0	AIN0	BIN0	ZIN0
1	AIN1	BIN1	ZIN1

■ Register Map

The following table lists the up/down counter register map.

Address	Register				Register function
	+0	+1	+2	+3	
0x02F0	RCRH0	RCRL0	UDCRH0	UDCRL0	Reload compare register high-order 0 Reload compare register low-order 0 Up/down count register high-order 0 Up/down count register low-order 0
0x02F4	CCR0		Reserved	CSR0	Counter control register 0 Counter status register 0
0x02F8	RCRH1	RCRL1	UDCRH1	UDCRL1	Reload compare register high-order 1 Reload compare register low-order 1 Up/down count register high-order 1 Up/down count register low-order 1
0x02FC	CCR1		Reserved	CSR1	Counter control register 1 Counter status register 1

39.4.1 Reload compare register (RCR0, RCR1)

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 $02F0_H$ (Access: Half-word, Word)**
- **RCRH1: Address $02F8_H$ (Access: Half-word, Word)**

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 values
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attributes

- **RCRL0: Address $02F1_H$ (Access: Byte, Half-word, Word)**
- **RCRL1: Address $02F9_H$ (Access: Byte, Half-word, Word)**

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 values
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attributes

As shown below, this register consists of high-order byte and 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.

39.4.2 Up/down count register (UDCR0, UDCR1)

This register operates as the counter for the up/down counter. The counter value can be checked by reading these registers.

- **UDCRH0: Address $02F2_H$ (Access: Half-word, Word)**
- **UDCRH1: Address $02FA_H$ (Access: Half-word, Word)**

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

- **UDCRL0: Address $02F3_H$ (Access: Byte, Half-word, Word)**
- **UDCRL1: Address $02FB_H$ (Access: Byte, Half-word, Word)**

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 values
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attributes

As shown below, this register consists of high-order byte and 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 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 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 counter control register (CCR) (M16E=0), the low-order side of up/down count register (UDCRL) must be read.

39.4.3 Counter control register (CCR0, CCR1)

This register controls the up/down counter operations.

- **CCR0: Address $02F4_H$ (Access: Byte, Half-word)**
- **CCR1: Address $02FC_H$ (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	bit
M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	
0	0	0	0	0	0	0	0	Initial values
R/W	R (RM1), W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
Reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	
0	0	0	0	1	0	0	0	Initial values
R0,W0	R0,W	R/W	R/W	R1,W	R/W	R/W	R/W	Attributes

[bit15] M16E (16-bit mode selection bit)

Uses the up/down counter in the 8-bit mode or 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 (Counter 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)

Uses the peripheral clock (PCLK) divided by the division ratio (set by this bit) as the count clock when the timer mode is selected.

Write value	Description
0	Peripheral clock (PCLK) divided by 2
1	Peripheral clock (PCLK) divided by 8

(Note) This bit is valid only if the timer mode has been set with the CMS1 and CMS0 bits (CMS1, CMS0=00). This bit setting is ignored if another operating mode has been selected.

[bit11, bit10] CMS1, CMS0 (Operating mode selection bits)

Select an operating mode of up/down counter as follows.

- Timer mode
  The time is counted down in synchronous with the count clock.
- Up/down count mode
  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	Operating mode
0	0	Timer mode
0	1	Up/down count mode
1	0	Phase difference count mode (multiply-by-two)
1	1	Phase difference count mode (multiply-by-four)

[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)

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)

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.

(Notes) 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 "0" to the UDCC bit. (Software-triggered clear)

[bit4] RLDE (Reload enable 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 "0000H".

UDCC	Read	Write
0	"1" is read.	The counter is cleared.
1		Ignored

[bit2] CGSC (Counter clear/gate selection 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 bit)

Selects an effective edge or an effective level of signal at the ZIN pin. The meaning of this bit 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

39.4.4 Counter Status Register (CSR0, CSR1)

This register is used to check the status of the up/down counter and control interrupt requests.

- **CSR0: Address $02F7_H$ (Access: Byte)**
- **CSR1: Address $02FF_H$ (Access: Byte)**

7	6	5	4	3	2	1	0	bit
CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R (RM1), W	R (RM1), W	R (RM1), W	R,WX	R,WX	Attributes

[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

(Notes) 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 underflowed.

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	Counting down
1	0	Counting up
1	1	Counting up and down at the same time

39.5 Interrupt

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 39.5-1 shows the correspondence between operating modes and interrupt requests.

Table 39.5-1 Correspondence between Operating Modes and Interrupt Requests

Interrupt request	Timer mode	Up/down count mode	Phase difference count mode (multiply-by-two/ multiply-by-four)
Counting direction change interrupt request	×	Yes	Yes
Compare result match interrupt request	Yes	Yes	Yes
Overflow interrupt request	×	Yes	Yes
Underflow interrupt request	Yes	Yes	Yes

Table 39.5-2 shows interrupts that can be used for the up/down counter.

Table 39.5-2 Up/Down Counter Interrupts

Interrupt request	Interrupt request flag	Enabling of interrupt request	Clearing of interrupt request
Counting direction change interrupt request	CDCF=1 in CCR	CFIE=1 in CCR	Write "0" to the CDCF bit in CCR
Compare result match interrupt request	CMPF=1 in CSR	CITE=1 in CSR	Write "0" to the CMPF bit in CSR
Overflow interrupt request	OVFF=1 in CSR	UDIE=1 in CSR	Write "0" to the OVFF bit in CSR
Underflow interrupt request	UDFF=1 in CSR	UDIE=1 in CSR	Write "0" to the UDFF bit 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 "39.6 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 C. 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 "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

39.6 Operation and Setting Procedure Examples

This section explains the operation of the up/down counter. An example is also given to set operating state.

■ Overview

● Counter mode

Depending on the setting, the up/down counter can be used as a 16-bit up/down counter or an 8-bit up/down counter.

Set the counter mode in the M16E bit in the counter control register (CCR).

- 8-bit mode (M16E=0)

  Only the up/down count register low-order bit (UDCRL) is used. Write the reload and compare values only to the reload compare register low-order bit (RCRL) using byte access.
- 16-bit mode (M16E=1)

  Both the high-order and low-order bytes of the up/down count register (UDCR) are used. Write the reload and compare values to the reload compare register (RCR) using half-word access.

● Operating mode

One of the following three modes (four types) can be selected as the operating mode of the up/down counter using the CMS1 and CMS0 bits of the counter control register (CCR).

- Timer mode (CMS1, CMS0=00)

  The counter decrements from a preset value in synchronization with the count clock.

  The count clock is generated by dividing the peripheral clock (PCLK) by 2 or 8 using the prescaler.
- Up/down count mode (CMS1, CMS0=01)

  The counter increments or decrements based on signals supplied from the external signal input pin.
- Phase difference count mode (multiply-by-two) (CMS1, CMS0=10)/Phase difference count mode (multiply-by-four) (CMS1, CMS0=11)

  The counter increments or decrements based on phase differences of signals supplied from the external signal input pin. This mode is suitable for counting of encoders such as motors because it enables high-precision counting of rotation angles and number of rotations and detection of the rotation direction by entering the encoder A-phase to the AIN pin, B-phase to the BIN pin, and Z-phase to the ZIN pin.

■ Available Functions

● Reload/compare functions

For the 8/16-bit up/down counter, the reload and compare functions can be enabled and disabled using the RLDE and UCRE bits of the counter control register (CCR).

- Reload function

  When an underflow occurs during countdown, the value set in the reload compare register (RCR) is reloaded and counting down is restarted. For the operations, see "■ Counting" in Section "39.6.1 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 Section "39.6.2 Operation in Up/down Count Mode".

  This function is not available in timer mode.

- Reload compare function

  This function is a combination of the reload and compare functions. The counter decrements and increments between "0000_H" and a value set in the reload compare register (RCR), enabling counting in any range. See "■ Counting" in Section "39.6.2 Operation in Up/down Count Mode".

  This function is not available in timer mode.

  Table 39.6-1 shows the setting method for the reload/compare functions.

Table 39.6-1 Setting Method for Reload/Compare Functions

RLDE bit	UCRE bit	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

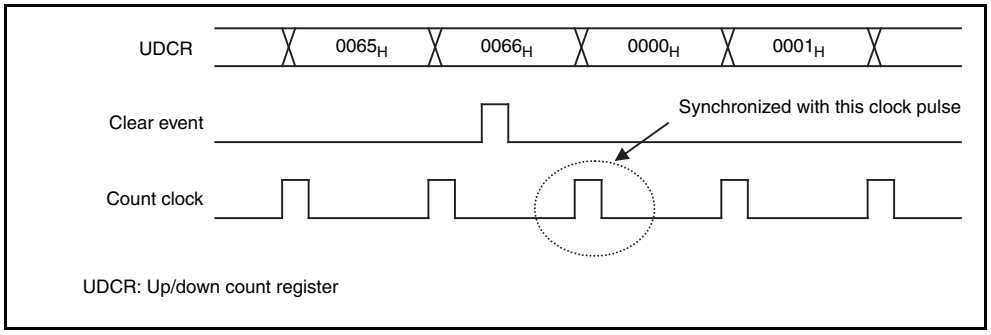
The counter value is cleared to "0000_H" in one of the following events.

- 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 "FFH" in 8-bit mode).

The time the counter value is cleared to "0000_H" depends on the up/down counter operating status as follows.

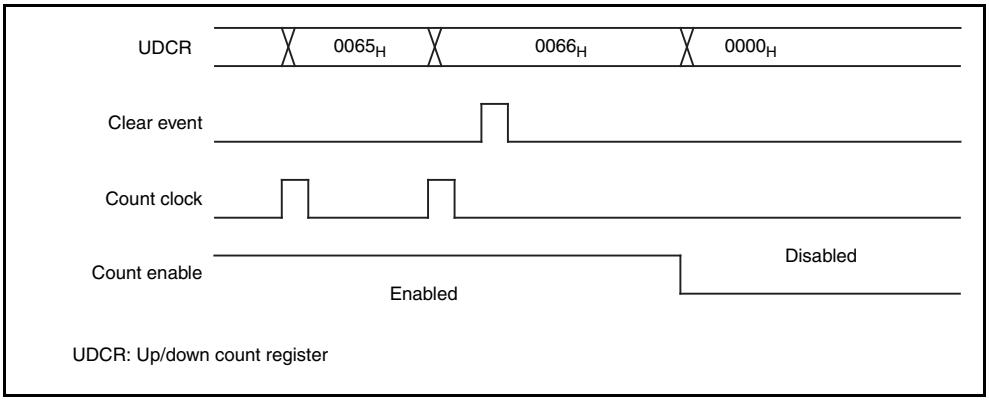
- If a clear event occurs during counting, the counter value will be cleared in synchronization with the count clock.
  Figure 39.6-1 shows clear event occurrence timing.

Figure 39.6-1 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 counter value will be cleared when the up/down counter stops.
  Figure 39.6-2 shows clear event occurrence timing.

Figure 39.6-2 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 time 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.

39.6.1 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 Section "39.4.3 Counter control register (CCR0, CCR1)".

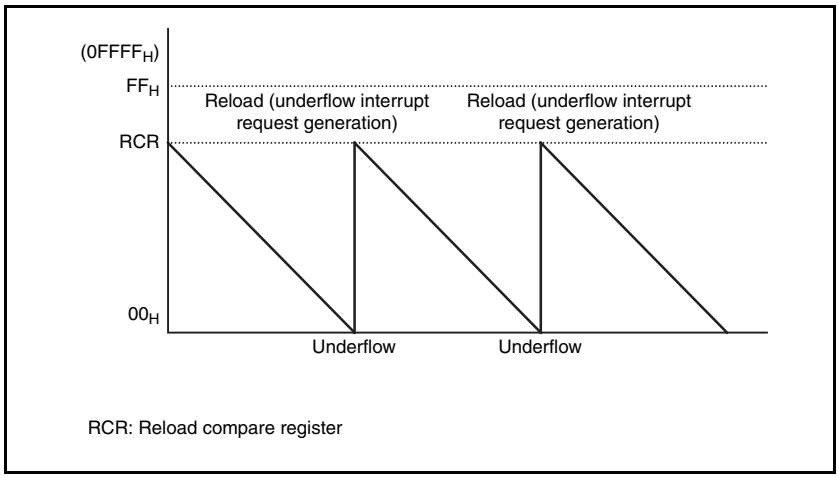
(Note) The minimum pulse width required at the ZIN pin is 2T (T is the cycle of the peripheral clock (PCLK)).

● 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 39.6-3 shows the operation performed when the reload function is in use.

Figure 39.6-3 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 of the reload compare register (RCR) is reloaded, the CMPF bit of the counter status register (CSR) also changes to "1".

39.6.2 Operation in Up/down Count Mode

This section explains operation in up/down count mode.

■ Overview

In this mode, the up/down counter counts up/down in synchronization 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 Section "39.4.3 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 Section "39.6.1 Operation in Timer Mode".

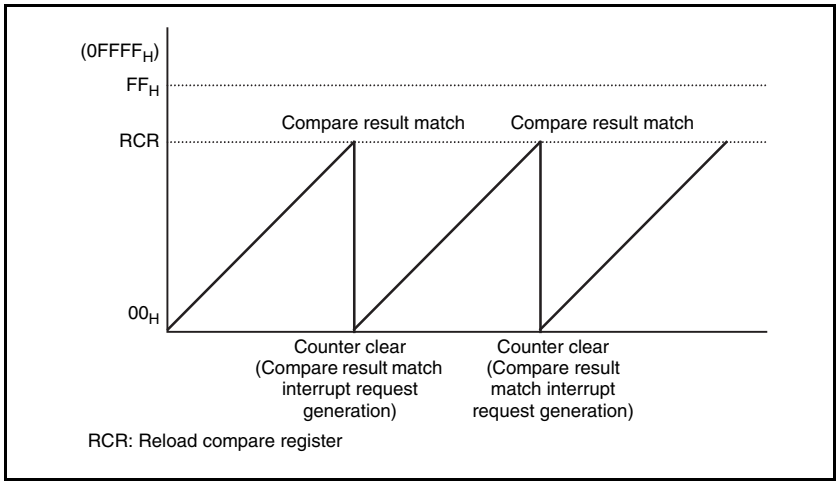
● Operation performed when the compare function is in use

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 39.6-4 shows the operation performed when the compare function is in use.

Figure 39.6-4 Operation Performed When the Compare Function is in Use



(Notes) If the compare function is in use, the up/down counter value will be cleared to "0000_H" when one of the following conditions is fulfilled.

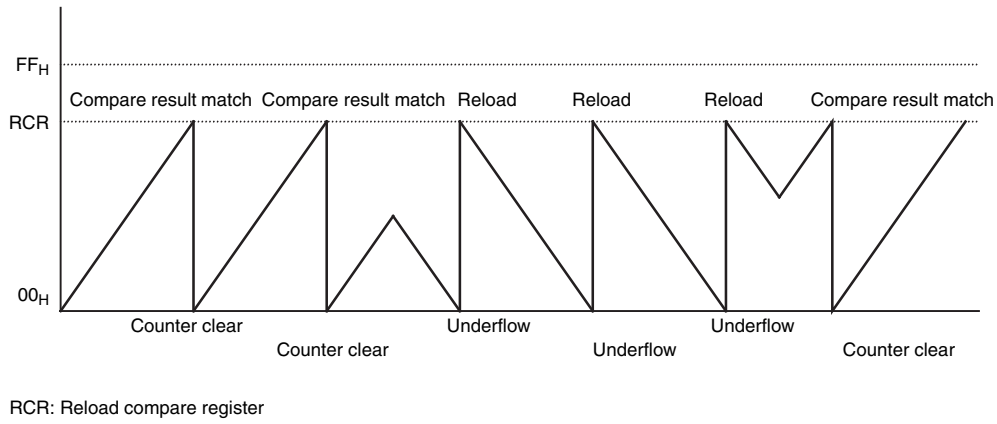
- The up/down counter value matches the value set in the reload compare register (RCR) (compare result match) and further, the next counting up operation is performed. However, a comparison result match does not cause clearing of the up/down counter value if one of the following conditions is fulfilled:
- The next operation is counting down.
- The up/down counter is inactive.

● Operation performed when the reload compare function is in use

The reload function is used at counting down and the compare function is used at counting up.

Figure 39.6-5 shows operation performed when the reload compare function is in use.

Figure 39.6-5 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 39.6-2 lists the counting directions indicated with the UDF1 and UDF0 bits.

Table 39.6-2 Correspondence between UDF1 and UDF0 Bits and Counting Directions

UDF1	UDF0	Counting 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".

39.6.3 Operation in the Phase Difference Count Mode (Multiply-by-Two)

This section explains 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 39.6-3 lists the count methods.

Table 39.6-3 Count Methods

BIN pin	AIN pin	Counting 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

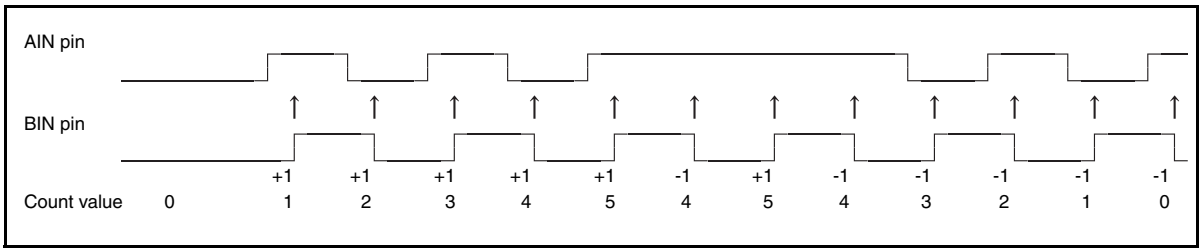
■ Counting

● 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 39.6-6 shows the operation in the phase difference count mode (multiply-by-two).

Figure 39.6-6 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 Section "39.4.3 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 Section "39.6.1 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 Section "39.6.2 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 Section "39.6.2 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 Section "39.6.2 Operation in Up/down Count Mode".

39.6.4 Operation in the Phase Difference Count Mode (Multiply-by-Four)

This section explains 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 39.6-4 lists the count methods.

Table 39.6-4 Count Methods

Edge detection pin	Detection edge	Level check pin	Input level	Counting 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

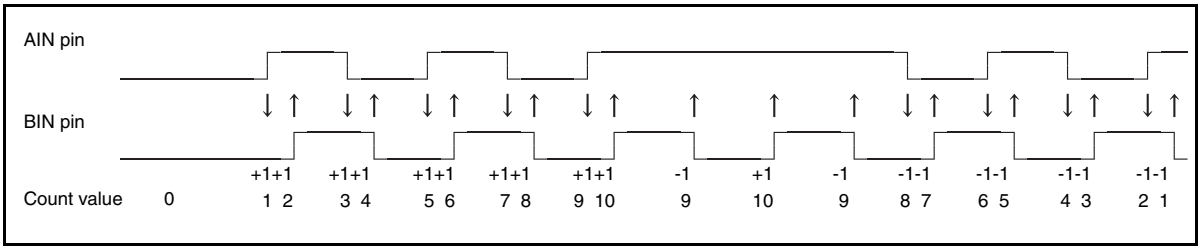
■ Counting

● 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 39.6-7 shows the operation in the phase difference count mode (multiply-by-four).

Figure 39.6-7 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 Section "39.4.3 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 Section "39.6.1 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 Section "39.6.2 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 Section "39.6.2 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 Section "39.6.2 Operation in Up/down Count Mode".

CHAPTER 40 BASE TIMERS

40.1 Overview

This product includes base timers for two 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

40.2 Features

This product includes base timers for two channels. Each channel selects and uses appropriate ones of the following functions:

■ 16/32-bit Reload Timer

A base timer can be used as a 16/32-bit reload timer. The 16/32-bit reload timer is a timer that decrements 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 modes:
  - Reload mode: 3In this mode, when the down counter underflows, the preset value (cycle) is reloaded to allow the timer to repeat 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 clocks (peripheral clock): Peripheral clock (PCLK) frequency divided by 1, 4, 16, 128, 256, 512, 1024, and 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

Code: FM10-3v1-91580MS-1-E

■ 16-bit PWM 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 modes:
  - Reload mode: In this mode, when the 16-bit down counter underflows, the preset cycle is reloaded to allow the timer to repeat 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 clocks (peripheral clock): Peripheral clock (PCLK) frequency divided by 1, 4, 16, 128, 256, 512, 1024, and 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: Detection of rising edge, falling edge, and both edges
- **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

■ 16/32-bit PWC 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 modes:
  - Single measurement mode: In this mode, measurement is conducted only once.
  - Continuous measurement mode: In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
- **Count clock:** You can select one of the internal (peripheral) clocks obtained by dividing the frequency of the peripheral clock (PCLK) by eight types.
  - Peripheral clock (PCLK) frequency divided 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: Time from the detection of a rising edge to the detection of the next rising edge
  - Falling edge interval: Time 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:
  --- Time from the detection of a rising edge to the detection of the falling edge
  --- Time 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

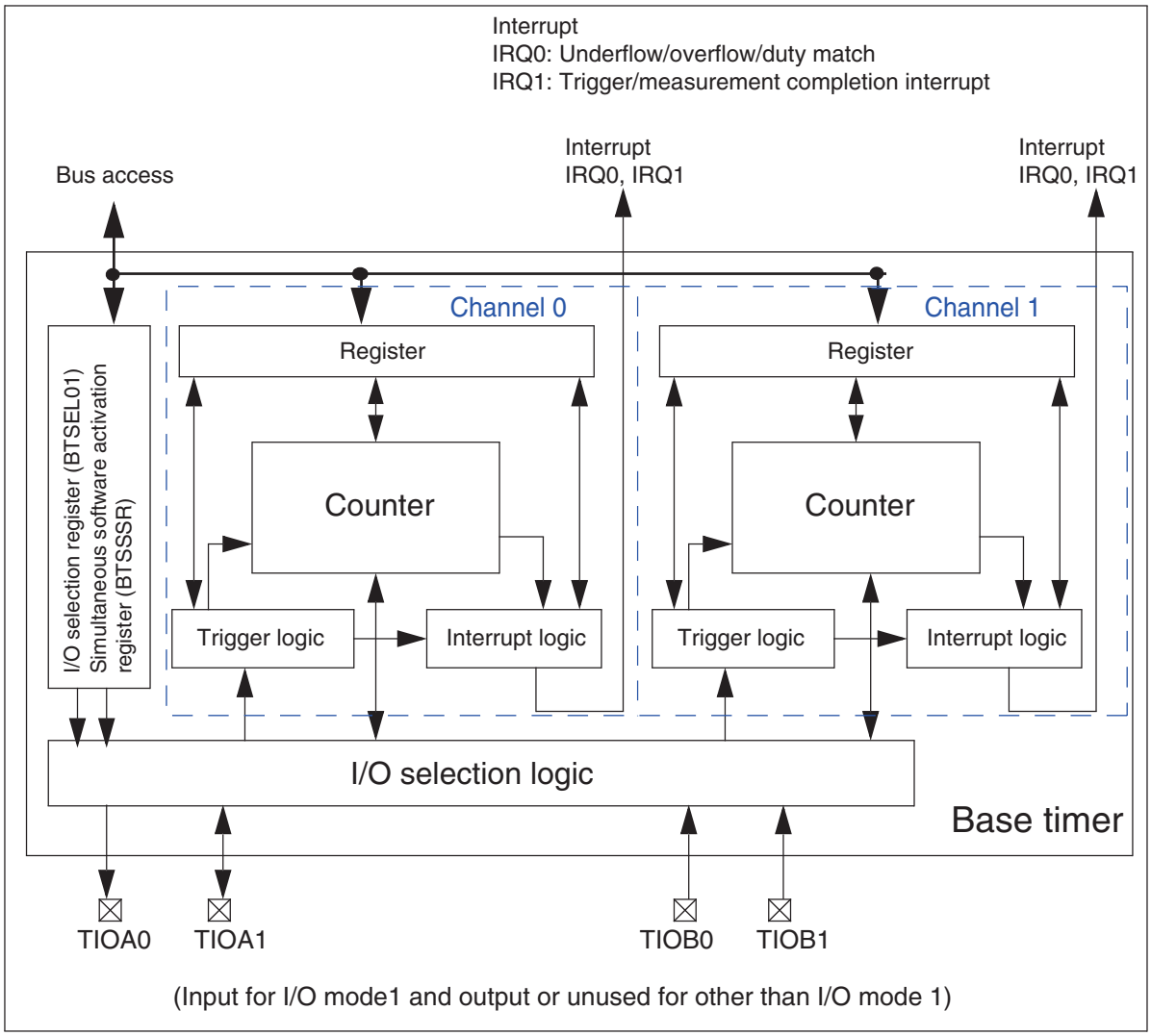
■ 16-bit PPG Timer

The 16-bit PPG timer, PPG standing for Programmable Pulse Generator, is a timer that generates a waveform with a desired pulse width.

- **I/O mode:** You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.
- **Operation mode:** You can select one of the following two modes:
  - Reload mode: A sequence of "L"-level and "H"-level signals (consecutive pulses) is output.
  - One-shot mode: A string of one "L"-level signal and one "H"-level signal (single pulses) is output.
- **Count clock:** You can select one of eight internal (peripheral) clocks and three external clocks (ECK signals).
  - Internal clocks (peripheral clock): Peripheral clock (PCLK) frequency divided by 1, 4, 16, 128, 256, 512, 1024, and 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 H width setting reload register (BT0PRLH/BT1PRLH).
  - IRQ1: When a 16-bit PPG timer activation trigger is detected

40.3 Configuration

Figure 40.3-1 Block Diagram of Base Timers (Overview)



40.4 Registers

■ List of Base Addresses (Base-addr) and External Pins

Table 40.4-1 Table of Base Addresses (Base_addr) and External Pins

Channel number	Base address	External pin
0	0x0080	TIOA0, TIOA1, TIOB0, and TIOB1 are assigned based on the BTSEL01 register setting.
1	0x0090	

■ Register Map

Table 40.4-2 Register Map

Address	Register				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 2 0 [Reload timer] Status control register 0 [PWM] Status control register 0 [PPG] Status control register 0 [PWC] Status control register 0
0x0088	[Reload timer] BT0PCSR [PWM] BT0PCSR [PPG] BT0PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT0PDUT [PPG] BT0PRLH [PWC] BT0DTBF		[Reload timer] Cycle setting register 0 [PWM] Cycle setting register 0 [PPG] L width setting reload register 0 [PWM] Duty setting register 0 [PPG] H width setting reload register 0 [PWC] Data buffer register 0
0x008C	Reserved				
0x0090	[Common] BT1TMR		[Common] BT1TMCR		[Common] Timer register 1 [Common] Control register 1
0x0094	[Common] BT1TMCR 2	[Reload timer] BT1STC [PWM] BT1STC [PPG] BT1STC [PWC] BT1STC	Reserved		[Common] Control register 2 1 [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

40.4.1 Common Registers

The registers described here are common to various operations.

40.4.1.1 Timer Register 0, 1: BT0TMR, BT1TMR (Base Timer 0/1 Timer Register)

These registers are used to read the counter value on the timer. The registers are only valid when its content represents a reload, PWM, or PPG timer. The value read from the registers is indefinite if a PWC timer is read. For information on the values that will be read, see the section of Operation Description.

(Note) These registers must be accessed in 16-bit mode.

- **BT0TMR, BT1TMR: Address 0080_H, 0090_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R,WX	R,WX	• • •	R,WX	R,WX	R,WX	Attribute

40.4.1.2 Timer Control Register 0, 1: BT0TMCR, BT1TMCR (Base Timer 0/1 TiMer Control Register), BT0TMCR2, BT1TMCR2 (Base Timer 0/1 TiMer Control Register2)

These registers are used to variously configure, start, and stop a base timer and to issue software triggers.

(Notes)
- If you need to change the FMD[2:0] setting, once reset it to FMD[2:0]=000, and then set FMD[2:0] to the desired value.
- Reserved bits must be set to "0".
- If you want to set bits of these registers except for the software trigger (STRG) bit, proceed as follows:
  (1) Once stop operation by writing FMD[2:0]=000 or CTEN=0.
  (2) Write desired values to the timer function selection bits (FMD[2:0]) and other bits.
- When writing to the software trigger bit (STRG), be careful not to clear other bits.
- Since FMD[2:0]=000 specifies reset mode, you cannot set other bits when setting FMD[2:0]=000.
- These registers must be accessed in 16 -bit mode.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0TMCR, BT1TMCR: Address 0082_H, 0092_H (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
Reserved	CKS[2:0]			[PWM - PPG] RTGEN [Others] Reserved	[PWM - PPG] PMSK [PWC] EGS[2] [Others] Reserved	EGS[1:0]		
0	0	0	0	0	0	0	0	Initial value
R/W R0,WX(*3)	R/W	R/W	R/W	R/W R0,WX(*1)	R/W R0,WX(*1)	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
[Reload timer - PWC] T32 [Others] Reserved	FMD[2:0]			[Reload timer - PWM - PPG] OSEL [Others] Reserved	MDSE	CTEN	STRG	
0	0	0	0	0	0	0	0	Initial value
R/W R0,WX(*1) R0,WX(*2)	R/W	R/W	R/W	R/W R/W0(*1)	R/W	R,W	R0,W R0,W(*1)	Attribute

(*1) Attribute assumed for "Reserved"
(*2) Attribute assumed for a 32-bit timer serving an odd-numbered channel
(*3) Attribute assumed for a 32-bit timer serving an odd-numbered channel or for a 16/32-bit PWC timer

- **BT0TMCR2-BT1TMCR2: Address 0084_H, 0094_H (Access: Byte)**

15	14	13	12	11	10	9	8	bit
-	-	-	-	-	-	-	CKS3	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	Attribute

[BTxTMCR2: bit15 to bit9, BTxTMCR: bit15] Reserved

There will be no effect on the operation by writing to these bits.

[BTxTMCR2: bit8, BTxTMCR: bit14 to bit12] CKS3-CKS0 (Clock Select):Count clock selection bits

Select a count clock.

CKS3 to CKS0	Description	
	Clock source	Description
0000	Internal clock (peripheral clock (PCLK))	Frequency divided by 1
0001		Frequency divided by 4
0010		Frequency divided by 16
0011		Frequency divided by 128
0100		Frequency divided by 256
0101	[Reload timer PWM PPG] External clock (ECK signal) [PWC] Disabled	Rising edge
0110		Falling edge
0111		Both edges
1000	Internal clock (peripheral clock (PCLK))	Frequency divided by 512
1001		Frequency divided by 1024
1010		Frequency divided by 2048
Other	Disabled	

In the PWC mode, settings of 0101, 0110, and 0111 are prohibited.

[PWM/PPG] [BTxTMCR: 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 (BT0PCSR/BT1PCSR)/L width setting reload register(BT0PRLL/BT1PRLL) by reloading it to the 16-bit down counter.

RTGEN	Description of operation
0	Does not reactivate.
1	Reactivates.

[PWM/PPG] [BTxTMCR: bit10] PMSK (Pulse MaSK) Pulse output mask bit

Select a level of waveform to output (TOUT signal) from the followings:

- Normal output: Output the waveform output from the 16-bit PWM/PPG timer without modification.
- Fixed output: Output a sequence of "L" level or "H" level signals regardless of the settings of cycle or duty.

PMSK	Description
0	Normal output
1	Fixed output

If the fixed output is selected by writing "1" to this bit, the level being output will vary depending on the settings of OSEL bit.

- If OSEL=0: "L" level will be output.
- If OSEL=1: "H" level will be output.

[Reload timer/PWM/PPG] [BTxTMCR: bit9, bit8] EGS1, EGS0 (EdGe Select) Trigger input selection bits

Select an effective edge for the external activation trigger (TGIN) signal.

EGS[1:0]	Description
00	Trigger input disabled
01	Rising edge
10	Falling edge
11	Both edges

[PWC] [BTxTMCR: bit10 to bit8] EGS2 to EGS0 (EdGe Select) Measurement mode selection bits

Select a measurement mode.

EGS[2:0]	Description
000	**"H" pulse width measurement:** Duration in which the input signal is maintained at the "H" level
001	**Rising edge interval measurement:** Time from the detection of a rising edge to the detection of the next rising edge
010	**Falling edge interval measurement:** Time from the detection of a falling edge to the detection of the next falling edge
011	**Edge-to-edge pulse width measurement:** The width between consecutive input edges is either of. (1) Time from the detection of a rising edge to the detection of the falling edge (2) Time from the detection of a falling edge to the detection of the rising edge
100	**"L" pulse width measurement:** Duration in which the input signal is maintained at the "L" level (Time from the detection of a falling edge to the detection of the rising edge)
101 110 111	Setting prohibited

[Reload timer/PWC] [BTxTMCR: bit7] T32 (Timer 32bit) 32-bit timer selection bit

Select whether to run the 16/32-bit timer individually by each channel or use the two channels as 32-bit timer through a cascade connection. Set this bit for both channel 0 and channel 1.

T32(channel0)	T32(channel1)	Description
0	0	16-bit independent operation respectively
0	1	Setting prohibited
1	0	32-bit timer
1	1	Setting 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.)

[BTxTMCR: bit6 to bit4] FMD2 to FMD0 (Function MoDe) Timer function selection bits

These bits are used to select a function of base timer. To change these bits, go to 000(reset mode) first, and set it to another mode.

FMD[2:0]	Description
000	Reset mode (Writing FMD=000 will reverse the state of the base timer after the reset. Each register will be reset to the initial value.)
001	16-bit PWM timer
010	16-bit PPG timer
011	16/32-bit reload timer
100	16/32-bit PWC Timer
101 110 111	Setting prohibited

[BTxTMCR: 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

[BTxTMCR: bit2] MDSE (MoDe Select) Mode selection bit

[Reload timer-PWM]

MDSE	Description
0	**Reload mode:** When the down counter underflows, the value of the cycle setting register (BT0PCSR/BT1PCSR) 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.

[BTxTMCR: bit1] CTEN (CounT ENable) Counter operation enable bit

Enables/disables the counter operation.

This bit is cleared to "0" when measurement is completed in single measurement mode of the PWC mode.

CTEN	Description
0	Disables/stops the operation.
1	Enables the operation.

[BTxTMCR: bit0] STRG (Software TRiGger) Software trigger bit

Functions as a trigger for timer activation, etc.

In the PWC mode the read value is "0"; this bit should be cleared to "0".

(Notes)
- When writing to this bit, be careful not to clear other bits.
- When writing to CTEN and FMD[2:0] simultaneously, issue a trigger as soon as the operation is enabled.

STRG	Description
0	Ignores.
1	Issues a trigger.

40.4.1.3 I/O selection register: BTSEL01 (Base Timer SElect register ch.0 and ch.1)

These bits are used to set the I/O mode of ch.0 and ch.1 for the base timer.

(Notes)
- This register must be accessed in 8 -bit mode.
- Rewrite these registers after selecting base timer reset mode (FMD2 to FMD0 = 000) by setting the base timer x timer control registers (BTxTMCR).

- **BTSEL01: Address 009C$_H$ (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-				SEL01[3:0]				
1	1	1	1	0	0	0	0	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Undefined

Writing has no effect on operation.

[bit3 to bit0] SEL013 to SEL010 (SELect) ch.0/ch.1 I/O selection bits

These bits are used to set the I/O mode of ch.0 and ch.1 for the base timer.

SEL013 to SEL010	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 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 prohibited

40.4.1.4 Simultaneous software activation register: BTSSSR (Base Timer Software Synchronous Start Register)

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)**

15	14	13	12	11	10	9	8	bit
-								
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	Attribute

7	6	5	4	3	2	1	0	bit
-						SSSR1	SSSR0	
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,W	R1,W	Attribute

[bit15 to bit2] Undefined

Writing has no effect on operation.

[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 details of connection, see "Figure 40.5-2 Wiring Diagram of Each I/O Mode (2)".

SSSR0/1	Description
0	Does nothing.
1	"1" pulse will be allocated for input and activate the corresponding channel.

40.4.2 Register for 16/32-bit Reload Timer

40.4.2.1 Status control register 0, 1: BT0STC, BT1STC (Base Timer 0/1 STatus Control)

These registers control interrupt requests.

(Notes)
- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8 -bit mode.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0STC, BT1STC: Address 0085_H, 0095_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1), W	R0,W0	R(RM1), W	Attribute

[bit7] Reserved

Always write "0" to this bit.

[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 bit = 1).

[bit5] Reserved

Always write "0" to this bit.

[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.

[bit3] Reserved

Always write "0" to this bit.

[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 TGIE bit is set to "1" while this bit is "1", a trigger interrupt request will be generated.

[bit1] Reserved

Always write "0" to this bit.

[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.	Ignores.

40.4.2.2 Cycle setting register 0, 1: BT0PCSR, BT1PCSR (Base Timer 0/1 Pulse Counter Start Register)

These registers with a buffer set the cycle for 16/32-bit reload timer. The down counter counts down from the value set to these registers.

(Notes)
- These registers must be accessed in 16 -bit mode.
- Set these registers after selecting a base timer function to the 16/32-bit reload timer (FMD2 to FMD0=011)using the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR).
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0PCSR, BT1PCSR: Address 0088_H, 0098_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

These registers with a buffer set the cycle for the 16/32-bit reload timer. The down counter counts down from the value set to these registers.

The value set to these registers is loaded to the 16-bit down counter in the following cases:

- When the 16/32-bit reload timer is started
- When the down counter underflows

The following values are set to these registers when two channels of a 16-bit reload timer are cascaded and it is used as the 32-bit reload timer.

- Value of even-number channel cycle setting register (BT0PCSR/BT1PCSR): Value of lower 16 bits
- Value of the odd-number channel cycle setting register (BT0PCSR/BT1PCSR): Value of upper 16 bits

For this reason, in the 32-bit timer mode, write values into these registers in the following order.

1. Odd-number channel cycle setting register (BT0PCSR/BT1PCSR)
2. Even-number channel cycle setting register (BT0PCSR/BT1PCSR)

40.4.3 Register for 16-bit PWM Timer

40.4.3.1 Status control register 0, 1: BT0STC, BT1STC (Base Timer 0/1 STatus Control)

These registers control interrupt requests.

(Notes)
- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR, DTIR, and UDIR, "1" is read out.
- These registers must be accessed in 8 -bit mode.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0STC, BT1STC: Address 0085_H, 0095_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
Reserved	TGIE	DTIE	UDIE	Reserved	TGIR	DTIR	UDIR	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R(RM1), W	R (RM1), W	R(RM1), W	Attribute

[bit7] Reserved

Always write "0" to this bit.

[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 bit = 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 duty setting register (BT0PDUT/BT1PDUT) (DTIR bit = 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.

[bit3] Reserved

Always write "0" to this bit.

[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 (BT0PDUT/BT1PDUT) (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.	Ignores.

40.4.3.2 Cycle setting register 0, 1: BT0PCSR, BT1PCSR (Base Timer 0/1 Pulse Counter Start Register)

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 signal) is inverted.

(Notes)
- These registers must be accessed in 16 -bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR).
- Be sure to rewrite the duty setting register (BT0PDUT/BT1PDUT) when these registers are rewritten.
- Do not set a value smaller than the value set to the duty setting register (BT0PDUT/ BT1PDUT).
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0PCSR, BT1PCSR: Address 0088_H, 0098_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

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 signal) 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 duty setting register (BT0PDUT/BT1PDUT), the level of the output signal (TOUT signal) can be fixed. The output signal level is as follows according to the setting of OSEL bit of the timer control register (BT0TMCR/BT1TMCR):

- OSEL=0: "H" level
- OSEL=1: "L" level

40.4.3.3 Duty Setting Register 0, 1: BT0PDUT, BT1PDUT (Base Timer 0/1 Pulse DuTy register)

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 signal) is inverted.

(Notes)
- These registers must be accessed in 16 -bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR).
- Do not set the value higher than the value set to the cycle setting register (BT0PCSR/BT1PCSR) when these registers are rewritten.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0PDUT, BT1PDUT: Address 008A$_H$, 009A$_H$ (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

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 signal) 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 cycle setting register (BT0PCSR/BT1PCSR), the level of the output signal (TOUT signal) can be fixed. The output signal level is as follows according to the setting of OSEL bit of the timer control register (BT0TMCR/BT1TMCR):

- OSEL=0: All "H" level
- OSEL=1: All "L" level

40.4.4 Register for 16-bit PPG Timer

40.4.4.1 Status control register 0, 1: BT0STC, BT1STC (Base Timer 0/1 STatus Control)

These registers control interrupt requests.

(Notes)
- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8 -bit mode.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0STC, BT1STC: Address 0085_H, 0095_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1), W	R0,W0	R(RM1), W	Attribute

[bit7] Reserved

Always write "0" to this bit.

[bit6] TGIE (TriGger Interrupt Enable) Trigger interrupt request enable bit

This bit sets whether or not to generate trigger interrupt request when a 16-bit PPG timer activation trigger is detected (TGIR bit = 1).

[bit5] Reserved

Always write "0" to this bit.

[bit4] UDIE (UnDerflow Interrupt Enable) Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the H width setting reload register (BT0PRLH/BT1PRLH) completed counting down and the counter underflows (UDIR bit =1).

TGIE/UDIE	Description
0	Disables.
1	Enables.

[bit3] Reserved

Always write "0" to this bit.

[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.

[bit1] Reserved

Always write "0" to this bit.

[bit0] UDIR (UnDerflow Interrupt Register) Underflow interrupt request flag bit

This bit indicates that the H width setting reload register (BT0PRLH/BT1PRLH) 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.	Ignores.

40.4.4.2 L width setting register 0, 1: BT0PRLL, BT1PRLL (Base Timer 0/1 Pulse Length of "L" register)

These registers set the width of signal level output from the 16-bit PPG timer.

(Notes)
- These registers must be accessed in 16 -bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR).
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0PRLL, BT1PRLL: Address 0088_H, 0098_H (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

These registers set the width of signal level 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 signal) will be inverted. Setting these registers and the H width setting reload register (BT0PRLH/BT1PRLH) 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 OSEL bit of the timer control register (BT0TMCR/BT1TMCR) as follows:

- OSEL=0: "L" level width
- OSEL=1: "H" level width

The value set to registers is loaded to the 16-bit down counter when a 16-bit PPG timer activation trigger is detected or when the H width setting reload register (BT0PRLH/BT1PRLH) completed counting values and underflow occurs.

40.4.4.3 H width setting register 0, 1: BT0PRLH, BT1PRLH (Base Timer 0/1 Pulse Length of "H" register)

These registers with a buffer set the width of signal level output when the L width setting reload register (BT0PRLL/BT1PRLL) completes counting values.

(Notes)
- These registers must be accessed in 16 -bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR).
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0PRLH, BT1PRLH: Address 008A$_H$, 009A$_H$ (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R/W	R/W	• • •	R/W	R/W	R/W	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

These registers with a buffer set the width of signal level output when the L width setting reload register (BT0PRLL/BT1PRLL) completes counting values. When the 16-bit down counter completes counting down the value set to these registers, the level of the output waveform (TOUT signal) will be inverted.

Setting these registers and the L width setting reload register (BT0PRLL/BT1PRLL) 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 OSEL bit of the timer control register (BT0TMCR/BT1TMCR) 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 H width setting reload register (BT0PRLH/BT1PRLH) completes counting down values and underflow occurs

- Transfer to the 16-bit down counter

- When counting down from the value of the L width setting reload register (BT0PRLL/BT1PRLL) is completed.

For rewriting timing, see "■ Write Timing" in Section "40.5.6.3 Operation in Reload Mode".

40.4.5 Register for 16/32-bit PWC Timer

40.4.5.1 Status control register 0, 1: BT0STC, BT1STC (Base Timer 0/1 STatus Control)

These registers control interrupt requests.

(Notes)
- Reserved bits must be set to "0".
- For the read-modify-write instruction to OVIR, "1" is read out.
- These registers must be accessed in 8 -bit mode.
- These registers will also be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0STC, BT1STC: Address 0085_H, 0095_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
ERR	EDIE	Reserved	OVIE	Reserved	EDIR	Reserved	OVIR	
0	0	0	0	0	0	0	0	Initial value
R,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R(RM1), W	Attribute

[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 (BT0DTBF/BT1DTBF) 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 (BT0DTBF/BT1DTBF).

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 bit = 1).

[bit5] Reserved

Always write "0" to this bit.

[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	Disables.
1	Enables.

[bit3] Reserved

Always write "0" to this bit.

[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 (BT0DTBF/BT1DTBF) is read out.

[bit1] Reserved

Always write "0" to this bit.

[bit0] OVIR (OVerflow Interrupt Register) Overflow interrupt request flag bit

This bit indicates that the up counter value has changed from "$FFFF_H$" to "0000_H" and an overflow occurred. When this bit is "1" and the OVIE bit is set to "1", an overflow interrupt request is generated. This bit is cleared when "0" is written.

EDIR/OVIR	Read	Write
0	Measurement completion/overflow has not been occurred.	(EDIR) Ignores (OVIR) This bit is cleared.
1	Measurement completion/overflow has been occurred.	Ignores.

40.4.5.2 Data Buffer Register 0-1: BT0DTBF, BT1DTBF (Base Timer 0/1 DaTa BuFfer register)

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 (BT0TMCR.FMD=000/BT1TMCR.FMD=000)).

- **BT0DTBF, BT1DTBF: Address $008A_H$, $009A_H$ (Access: Half-word)**

15	14	• • •	2	1	0	bit
D[15:0]						
0	0	• • •	0	0	0	Initial value
R,WX	R,WX	• • •	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] D[15:0] (Data) Data bit

These registers are used to read out the measurement value of the 16/32-bit PWC timer and the up counter value. The value read from these registers is different in the single measurement mode and continuous measurement mode.

- Single measurement mode: The up counter value is read during counting and the measurement result is read after the measurement completion.
- Continuous measurement mode: The value measured previously is read both during counting and after the measurement completion. The up counter value cannot be read.

The following values are set to these registers when two channels of a 16-bit PWC timer are cascaded and it is used as the 32-bit PWC timer.

- Value of even-number channel data buffer register (BT0DTBF): Value of lower 16 bits
- Value of odd-number channel data buffer register (BT1DTBF): Value of upper 16 bits

In the 32-bit timer mode, read these registers value in the following order.

1. Even-channel data buffer register (BT0DTBF)
2. Odd-channel data buffer register (BT1DTBF)

40.5 Operation

40.5.1 Selection of Timer Function

Select the timer function for BT0TMCR.FMD[2:0]/BT1TMCR.FMD[2:0].

40.5.2 I/O Assignment

Set I/O of the base timer for the BTSEL01 register before using the timer. You can select one of the following seven modes:

I/O mode 0: 16-bit timer standard mode
The base timer operates separately for each channel in this mode.
I/O mode 1: 32-bit timer full mode
The even-channel signals of the base timer are separately 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 channel will start 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 (BT0TMCR.EGS/BT1TMCR.EGS).

Figure 40.5-1 Wiring Diagram of Each I/O Mode (1)

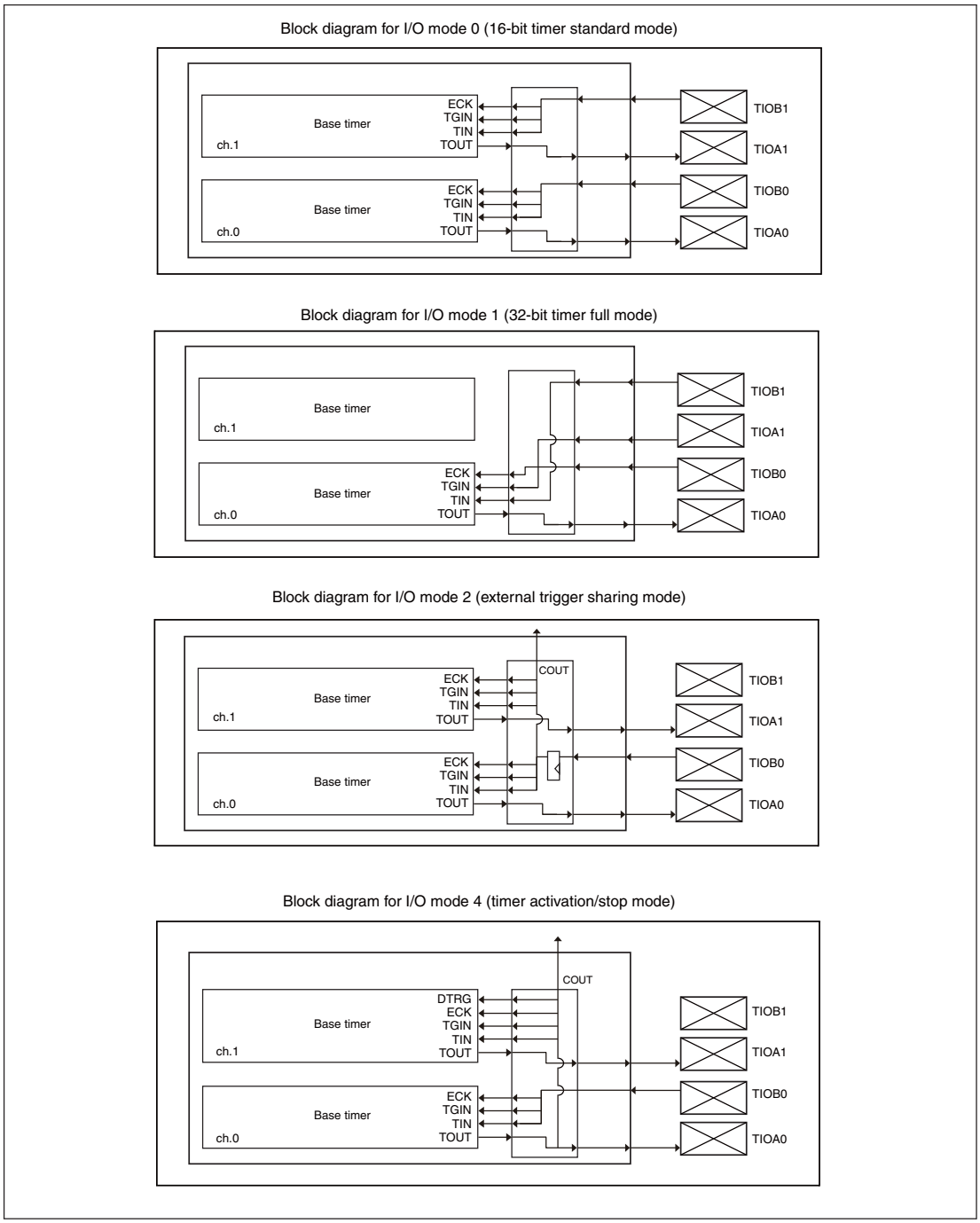
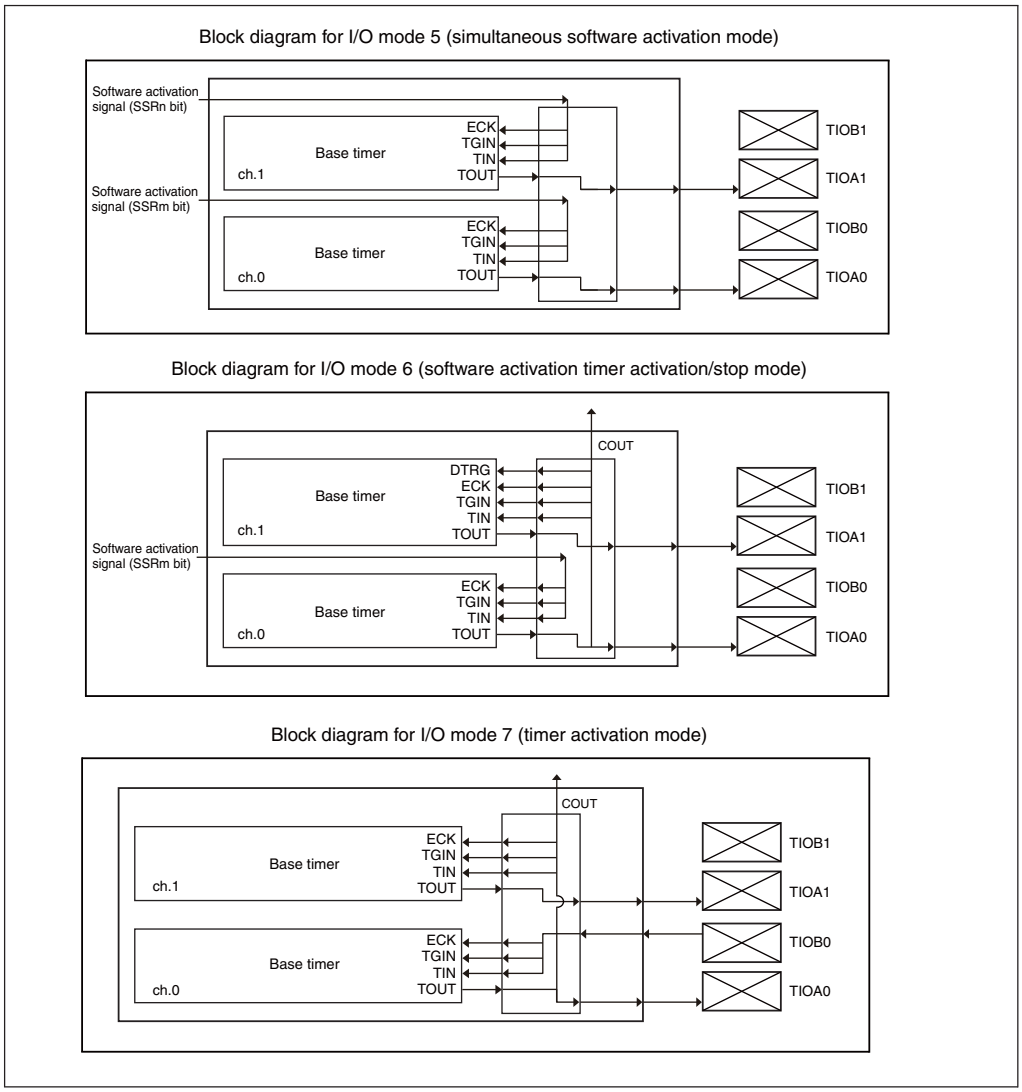


Figure 40.5-2 Wiring Diagram of Each I/O Mode (2)



40.5.3 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.

■ 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.

■ 32-bit Mode Setting

First, set "000" to the FMD bit of the BT0TMCR/BT1TMCR 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 BT0TMCR/BT1TMCR 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 bits 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 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 bit of the BT0TMCR/BT1TMCR register of the even-number channel to reset to the reset mode for both the even-number and odd-number channels, and make a setting in the 16-bit mode for each channel.

■ 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 bits and the timer/counter of the odd-number channel operates with upper 16 bits.

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 40.5-11 Configuration in 32-bit Timer Mode" and "Figure 40.5-27 Configuration in 32-bit Timer Mode".

40.5.4 16/32-bit Reload Timer Operation

This section explains the operation performed when the base timer included in this product is used as the 16/32-bit reload timer. An example is also given to set various operation conditions.

Figure 40.5-3 Block Diagram of 16/32-bit Reload Timer (16-bit Reload Timer Operation)

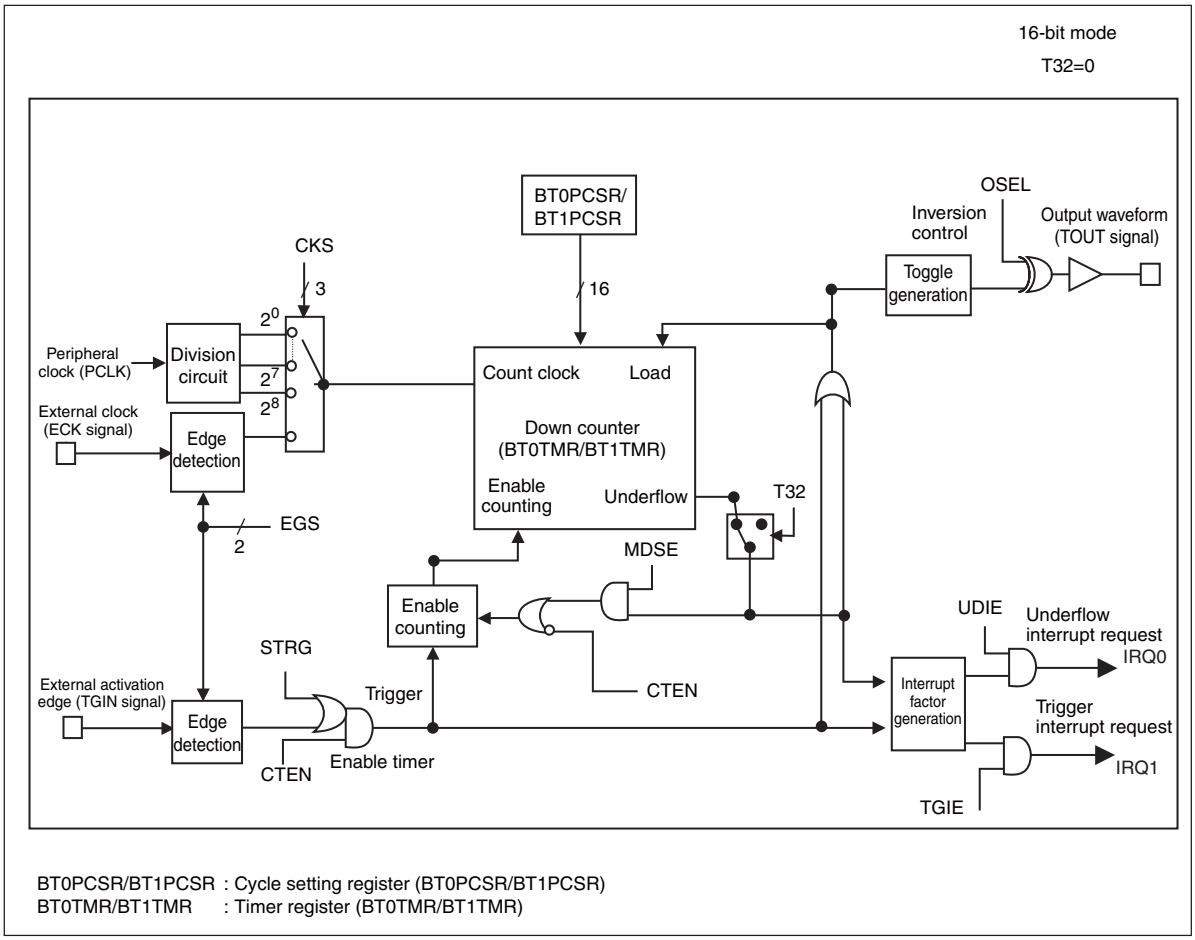
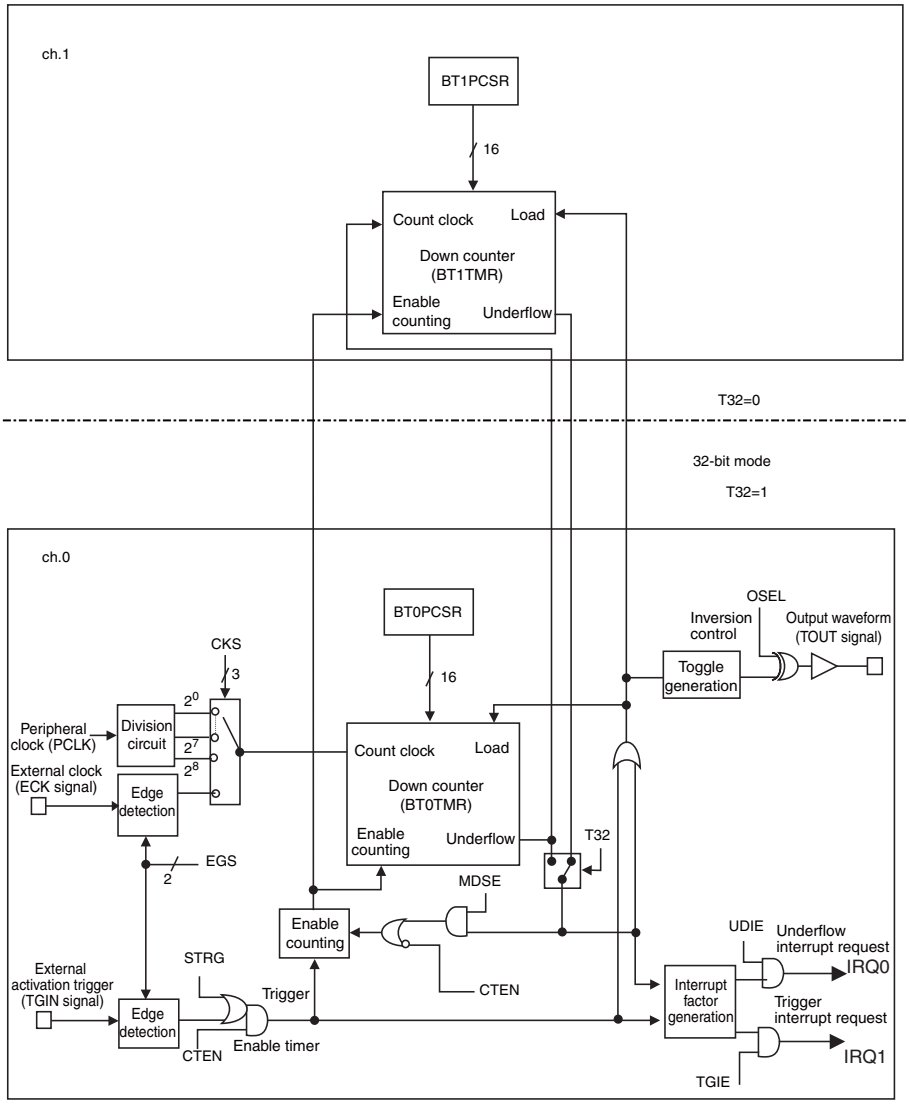


Figure 40.5-4 Block Diagram of 16/32-bit Reload Timer (32-bit Reload Timer Operation)



40.5.4.1 Overview

The 16/32-bit reload timer is a timer that decrements from the value set in the cycle setting register (BT0PCSR/BT1PCSR). 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 timer control register (BT0TMCR/BT1TMCR).
  - 16-bit timer mode (T32=0): A 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 MDSE bit of the timer control register (BT0TMCR/BT1TMCR).
  - Reload mode (MDSE=0): In this mode, when the down counter underflows, the preset value (cycle) is reloaded to allow the timer to repeat counting.
  - One-shot mode (MDSE=1): Once the down counter underflows, the counter will no longer count.

40.5.4.2 Operation in Reload Mode

This section explains the operation in reload mode.

■ Overview

In this mode, the value set in the cycle setting register (BT0PCSR/BT1PCSR) 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 timer control register (BT0TMCR/BT1TMCR) to "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 timer control register (BT0TMCR/BT1TMCR) to "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 timer control register (BT0TMCR/BT1TMCR) 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 Section "40.5.2 I/O Assignment".
- To start counting as soon as the operation is permitted, set both CTEN and STRG bits of the timer control register (BT0TMCR/BT1TMCR) to "1".

● Counting Operation

When an activation trigger is input, the value (cycle) set in the cycle setting register (BT0PCSR/BT1PCSR) 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)

The following figures show the count start timing.

Figure 40.5-5 Count Start Timing (Software Trigger)

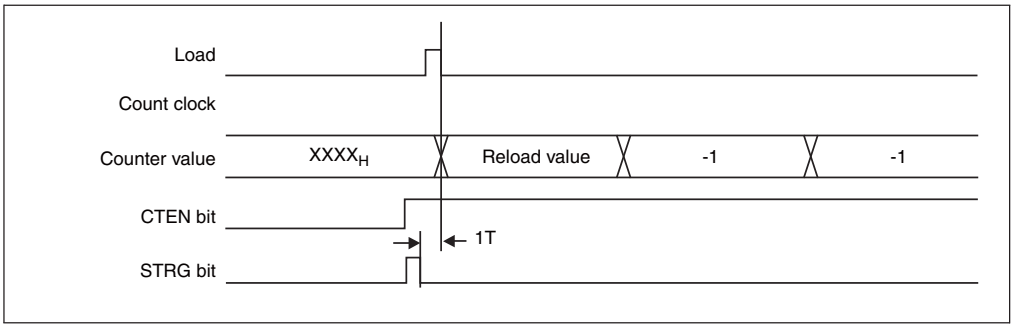
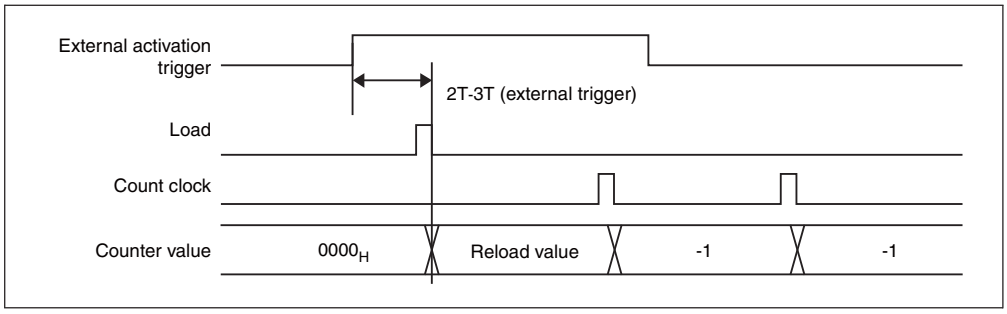


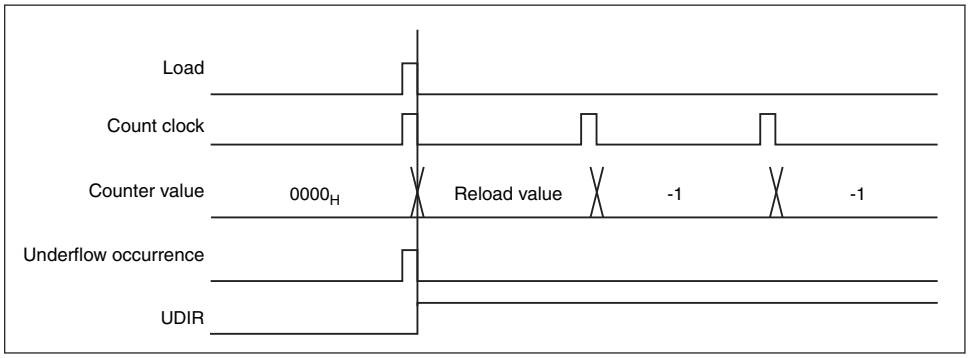
Figure 40.5-6 Count Start Timing (External Activation Trigger (TGIN Signal), Effective Edge = Rising Edge)



(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 Section "40.5.2 I/O Assignment".

When the down counter underflows after attempting to count down further from the value of "0000$_H$", the value (cycle) set in the cycle setting register (BT0PCSR/BT1PCSR) is reloaded to the down counter, which continues to count down. If an underflow occurs, UDIR bit of the status control register (BT0STC/BT1STC) changes to "1". At this time, an underflow interrupt request occurs if UDIE bit is set to "1". The following shows the operation in case of an underflow.

Figure 40.5-7 Operation in Case of an Underflow



■ Output Waveform

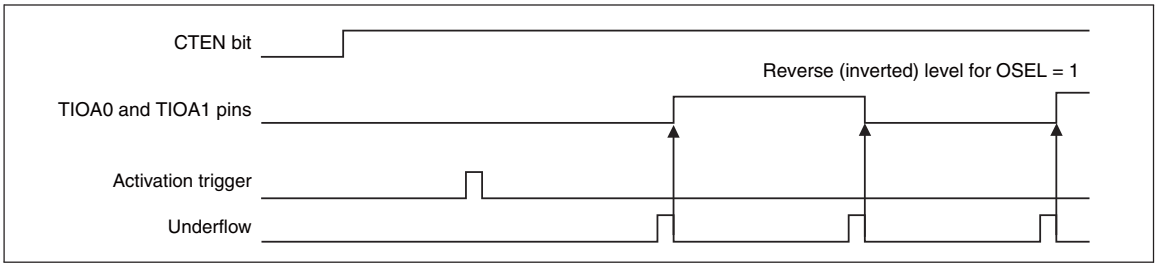
The waveform (TOUT signal) of the 16/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of OSEL bit of the timer control register (BT0TMCR/BT1TMCR).

Table 40.5-1 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.

The output waveform in reload mode is shown below.

Figure 40.5-8 Output Waveform in Reload Mode (Normal Polarity)



40.5.4.3 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

■ Overview

In this mode, the counter will no longer count down once an underflow occurs.

To use this mode, set one-shot mode by setting the MDSE bit of the timer control register (BT0TMCR/BT1TMCR) to "1".

■ Operations

● Activation

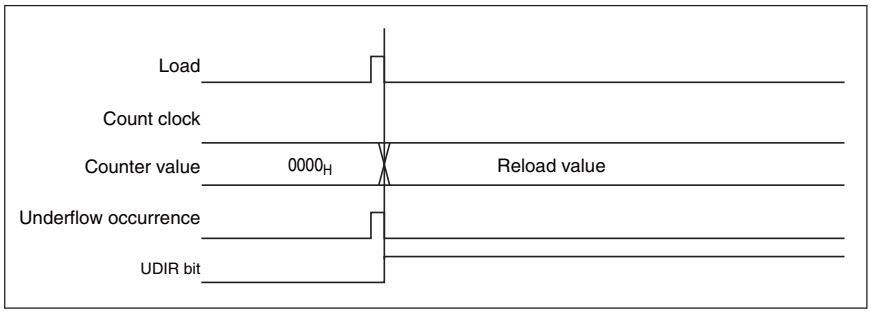
The same operation as in reload mode. See "■ Operation" in Section "40.5.4.2 Operation in Reload Mode".

● Counting Operation

The operation is the same as in reload mode until an underflow occurs. See "■ Operation" in Section "40.5.4.2 Operation in Reload Mode". When the down counter underflows, the value (cycle) set in the cycle setting register (BT0PCSR/BT1PCSR) is reloaded to the down counter. However, the down counter stops counting. If an underflow occurs, the UDIR bit of the status control register (BT0STC/BT1STC) changes to "1". At this time, an underflow interrupt request occurs if the UDIE bit of the status control register (BT0STC/BT1STC) is set to "1".

The following shows the operation in case of an underflow.

Figure 40.5-9 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 OSEL bit of the timer control register (BT0TMCR/BT1TMCR).

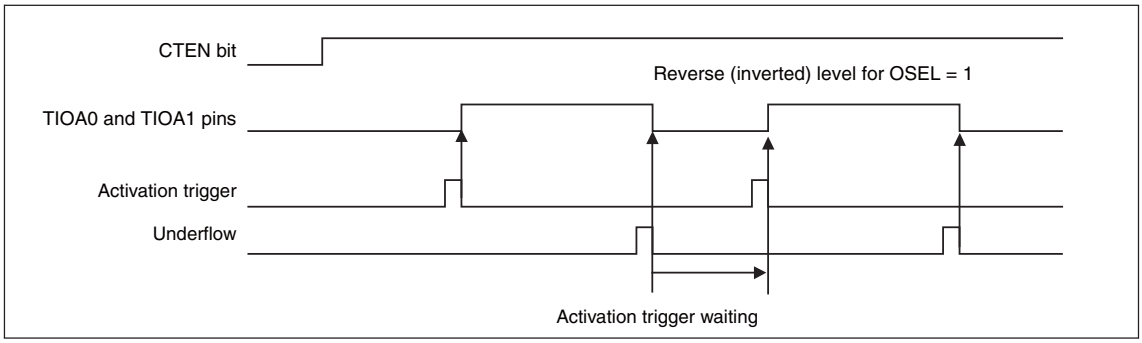
The correspondence between output polarities and output waveforms is shown below.

Table 40.5-2 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.

The output waveform in one-shot mode is shown in Figure 40.5-10.

Figure 40.5-10 Output Waveform in One-shot Mode (Normal Polarity)



40.5.4.4 Operation in 32-bit Timer Mode

This section explains the setting and operation for cascading 2 channels of a 16-bit reload timer and using it as a 32-bit reload timer.

■ Overview

Using the T32 bit of the timer control register (BT0TMCR/BT1TMCR), 2 channels of a 16-bit reload timer can be cascaded and used as a 32-bit reload timer. In this mode, even-number channels correspond to the operations of the lower 16 bits and odd-number channels to the operations of the upper 16 bits. 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 Examples

To set 32-bit timer mode, set the T32 bit of the timer control register (BT0TMCR/BT1TMCR) of even-number channels to "1" and the T32 bit of the timer control register (BT0TMCR/BT1TMCR) 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. Select ch.0 to reset mode by setting the FMD2 to FMD0 bits of the base timer 0 timer control register (BT0TMCR). (FMD2 to FMD0 = 000)
2. Select 16/32-bit reload timer for ch.0 and ch.1 by setting the FMD2 to FMD0 bits of the 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 timer control register (BT0TMCR/BT1TMCR) to "0".
- A reload value in the cycle setting register (BT0PCSR/BT1PCSR) must be set in the order of the odd-number → even-number channels.

■ Operation

In 32-bit timer mode, the counting operation is basically the same as in 16-bit timer mode.

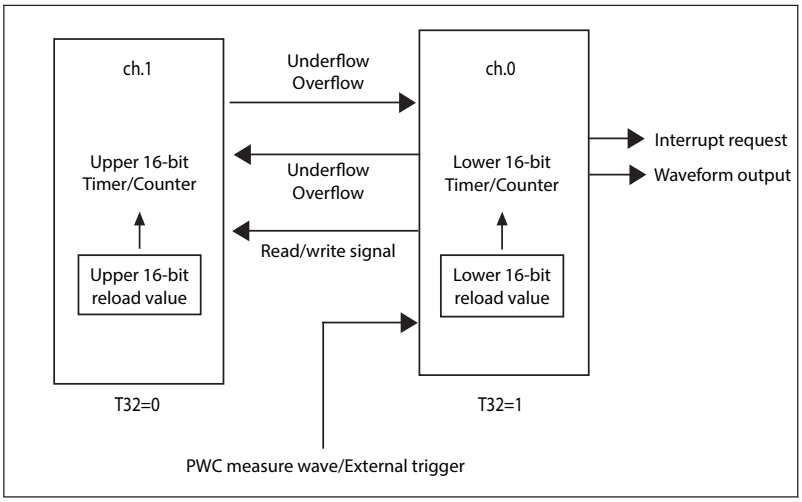
However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the registers next to the odd-number channels.

- Timer control register (BT0TMCR/BT1TMCR)
- Status control register (BT0STC/BT1STC)

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 cycle setting register (BT0PCSR/BT1PCSR) and the even-number channel cycle setting register (BT0PCSR/BT1PCSR) (lower 16 bits) 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 bits and the odd-number channels as the upper 16 bits.
3. When the down counter underflows, the UDIR bit of the timer control register (BT0TMCR/BT1TMCR) of the even-number channels changes to "1". The channel configuration in 32-bit timer mode is shown below.

Figure 40.5-11 Configuration in 32-bit Timer Mode



(Notes)
- The value of the down counter can be checked by reading the timer register (BT0TMR/BT1TMR). In the 32-bit timer mode, it must be read in the order of the lower 16 bits (even-numbered channel) → upper 16 bits (odd-number channel).
- In 32-bit timer mode, the operation of the 32-bit 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 signal) from an odd-number channel pin is fixed to "L" level.

40.5.4.5 Interrupts

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 40.5-3 Conditions for Interrupt Generation

Interrupt request	Interrupt request flag	Permission of interrupt request	Clearing of interrupt request
Trigger interrupt request	BT0STC/BT1STC TGIR=1	BT0STC/BT1STC TGIE=1	Set the TGIR bit of BT0STC/ BT1STC to "0".
Underflow interrupt request	BT0STC/BT1STC UDIR=1	BT0STC/BT1STC UDIE=1	Set the UDIR bit of BT0STC/ BT1STC 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 "Table C-1" in "APPENDIX C. 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 "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

40.5.4.6 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 timer control register (BT0TMCR/BT1TMCR) after stopping the 16-bit down counter by resetting CTEN bit to "0".
  - CKS3 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 (BT0TMCR/ BT1TMCR) 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 FMD2 to FMD0 bits or T32 bit of the timer control register (BT0TMCR/BT1TMCR) after a reset, be sure to once reset the FMD2 to FMD0 bits to "000" to select reset mode, and then rewrite these bits.

■ Notes on Operation

- 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 cycle setting register (BT0PCSR/BT1PCSR) is loaded to the 16-bit down counter, which begins counting.
- A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

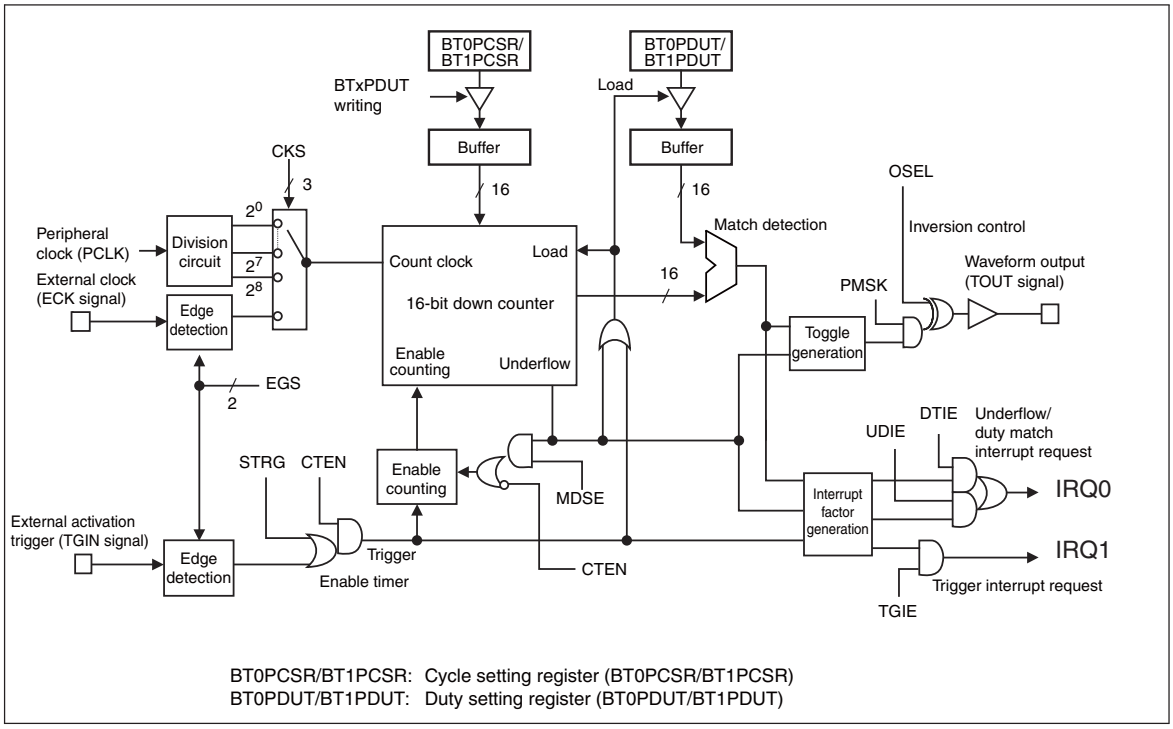
■ Notes on Interrupts

- If to clear the interrupt request flag and to change the interrupt request flag to "1" occur at the same time, the flag clear is ignored. The interrupt request flag is held to "1".

40.5.5 16-bit PWM Timer Operation

This section explains the operation performed when the base timer included in this product is used as the 16-bit PWM timer. Examples of procedures for setting various operating conditions are also provided.

Figure 40.5-12 Block Diagram of 16-bit PWM Timer (16-bit PWM Timer Operation)



40.5.5.1 Overview

The 16-bit PWM timer sets the cycle in the cycle setting register (BT0PCSR/BT1PCSR) and the duty in the duty setting register (BT0PDUT/BT1PDUT). A desired waveform (TOUT signal) can be output by setting values in these registers. The 16-bit PWM timer starts decrementing from the value set in the cycle setting register (BT0PCSR/BT1PCSR). When the value of the down counter matches the value of the duty setting register (BT0PDUT/BT1PDUT), the output signal (TOUT signal) level is inverted. When the down counter underflows, the output level is inverted again. This method enables output of a waveform (TOUT signal) with a desired cycle and duty.

One of two 16-bit PWM timer operation modes can be selected using the MDSE bit of the timer control register (BT0TMCR/BT1TMCR) 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 repeat counting.
- One-shot mode (MDSE=1): Once the 16-bit down counter underflows, the counter will no longer count.

40.5.5.2 Operation in Reload Mode

This section explains the operation in reload mode.

■ Overview

In this mode, the value set in the cycle setting register (BT0PCSR/BT1PCSR) 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 timer control register (BT0TMCR/BT1TMCR) to "0".

■ Operation

● Activation

Activate the 16-bit PWM timer with the following procedure:

1. Permit 16-bit PWM timer operation by setting the CTEN bit of the timer control register (BT0TMCR/BT1TMCR) to "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 timer control register (BT0TMCR/BT1TMCR) 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 decrementing from the value set in the cycle setting register (BT0PCSR/BT1PCSR).

(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 cycle setting register (BT0PCSR/BT1PCSR) can be loaded to the 16-bit down counter:
  - If a software trigger is used: 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 decrementing from the value set in the cycle setting register (BT0PCSR/BT1PCSR).

When the value of the 16-bit down counter matches the value of the duty setting register (BT0PDUT/BT1PDUT), the operation is performed as follows:

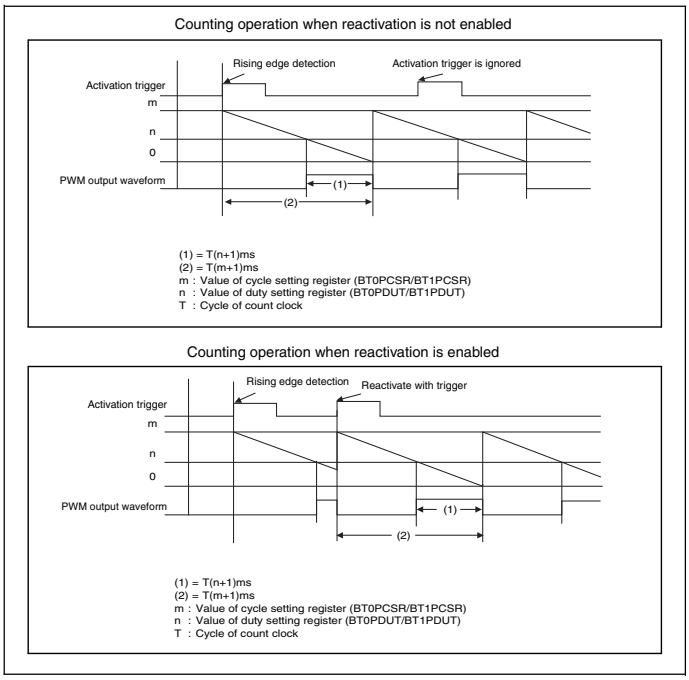
- The DTIR bit of the status control register (BT0STC/BT1STC) changes to "1".
- The level of the output signal (TOUT signal) is inverted.
- Countdown is continued. Later, when the 16-bit down counter underflows, the operation is performed as follows:
-The UDIR bit of the status control register (BT0STC/BT1STC) changes to "1" and the level of the output signal (TOUT signal) is inverted.
- The value of the cycle setting register (BT0PCSR/BT1PCSR) is reloaded to continue countdown.

Every time an underflow occurs, the value of the cycle setting register (BT0PCSR/BT1PCSR) 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 (BT0TMCR/BT1TMCR).

- 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 (BT0STC/BT1STC) changes to "1". In addition, the value set in the cycle setting register (BT0PCSR/BT1PCSR) is reloaded to the 16-bit down counter, which begins counting.

These operations are shown below.

Figure 40.5-13 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 OSEL bit of the timer control register (BT0TMCR/BT1TMCR).

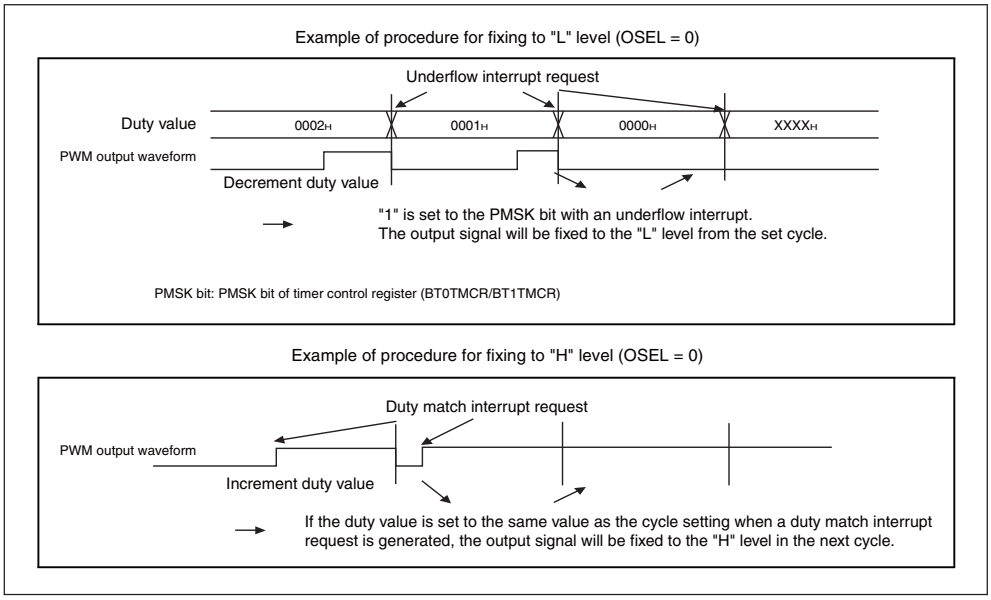
Normal polarity (OSEL=0)
- When the 16-bit PWM timer is activated: "L" level
- When a duty match occurs: "H" level
- When an underflow occurs: "L" level

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 timer control register (BT0TMCR/BT1TMCR). Examples of procedures are shown below.

Figure 40.5-14 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

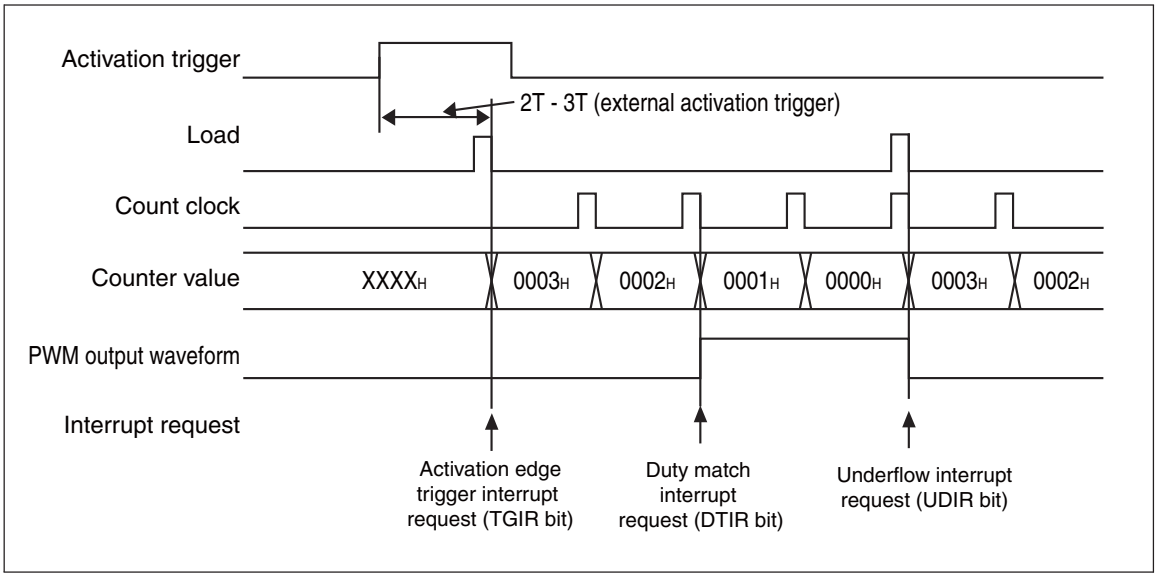
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- An activation trigger is detected.
- The value of the 16-bit down counter matches the value of the duty setting register (BT0PDUT/BT1PDUT)
- When an underflow occurs:

An example of interrupt request generation timing using the following settings is shown below.

- Value of the cycle setting register (BT0PCSR/BT1PCSR) = 0003_H
- Value of the duty setting register (BT0PDUT/BT1PDUT) = 0001_H

Figure 40.5-15 Interrupt Request Generation Timing Chart



40.5.5.3 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

■ Counting Operation

In this mode, counting stops if an underflow occurs when the value of the 16-bit down counter changes from the value set in the cycle setting register (BT0PCSR/BT1PCSR) to "$FFFF_H$".

To use this mode, set one-shot mode by setting the MDSE bit of the timer control register (BT0TMCR/BT1TMCR) to "1".

● Activation

The same operation as in reload mode. See "■ Operation" in the section entitled "40.5.5.2 Operation in Reload Mode".

● Counting Operation

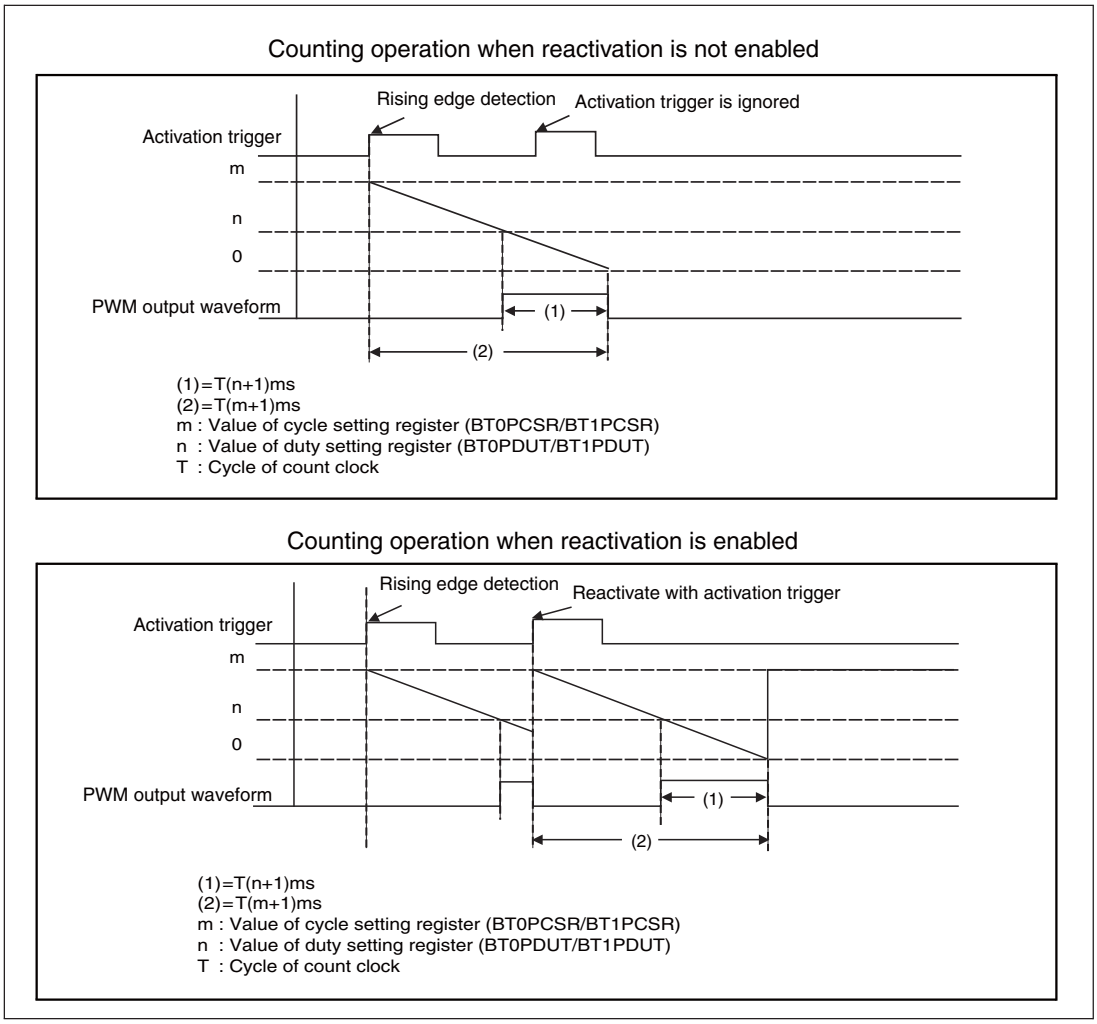
When an activation trigger is input, the 16-bit down counter, in synchronization with the count clock, starts decrementing from the value set in the cycle setting register (BT0PCSR/BT1PCSR). When the value of the 16-bit down counter matches the value of the duty setting register (BT0PDUT/BT1PDUT), the operation is performed as follows:

- The DTIR bit of the status control register (BT0STC/BT1STC) 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 status control register (BT0STC/BT1STC) 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 (BT0TMCR/BT1TMCR).

- 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 (BT0STC/BT1STC) changes to "1". In addition, the value set in the cycle setting register (BT0PCSR/BT1PCSR) is reloaded to the 16-bit down counter, which begins counting.

Figure 40.5-16 Counting Operation



(Note) If a 16-bit PWM timer activation trigger is detected when counting ends, the value set in the cycle setting register (BT0PCSR/BT1PCSR) is loaded to the 16-bit down counter, which begins counting.

■ Output Waveform

The same operation as in reload mode. See "■ Output Waveform" in Section "40.5.5.2 Operation in Reload Mode".

■ Interrupt Generation Timing

The same operation as in reload mode. See "■ Interrupt Generation Timing" in Section "40.5.5.2 Operation in Reload Mode".

40.5.5.4 Interrupt

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 duty setting register (BT0PDUT/BT1PDUT)) (duty match interrupt request).
- An underflow occurs (underflow interrupt request).

Table 40.5-4 Conditions for Interrupt Generation

Interrupt request	Interrupt request flag	Permission of interrupt request	Interrupt request clear
Trigger interrupt request	BT0STC/BT1STC TGIR=1	BT0STC/BT1STC TGIE=1	Set the TGIR bit of BT0STC/BT1STC to "0".
Duty match interrupt request	BT0STC/BT1STC DTIR=1	BT0STC/BT1STC DTIE=1	Set the DTIR bit of BT0STC/BT1STC to "0".
Underflow interrupt request	BT0STC/BT1STC UDIR=1	BT0STC/BT1STC UDIE=1	Set the UDIR bit of BT0STC/BT1STC 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 "Table C-1" in "APPENDIX C. 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 "CHAPTER 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

40.5.5.5 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 (BT0TMCR/BT1TMCR) after stopping the 16-bit down counter by resetting CTEN bit to "0".
  - CKS3 to CKS0 bits
  - EGS1 and EGS0 bits
  - FMD2 to FMD0 bits
  - MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the timer control register (BT0TMCR/BT1TMCR) 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 timer control register (BT0TMCR/BT1TMCR) after a reset, be sure to once reset the FMD2 to FMD0 bits to "000" to select reset mode, and then select a base timer function using the FMD2 to FMD0 bits again.

- 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 timer control register (BT0TMCR/BT1TMCR) to "001".
  2. Set the cycle in the cycle setting register (BT0PCSR/BT1PCSR).
  3. Set the duty in the duty setting register (BT0PDUT/BT1PDUT).

■ 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 cycle setting register (BT0PCSR/BT1PCSR) is loaded to the 16-bit down counter, which then starts counting.
- A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

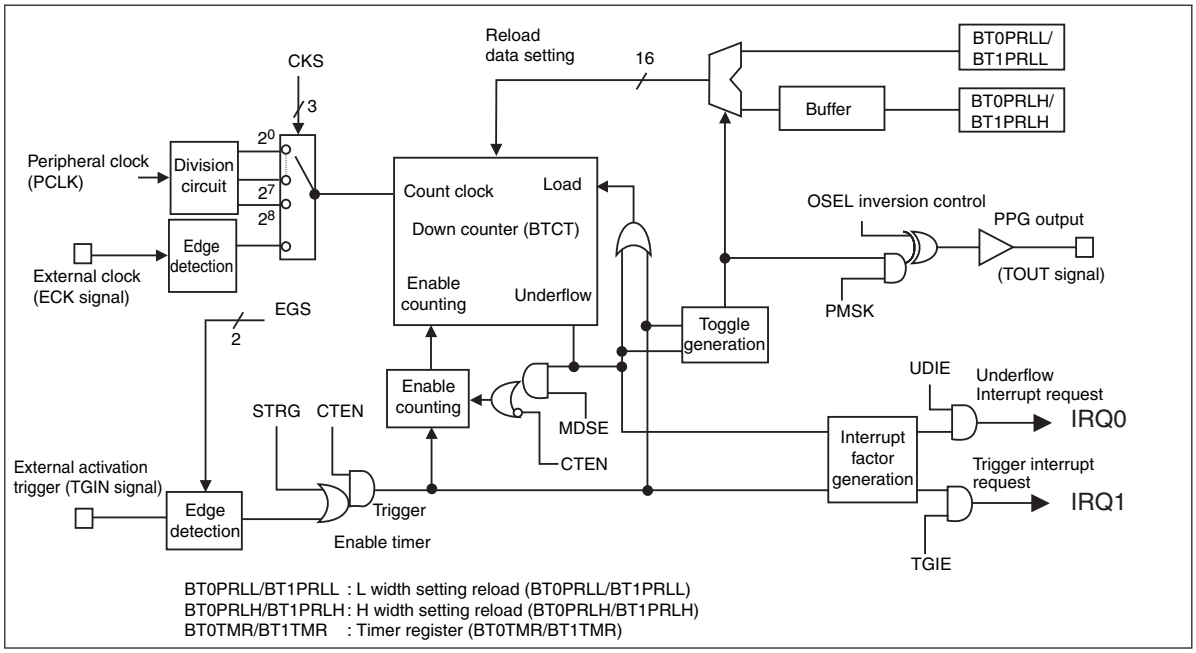
■ Notes on Interrupts

- If to clear the interrupt request flag and to change the interrupt request flag to "1" occur at the same time, the flag clear is ignored. The interrupt request flag is held to "1".

40.5.6 16-bit PPG Timer Operation

This section explains the operation performed when the base timer included in this product is used as the 16-bit PPG timer. Examples of procedures for setting various operating conditions are also provided.

Figure 40.5-17 Block Diagram of 16-bit PPG Timer (16-bit PPG Timer Operation)



40.5.6.1 Overview

The 16-bit PPG timer, once activated, decrements from the value initially specified by the L width setting reload register (BT0PRLL/BT1PRLL). When counting down from the value set in the L width setting reload register (BT0PRLL/BT1PRLL) is completed, the timer begins counting down from the value set in the H width setting reload register (BT0PRLH/BT1PRLH).

When counting down from the value set in each register is completed, the output signal (TOUT signal) inverts its level. Therefore, by configuring the L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH), 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 (BT0TMCR/BT1TMCR) 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.

40.5.6.2 Pulse Width Calculation Method

When the 16-bit PPG timer has counted down by the value set in the L width setting reload register (BT0PRLL/BT1PRLL) or H width setting reload register (BT0PRLH/BT1PRLH) plus 1, the output signal (TOUT signal) 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 L width setting reload register (BT0PRLL/BT1PRLL)
H: Value set in the H width setting reload register (BT0PRLH/BT1PRLH)

This means that when the L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH) 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.

40.5.6.3 Operation in Reload Mode

This section explains the operation in reload mode.

■ Overview

In this mode, the values set in the L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH) 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 L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH) each time an underflow interrupt request is issued.

To use this mode, set reload mode by resetting the MDSE bit of the timer control register (BT0TMCR/BT1TMCR) to "0".

■ Operation

● Activation

Activate the 16-bit PPG timer with the following procedure:

1. Permit 16-bit PPG timer operation by setting the CTEN bit of the timer control register (BT0TMCR/BT1TMCR) to "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 timer control register (BT0TMCR/BT1TMCR) 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 (BT0PRLL/BT1PRLL) can be loaded to the 16-bit down counter:
  - If a software trigger is used: 1T (T: Count clock cycle)
  - If an external event trigger is used: 2T to 3T (T: Count clock cycle)

Counting Operation

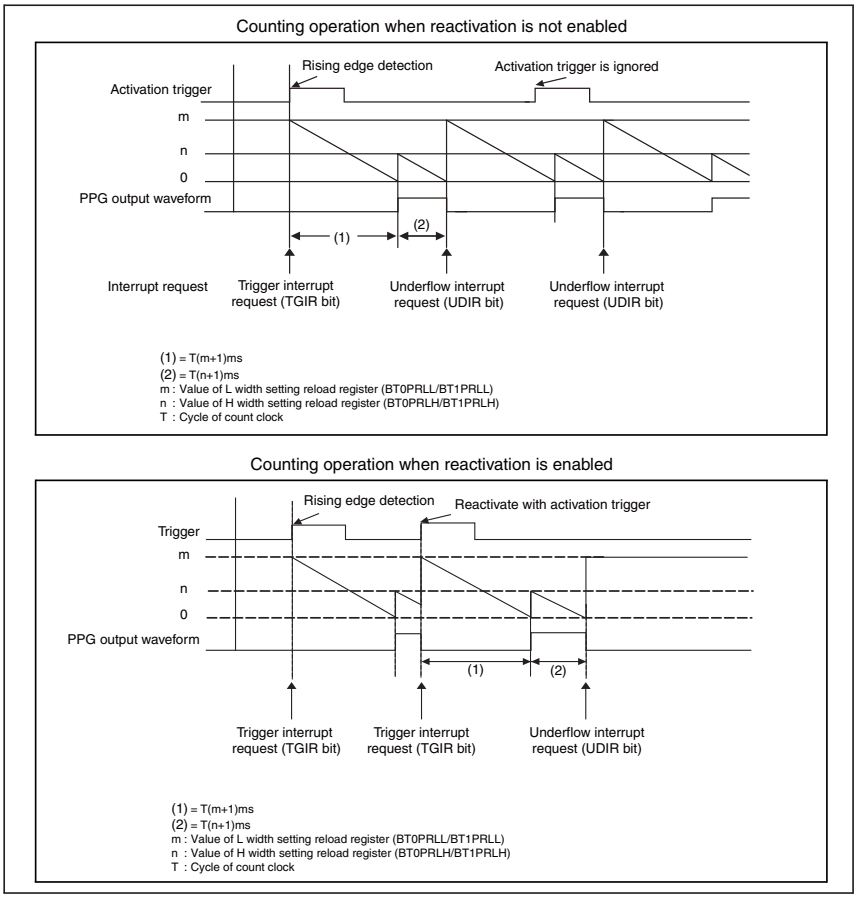
Counting operation initiated by the entry of an activation trigger is explained below using an example where the OSEL bit of the timer control register (BT0TMCR/BT1TMCR) is set for normal polarity (OSEL=0).

1. The value set in the L width setting reload register (BT0PRLL/BT1PRLL) is transferred to the 16-bit down counter and the value set in the H width setting reload register (BT0PRLH/BT1PRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting reload register (BT0PRLL/BT1PRLL). The output signal (TOUT signal) is at the "L" level.
2. The 16-bit down counter completes counting down from the value of L width setting reload register (BT0PRLL/BT1PRLL).
3. The buffered value of H width setting reload register (BT0PRLH/BT1PRLH) is reloaded to the 16-bit down counter, which continues counting down. The output signal (TOUT signal) is at the "H" level.
4. The 16-bit down counter completes counting down from the value of H width setting reload register (BT0PRLH/BT1PRLH), thus causing an underflow.
5. The value of the L width setting reload register (BT0PRLL/BT1PRLL) is reloaded to the 16-bit down counter, which continues to count down. The output signal (TOUT signal) is at the "L" level. In addition, the value of the H width setting reload register (BT0PRLH/BT1PRLH) 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 (BT0TMCR/BT1TMCR).

- 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 (BT0STC/BT1STC) changes to "1". In addition, the value set in the L width setting reload register (BT0PRLL/BT1PRLL) is reloaded to the 16-bit down counter, which begins counting.

Figure 40.5-18 Example of Counting Operation in Reload Mode



(Notes) • The output method and output destination of the output signal (TOUT signal) 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 L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH) are reloaded at the following timing:

● Value set in the L width setting reload register (BT0PRLL/BT1PRLL)

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 H width setting reload register (BT0PRLH/BT1PRLH) is completed.

● Value set in the H width setting reload register (BT0PRLH/BT1PRLH)

It is transferred to the buffer in one of the following events:

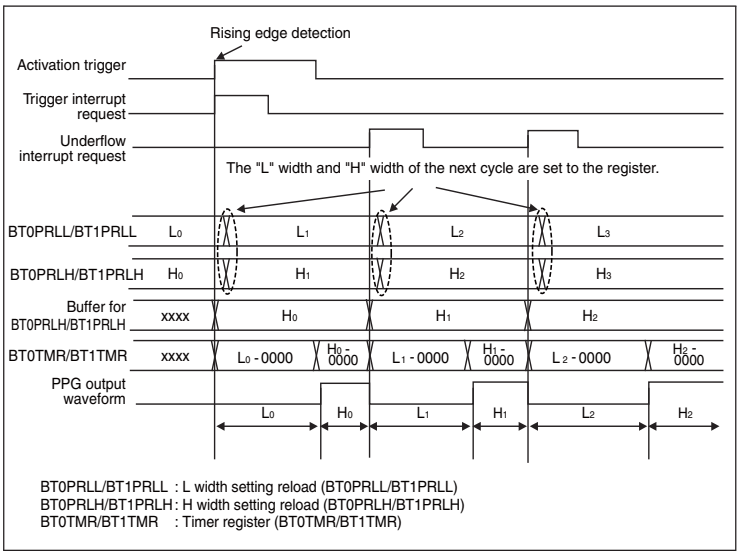
- An activation trigger is detected.
- An underflow occurs after counting down from the value of the H width setting reload register (BT0PRLH/BT1PRLH) is completed.

The content of the buffer is loaded to the 16-bit down counter in the following event:

- Counting down from the value of the L width setting reload register (BT0PRLL/BT1PRLL) is completed.

Therefore, rewrite the L width setting reload register (BT0PRLL/BT1PRLL) and H width setting reload register (BT0PRLH/BT1PRLH) during the period from the time an underflow occurs (the UDIR bit of the status control register (BT0STC/BT1STC) changes to "1") to the time counting based on the next cycle begins. The new data will be effective as the next cycle.

Figure 40.5-19 Write Timing



■ Interrupt Generation Timing

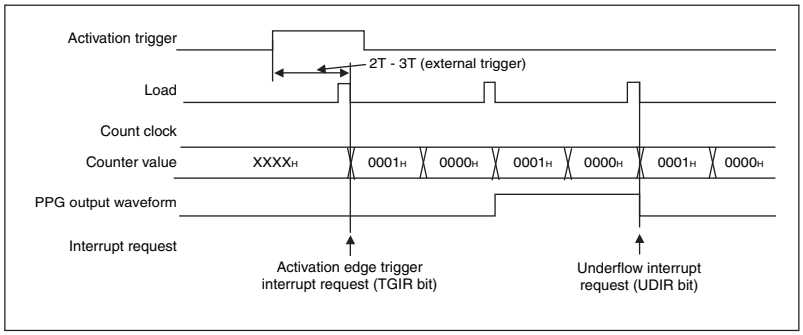
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- An activation trigger is detected.
- An underflow occurs based on the value of H width setting reload register (BT0PRLH/BT1PRLH).

An example of interrupt request generation timing using the following settings is shown below.

- Value of L width setting reload register (BT0PRLL/BT1PRLL) = 0001_H
- Value of H width setting reload register (BT0PRLH/BT1PRLH) = 0001_H

Figure 40.5-20 Interrupt Request Generation Timing Chart



40.5.6.4 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

■ Counting Operation

● Activation

The same operation as in reload mode. See "■ Operation" in Section "40.5.6.3 Operation in Reload Mode".

● Counting Operation

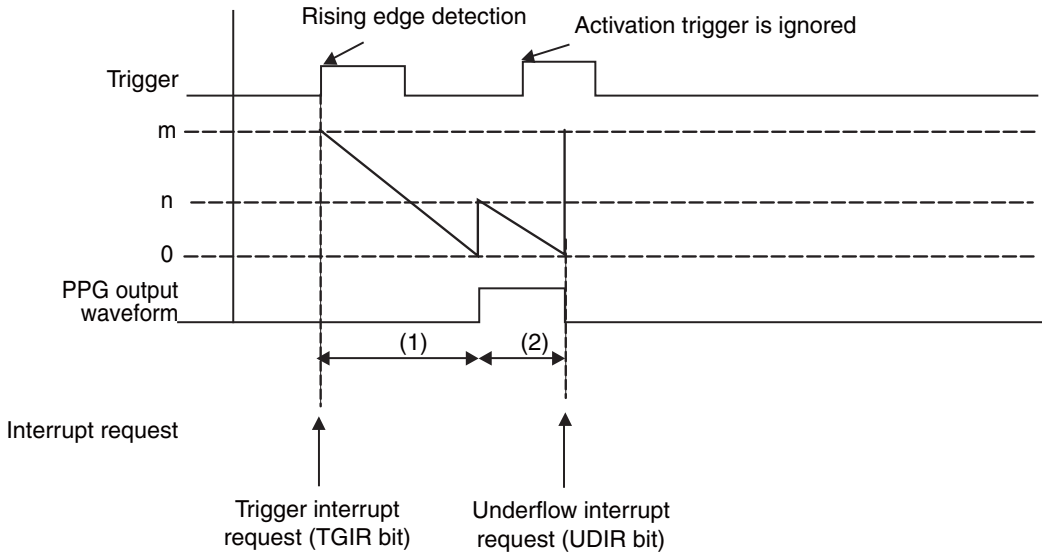
Counting operation initiated by the entry of an activation trigger is explained below, using an example where the OSEL bit of the timer control register (BT0TMCR/BT1TMCR) is set for normal polarity (OSEL=0).

1. The value set in the L width setting reload register (BT0PRLL/BT1PRLL) is transferred to the 16-bit down counter and the value set in the H width setting reload register (BT0PRLH/BT1PRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting reload register (BT0PRLL/BT1PRLL). The output signal (TOUT signal) is at the "L" level.
2. The 16-bit down counter completes counting down from the value of L width setting reload register (BT0PRLL/BT1PRLL).
3. The buffered value of H width setting reload register (BT0PRLH/BT1PRLH) is reloaded to the 16-bit down counter, which continues counting down. The output signal (TOUT signal) is at the "H" level.
4. The 16-bit down counter completes counting down from the value of H width setting reload register (BT0PRLH/BT1PRLH), 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 (BT0TMCR/BT1TMCR).

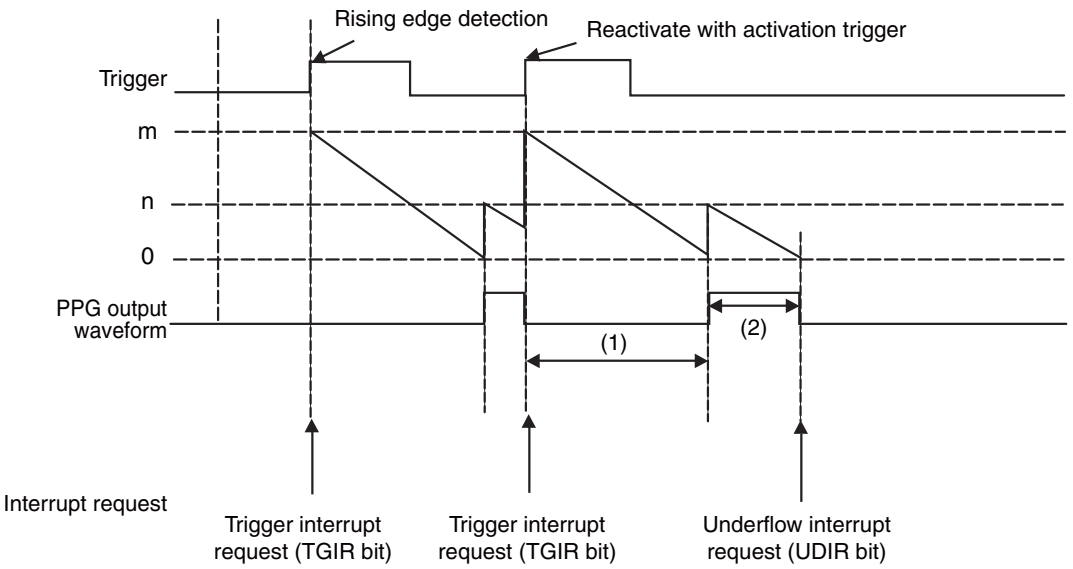
- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.
- If reactivation is enabled (RTGEN=1): The TGIR bit of status control register (BT0STC/BT1STC) changes to "1". In addition, the value of L width setting reload register (BT0PRLL/BT1PRLL) is reloaded to the 16-bit down counter, which starts counting.

Figure 40.5-21 Example of Counting Operation If Reactivation Is Not Enabled



(1) = T(m+1)ms
(2) = T(n+1)ms
m : Value of L width setting reload register (BT0PRLL/BT1PRLL)
n : Value of H width setting reload register (BT0PRLH/BT1PRLH)
T : Cycle of count clock

Figure 40.5-22 Example of Counting Operation If Reactivation Is Enabled



(1) = T(m+1)ms
(2) = T(n+1)ms
m : Value of L width setting reload register (BT0PRLL/BT1PRLL)
n : Value of H width setting reload register (BT0PRLH/BT1PRLH)
T : Cycle of count clock

(Notes)
- The output method and output destination of the output signal (TOUT signal) 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 (BT0PRLL/BT1PRLL) is loaded to the 16-bit down counter, which starts counting.

■ Interrupt Generation Timing

The same operation as in reload mode. See "■ Interrupt Generation Timing" in Section "40.5.6.3 Operation in Reload Mode".

40.5.6.5 Interrupt

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 (BT0PRLH/BT1PRLH). (underflow interrupt request)

Table 40.5-5 Conditions for Interrupt Generation

Interrupt request	Interrupt request flag	Permission of interrupt request	Interrupt request clear
Trigger interrupt request	BT0STC/BT1STC TGIR=1	BT0STC/BT1STC TGIE=1	Set the TGIR bit of BT0STC/BT1STC to "0".
Underflow interrupt request	BT0STC/BT1STC UDIR=1	BT0STC/BT1STC UDIE=1	Set the UDIR bit of BT0STC/BT1STC 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 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

40.5.6.6 Application Notes

Note the following when using the 16-bit PPG timer:

■ Notes on Program Setting

- Change the following bits of the timer control register (BT0TMCR/BT1TMCR) only after stopping the 16-bit down counter by resetting the CTEN bit to "0".
  - CKS3 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 (BT0TMCR/BT1TMCR) 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 (BT0TMCR/BT1TMCR) 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 (BT0TMCR/BT1TMCR) to "010".
  2. Set the L width setting reload register (BT0PRLL/BT1PRLL).
  3. Set the H width setting reload register (BT0PRLH/BT1PRLH).

■ 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 (BT0PRLL/BT1PRLL) is loaded to the 16-bit down counter, which starts counting.
- A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

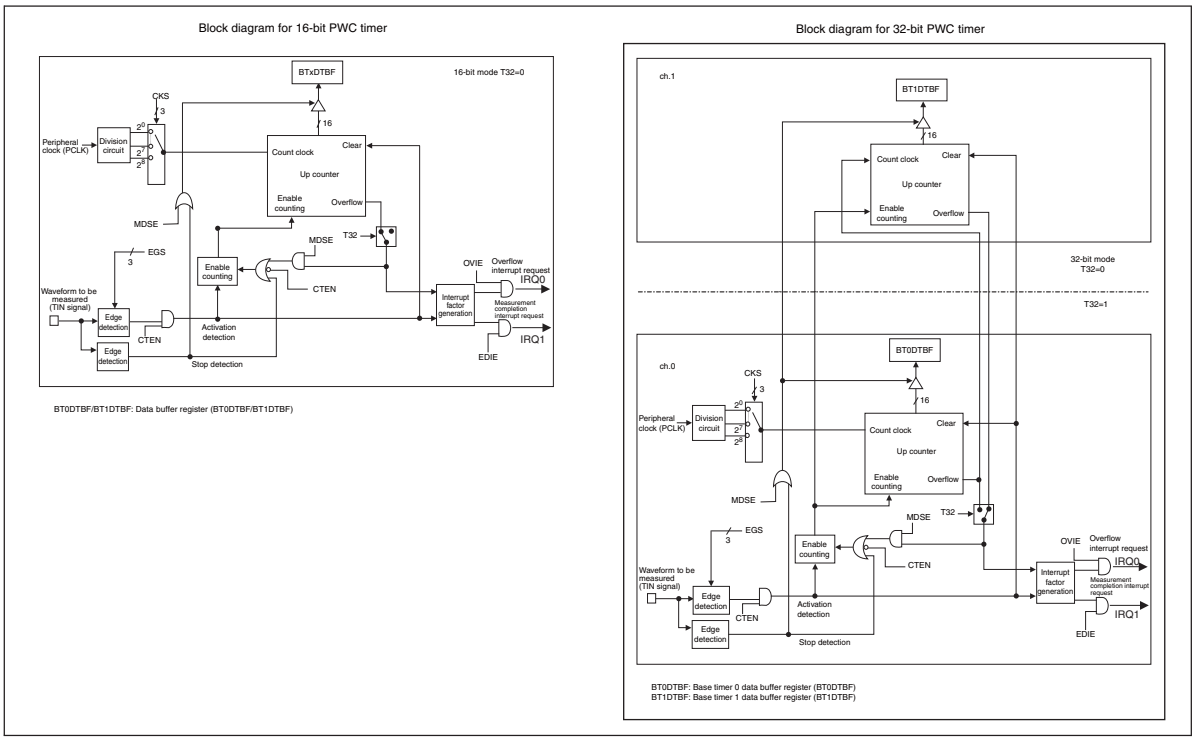
■ Notes on Interrupts

- If to clear the interrupt request flag and to change the interrupt request flag to "1" occur at the same time, the flag clear is ignored. The interrupt request flag is held to "1".

40.5.7 16/32-bit PWC Timer Operation

This section explains the operation performed when the base timer included in this product is used as the 16/32-bit PWC timer. An example is also given to set various operation conditions.

Figure 40.5-23 Block Diagram of 16-bit PWC Timer (16/32-bit PWC Timer Operation)



40.5.7.1 Overview

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 signal), 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 (BT0DTBF/BT1DTBF).

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 TIN signal varies depending on the I/O mode that has been set by the I/O select register (BTSEL01). See Section "40.5.2 I/O Assignment".

■ Timer Mode

Either of the following timer modes can be selected using the T32 bit of timer control register (BT0TMCR/BT1TMCR).

- 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.

For the 32-bit timer mode operations, see Section "40.5.7.3 32-bit Timer Mode Operation".

(Note) The T32 bit setting differs between odd-numbered and even-numbered channels when the 32-bit timer mode is selected. For details, see Section "40.5.7.3 32-bit Timer Mode Operation".

■ Operation Mode

Either of the following two modes can be selected using the MDSE bit of timer control register (BT0TMCR/BT1TMCR).

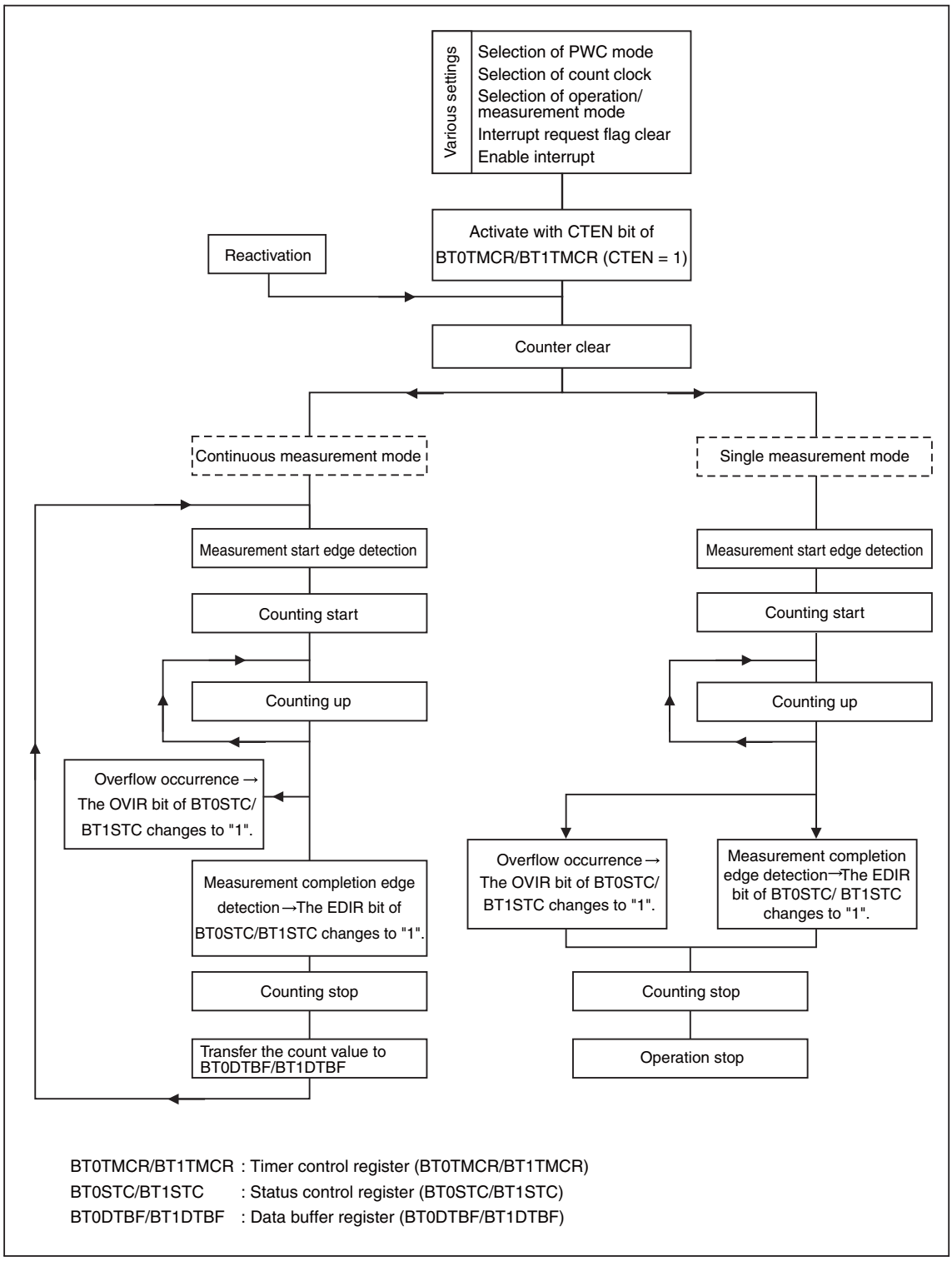
- Continuous measurement mode (MDSE=0): In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
- Single measurement mode (MDSE=1): In this mode, measurement is conducted only once. Differences between the single and continuous measurement modes are listed on the table below.

Table 40.5-6 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.
BT0DTBF/BT1DTBF 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 0000_H

The following shows the standard operation flow.

Figure 40.5-24 Operation Flow

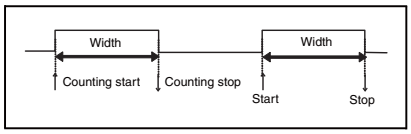
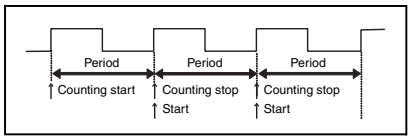
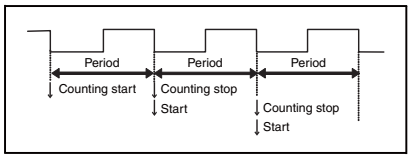
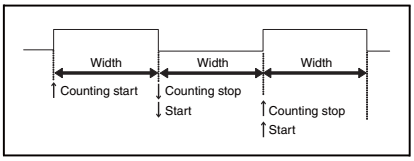
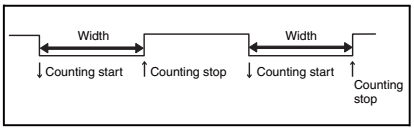


(Notes) In the continuous measurement mode, if the next measurement is completed before the measurement result has been read from the data buffer register (BT0DTBF/BT1DTBF), the value being held by the data buffer register (BT0DTBF/BT1DTBF) is overwritten by the new value. The old value is discarded. If it has occurred, the ERR bit of status control register (BT0STC/BT1STC) changes to "1". This ERR bit is cleared to "0" when a value is read from the data buffer register (BT0DTBF/BT1DTBF).

■ Measurement Mode

Either of the following five modes can be selected using EGS2 to EGS0 bits of timer control register (BT0TMCR/BT1TMCR).

Figure 40.5-25 Measurement Modes and their Explanation

Measurement mode (EGS2-EGS0)	Measurement content
Measurement of H pulse width (EGS2-EGS0=000)	The width of the "H" level signal period is measured.  Counting (measurement) start: at rising edge detection Counting (measurement) completion: at falling edge detection
Measurement of the cycle between rising edges (EGS2-EGS0=001)	The cycle from the rising edge detection to the next rising edge detection is measured.  Counting (measurement) start: at rising edge detection Counting (measurement) completion: at rising edge detection
Measurement of the cycle between falling edges (EGS2-EGS0=010)	The cycle from the falling edge detection to the next falling edge detection is measured.  Counting (measurement) start: at falling edge detection Counting (measurement) completion: at falling edge detection
Measurement of the pulse width between all edges (EGS2-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  Counting (measurement) start: at edge detection Counting (measurement) completion: at edge detection
Measurement of L pulse width (EGS2-EGS0=100)	The width of the "L" level signal period is measured.  Counting (measurement) start: at falling edge detection Counting (measurement) completion: at rising edge detection

40.5.7.2 Operation during PWC Measurement

This section explains the operations during measurement. For explanation of "sensitive edges" (1) and (2) described below, see "Figure 40.5-25 Measurement Modes and their Explanation".

■ 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 timer control register (BT0TMCR/BT1TMCR) to "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 signal) 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 signal), the up counter stops from operating. During this time, the up counter value is stored in the data buffer register (BT0DTBF/BT1DTBF). An interrupt request can be generated at the end of measurement or at an occurrence of overflow.

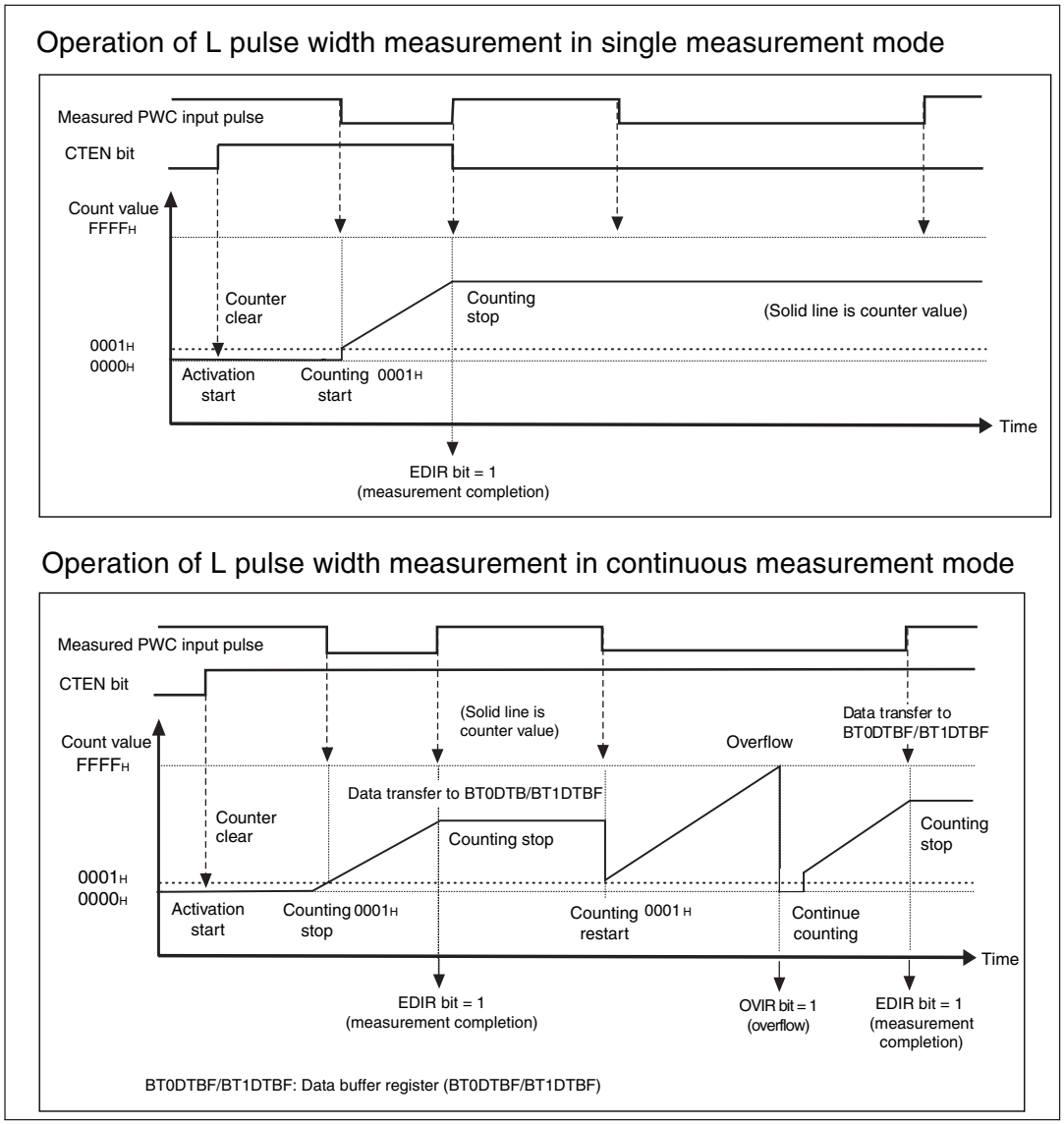
(Notes)
- In the single measurement mode, the counting stops if an overflow occurs.
- The input method of waveforms to be measured (TIN signal) varies depending on the I/O mode that has been set by the I/O select register (BTSEL01).

● Operation in continuous measurement mode

If sensitive edge (1) is detected in the input signal (TIN signal) 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 signal), 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 (BT0DTBF/BT1DTBF). If a rising edge of input signal (TIN signal) 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 select register (BTSEL01).

Figure 40.5-26 Operation Example



■ Reactivation

If the CTEN bit of timer control register (BT0TMCR/BT1TMCR) 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 status control register (BT0STC/BT1STC) 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 status control register (BT0STC/BT1STC) is set to "1". Also, the current measurement result is transferred to the data buffer register (BT0DTBF/BT1DTBF).

• 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 signal) simultaneously, the timer immediately starts counting from value "0001_H".

■ Calculating the Pulse Width

After the measurement, the measurement result can be read from the data buffer register (BT0DTBF/BT1DTBF) and the measured pulse width can be calculated using the following formula.

Pulse width = n × T

n: Data buffer register (BT0DTBF/BT1DTBF) value

T: Count clock cycle

40.5.7.3 32-bit Timer Mode Operation

This section explains the setting and operation for cascading 2 channels of a 16-bit PWC timer and using it as a 32-bit PWC timer.

■ Overview

Using the T32 bit of the timer control register (BT0TMCR/BT1TMCR), 2 channels of a 16-bit PWC timer can be cascaded and used as a 32-bit PWC timer.

In this mode, the even-numbered channel corresponds to the lower 16-bit operation, and the odd-numbered channel corresponds to the upper 16-bit operation. Therefore, the up counter must be read in the order of the lower 16 bits (even-numbered channel) → the upper 16 bits (odd-numbered channel).

■ Setting Procedure Example

To select the 32-bit timer mode, set the T32 bit of timer control register (BT0TMCR/BT1TMCR) of the even-numbered channel to "1". Also, set the T32 bit of the odd-numbered channel to "0". When selecting the 32-bit timer mode, set each register as follows.

The register setting differs between even-numbered and odd-numbered channels. In this example, channel 0 and channel 1 are connected by cascading.

1. Specify ch.0 to reset mode by setting FMD2 to FMD0 bits of base timer 0 timer control register (BT0TMCR). (FMD2 to FMD0 = 000)
2. Set the 16/32-bit PWC timer for channels 0 and 1 by setting the FMD2 to FMD0 bits of timer control registers (BT0TMCR/BT1TMCR) of channels 0 and 1. (FMD2 to FMD0 = 100) At the same time, select the 32-bit timer mode by setting the T32 bit of base timer 0 timer control register (BT0TMCR). (T32=1)

(Note) Rewrite the T32 bit while the operation of both of the even-numbered and odd-numbered channels is stopped. Whether the counting operation is stopped can be checked by setting the CTEN bit of the timer control register (BT0TMCR/BT1TMCR) to "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 is based on the even-numbered channel settings and the following register settings of odd-numbered channel are ignored.

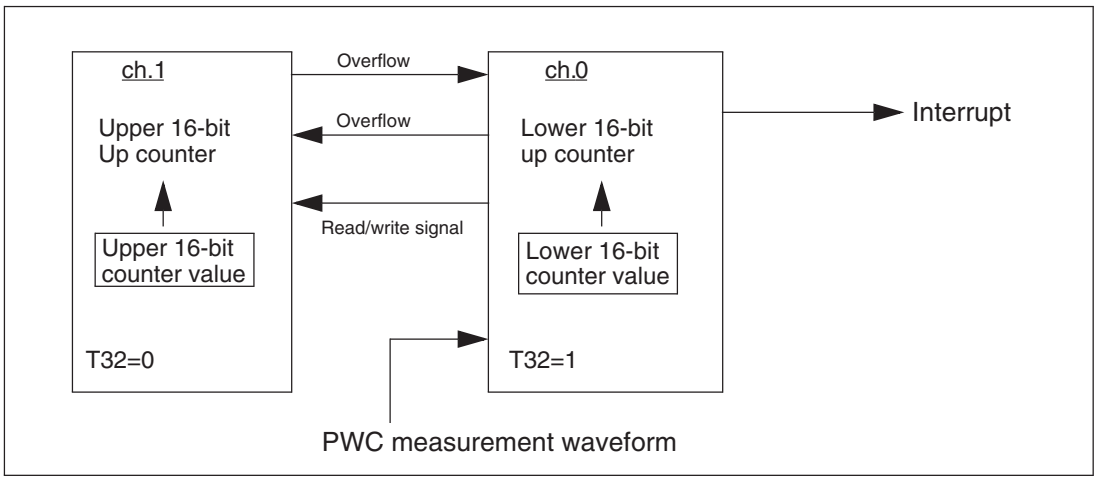
- Timer control register (BT0TMCR/BT1TMCR)
- Status control register (BT0STC/BT1STC)

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 timer control register (BT0TMCR/BT1TMCR) (by setting CTEN=1) of the even-numbered channel, the 32-bit PWC timer starts.
2. When a measurement start edge is detected in the input signal (TIN signal), 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 is stored in the data buffer register (BT0DTBF) of the even-numbered channel. Also, the upper 16-bit data is stored in the data buffer register (BT1DTBF) of the odd-numbered channel.

The channel configuration in 32-bit timer mode is shown below.

Figure 40.5-27 Configuration in 32-bit Timer Mode



(Notes)
- The down counter value can be checked by reading the data buffer register (BT0DTBF/BT1DTBF). In the 32-bit timer mode, it must be read in the order of the lower 16 bits (even-numbered channel) → upper 16 bits (odd-number channel).
- In 32-bit timer mode, the operation of the 32-bit PWC timer conforms to the settings of the even-number channel. Therefore, an interrupt request of the even-numbered channel is effective.

40.5.7.4 Interrupt

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 40.5-7 Conditions for Interrupt Generation

Interrupt request	Interrupt request flag	Permission of interrupt request	Interrupt request clear
Overflow interrupt request	BT0STC/BT1STC OVIR=1	BT0STC/BT1STC OVIE=1	Write the OVIR bit of BT0STC/ BT1STC to "0"
Measurement end interrupt request	BT0STC/BT1STC EDIR=1	BT0STC/BT1STC EDIE=1	Read BT0DTBF/BT1DTBF

(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 "Table C-1" in "APPENDIX C. Interrupt Vector".
- 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 22 INTERRUPT CONTROL (INTERRUPT CONTROLLER)".

40.5.7.5 Application Notes

Note the following when using the 16/32-bit PWC timer:

■ Notes on Program Setting

- Change the following bits of the timer control register (BT0TMCR/BT1TMCR) after stopping the up counter by resetting CTEN bit to "0".
  - CKS3 to CKS0 bits
  - EGS2 to EGS0 bits
  - T32 bit
  - FMD2 to FMD0 bits
  - MDSE bit
  - All registers are initialized if the FMD2 to FMD0 bits of timer control register (BT0TMCR/ BT1TMCR) 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 FMD2 to FMD0 bits or T32 bit of the timer control register (BT0TMCR/ BT1TMCR) after a reset, be sure to reset the FMD2 to FMD0 bits to "000" to select the reset mode. Then, rewrite 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 timer control register (BT0TMCR/BT1TMCR) to "100".

- Enable 16/32-bit PWC timer operation by setting the CTEN bit of the timer control register (BT0TMCR/BT1TMCR) to "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 timer control register (BT0TMCR/BT1TMCR) to "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 signal) simultaneously, the timer immediately starts counting from value "0001_H".
- If two channels of PWC timers are used as a single 32-bit PWC timer, the 16-bit PWC timer setting of the even-numbered channel is made valid. The timer setting of odd-numbered channel is ignored.
- The input operation of measurement waveforms varies depending on the base timer I/O selection function.

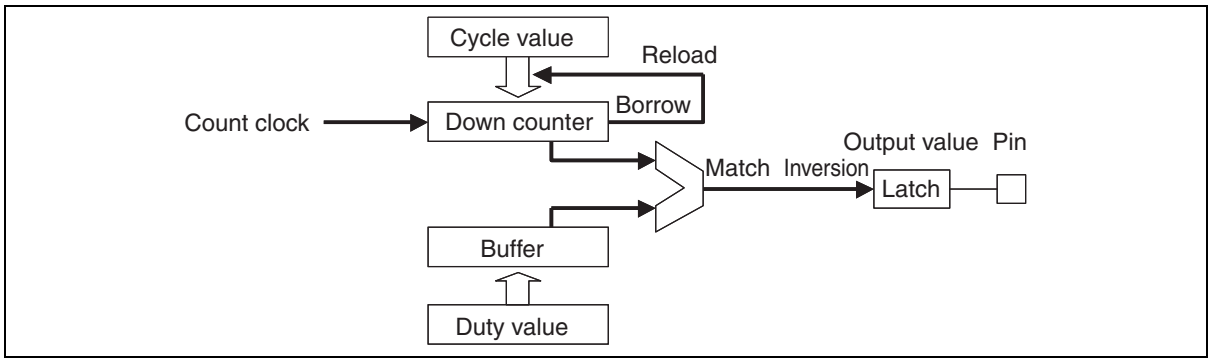
■ Notes on Interrupts

- If to clear the interrupt request flag and to change the interrupt request flag to "1" occur at the same time, the flag clear 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 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 (BT0DTBF/BT1DTBF).

CHAPTER 41 PPG

41.1 Overview

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.

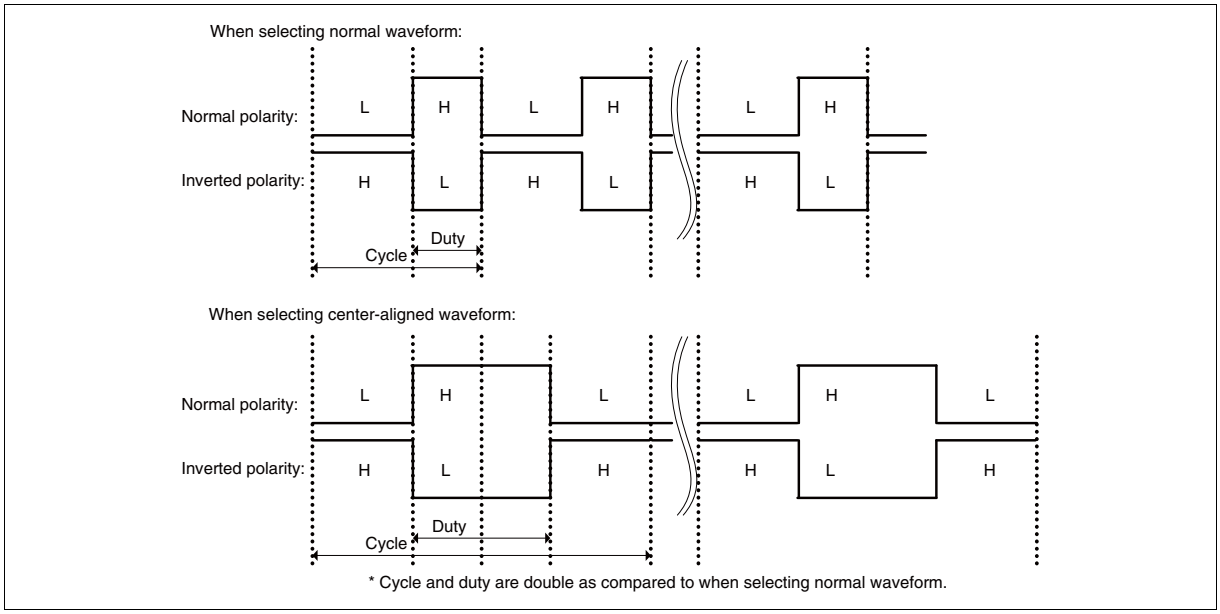


41.2 Features

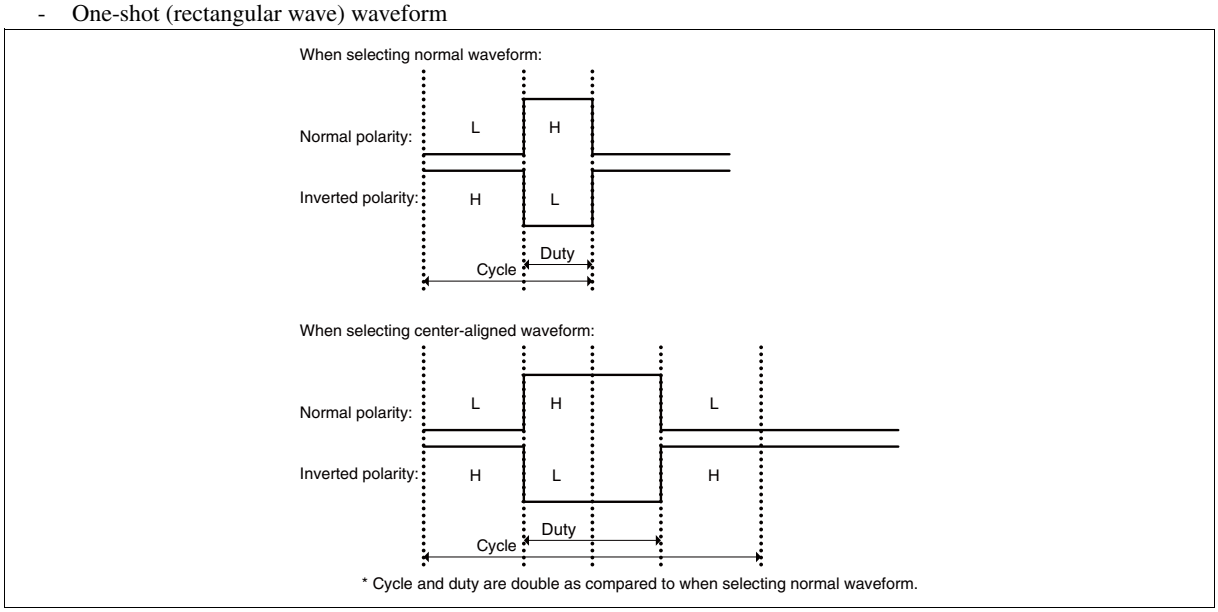
- Clamp output
  - Normal polarity: Output clamped to "L"
  - Inverted polarity: Output clamped to "H"
- Count clock
  - One of the following four count clocks is selected:
    Outputs obtained by dividing the frequency of the peripheral clock by 1, 4, 16, and 64.
- Cycle
  - Setting range is from the duty value to 65535 (specified by a 16-bit register).
  - Cycle = count clock x (PCSR register value + 1)
  - Example: Count clock = 32 MHz (31.25 ns), PCSR value = 63999
    Cycle = 31.25ns × (63999+1) = 2ms
- Duty
  - Setting range is from 0 to cycle value (specified by a 16-bit register).
  - Duty = count clock (PDUT register value + 1)

Code: FS30-3v1-91580MS-1-E

- The PPG can output the following types of waveforms:
  - PWM waveform



  - One-shot (rectangular wave) waveform

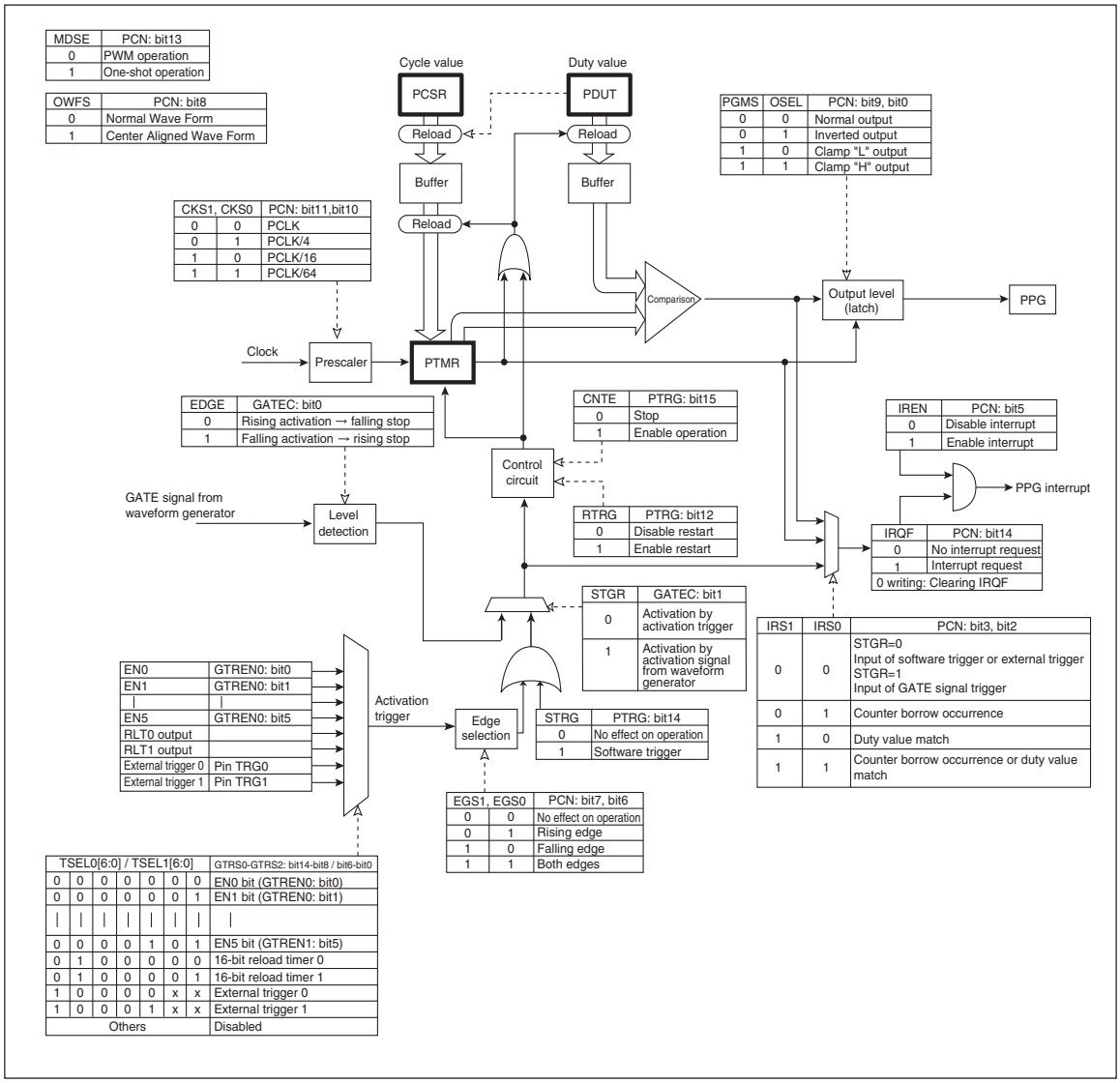


- Interrupts
  - One of the following four interrupts is selected:
    - Software trigger or external trigger (TRG pin)
    - Borrow occurrence on the counter (match with the specified cycle)
    - Duty match
    - Borrow occurrence on the counter (match with the specified cycle) or duty match
- Activation trigger
  - Software trigger
  - Activation trigger
    The activation trigger is selected from one of the following triggers:
    - Internal trigger (EN0 to EN5)
    - External trigger (TRG pins 0, 1)
    - Reload timer 0/1
- GATE function
  - PPG is activated/stopped by GATE signals from the waveform generator.

41.3 Configuration

■ Block Diagram of PPG

Figure 41.3-1 Block Diagram of PPG



41.4 Registers

The PPG, PPG control, and PPG SEL registers are listed below.

■ List of PPG Registers

Table 41.4-1 List of PPG Registers

Address	+0	+1	+2	+3
0x0200	PPG (ch.0) control status register (PCN0)		PPG (ch.0) cycle setting register (PCSR0)	
0x0204	PPG (ch.0) duty setting register (PDUT0)		PPG (ch.0) timer register (PTMR0)	
0x0208	PPG (ch.1) control status register (PCN1)		PPG (ch.1) cycle setting register (PCSR1)	
0x020C	PPG (ch.1) duty setting register (PDUT1)		PPG (ch.1) timer register (PTMR1)	
0x0210	PPG (ch.2) control status register (PCN2)		PPG (ch.2) cycle setting register (PCSR2)	
0x0214	PPG (ch.2) duty setting register (PDUT2)		PPG (ch.2) timer register (PTMR2)	
0x0218	PPG (ch.3) control status register (PCN3)		PPG (ch.3) cycle setting register (PCSR3)	
0x021C	PPG (ch.3) duty setting register (PDUT3)		PPG (ch.3) timer register (PTMR3)	
0x0220	PPG (ch.4) control status register (PCN4)		PPG (ch.4) cycle setting register (PCSR4)	
0x0224	PPG (ch.4) duty setting register (PDUT4)		PPG (ch.4) timer register (PTMR4)	
0x0228	PPG (ch.5) control status register (PCN5)		PPG (ch.5) cycle setting register (PCSR5)	
0x022C	PPG (ch.5) duty setting register (PDUT5)		PPG (ch.5) timer register (PTMR5)	

■ List of PPG Control Registers

Table 41.4-2 List of PPG Control Registers

Address	+0	+1	+2	+3
0x02C0	General-purpose trigger selection register 0 (GTRS0)		General-purpose trigger selection register 1 (GTRS1)	
0x02C4	General-purpose trigger selection register 2 (GTRS2)		Reserved	
0x02D8	General-purpose trigger setting register 0 (GTREN0)		Reserved	

■ List of GATE Function Control Registers

Table 41.4-3 GATE Function Registers List

Address	+0	+1	+2	+3
0x02E0	Reserved	GATE function control register 0 (GATEC0)	Reserved	GATE function control register 2 (GATEC2)
0x02E4	Reserved	GATE function control register 4 (GATEC4)	Reserved	Reserved

41.4.1 PPG Control Status Register (PCN0-5)

The PPG control status register (PCN) controls the operation and status of the PPG.

- **PCN0 to PCN5: Address 0200_H, 0208_H, . . .0228_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	OWFS	
0	0	0	0	0	0	0	0	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
EGS1	EGS0	IREN	IRQF	IRS1	IRS0	Reserved	OSEL	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R(RM1)/W	R/W	R/W	R/W0	R/W	Attribute

[bit15] CNTE: Timer operation enable bit

CNTE	Function
0	Inactive
1	Active

Timer operation enable bit

If this bit is set to "0", the PPG is deactivated.

If this bit is set to "1", the PPG is activated.

[bit14] STRG: Software trigger bit

STRG	Function
0	PPG operation is not influenced by the value written to this bit (which is always "0" when read).
1	A software trigger is generated.

Software trigger bit

If this bit is set to "0", PPG operation is not influenced by the 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	Function
0	PWM operation
1	One-shot operation

This bit selects types of output waveform.

If this bit is set to "0", the PPG is enabled to perform PWM operation, thus generating a sequence of pulses.

If this bit is set to "1", the PPG generates only one pulse.

(Note) This bit is effective for each cycle (trigger generation or counter borrow occurrence).

[bit12] RTRG: Restart enable bit

RTRG	Function
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, an internal factor, or an external factor).

[bit11, bit10] CKS1, CKS0 : Count clock selection bits

CKS1	CKS0	Function
0	0	Peripheral clock (PCLK)
0	1	Division by 4 of the peripheral clock frequency
1	0	Division by 16 of the peripheral clock frequency
1	1	Division by 64 of the peripheral clock frequency

Down counter count clock selection bits

[bit9] PGMS: PPG output mask selection bit

PGMS	Function
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.

(Notes)
- 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 setting this bit from "1" to "0" to release the PPG output mask, set it in a period from the start of a cycle to a duty match.

[bit8] PPG output waveform selection bit

OWFS	Function
0	Normal waveform
1	Center-aligned waveform

This bit selects a PPG output waveform.

If this bit is set to "0", the PPG outputs a normal waveform.

If this bit is set to "1", the PPG outputs a center-aligned waveform.

(Note) This bit is effective for each cycle (trigger generation or counter borrow occurrence).

[bit7, bit6] EGS1, EGS0: Trigger input edge selection bits

EGS1	EGS0	Function
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)

For the trigger input selected with each PPG activation trigger selection bit (GTRS0 to GTRS2.TSEL1/0), select the edge causing the activation with this bit.

(Note)
- If EGS1 = 0 and EGS0 = 0, only a software trigger (PCN:STRG) is enabled. Input from other than a software trigger is disabled.
- The EGS1=1 or EGS0=1 setting affects only the input from a trigger other than a software trigger.
- Writing "1" to the software trigger (PCN:STRG) is not influenced by the PCN:EGS1, EGS0 setting.

[bit5] IREN: Interrupt request enable bit

IREN	Function
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	Read	Write
0	No interrupt request	Clears the interrupt request flag.
1	Interrupt request present	Writing of "1" does not influence operation.

Interrupt request flag bit

(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	Function
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

These bits select the operation that generates an interrupt request.

(Notes) 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 41.5-1 Example of PWM Operation (Normal Waveform Selected),
  - Figure 41.5-3 Example of One-shot Operation (Normal Waveform Selected)
- In the case of the PPG output waveform selection bit (OWFS = 1):
  - Figure 41.5-2 Example of PWM Operation (Center-aligned Waveform Selected),
  - Figure 41.5-4 Example of One-shot Operation (Center-aligned Waveform Selected)

[bit1] Reserved

Should be written "0".

[bit0] OSEL: PPG output polarity selection bit

OSEL	Function
0	Normal polarity
1	Inverted polarity

This bit selects a 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.

* The 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 when they are written.

41.4.2 PPG Cycle Setting Register (PCSR 0 to PCSR5)

The PPG cycle setting register (PCSR) specifies the cycle of the PPG output waveform.

- **PCSR0 to PCSR5:0202$_H$, 020A$_H$, . . .022A$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
x	x	x	x	x	x	x	x	Initial value
R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
x	x	x	x	x	x	x	x	Initial value
R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	Attribute

[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.

If you write to the PPG cycle setting register, be sure also to set the PPG duty setting register (PDUT).

(Notes)
- The set cycle value must be the value that is effective when the PPG output waveform selection bit (PCN:OWFS) = 0 is selected (normal waveform).
- If the PPG output waveform selection bit (PCN:OWFS) = 1 is selected (center-aligned waveform), the cycle of the PPG output waveform is two times the value set in the PPG cycle setting register.

41.4.3 PPG Duty Setting Register (PDUT0 to PDUT5)

The PPG duty setting register (PDUT) specifies the duty of the PPG output waveform.

- **PDUT0 to PDUT5: Address 0204_H, $020C_H$, . . .$022C_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
x	x	x	x	x	x	x	x	Initial value
R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
x	x	x	x	x	x	x	x	Initial value
R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	R1, W	Attribute

[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.

(Notes)
- In the PPG duty setting register, set a smaller value than the value set in the PPG cycle setting register (PCSR).
- The set duty value must be the value that is effective when the PPG output waveform selection bit (PCN:OWFS) = 0 is selected (normal waveform).
- If the PPG output waveform selection bit (PCN:OWFS) = 1 is selected (center-aligned waveform), the duty of the PPG output waveform is two times the value set in the PPG duty setting register.

41.4.4 PPG Timer Register (PTMR0 to PTMR5)

The PPG timer register (PTMR) allows the PPG timer count down value to be read.

- **PTMR0 to PTMR5: Address 0206_H, $020E_H$, . . .$022E_H$ (Access: Half-word, Word)**

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
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
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	R, WX	Attribute

[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 16-bit down counter can be read from these bits.

•If the normal waveform (OWFS = 0) is selected, the lower 16 bits are read.

•If the center-aligned waveform (OWFS = 1) is selected, the upper 16 bits are read.

41.4.5 General-purpose Trigger Selection Register (GTRS0 to GTRS2)

The general-purpose trigger selection register (GTRS) is used to select a trigger input to the PPG.

- **GTRS0 to GTRS2: Address $02C0_H$ to $02C4_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
-	TSEL1[6:0]							
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

7	6	5	4	3	2	1	0	bit
-	TSEL0[6:0]							
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

[bit15] Reserved

Always set this bit to "0".

[bit14 to bit8] TSEL1: Activation trigger selection bits

These bits select an activation trigger for PPG1, 3, and 5.

TSEL1 [6:0]							Activation trigger selection bits
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	1	0	0	0	0	0	16-bit reload timer 0 selected
0	1	0	0	0	0	1	16-bit reload timer 1 selected
1	0	0	0	0	x	x	External trigger 0 selected
1	0	0	0	1	x	x	External trigger 1 selected
Others							Disabled

(Note) A trigger input to PPG is selected.
On selected PPG, the PPG will be activated when the edge selected by the trigger input edge selection bits (PCN:EGS1, EGS0) is detected upon the selected activation trigger.

[bit7] Reserved

Always set this bit to "0".

[bit6 to bit0] TSEL0: Activation trigger selection bits

These bits select an activation trigger for PPG0, 2, and 4.

TSEL0 [6:0]							Activation trigger selection bits
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	1	0	0	0	0	0	16-bit reload timer 0 selected
0	1	0	0	0	0	1	16-bit reload timer 1 selected
1	0	0	0	0	x	x	External trigger 0 selected
1	0	0	0	1	x	x	External trigger 1 selected
Others							Disabled

(Note) A trigger input to PPG is selected.
On selected PPG, the PPG will be activated when the edge selected by the trigger input edge selection bits (PCN:EGS1, EGS0) is detected upon the selected activation trigger.

41.4.6 General-purpose trigger setting register (GTREN0)

The general-purpose trigger setting register (GTREN) controls the generation of internal trigger to the PPG.

- **GTREN0: Address 02D8$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	Attribute

7	6	5	4	3	2	1	0	bit
Reserved		EN5	EN4	EN3	EN2	EN1	EN0	
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W0	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit6] Reserved

Always set these bits to "0".

[bit5 to bit0] EN5 to EN0: Internal trigger input bits

EN5 to EN0	Function
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 EN5) is selected by the PPG activation trigger selection bits (TSEL0[6:0], TSEL1[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 EN5), the trigger input bit services as the PPG activation trigger.

41.4.7 GATE Function Control Register GATEC0, 2, and 4

- **GATEC0, 2 and 4: Address $02E1_H$, $02E3_H$, and $02E5_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved						STGR	EDGE	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	Attribute

[bit7 to bit2] Reserved

Always set these bits to "0".

[bit1] STGR:GATE function selection bit

STGR	Function
0	PPG is activated by an activation trigger.
1	PPG is activated/stopped by GATE signals from the waveform generator.

[bit0] EDGE: Activation effective edge selection bit

EDGE	Function
0	PPG is activated at the rising of GATE signals and stopped at the falling of them. It is activated during "H".
1	PPG is activated at the falling of GATE signals and stopped at the rising of them. It is activated during "L".

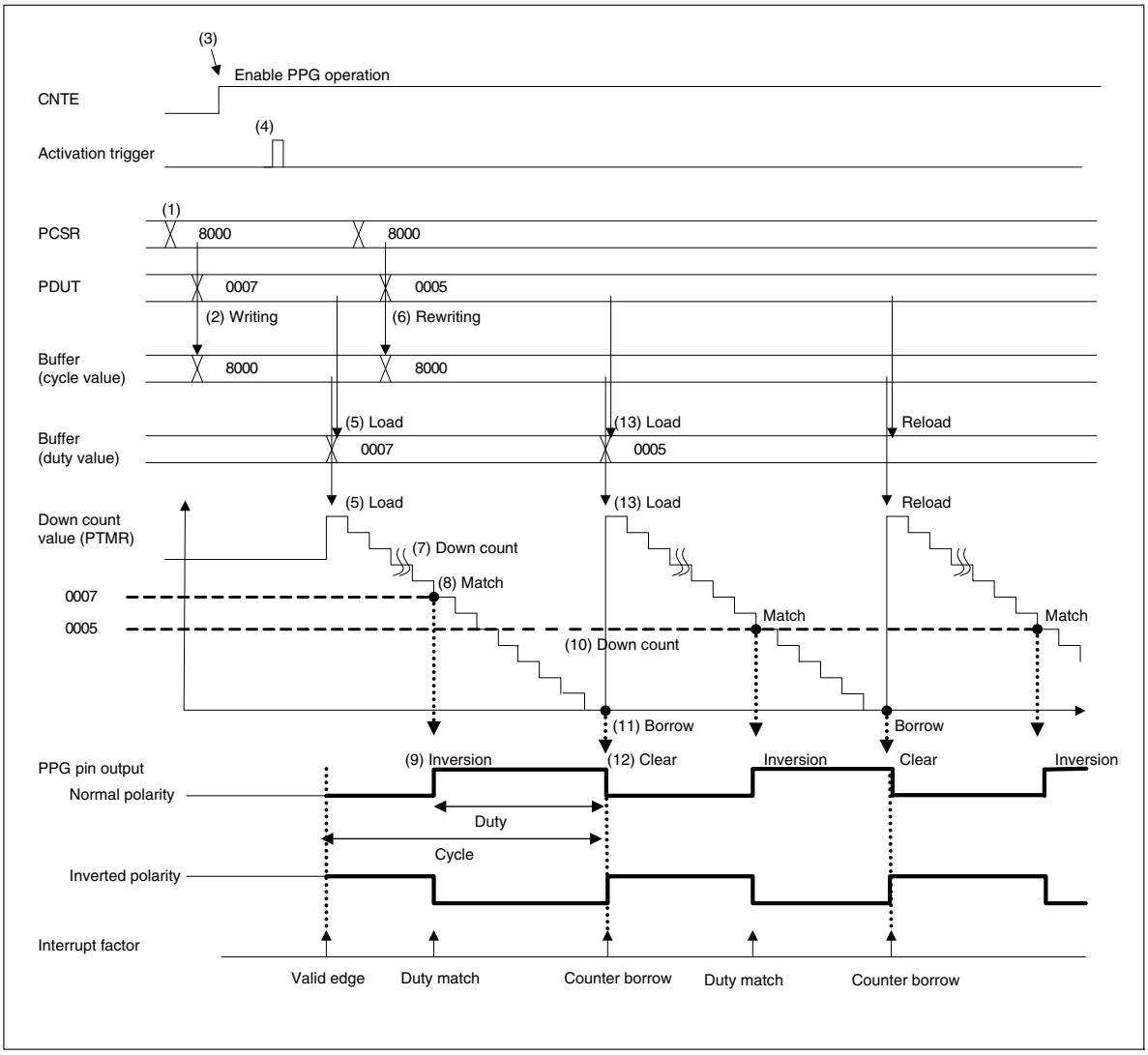
41.5 Operation Explanation

This section explains PPG operation.

41.5.1 PWM Operation (Normal Waveform Selected)

During PWM operation, variable-duty pulses are output at the PPG pin.

Figure 41.5-1 Example of PWM Operation (Normal Waveform Selected)



(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 (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 (PTMR) and the duty value to the buffer (duty value)
(14) Repetition of steps (7) to (13)

Calculation formulas:

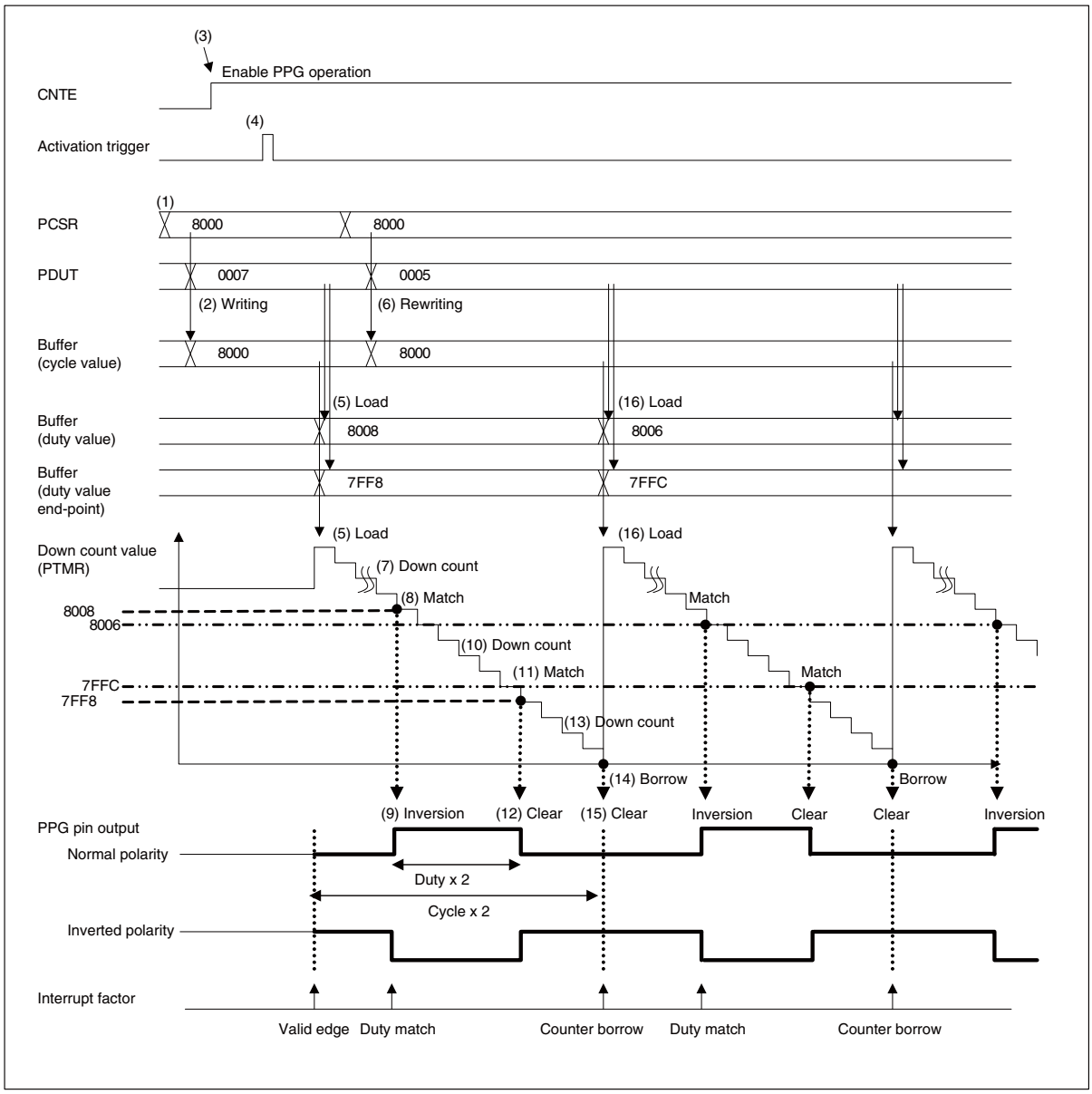
- Cycle = {Cycle value (PCSR) + 1} × [Count clock]
- Duty = {Duty value (PDUT) + 1} × [Count clock]
- Time to pulse output = [Cycle value (PCSR) - duty value (PDUT)] x [Count clock]

41.5.2 PWM Operation (Center-aligned Waveform Selected)

During 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 waveform), 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 41.5-2 Example of PWM Operation (Center-aligned Waveform Selected)



(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 (PTMR) and the duty value to the buffer (duty value) and 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 (PTMR) and the duty value to the buffer (duty value) and buffer (duty value end point).
(17) Repetition of steps (7) to (16)

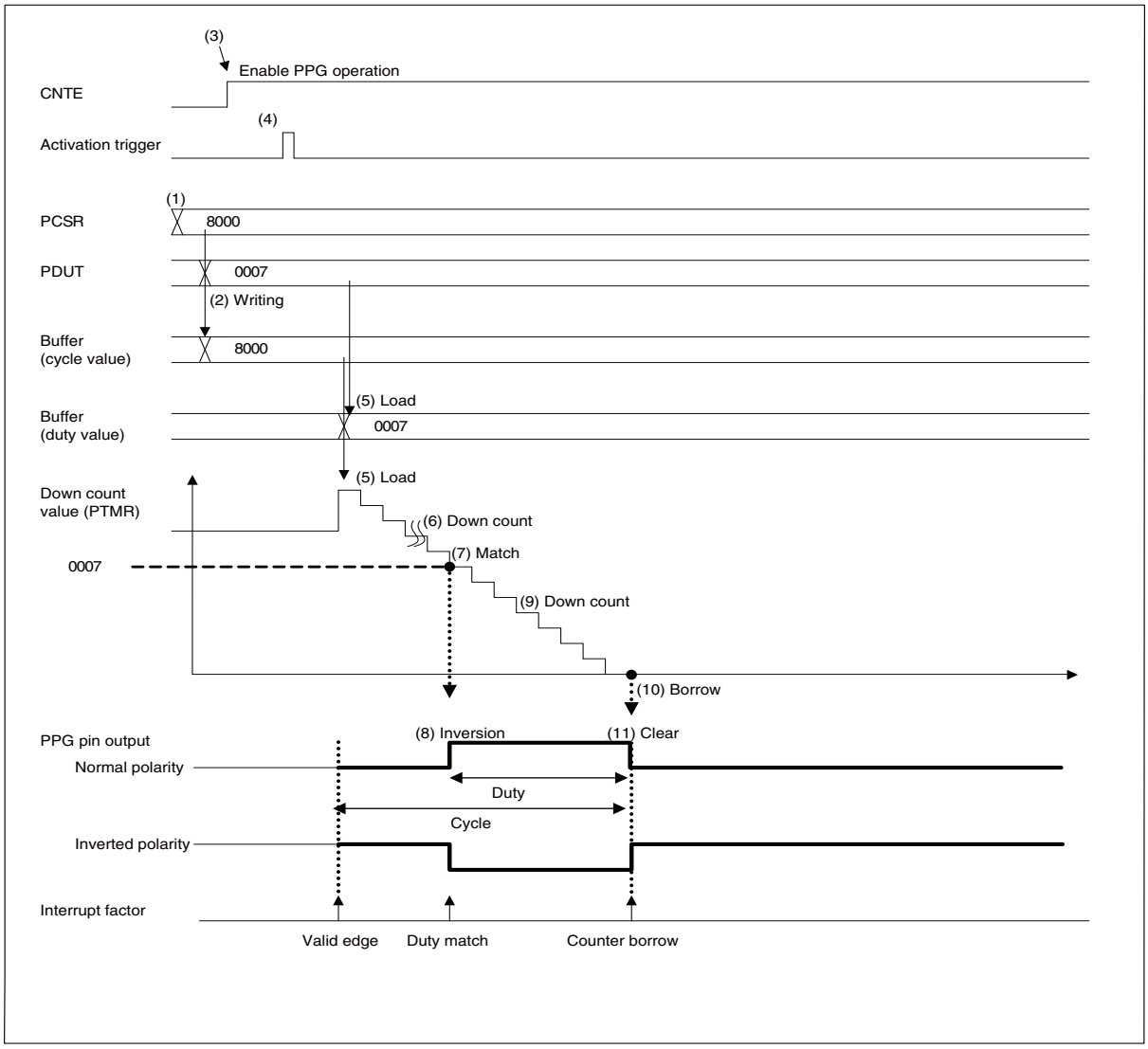
Calculation formulas:

- 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)] x [Count clock]

41.5.3 One-shot Operation (Normal Waveform Selected)

During one-shot operation, one-shot pulses are output at the PPG pin.

Figure 41.5-3 Example of One-shot Operation (Normal Waveform Selected)



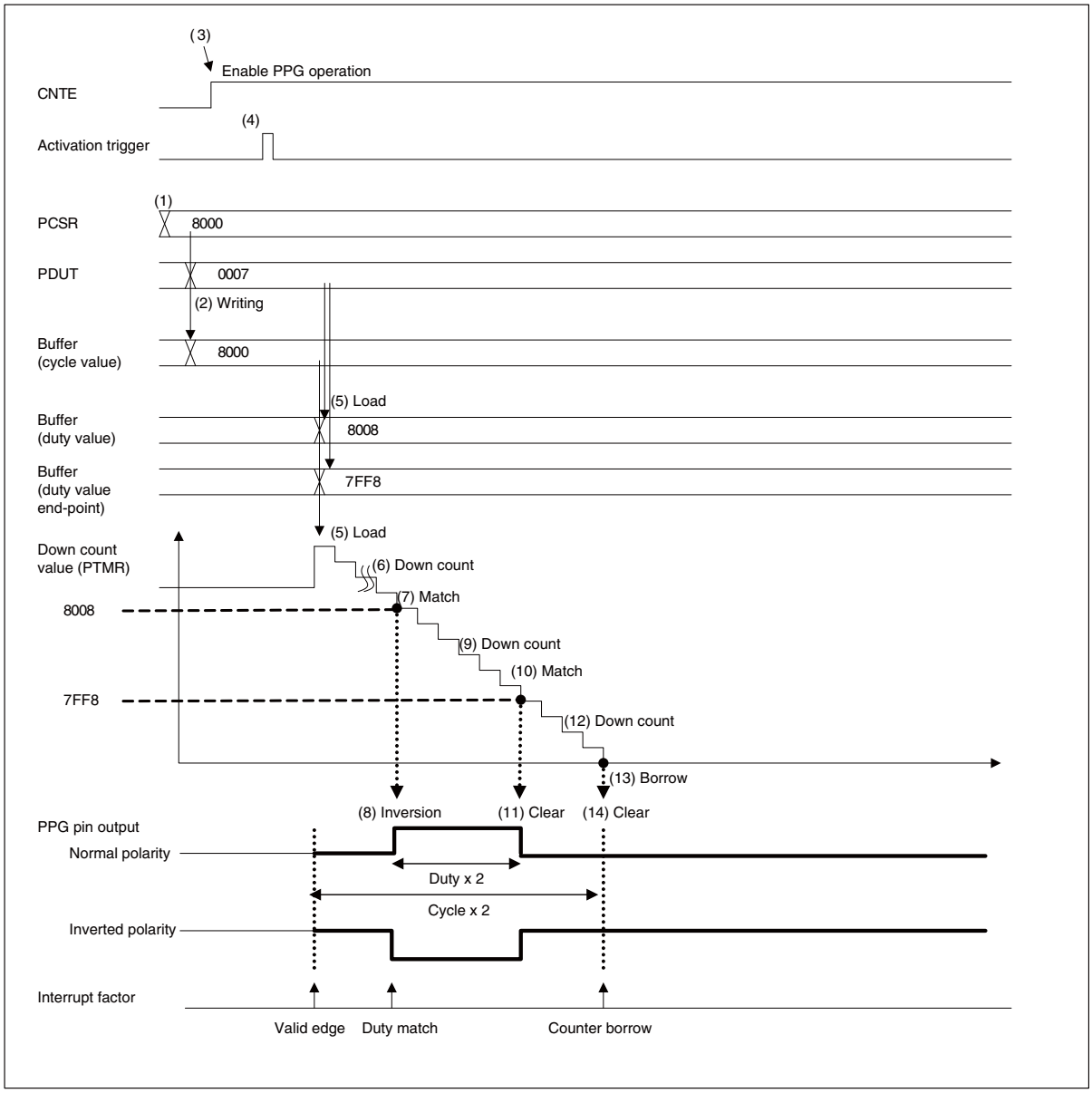
(1) Writing of PCSR (cycle value)
(2) Writing of PDUT (duty value) and transferring cycle value to the buffer (cycle value)
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading of the cycle value to the down count (PTMR) and the duty value to the buffer (duty value)
(6) Counter 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

41.5.4 One-shot Operation (Center-aligned Waveform Selected)

During 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 waveform), 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 41.5-4 Example of One-shot Operation (Center-aligned Waveform Selected)



(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 (PTMR) and the duty value to the buffer (duty value) and 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).

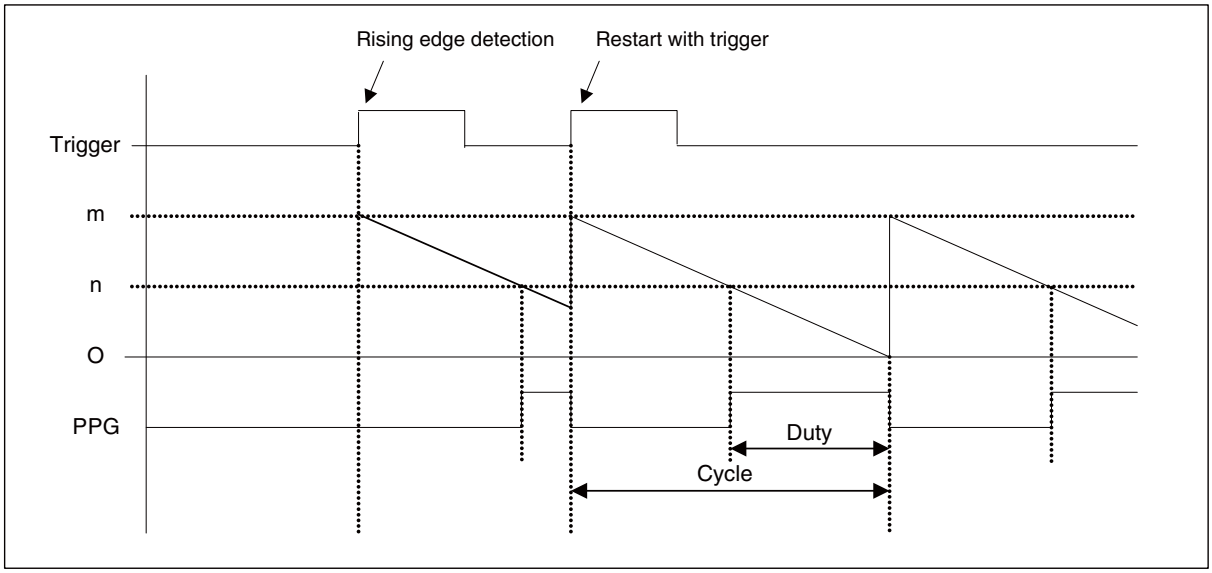
(8) Output level inversion at the PPG pin
(9) Counter decrement
(10) The down counter matches the duty value end point (output level clear timing).
(11) Clearing of PPG pin output level (restoration to normal state)
(12) Counter decrement
(13) Counter borrow occurrence
(14) Clearing of PPG pin output level (restoration to normal state)
(15) End of operation sequence

41.5.5 Restart Operation

Restart operation is as follows:

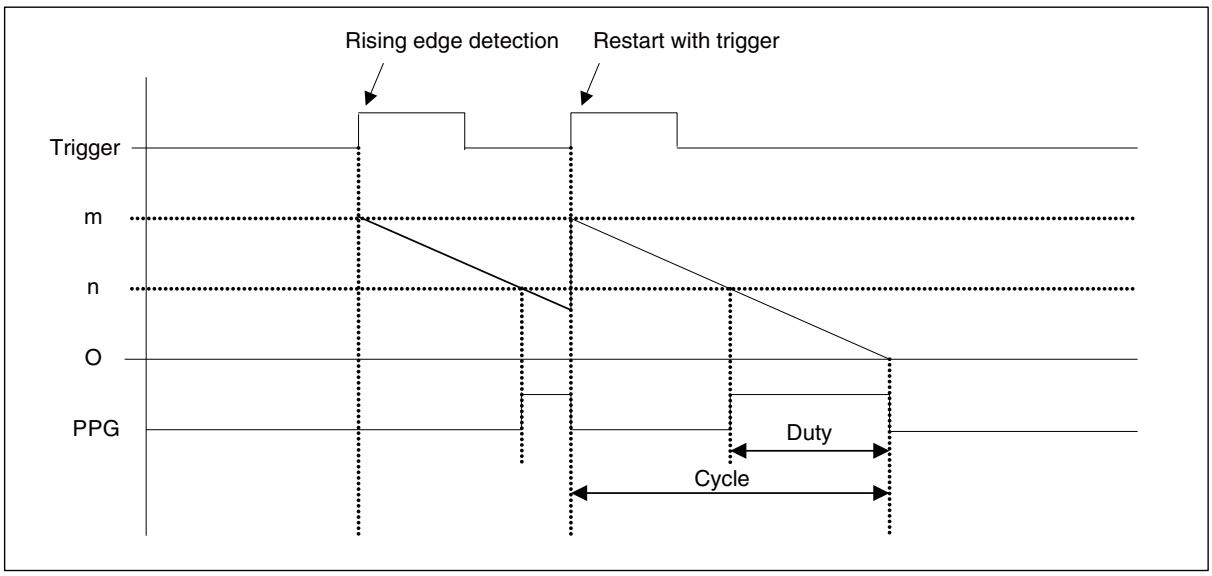
■ Restart during PWM Operation

Figure 41.5-5 Restart during PWM Operation (Normal Waveform Selected)



■ Restart during One-shot Operation

Figure 41.5-6 Restart during One-shot Operation (Normal Waveform Selected)



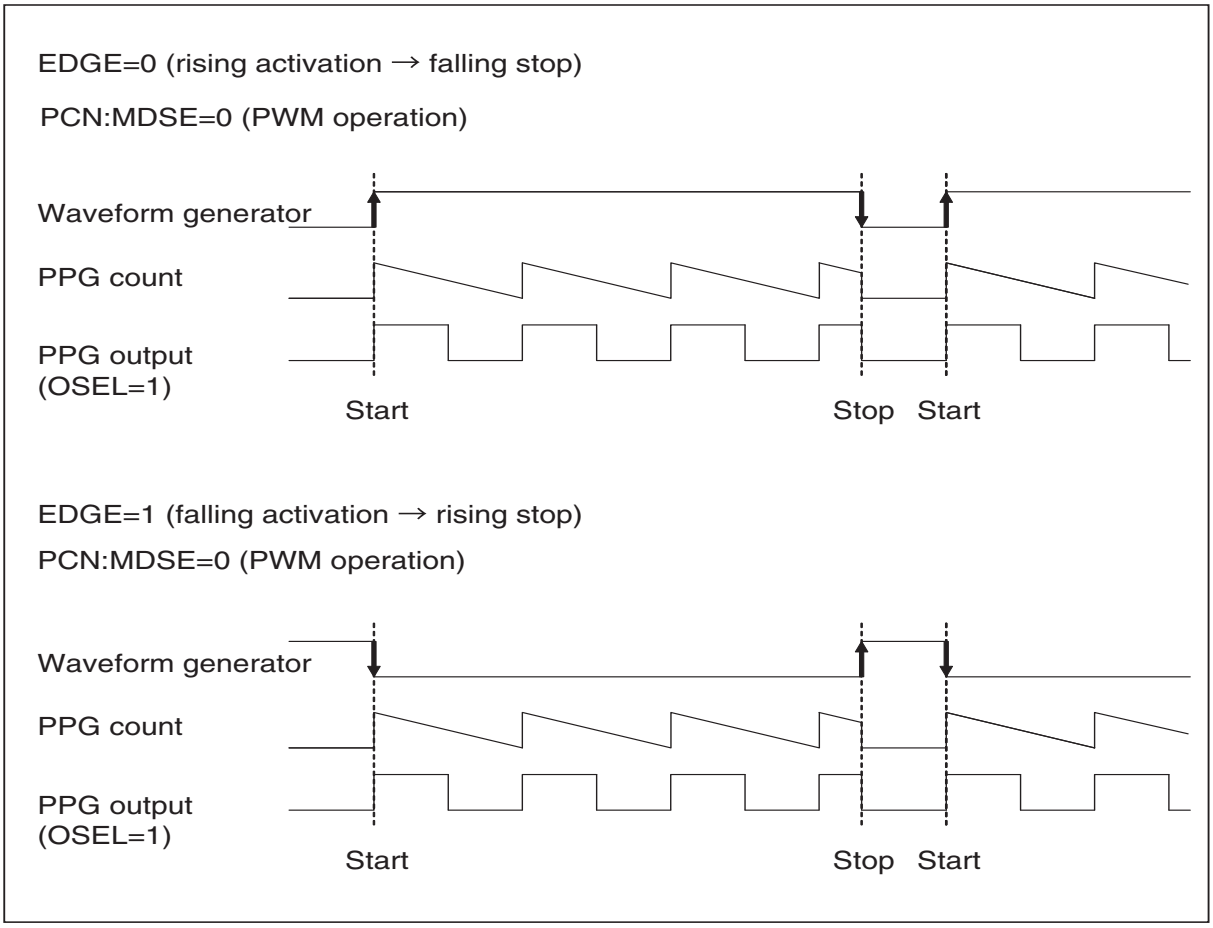
Regardless of whether PWM operation or one-shot operation is being performed, the second or subsequent trigger does not influence operation (does not cause restart) if the restart enable bit (PCN:RTRG) is "0" (restart disabled). However, if, in the case of one-shot operation, the second or subsequent trigger occurs after one-shot operation, it works as a restart trigger.

41.5.6 GATE Operation

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 41.5-7 PPG Count 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.

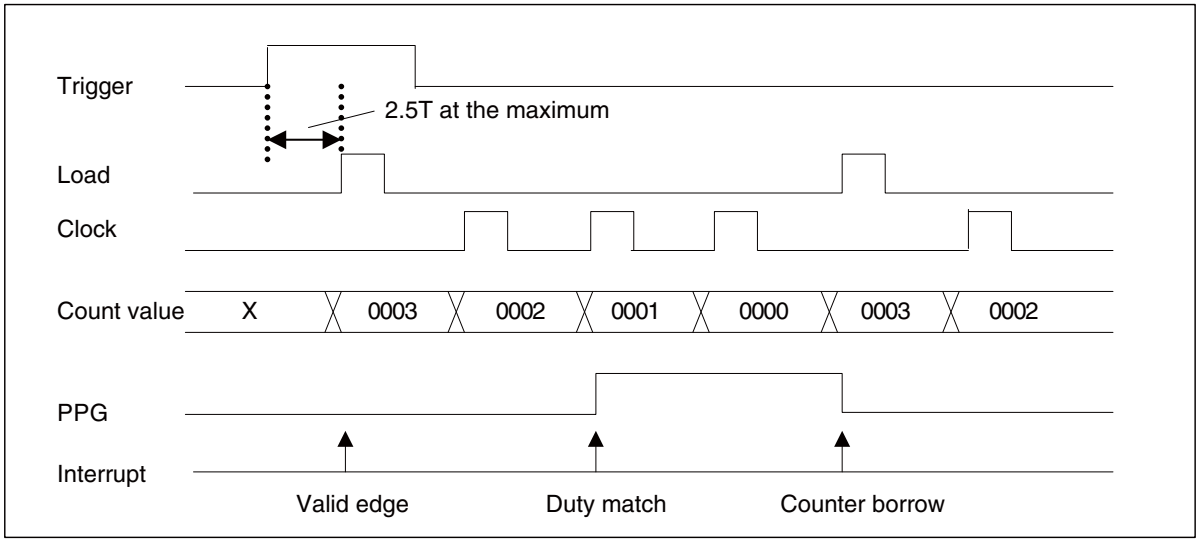
41.6 Notes

Note the following when using the PPG:

■ Application Notes

- If the timing that sets the interrupt request flag (PCNn:IRQF) to "1" and another timing that sets the same flag to "0" occur at the same time, the first event takes precedence, and request to clear the flag is disabled.
- 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).

Figure 41.6-1 Timing Chart



- When writing a cycle value (PCSR) and a duty value (PDUT), be sure to observe the sequence of (1) PCSR and (2) PDUT.
  Note the following when rewriting the cycle value (PCSR) and duty value (PDUT):
  1) A cycle value (PCSR) and a 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 PPG operation, the new value will be effective on the output waveform after the cycle next to the cycle after which the duty value (PDUT) is written.
  3) If only the cycle value (PCSR) needs to be changed, (1) PCSR and (2) PDUT must be reconfigured in that order. Then, the cycle value (PCSR) must be reset to the value that is equal to the duty value (PDUT).
  4) The duty value (PDUT) may be freely rewritten.

- For the PPG duty setting register value (PDUT), set a smaller value than the value set in the PPG cycle setting register (PCSR).

- 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, values are not written at high-order and low-order bit positions.

- To activate the PPG, the timer operation enable bit (PCN:CNTE) must be set to "1" before or when PPG operation is enabled.

- During PPG operation, do not change any of the following: the mode selection bit (PCN:MDSE), the restart enable bit (PCN:RTRG), counter clock selection bits (PCN:CKS1, CKS0), trigger input edge selection bits (PCN:EGS1, EGS0), interrupt factor selection bits (PCN:IRS1, IRS0), 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 PPG operation, disable PPG operation before reconfiguring the register.

- If the timer operation enable bit (PCN:CNTE) is set to "0" to disable the PPG during PPG operation the PPG down count value (PTMR) is maintained, and PPG output becomes "L" and stops.
  Thereafter, to restart the PPG, set the timer operation enable bit (PCN:CNTE) to "1" to enable the PPG and reload the cycle and duty values by entering an activation trigger.

- If the GATE signal changes from "1" to "0" (when EDGE="0") during PPG operation, the PPG down count value (PTMR) is maintained, and PPG output becomes "L" and stops.
  Afterward, to restart operation, the GATE signal is set from "0" to "1", the cycle and duty values are reloaded.

- The GATE function control register (GATEC) is set 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 PPG operation.

- It takes four clock in internal clock until the PPG output is stopped after GATE signal is negated.

- If the timer operation enable bit (PCN:CNTE) is set to "0" and the PPG has been prohibited, it takes 3 internal clocks until PPG output stops.

- If the PPG output waveform selection bit (OWFS) is rewritten during PPG operation, the new setting is reflected from the next cycle.

- When setting the PPG output mask (PCN.PGMS) from "1" to "0" during PPG operation to release the mask, set it in a period from the start of a cycle to a duty match.

- The following applies within the PPG output mask (PCN.PGMS="1").
  A duty match causes an interrupt but does not set the interrupt flag to "1".
  A counter borrow occurrence causes an interrupt and sets the interrupt flag to "1".
  A trigger (software trigger, external trigger, or trigger generated by GATE signal input) causes an interrupt and sets the interrupt flag to "1".

- The restart operation does not start when the GATE function is enabled (STGR="1") and restart is enabled (RTRG="1"), even if another activation trigger is input.

- When PGMS is cleared from "1" to "0" after a matching, the pulse with specified duty is outputted in the next period (PPG cycle).

CHAPTER 42 FREE-RUN TIMER

42.1 Overview

The free-run timer consists of one free-run timer simultaneous activation, six 16-bit free-run timers (1ch each and total of 6ch), and one free-run timer selector.

For the 128 MHz operation setting for and notes on the motor control function, see CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION.

42.2 Features

■ Function of the Free-run Timer Simultaneous Activation

- Of the six 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 operating 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.
- The counter value can be initialized to "0000_H" upon a reset, software clear or if a compare match is generated between this timer and the compare register values.
- 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

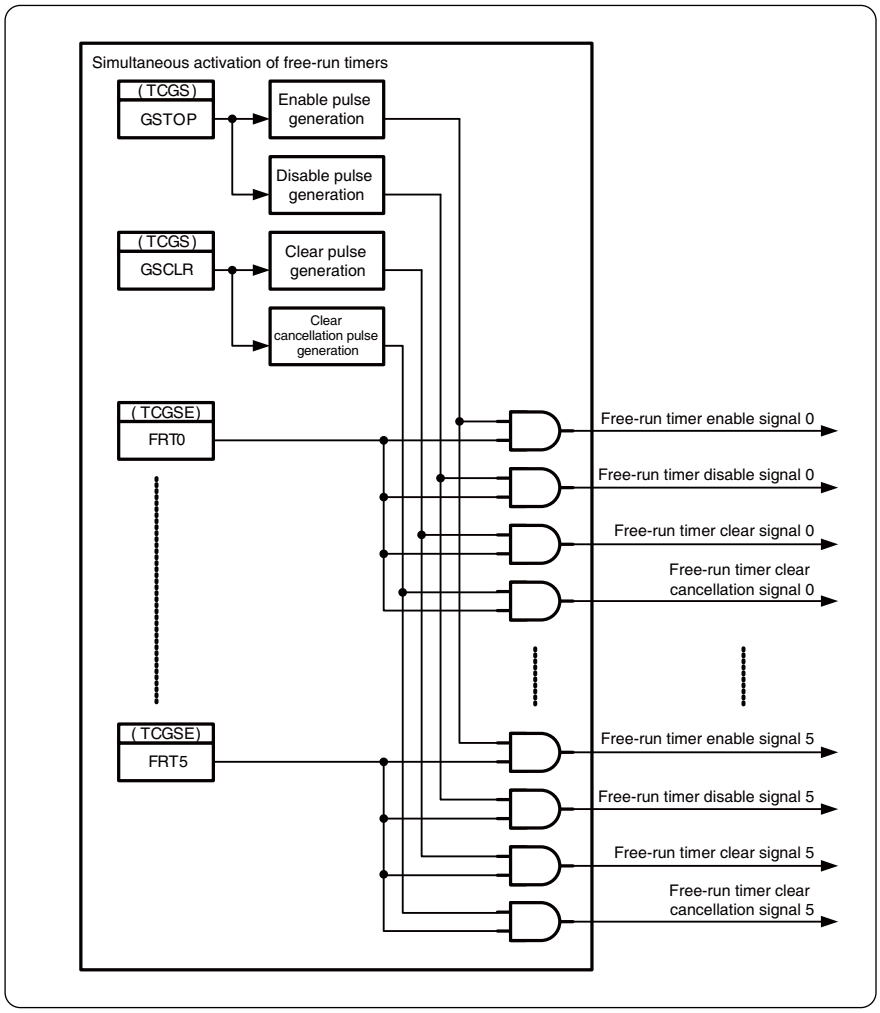
- The free-run timer selection register allows you to select the assignment of the free-run timer from among the output capture, the input capture, and the A/D activation compare.

Code: FS17-1v0-91580MS-1-E

42.3 Configuration

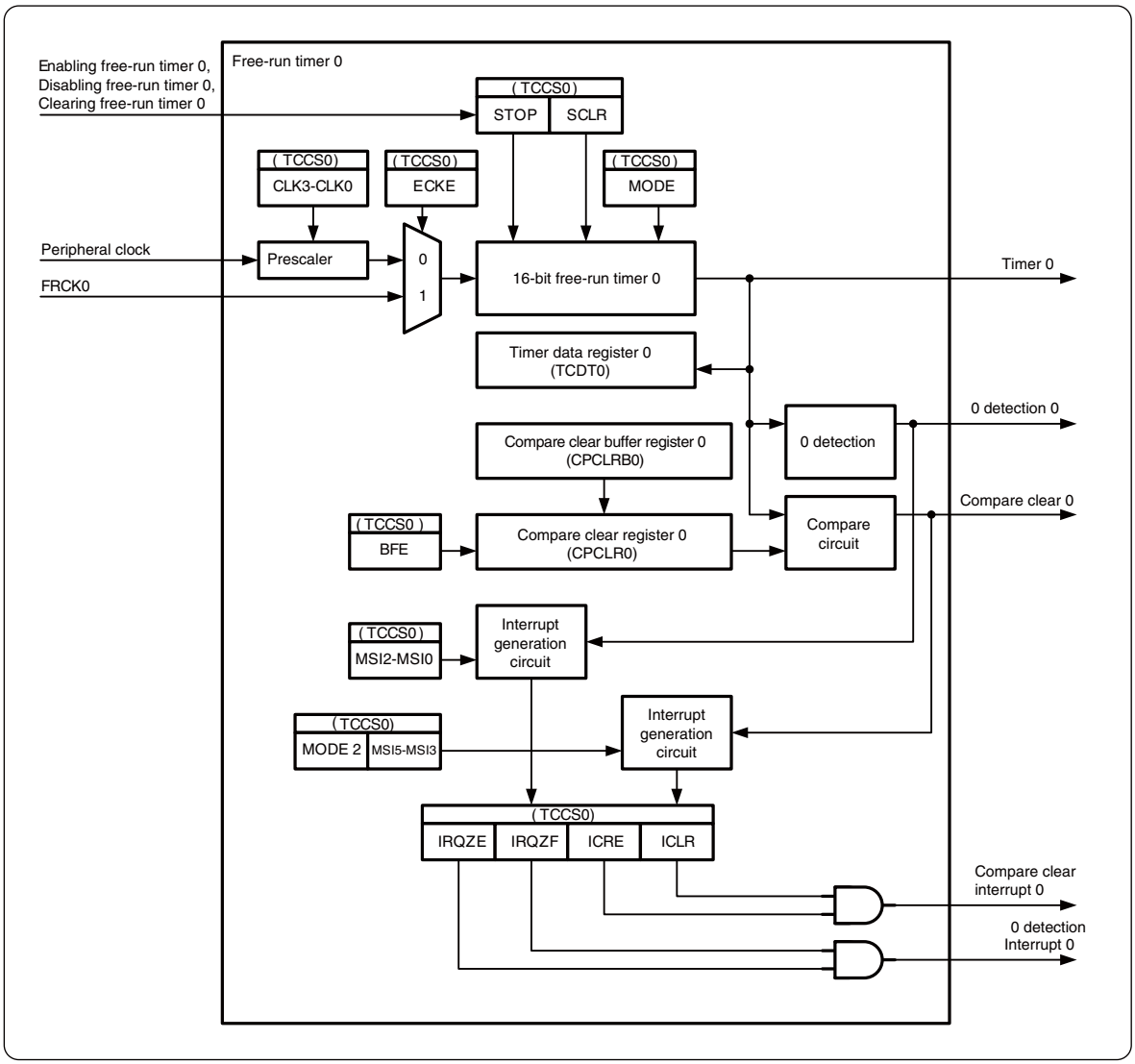
■ Configuration of the Free-run Timer Simultaneous Activation

Figure 42.3-1 Configuration of the Free-run Timer Simultaneous Activation



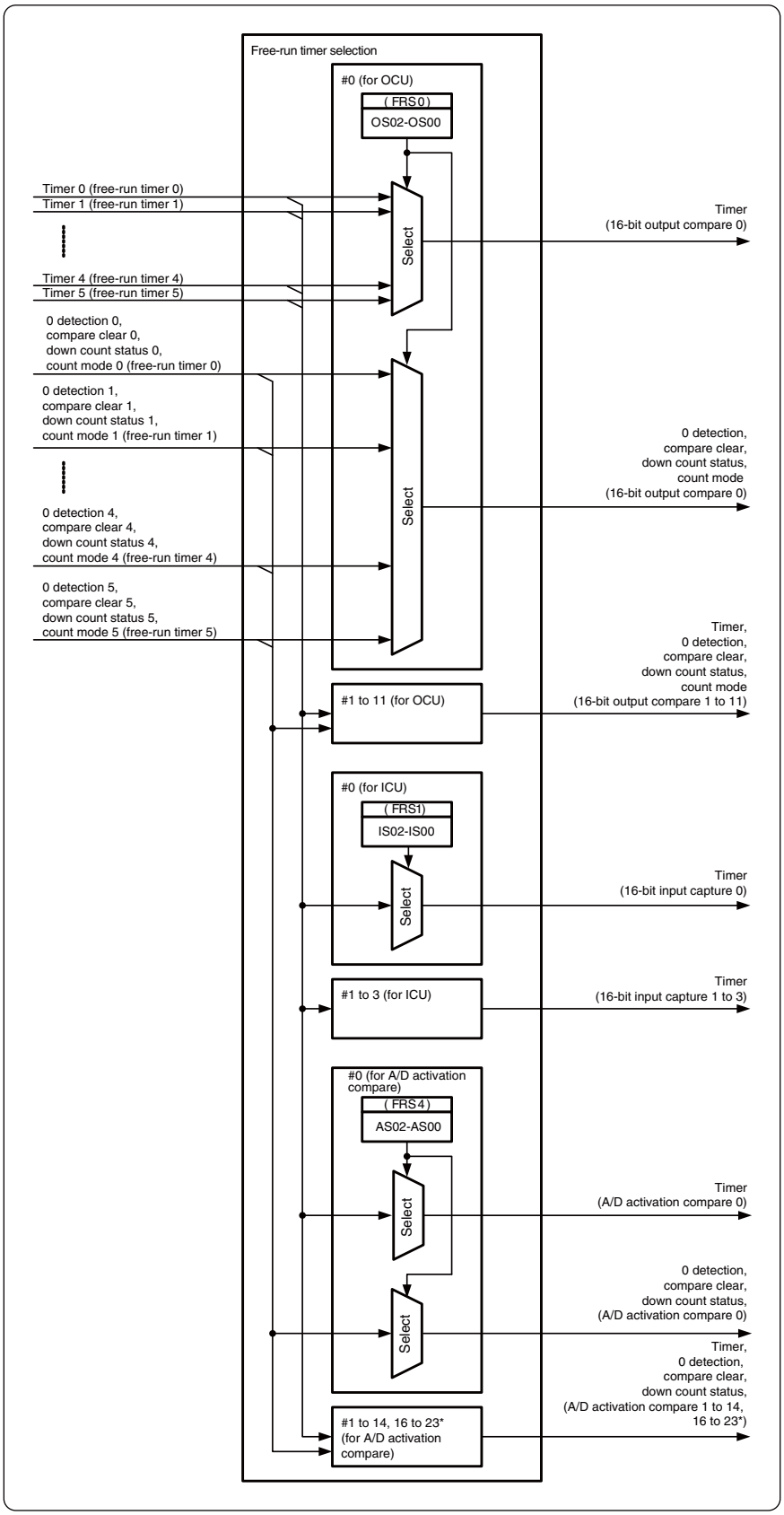
■ Configuration of the 16-bit Free-run Timer

Figure 42.3-2 Configuration of the 16-bit Free-run Timer (for one channel)



■ Configuration of the Free-run Timer Selector

Figure 42.3-3 Configuration of the Free-run Timer Selector



*: Since the MB91F583AS/F584AS/F585AS A/D converter contains 17 channels, they are 1 to 14, 19 and 20.

42.4 Registers

Table 42.4-1 List of Registers for the Free-run Timer Simultaneous Activation

Address	+0	+1	+2	+3
$0x1100_H$	Timer synchronous activation register (TCGS)	Reserved	Reserved	Timer synchronous activation Enable register (TCGSE)

Table 42.4-2 List of Registers for the 16-bit Free-run Timer

Address	+0	+1	+2	+3
$0x1104_H$	Compare clear buffer register 0 (CPCLRB0) Compare clear register 0 (CPCLR0)		Timer data register (TCDT0)	
$0x1108_H$	Timer state control register 0 (TCCS0)			Reserved
$0x110C_H$	Compare clear buffer register 1 (CPCLRB1) Compare clear register 1 (CPCLR1)		Timer data register (TCDT1)	
$0x1110_H$	Timer state control register 1 (TCCS1)			Reserved
$0x1114_H$	Compare clear buffer register 2 (CPCLRB2) Compare clear register 2 (CPCLR2)		Timer data register (TCDT2)	
$0x1118_H$	Timer state control register 2 (TCCS2)			Reserved
$0x111C_H$	Compare clear buffer register 3 (CPCLRB3) Compare clear register 3 (CPCLR3)		Timer data register (TCDT3)	
$0x1120_H$	Timer state control register 3 (TCCS3)			Reserved
$0x1124_H$	Compare clear buffer register 4 (CPCLRB4) Compare clear register 4 (CPCLR4)		Timer data register (TCDT4)	
$0x1128_H$	Timer state control register 4 (TCCS4)			Reserved
$0x112C_H$	Compare clear buffer register 5 (CPCLRB5) Compare clear register 5 (CPCLR5)		Timer data register (TCDT5)	
$0x1130_H$	Timer state control register 5 (TCCS5)			Reserved

Table 42.4-3 List of Registers for the Free-run Timer Selector

Address	+0	+1	+2	+3
$0x1134_H$	Free-run timer selection register 0 (FRS0)			
$0x1138_H$	Free-run timer selection register 1 (FRS1)			
$0x113C_H$	Free-run timer selection register 2 (FRS2)			
$0x1140_H$	Reserved			
$0x1144_H$	Free-run timer selection register 4 (FRS4)			
$0x1148_H$	Free-run timer selection register 5 (FRS5)			
$0x114C_H$	Free-run timer selection register 6 (FRS6)			

42.4.1 Registers for the Free-run Timer Simultaneous Activation

The free-run simultaneous activation consists of the timer synchronous activation register and the timer synchronous enable register.

42.4.1.1 Timer Synchronous Activation Register (TCGS)

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 1100_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved						GSTOP	GSCLR	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	R0,W	Attribute

[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":
  Start 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":
  Stop 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":
  Initialize 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":
  Cancel the instruction of 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".

42.4.1.2 Timer Synchronous Activation Enable Register (TCGSE)

The timer synchronous activation enable register (TCGSE) allows you to set the free-run timer that enables the simultaneously activation/clear.

- **TCGSE: Address 1103_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved		FRT5	FRT4	FRT3	FRT2	FRT1	FRT0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] Reserved

Always write 0 to these bits.

[bit5 to bit0] FRT5 to FRT0: Simultaneous activation/clear setting bits

FRT5-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).

42.4.2 Registers for the 16-bit Free-run Timer

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.

42.4.2.1 Compare Clear Buffer Register (CPCLRB)/Compare Clear Register (CPCLR)

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.

- **CPCLRB0,1,2,3,4,5: Address 1104_H, $110C_H$, . . . $112C_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	
1	1	1	1	1	1	1	1	Initial value
W	W	W	W	W	W	W	W	Attribute

7	6	5	4	3	2	1	0	bit
CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	
1	1	1	1	1	1	1	1	Initial value
W	W	W	W	W	W	W	W	Attribute

[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.

(Notes) 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.

- **CPCLR0,1,2,3,4,5: Address 1104_H, $110C_H$, ... $112C_H$ (Access: Half-word, Word))**

15	14	13	12	11	10	9	8	bit
CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	
1	1	1	1	1	1	1	1	Initial value
R	R	R	R	R	R	R	R	Attribute

7	6	5	4	3	2	1	0	bit
CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	
1	1	1	1	1	1	1	1	Initial value
R	R	R	R	R	R	R	R	Attribute

[bit15 to bit0] CL15 to CL00:Compare clear value bits

	Function
CL15 to CL00	Compare clear value

- The compare clear register is used for comparison with the count value of the 16-bit free-run timer.
- In the up-count mode, if this register matches the count value of the 16-bit free-run timer, the 16-bit free-run timer will be reset to "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.

(Notes) When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

42.4.2.2 Timer Data Register (TCDT0 to TCDT5)

The timer data register (TCDT) is used for reading 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 1106_H (Access: Half-word, Word)**
- **TCDT1: Address $110E_H$ (Access: Half-word, Word)**
- **TCDT2: Address 1116_H (Access: Half-word, Word)**
- **TCDT3: Address $111E_H$ (Access: Half-word, Word)**
- **TCDT4: Address 1126_H (Access: Half-word, Word)**
- **TCDT5: Address $112E_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
T15	T14	T13	T12	T11	T10	T09	T08	
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
T07	T06	T05	T04	T03	T02	T01	T00	
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] T15 to T00:Count value bits

	Function
T15 to T00	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)

(Notes) 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 counter 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 counter 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 counter 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.

42.4.2.3 Timer State Control Register (TCCS0 to TCCS5)

The timer state control register (TCCS) is a register used for controlling the operation of the 16-bit free-run timer.

- **TCCS0: Address 1108_H (Access: Byte, Half-word, Word)**
- **TCCS1: Address 1110_H (Access: Byte, Half-word, Word)**
- **TCCS2: Address 1118_H (Access: Byte, Half-word, Word)**
- **TCCS3: Address 1120_H (Access: Byte, Half-word, Word)**
- **TCCS4: Address 1128_H (Access: Byte, Half-word, Word)**
- **TCCS5: Address 1130_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R(RM1),W	R/W	R,W	R,W	R,W	R(RM1),W	R/W	Attribute

23	22	21	20	19	18	17	16	bit
BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0	
0	1	0	0	0	0	0	0	Initial value
R/W	R,W	R/W	R0,W	R/W	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved				MODE2	MSI5	MSI4	MSI3	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R/W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attribute

[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 immediately as soon as this bit is set. Therefore, you need to 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".

(Notes) 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.

(Notes) 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".

(Notes) 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

CLK3	CLK2	CLK1	CLK0	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				-	-	-	-	-	-

Note) ϕ : Peripheral clock

•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 bit 2

MODE2	MODE*	Function
0	0	Value set for MSI5 to MSI3 will be invalid
0	1	Value set for MSI5 to MSI3 will be invalid
1	0	Setting disabled (operation is not guaranteed)
1	1	Value set for MSI5 to MSI3 will be valid

- 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.

(Notes) 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 reload 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.

42.4.3 Register for the Free-run Timer Selector

The free-run timer selector has the free-run timer selection register.

42.4.3.1 Free-run Timer Selection Register (FRS)

The free-run timer selection register (FRS) is used to set any of 6 channels in the free-run timers for the input capture, output compare, A/D activation compare respectively.

- **FRS0: Address 1134_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attribute

23	22	21	20	19	18	17	16	bit
Reserved	OS52	OS51	OS50	Reserved	OS42	OS41	OS40	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	OS32	OS31	OS30	Reserved	OS22	OS21	OS20	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	OS12	OS11	OS10	Reserved	OS02	OS01	OS00	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

- **FRS2: Address $113C_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attribute

23	22	21	20	19	18	17	16	bit
Reserved	OS112	OS111	OS110	Reserved	OS102	OS101	OS100	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	OS92	OS91	OS90	Reserved	OS82	OS81	OS80	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	OS72	OS71	OS70	Reserved	OS62	OS61	OS60	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

[bit31 to bit24] Reserved

Always write "1" to these bits.

[bit23] Reserved

Always write "0" to this bit.

[bit22 to bit20] OS52 to OS50, OS112 to OS110:
Output compare free-run timer selector selection bits

OS52/ OS112	OS51/ OS111	OS50/ OS110	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.5/ch.11.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19] Reserved

Always write "0" to this bit.

[bit18 to bit16] OS42 to OS40, OS102 to OS100:
Output compare free-run timer selector selection bits

OS42/ OS102	OS41/ OS101	OS40/ OS100	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.4/ch.10.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] OS32 to OS30, OS92 to OS90:
Output compare free-run timer selector selection bits

OS32/ OS92	OS31/ OS91	OS30/ OS90	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.3/ch.9.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11] Reserved

Always write "0" to this bit.

[bit10 to bit8] OS22 to OS20, OS82 to OS80:
Output compare free-run timer selector selection bits

OS22/ OS82	OS21/ OS81	OS20/ OS80	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.2/ch.8.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] OS12 to OS10, OS72 to OS70:

Output compare free-run timer selector selection bits

OS12/ OS72	OS11/ OS71	OS10/ OS70	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.1/ch.7.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3] Reserved

Always write "0" to this bit.

[bit2 to bit0] OS02 to OS00, OS62 to OS60:

Output compare free-run timer selector selection bits

OS02/ OS62	OS01/ OS61	OS00/ OS60	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the output compare ch.0/ch.6.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

- **FRS1: Address 1138_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attribute

23	22	21	20	19	18	17	16	bit
Reserved								
1	1	1	1	1	1	1	1	Initial value
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	IS32	IS31	IS30	Reserved	IS22	IS21	IS20	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	IS12	IS11	IS10	Reserved	IS02	IS01	IS00	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

[bit31 to bit16] Reserved

Always write "1" to these bits.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] IS32 to IS30: Input capture free-run timer selector selection bits

IS32	IS31	IS30	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the input capture ch.3.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11] Reserved

Always write "0" to this bit.

[bit10 to bit8] IS22 to IS20:Input capture free-run timer selector selection bits

IS22	IS21	IS20	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the input capture ch.2.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] IS12 to IS10: Input capture free-run timer selector selection bits

IS12	IS11	IS10	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the input capture ch.1.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3] Reserved

Always write "0" to this bit.

[bit2 to bit0] IS02 to IS00: Input capture free-run timer selector selection bits

IS02	IS01	IS00	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
0	1	0	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the input capture ch.0.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

- **FRS4: Address 1144_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved	AS72	AS71	AS70	Reserved	AS62	AS61	AS60	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

23	22	21	20	19	18	17	16	bit
Reserved	AS52	AS51	AS50	Reserved	AS42	AS41	AS40	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	AS32	AS31	AS30	Reserved	AS22	AS21	AS20	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	AS12	AS11	AS10	Reserved	AS02	AS01	AS00	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

[bit31] Reserved

Always write "0" to this bit.

[bit30 to bit28] AS72 to AS70: A/D activation compare free-run timer selector selection bits

AS72	AS71	AS70	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.7.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit27] Reserved

Always write "0" to this bit.

[bit26 to bit24] AS62 to AS60: A/D activation compare free-run timer selector selection bits

AS62	AS61	AS60	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.6.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit23] Reserved

Always write "0" to this bit.

[bit22 to bit20] AS52 to AS50: A/D activation compare free-run timer selector selection bits

AS52	AS51	AS50	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.5.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19] Reserved

Always write "0" to this bit.

[bit18 to bit16] AS42 to AS40: A/D activation compare free-run timer selector selection bits

AS42	AS41	AS40	Function
0	0	0	Free-run timer 0
0	0	0	Free-run timer 1
0	0	1	Free-run timer 2
0	0	1	Free-run timer 3
0	1	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.4.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] AS32 to AS30: A/D activation compare free-run timer selector selection bits

AS32	AS31	AS30	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.3.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11] Reserved

Always write "0" to this bit.

[bit10 to bit8] AS22 to AS20: A/D activation compare free-run timer selector selection bits

AS22	AS21	AS20	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.2.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] AS12 to AS10: A/D activation compare free-run timer selector selection bits

AS12	AS11	AS10	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.1.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3] Reserved

Always write "0" to this bit.

[bit2 to bit0] AS02 to AS00: A/D activation compare free-run timer selector selection bits

AS02	AS01	AS00	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.0.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

- **FRS5: Address 1148_H (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved					AS142	AS141	AS140	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W0	R/W0	R/W0	R0,W0	R/W	R/W	R/W	Attribute

23	22	21	20	19	18	17	16	bit
Reserved	AS132	AS131	AS130	Reserved	AS122	AS121	AS120	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	AS112	AS111	AS110	Reserved	AS102	AS101	AS100	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	AS92	AS91	AS90	Reserved	AS82	AS81	AS80	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

[bit31 to bit27] Reserved

Always write "0" to these bits.

[bit26 to bit24] AS142 to AS140:

A/D activation compare free-run timer selector selection bits

AS142	AS141	AS140	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.14.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit23] Reserved

Always write "0" to this bit.

[bit22 to bit20] AS132 to AS130: A/D activation compare free-run timer selector selection bits

AS132	AS131	AS130	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.13.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit19] Reserved

Always write "0" to this bit.

[bit18 to bit16] AS122 to AS120: A/D activation compare free-run timer selector selection bits

AS122	AS121	AS120	Function
0	0	0	Free-run timer 0
0	0	0	Free-run timer 1
0	0	1	Free-run timer 2
0	0	1	Free-run timer 3
0	1	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.12.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] AS112 to AS110: A/D activation compare free-run timer selector selection bits

AS112	AS111	AS110	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.11.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11] Reserved

Always write "0" to this bit.

[bit10 to bit8] AS102 to AS100: A/D activation compare free-run timer selector selection bits

AS102	AS101	AS100	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.10.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] AS92 to AS90: A/D activation compare free-run timer selector selection bits

AS92	AS91	AS90	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.9.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit3] Reserved

Always write "0" to this bit.

[bit2 to bit0] AS82 to AS80: A/D activation compare free-run timer selector selection bits

AS82	AS81	AS80	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.8.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

- **FRS6: Address 114C$_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
Reserved	AS232	AS231	AS230	Reserved	AS222	AS221	AS220	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

23	22	21	20	19	18	17	16	bit
Reserved	AS212	AS211	AS210	Reserved	AS202	AS201	AS200	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

15	14	13	12	11	10	9	8	bit
Reserved	AS192	AS191	AS190	Reserved	AS182	AS181	AS180	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	AS172	AS171	AS170	Reserved	AS162	AS161	AS160	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R0,W0	R/W	R/W	R/W	Attribute

[bit31] Reserved

Always write "0" to this bit.

[bit30 to bit28] AS232 to AS230:

A/D activation compare free-run timer selector selection bits

AS232	AS231	AS230	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.23.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

[bit27] Reserved

Always write "0" to this bit.

[bit26 to bit24] AS222 to AS220: A/D activation compare free-run timer selector selection bits

AS222	AS221	AS220	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.22.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

[bit23] Reserved

Always write "0" to this bit.

[bit22 to bit20] AS212 to AS210: A/D activation compare free-run timer selector selection bits

AS212	AS211	AS210	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.21.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

[bit19] Reserved

Always write "0" to this bit.

[bit18 to bit16] AS202 to AS200: A/D activation compare free-run timer selector selection bits

AS202	AS201	AS200	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.20.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] AS192 to AS190: A/D activation compare free-run timer selector selection bits

AS192	AS191	AS190	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.19.

(Note) Before configuring these bits, make sure to verify that the free-run timer is inactive.

[bit11] Reserved

Always write "0" to this bit.

[bit10 to bit18] AS182 to AS180: A/D activation compare free-run timer selector selection bits

AS182	AS181	AS180	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.18.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] AS172 to AS170: A/D activation compare free-run timer selector selection bits

AS172	AS171	AS170	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.17.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

[bit3] Reserved

Always write "0" to this bit.

[bit2 to bit0] AS162 to AS160: A/D activation compare free-run timer selector selection bits

AS162	AS161	AS160	Function
0	0	0	Free-run timer 0
0	0	1	Free-run timer 1
0	1	0	Free-run timer 2
0	1	1	Free-run timer 3
1	0	0	Free-run timer 4
1	0	1	Free-run timer 5
Others			Setting disabled (operation is not guaranteed)

•These bits are used to configure the free-run timer assigned to the A/D activation compare ch.16.

(Note)
- Before configuring these bits, make sure to verify that the free-run timer is inactive.
- In models MB91F583AS/F584AS/F585AS, these bits are reserved bits. Always write "0" to these bits.

42.5 Operation

- Free-run timer simultaneous activation

  Of the six 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.

42.5.1 Interrupt for the 16-bit Free-run Timer

Table 42.5-1 shows the interrupt control bits and interrupt factor of the 16-bit free-run timer.

Table 42.5-1 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.

42.5.2 Operation of the 16-bit Free-run Timer

The 16-bit free-run timer starts counting up from the value configured at the timer data register (TCDT) after reset. The count value will be used as base time of the 16-bit output compare and 16-bit input capture.

42.5.2.1 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.

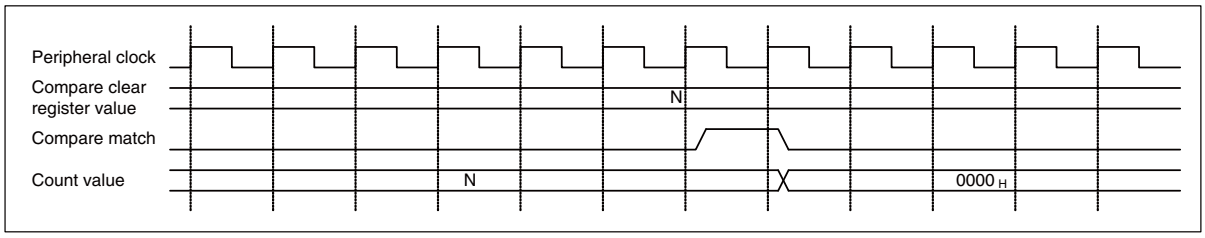
<Note:>

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 counter mode (MODE:bit21=1 of timer state control register (TCCS)), an unintended counting may be performed.

See Section "42.4.2.2 Timer Data Register (TCDT0 to TCDT5)" for the setting procedure of TCDT register during the up/down counter mode (MODE:bit21=1 of timer state control register (TCCS)).

Figure 42.5-1 Clear Timing of the 16-bit Free-run Timer



42.5.2.2 Timer Mode

For the 16-bit free-run timer, you will be able to select either one of the following modes:

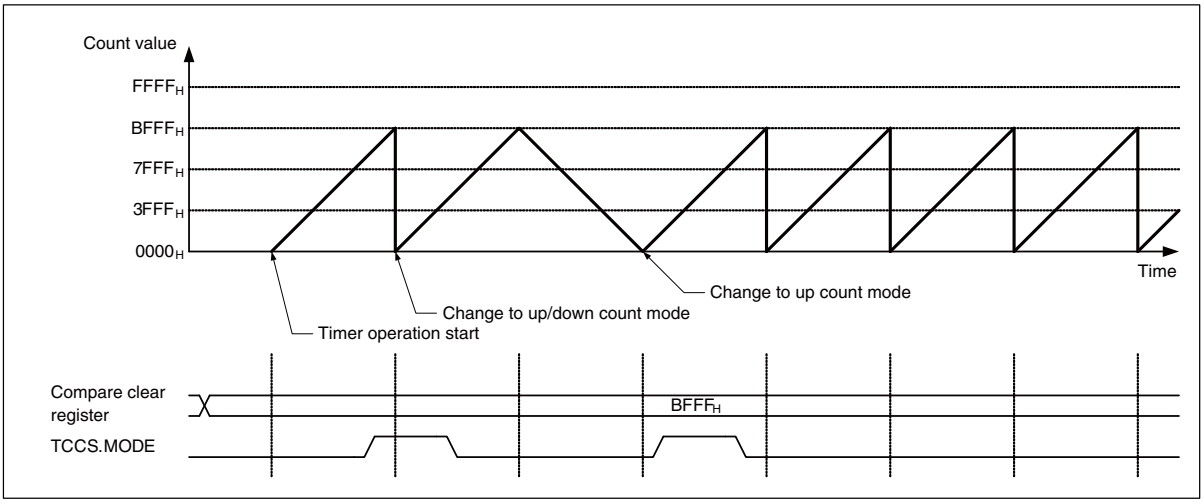
- Up count mode (MODE: bit21 of the TCCS register is 0)
- Up/down count mode (MODE: bit21 of the TCCS register is 1)

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 42.5-2 Changing the Timer Mode While the Timer Is Active



42.5.2.3 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 42.5-3 Operation in the Up Count Mode when the Compare Clear Buffer is Disabled (BFE: bit23 of the TCCS register is 0)

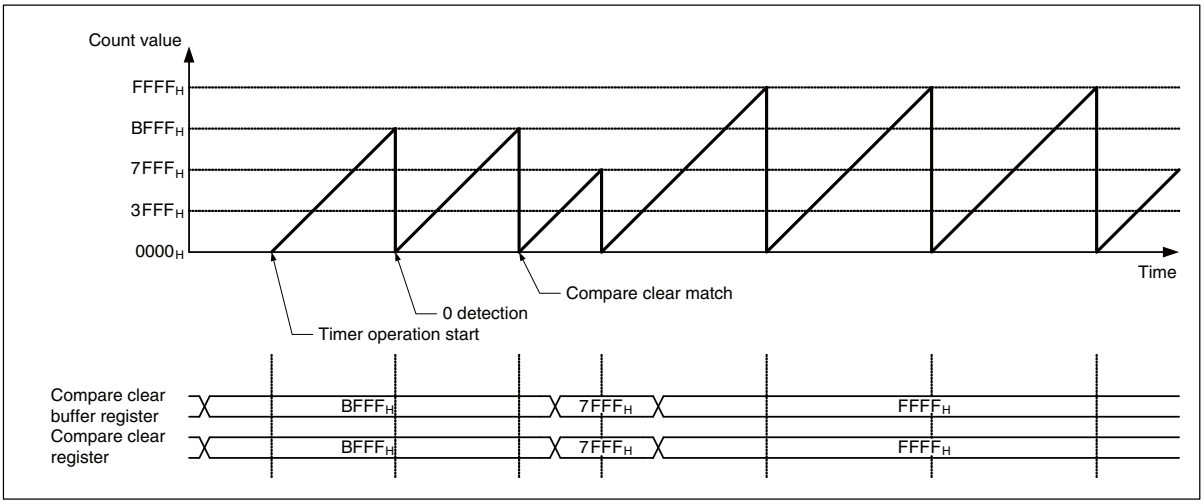


Figure 42.5-4 Operation in the Up Count Mode when the Compare Clear Buffer is Enabled (BFE: bit23 of the TCCS register is 1)

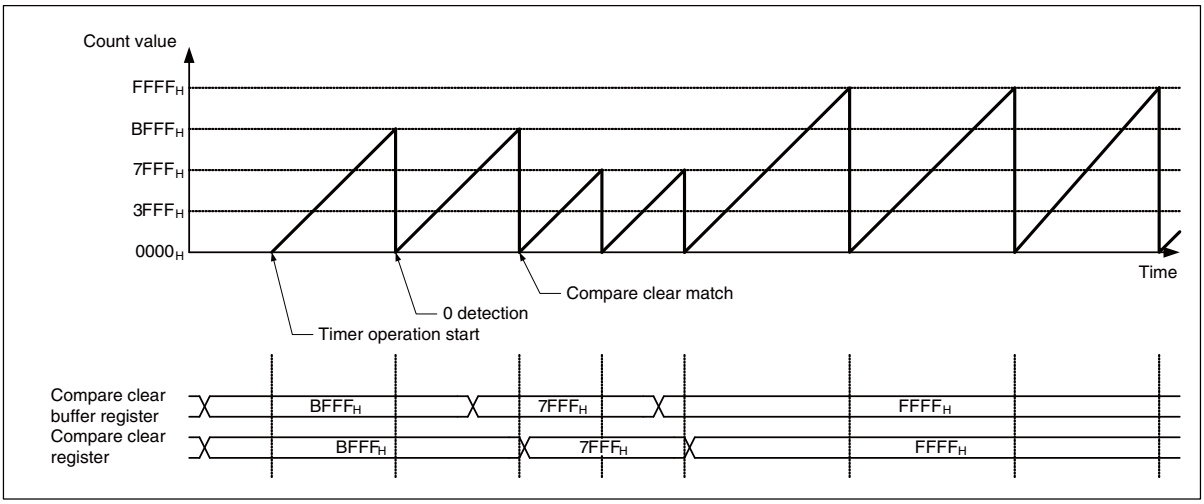
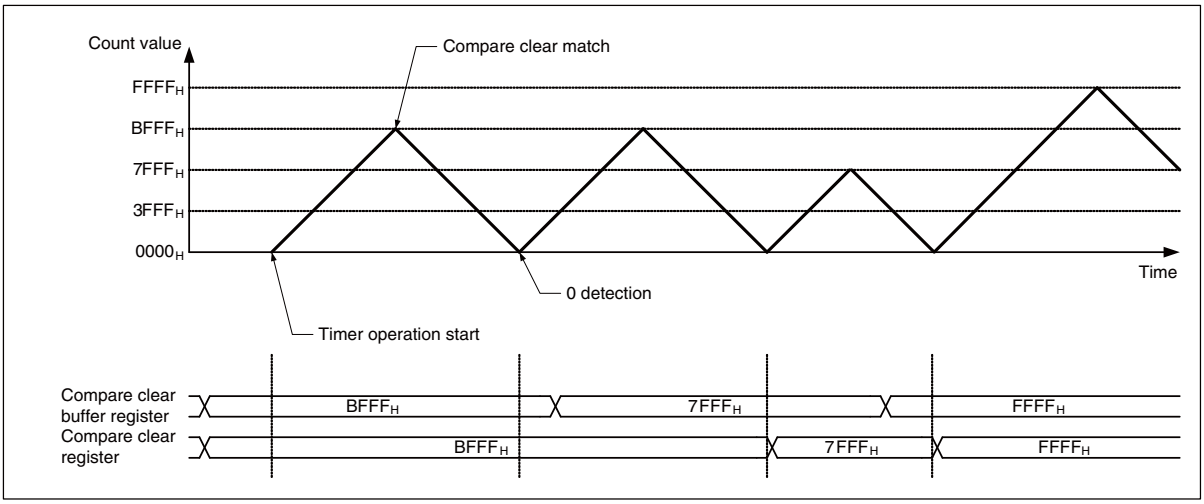


Figure 42.5-5 Operation in the Up/Down Count Mode when the Compare Clear Buffer is Enabled (BFE: bit23 of the TCCS register is 1)



42.5.2.4 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".

<Caution:>

Software clear (SCLR: bit20 of the TCCS register is 1) does not generate the 0 detection interrupt.

Figure 42.5-6 Interrupt Generated in the Up Count Mode (MODE: bit21 of the TCCS register is 0)

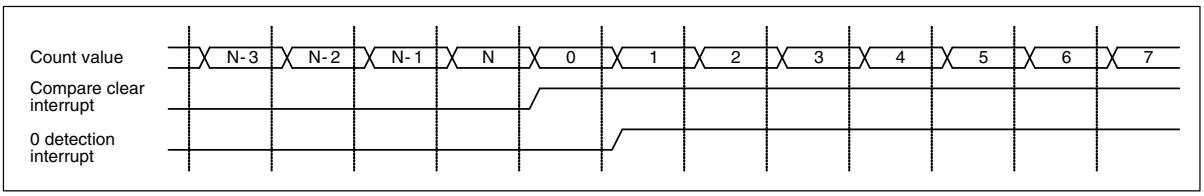
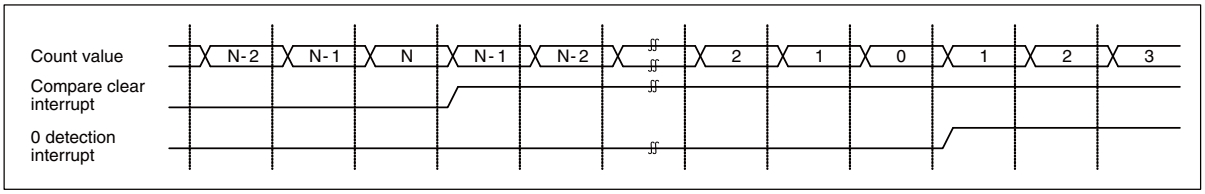


Figure 42.5-7 Interrupt Generated in the Up/Down Count Mode (MODE: bit21 of the TCCS register is 1)



42.5.2.5 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 while 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 42.5-8 Compare Clear Interrupt Masked in the Up Count Mode

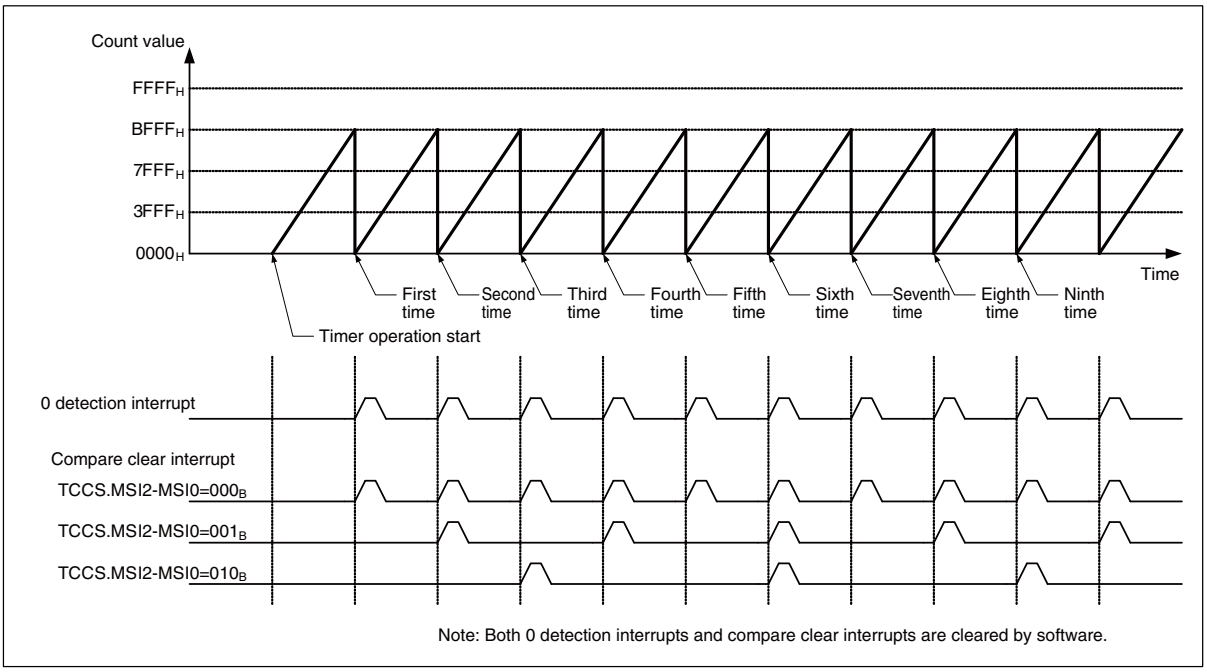


Figure 42.5-9 0 Detection Interrupt Masked in the Up/Down Count Mode

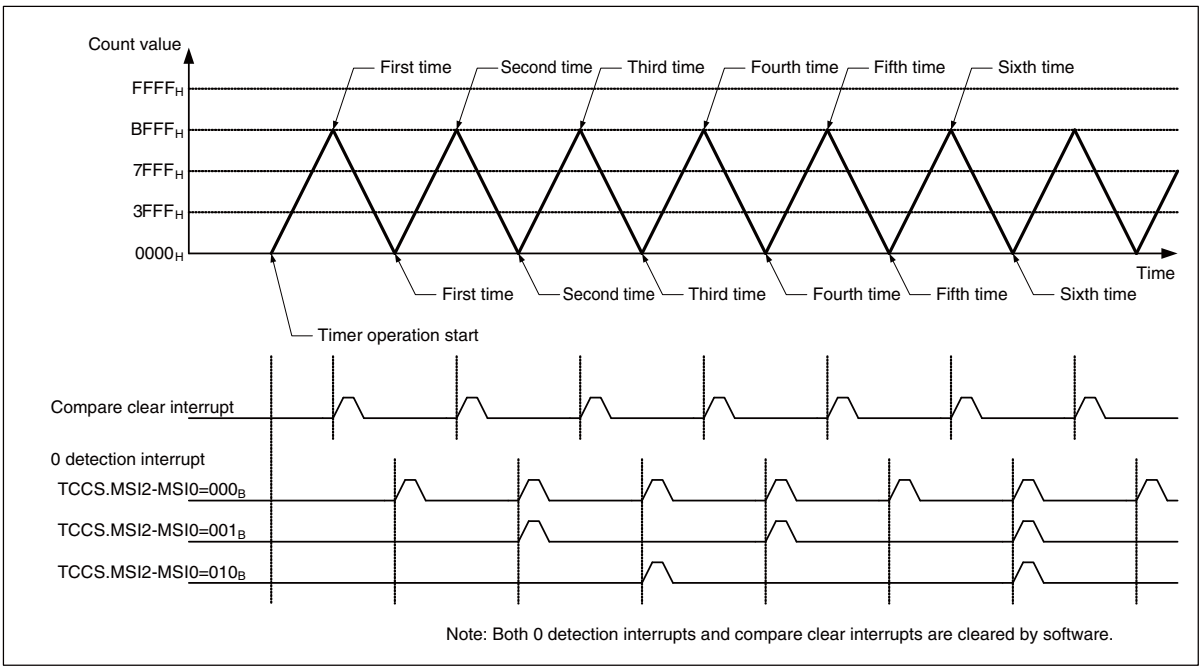
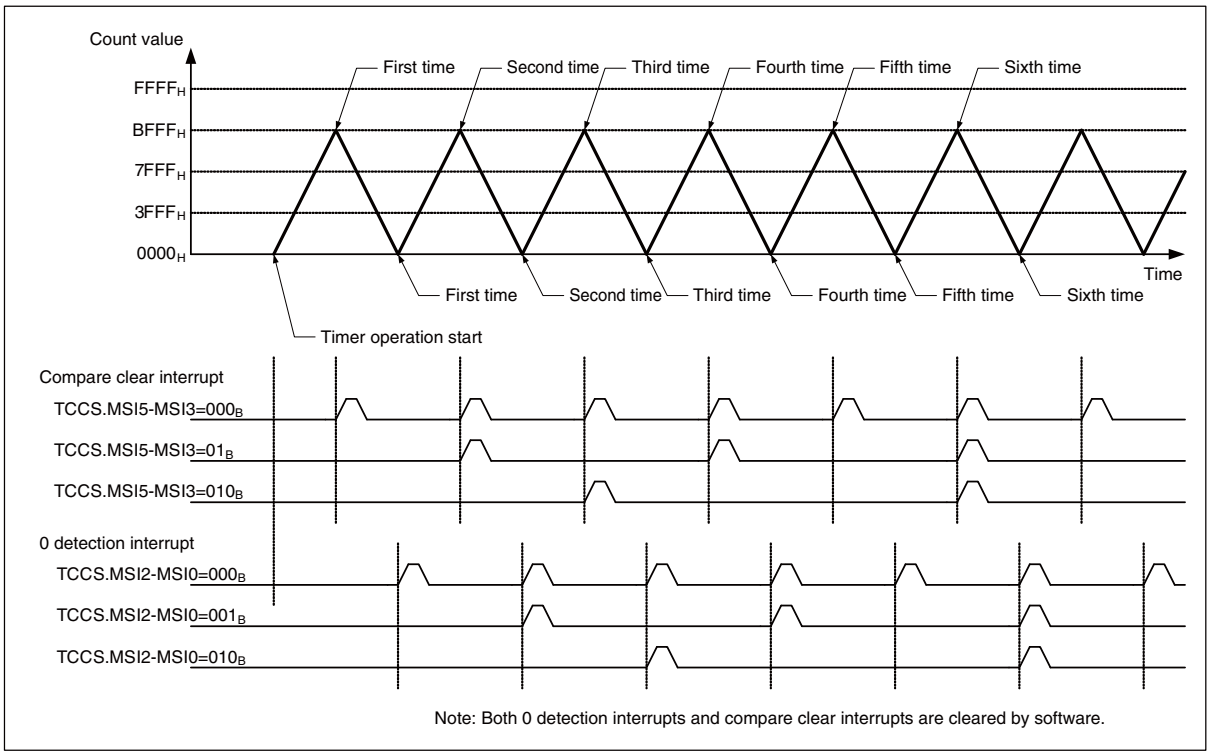


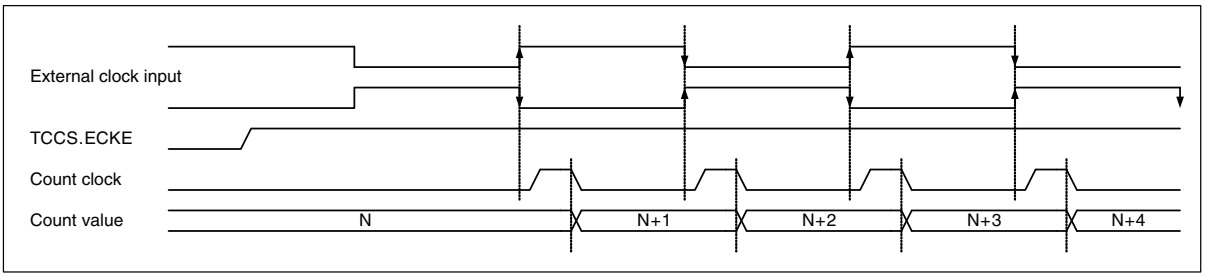
Figure 42.5-10 0 Detection Interrupt and Compare Clear Interrupt Masked in the Up/Down Count Mode



42.5.2.6 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 42.5-11 Count Timing of the 16-bit Free-run Timer



<Note>

Counting of the external clock input uses both edges of the external clock.

42.5.3 Operation of the Free-run Timer Selector

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 6 free-run timers, 6 (12ch) output compares, and 2 (4ch) input captures. In addition, the A/D activation compare consists of 23 channels*, so you will be able to select any of the following registers shown in the table below.

*: The MB91F583AS/F584AS/F585AS A/D activation compare has a 17-channel configuration.

Table 42.5-2 Table for Registers (1 / 2)

Resources	Register	Remarks
OCU0	FRS0:OS0[2:0]	Output compare
OCU1	FRS0:OS1[2:0]	
OCU2	FRS0:OS2[2:0]	
OCU3	FRS0:OS3[2:0]	
OCU4	FRS0:OS4[2:0]	
OCU5	FRS0:OS5[2:0]	
ICU0	FRS1:IS0[2:0]	Input capture
ICU1	FRS1:IS1[2:0]	
ICU2	FRS1:IS2[2:0]	
ICU3	FRS1:IS3[2:0]	
OCU6	FRS2:OS6[2:0]	Output compare
OCU7	FRS2:OS7[2:0]	
OCU8	FRS2:OS8[2:0]	
OCU9	FRS2:OS9[2:0]	
OCU10	FRS2:OS10[2:0]	
OCU11	FRS2:OS11[2:0]	

Table 42.5-2 Table for Registers (2 / 2)

Resources	Register	Remarks
ADT0	FRS4:AS0[2:0]	A/D activation compare
ADT1	FRS4:AS1[2:0]	
ADT2	FRS4:AS2[2:0]	
ADT3	FRS4:AS3[2:0]	
ADT4	FRS4:AS4[2:0]	
ADT5	FRS4:AS5[2:0]	
ADT6	FRS4:AS6[2:0]	
ADT7	FRS4:AS7[2:0]	
ADT8	FRS5:AS8[2:0]	
ADT9	FRS5:AS9[2:0]	
ADT10	FRS5:AS10[2:0]	
ADT11	FRS5:AS11[2:0]	
ADT12	FRS5:AS12[2:0]	
ADT13	FRS5:AS13[2:0]	
ADT14	FRS5:AS14[2:0]	
ADT16	FRS6:AS16[2:0]*	
ADT17	FRS6:AS17[2:0]*	
ADT18	FRS6:AS18[2:0]*	
ADT19	FRS6:AS19[2:0]	
ADT20	FRS6:AS20[2:0]	
ADT21	FRS6:AS21[2:0]*	
ADT22	FRS6:AS22[2:0]*	
ADT23	FRS6:AS23[2:0]*	

*: For the MB91F583AS/F584AS/F585AS, since A/D activation compare has a 17-channel configuration, the registers are reserved bits.

Table 42.5-3 Table for Setting Values

Setting value	Free-run timer
000_B	FRT0 (initial state)
001_B	FRT1
010_B	FRT2
011_B	FRT3
100_B	FRT4
101_B	FRT5
0110_B	Setting disabled (operation is not guaranteed)
0111_B	

<Note>

Before configuring the free-run timer selection register, make sure to verify that the free-run timer is inactive.

42.5.4 Notes on Operating Specifications

42.5.4.1 Notes at 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.

42.5.4.2 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.

42.5.4.3 Notes on Using the Free-run Timer Selector

Be sure to configure selection while the free-run timer is inactive.

CHAPTER 43 INPUT CAPTURE

43.1 Overview

This product includes four 16-bit input capture channels.

For the 128 MHz operation setting for and notes on the motor control function, see CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION.

43.2 Features

■ Functions of 16-bit Input Capture

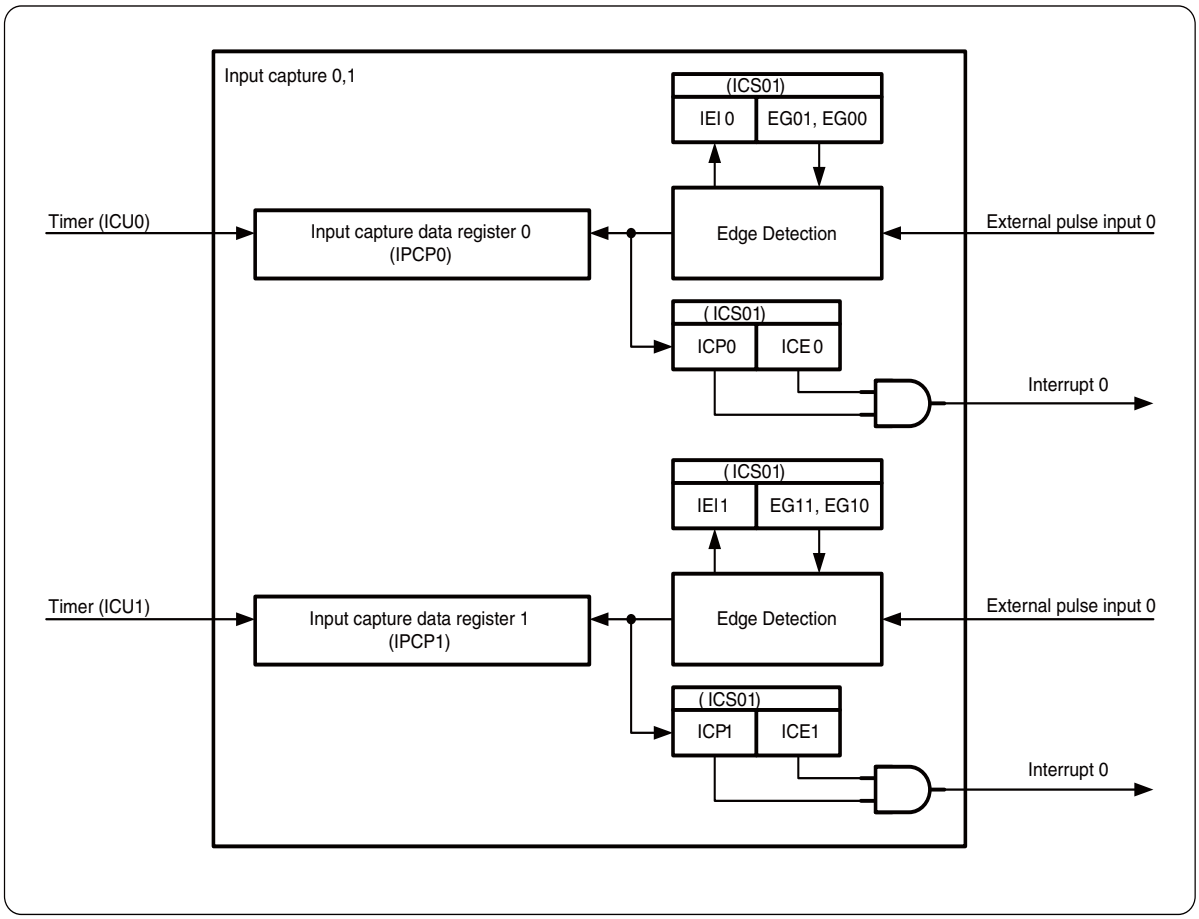
- Input capture consists of 4 independent external input pins, capture registers corresponding to this pin, and capture control registers. When an edge of the input signal from the external pin is detected, the value of the 16-bit free-run timer can be stored in the capture register and an interrupt 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 5 can be selected as the input. This can be set at the free-run timer selection register (FRS1). See Section "42.4.3.1 Free-run Timer Selection Register (FRS)" for details.

Code: FS23-1v0-91580MS-1-E

43.3 Configuration

■ Configuration of 16-bit Input Capture

Figure 43.3-1 Configuration of 16-bit Input Capture (For ch.0, ch.1)



43.4 Registers

■ List of Base Addresses (Base-addr) and External Pins

Channel	Base_addr	External pin name
0	0x1184	IN0
1	0x1184	IN1
2	0x118C	IN2
3	0x118C	IN3

■ List of 16-bit Input Capture Registers

Table 43.4-1 List of 16-bit Input Capture Registers

Address	+0	+1	+2	+3
0000_1184	Input capture data register 0 (IPCP0)		Input capture data register 1 (IPCP1)	
0000_1188	Input capture state control register 01 (ICS01)		Reserved	Sync switching register (LSYNS)
0000_118C	Input capture data register 2 (IPCP2)		Input capture data register 3 (IPCP3)	
0000_1190	Input capture state control register 23 (ICS23)		Reserved	Reserved

43.4.1 16-bit Input Capture Registers

The 16-bit input capture consists of input capture data registers and input capture state control registers.

43.4.1.1 Input Capture Data Register (IPCP0 to IPCP3)

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.

x: Channel number 0,2

y: Channel number 1,3

- **IPCPx: Address Base_addr+0x00$_H$ (Access: Half-word, Word)**
- **IPCPy: Address Base_addr+0x02$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	
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
CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	
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] 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.

43.4.1.2 Input Capture State Control Register (ICS)

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.

x: Channel number 0,2

y: Channel number 1,3

- **ICSxy: Address Base_addr+0x04$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved						IEIy	IEIx	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
ICPy	ICPx	ICEy	ICEx	EGy1	EGy0	EGx1	EGx0	
0	0	0	0	0	0	0	0	Initial value
R (RM1), W	R (RM1), W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit10] Reserved

Always write 0 to these bits.

[bit9, bit8] Effective edge indication bit

IEIn	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 EG1n, EG0n of the input capture state control register (ICS) are set to 00$_B$, the value read from this register is meaningless.
IEIn: The number n corresponds to the input capture channel number.

[bit7, bit6] 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 (ICEn) 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.
ICPn: The number n corresponds to the input capture channel number.

[bit5, bit4] Interrupt request enable bit

ICEn	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 (ICPn) is set while this bit is set to "1".
  ICEn: The number n corresponds to the input capture channel number.

[bit3 to bit0] Edge selection bits

EGn1	EGn0	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.
  EGn1, EGn0: The number n corresponds to the input capture channel number.

43.4.1.3 LIN SYNC FIELD Switching Register (LSYNS)

The LIN SYNCH FIELD switching register (LSYNS) is used for LIN linkage control.

- **LSYNS: Address 118B$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				LSYN3	LSYN2	LSYN1	LSYN0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R/W0	R/W	R/W	R/W	R/W	Attribute

[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 (IN3 to IN0)
1	Input Lin Sync Field detection signals from the multi-function serial interface ch.3 to ch.0.

•These bits are used to enable Lin Sync Field from the multi-function serial interface ch.3 to ch.0.

(Note) Since MB91F583AS/F584AS/F585AS multi-function serial interface contains 2 channels, LSYN3 and LSYN2 are reserved bits. Always write "0" to these bits.

43.5 Operation

43.5.1 Interrupts for 16-bit Input Capture

Table 43.5-1 shows the interrupt control bits and interrupt factor of the 16-bit input capture.

Table 43.5-1 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.

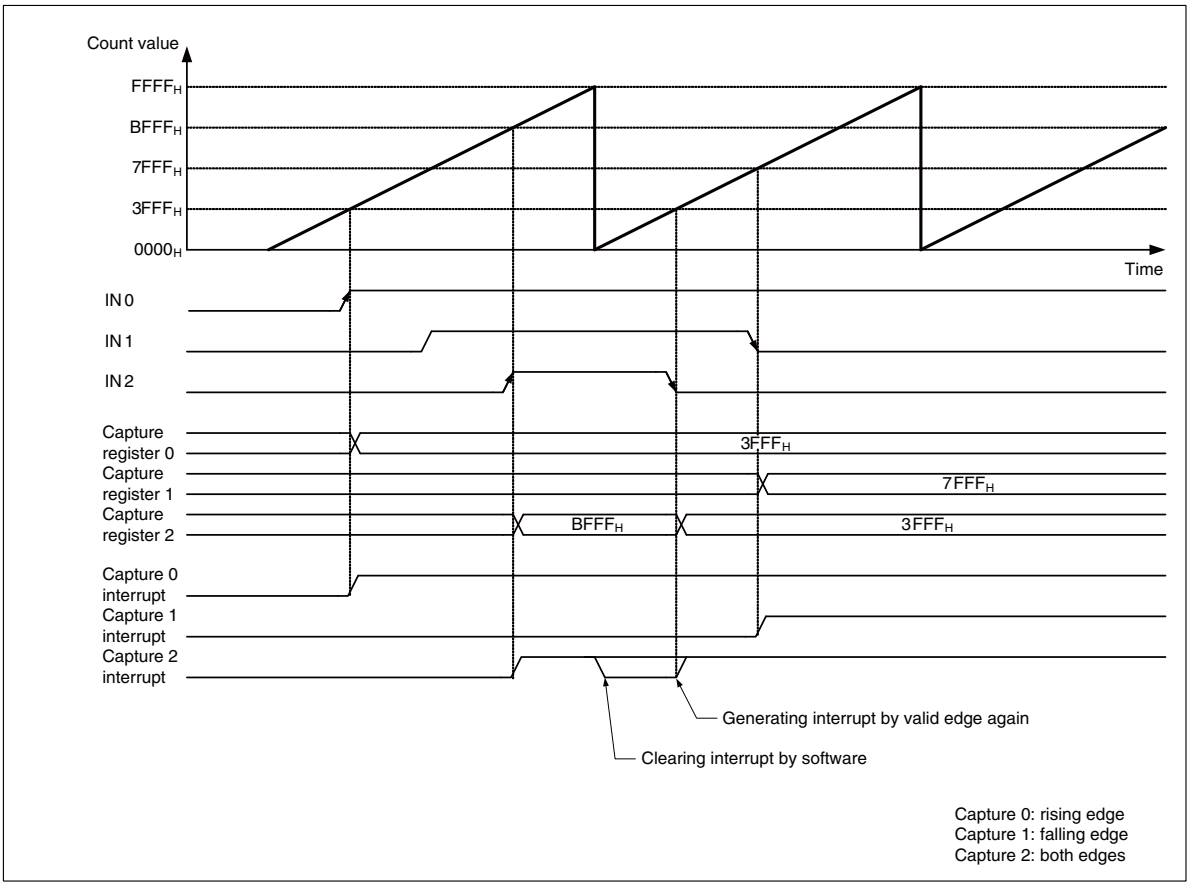
With 16-bit input capture, when an effective edge is detected at a pin, the input capture state control register (ICS) ICP1/ICP0: bit7/bit6 are set to "1". If interrupt requests are enabled (ICE1/ICE0:bit5 and bit4 of ICS01 is 1) with this state, an interrupt request is output to the interrupt controller.

43.5.2 Operation of 16-bit Input Capture

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.

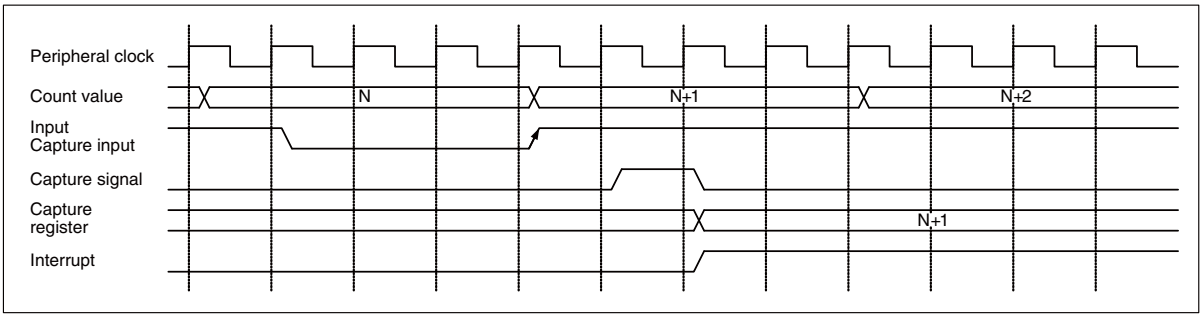
43.5.2.1 Operation of 16-bit Input Capture

Figure 43.5-1 Example of Input Capture Timing



43.5.2.2 16-bit Input Capture Input Timing

Figure 43.5-2 Example of 16-bit Input Capture Timing for Input Signals



43.5.3 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

The input capture register value is undefined after a reset.

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.

● Notes 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 44 OUTPUT COMPARE

44.1 Overview

This product includes twelve16-bit output compare channels.

For 128MHz operating configuration and notes of motor controls function, see "CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION".

44.2 Features

■ Functions of 16-bit Output Compare

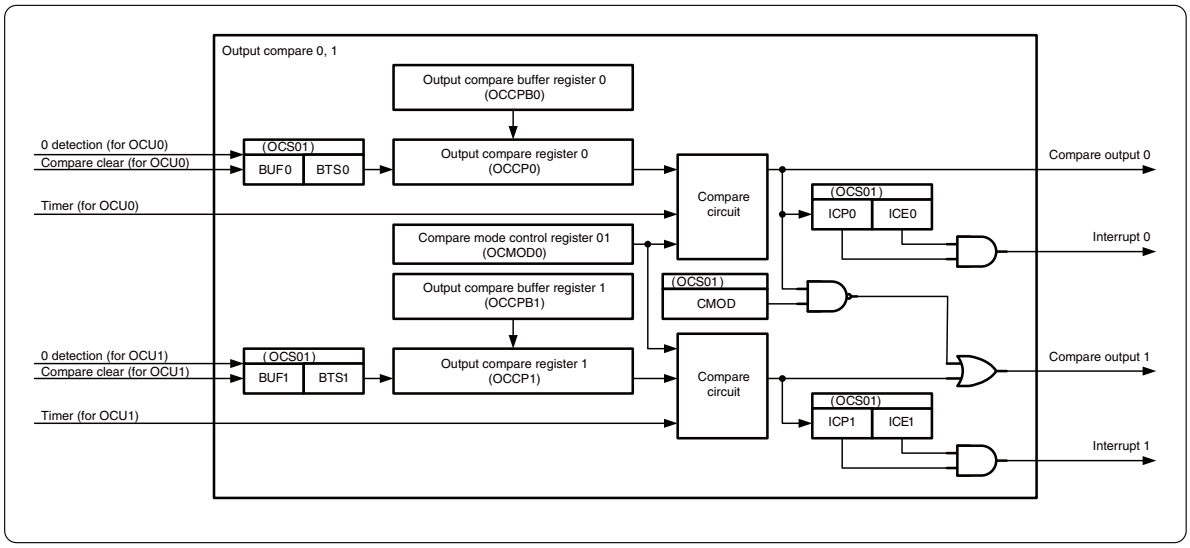
- 16-bit output compare consists of twelve 16-bit compare registers (with selectable buffer registers), compare output latch, 6 compare control registers, and compare mode control registers. When the 16-bit free-run timer value matches a compare register, an interrupt is generated and the output level is inverted.
- The 12 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 5 can be selected as the input for output compare 0 through 5, for output compare channels 0 through 11. This can be set at the free-run timer selection register (FRS0 and FRS2). See Section "42.4.3.1 Free-run Timer Selection Register (FRS)" 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 46 WAVEFORM GENERATOR" for details.

Code: FS18-2v5-91580MS-1-E

44.3 Configuration

■ Configuration of 16-bit Output Compare

Figure 44.3-1 Configuration of 16-bit Output Compare (For ch.0, ch.1)



44.4 Registers

■ Base address (Base-addr)

Channel	Base_addr
0	0x1154
1	0x1154
2	0x115C
3	0x115C
4	0x1164
5	0x1164
6	0x116C
7	0x116C
8	0x1174
9	0x1174
10	0x117C
11	0x117C

■ List of 16-bit Output Compare Registers

Table 44.4-1 List of 16-bit Output Compare Registers

Address	+0	+1	+2	+3
0000_1154	Output compare buffer register 0 (OCCPB0), Output compare register 0 (OCCP0)		Output compare buffer register 1 (OCCPB1), Output compare register 1 (OCCP1)	
0000_1158	Compare control register 01 (OCS01)		Reserved	Compare mode control register 01 (OCMOD01)
0000_115C	Output compare buffer register 2 (OCCPB2), Output compare register 2 (OCCP2)		Output compare buffer register 3 (OCCPB3), Output compare register 3 (OCCP3)	
0000_1160	Compare control register 23 (OCS23)		Reserved	Compare mode control register 23 (OCMOD23)
0000_1164	Output compare buffer register 4 (OCCPB4), Output compare register 4 (OCCP4)		Output compare buffer register 5 (OCCPB5), Output compare register 5 (OCCP5)	
0000_1168	Compare control register 45 (OCS[4][5])		Reserved	Compare mode control register 45 (OCMOD45)
0000_116C	Output compare buffer register 6 (OCCPB6), Output compare register 6 (OCCP6)		Output compare buffer register 7 (OCCPB7), Output compare register 7 (OCCP7)	
0000_1170	Compare control register 67 (OCS67)		Reserved	Compare mode control register 67 (OCMOD67)
0000_1174	Output compare buffer register 8 (OCCPB8), Output compare register 8 (OCCP8)		Output compare buffer register 9 (OCCPB9), Output compare register 9 (OCCP9)	
0000_1178	Compare control register 89 (OCS89)		Reserved	Compare mode control register 89 (OCMOD89)
0000_117C	Output compare buffer register 10 (OCCPB10), Output compare register 10 (OCCP10)		Output compare buffer register 11 (OCCPB11), Output compare register 11 (OCCP11)	
0000_1180	Compare control register 1011 (OCS1011)		Reserved	Compare mode control register 1011 (OCMOD1011)

44.4.1 16-bit Output Compare Registers

The 16-bit output compare consists of output compare buffer registers, output compare registers, compare control registers, and compare mode selection registers.

44.4.1.1 Output Compare Buffer Registers (OCCPB0 to OCCPB11)/ Output Compare Registers (OCCP0 to OCCP11)

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.

x: Channel number 0,2,4,6,8,10

y: Channel number 1,3,5,7,9,11

- **OCCPBx: Address Base_addr+0x00$_H$ (Access: Half-word, Word)**
- **OCCPBy: Address Base_addr+0x02$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	
0	0	0	0	0	0	0	0	Initial value
W	W	W	W	W	W	W	W	Attribute

7	6	5	4	3	2	1	0	bit
OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	
0	0	0	0	0	0	0	0	Initial value
W	W	W	W	W	W	W	W	Attribute

[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).

(Notes) When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

x: Channel number 0,2,4,6,8,10

y: Channel number 1,3,5,7,9,11

- **OCCPx: Address Base_addr+0x00$_H$ (Access: Half-word, Word)**
- **OCCPy: Address Base_addr+0x02$_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute

7	6	5	4	3	2	1	0	bit
OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute

[bit31 to bit16] OP15 to OP00: Compare value bits

	Function
OP15 to OP00	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".

(Notes) When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing the compare register.

44.4.1.2 Compare Control Register (OCS)

The compare control register (OCS) is used to control 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/OUT6 to OUT11.

x: Channel number 0,2,4,6,8,10

y: Channel number 1,3,5,7,9,11

- **OCSxy: Address Base_addr+ $0x04_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved	BTSy	BTSx	CMOD	Reserved		OTDy	OTDx	
0	1	1	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W0	R/W0	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
IOPy	IOPx	IOEy	IOEx	BUFy	BUFx	CSTy	CSTx	
0	0	0	0	1	1	0	0	Initial value
R(RM1),W	R(RM1),W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Reserved

Always write 0 to this bit.

[bit14, bit13] Buffer transfer selection bit

BTSn	Function
0	Transfer is activated when 0 is detected (ch.n).
1	Transfer is activated when a compare clear match occurs (ch.n).

- This bit is used to select the timing of data transfer from the output compare buffer register (OCCPBn) to the output compare register (OCCPn).
- 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.

BTSn: The number n corresponds to the output compare channel number.

[bit12] CMOD: Output level inversion mode bit

CMOD	Function	
0	For the compare mode control register: MODm=0 The compare output m is immediately inverted when there is a match with the output compare register (OCCPm). For the compare mode control register: MODm+1=0 The compare output m+1 is immediately inverted when there is a match with the output compare register (OCCPm+1).	For the compare mode control register: MODm=1 or MODm+1=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: MODm=0 The compare output m is immediately inverted when there is a match with the output compare register (OCCPm). For the compare mode control register: MODm+1=0 The compare output m+1 is immediately inverted when there is a match with the output compare register (OCCPm or OCCPm+1).	For the compare mode control register: MODm=1 or MODm+1=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 46 WAVEFORM GENERATOR" for the output setting method.
- When this bit is set to "0":
  When the compare mode control register: MODm/MODm+1=0
  - For the compare mode control register: MODm=0
    The compare output m is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCPm).
  - For the compare mode control register: MODm+1=0
    The compare output m+1 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCPm+1).

  For the compare mode control register: MODm/MODm+1=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: MODm/MODm+1=0
  - For the compare mode control register: MODm=0
    The compare output m is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCPm).
  - For the compare mode control register: MODm+1=0
    The compare output m+1 is immediately inverted when there is a match between the 16-bit free-run timer and the output compare register (OCCPm or OCCPm+1).
  - When the output compare registers (OCCPm and OCCPm+1) have the same value, the operation is the same as when one compare register is used.

  For the compare mode control register: MODm/MODm+1=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.

  m: Corresponds to the output compare channel number. (m = 0,2,4,6,8,10)

[bit11, bit10] Reserved

Always write 0 to these bits.

[bit9, bit8] Buffer transfer selection bits

OTDn	Function	
	Read	Write
0	Output value of the compare output n	The compare output n outputs "0".
1		The compare output n outputs "1".

- This bit is used to change the compare output n 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 n).

(Notes) A value can be written to this bit when CSTn=0 in the compare control register (OCS).
OTDn: The number n corresponds to the output compare channel number.

[bit7, bit6] Compare match interrupt flag bit

IOPn	Function	
	Read	Write
0	No compare match interrupt occurs for the output compare register (OCCPn).	This bit is cleared.
1	Compare match interrupt occurs for the output compare register (OCCPn).	This bit remains unaffected.

- This bit is an interrupt flag that indicates whether the value of the output compare register (OCCPn) 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 (IOEn) 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.

(Notes) 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.
IOPn: The number n corresponds to the output compare channel number.

[bit5, bit4] Compare match interrupt enable bit

IOEn	Function
0	Compare match interrupt is disabled for the output compare register (OCCPn).
1	Compare match interrupt is enabled for the output compare register (OCCPn).

- This bit is used to enable an output compare interrupt for the output compare register (OCCPn).
- An output compare interrupt occurs if the compare match interrupt flag bit (IOPn) is set while this bit is set to "1".

IOEn: The number n corresponds to the output compare channel number.

[bit3, bit2] Compare buffer invalidating bit

BUFn	Function
0	Validates the compare buffer of the output compare register (OCCPn).
1	Invalidates the compare buffer of the output compare register (OCCPn).

- This bit is used to invalidate the buffer function of the output compare register (OCCPn).
- When this bit is set to "0": This buffer function is validated.

BUFn: The number n corresponds to the output compare channel number.

[bit1, bit0] Compare operation enable bit

CSTn	Function
0	Disables the compare operation of the output compare register (OCCPn).
1	Enables the compare operation of the output compare register (OCCPn).

- This bit is used to enable the compare operation between the 16-bit free-run timer and the output compare register (OCCPn).
- Before enabling the compare operation, be sure to write a value to the output compare register (OCCPn) and the timer data register of the free-run timer (TCDT[x], where x is a pertinent free-run timer).

CSTn: The number n corresponds to the output compare channel number.

44.4.1.3 Compare Mode Control Register (OCMOD)

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.

x: Channel number 0,2,4,6,8,10

y: Channel number 1,3,5,7,9,11

- **OCMODxy: Address Base_addr+0x07$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved						MODy	MODx	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	RR0,W0	R/W	R/W	Attribute

[bit7 to bit2] Reserved

Always write 0 to these bits.

[bit1, bit0] Compare match mode setting bit

MODn	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 (OCSxy).

- 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 (OCSxy).

(Notes) Be sure to stop the compare operation before writing a value to this bit.
MODn: The number n corresponds to the output compare channel number.

44.5 Operation Explanation

44.5.1 Interrupts for 16-bit Output Compare

Table 44.5-1 shows the interrupt control bits and interrupt factor of the 16-bit output compare.

Table 44.5-1 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.

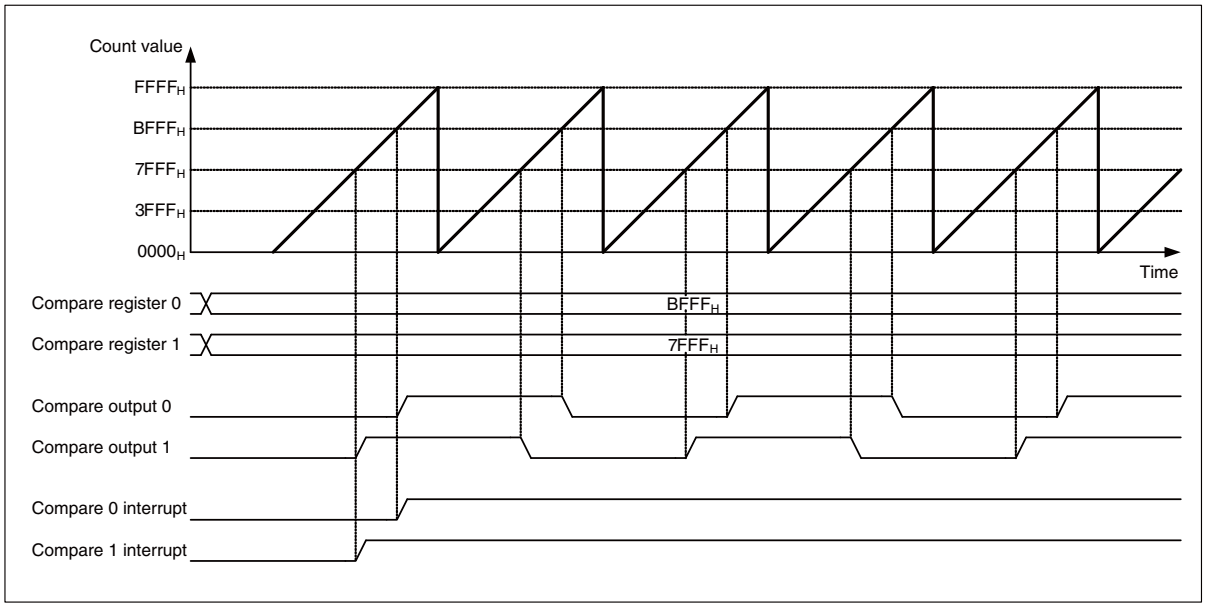
44.5.2 Operation of 16-bit Output Compare

The output compare is used to compare the value set in the specified compare 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.

44.5.2.1 Operation of 16-bit Output Compare (Inverted Mode, MOD0= 0 in OCMOD01 Register)

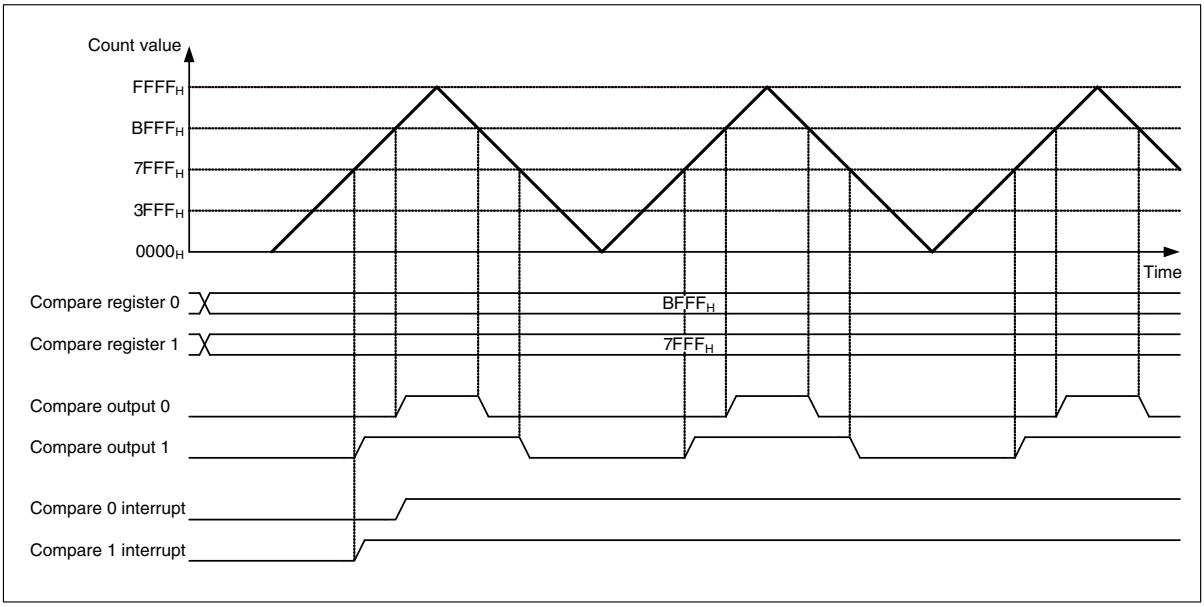
The compare operation can be performed in each of the channels (CMOD: bit12=0 in the compare control register (OCS01)).

Figure 44.5-1 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, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

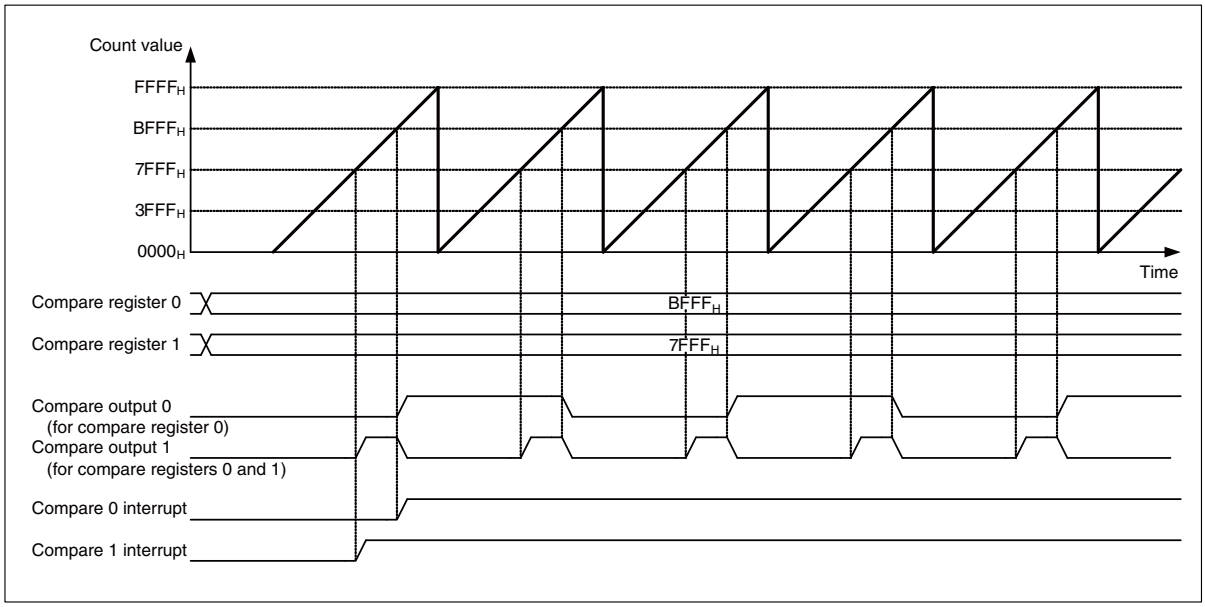
Figure 44.5-2 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used Independently (With the free-run timer in up/down count mode)



- For ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

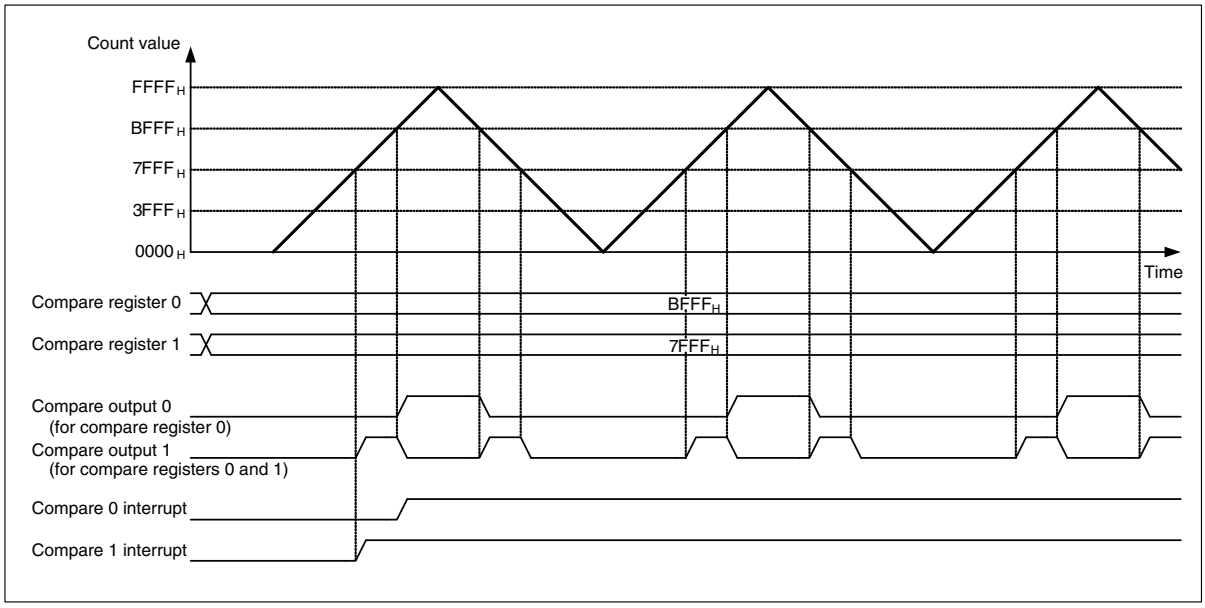
- The output level can be changed using a pair of compare registers (CMOD: bit12 = 1 in OCS01).

Figure 44.5-3 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, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

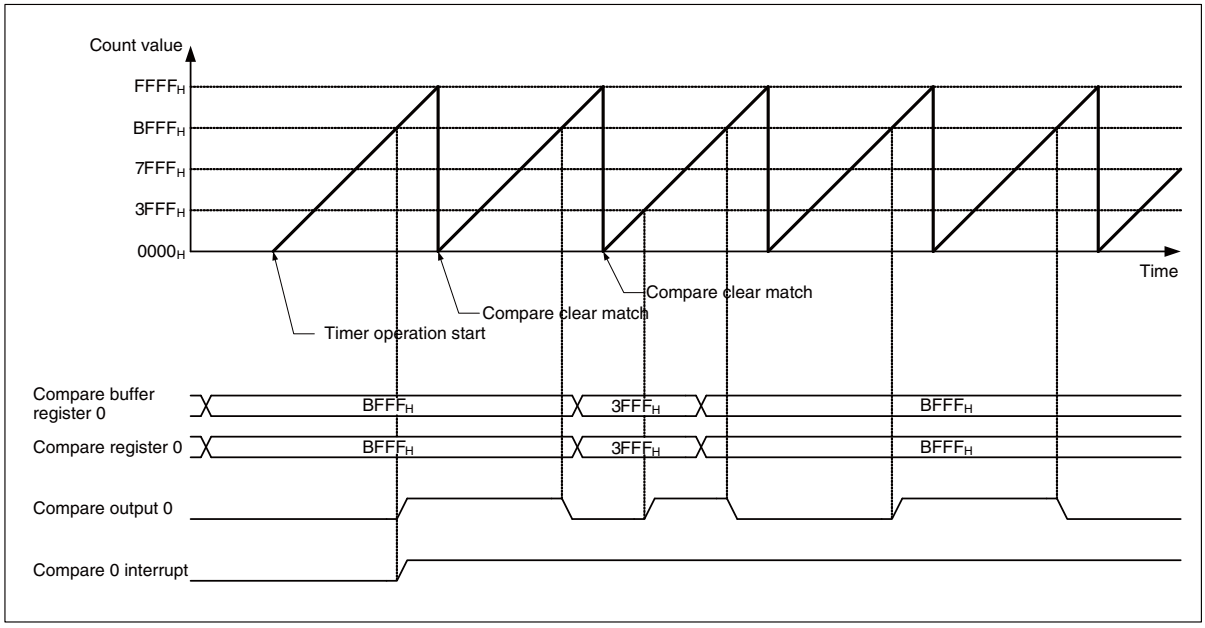
Figure 44.5-4 Example of Output Waveform When the Initial Output Value Is "0" and Compare Registers 0 and 1 Are Used Simultaneously (With the free-run timer in up/down count mode)



- For ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

● Output level when the compare buffer is invalid

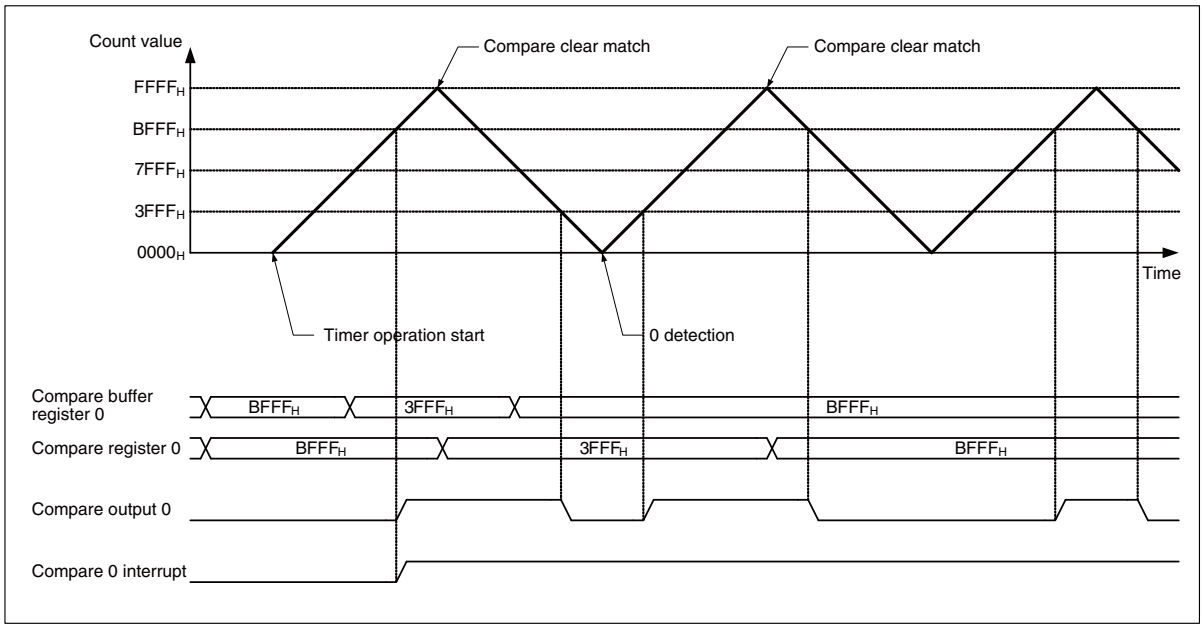
Figure 44.5-5 Example of Output Waveform When the Compare Buffer Is Invalid (With the free-run timer in up count mode)



- For ch.1,2,3,4,5,6,7,8,9,10 and 11, the operation is the same as ch.0.

● Output level when the compare buffer is selected upon a compare clear match

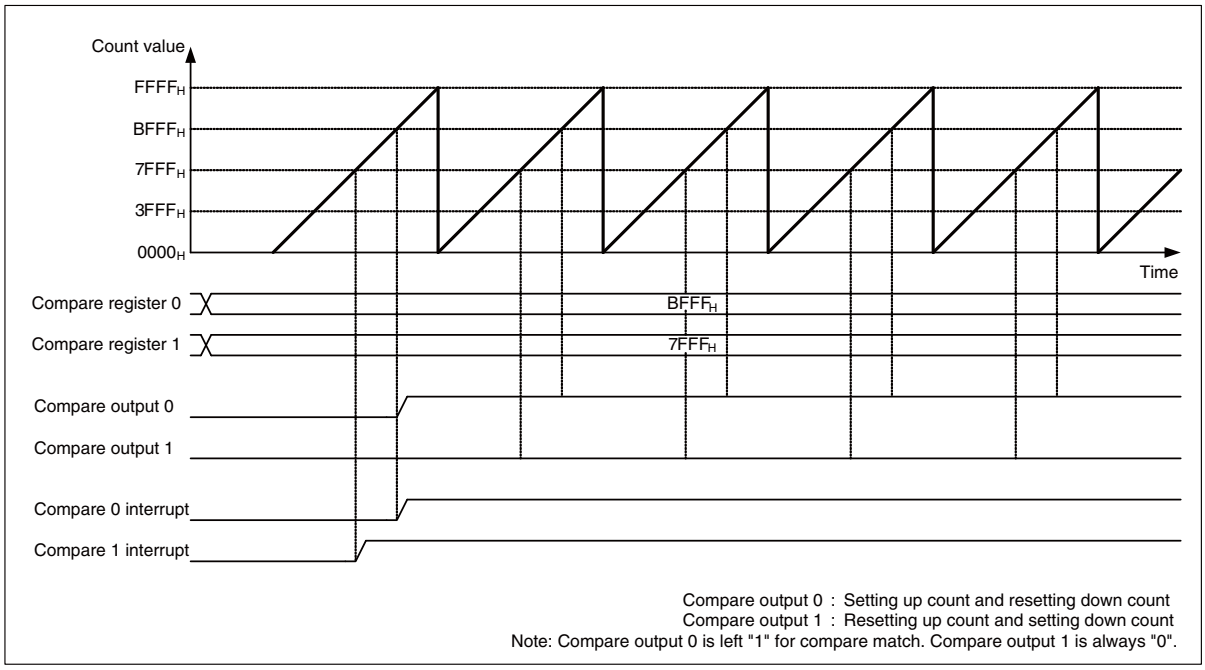
Figure 44.5-6 Example of Output Waveform When the Compare Buffer Is Valid (With the free-run timer in up/down count mode)



- For ch.1,2,3,4,5,6,7,8,9,10 and 11, the operation is the same as ch.0.

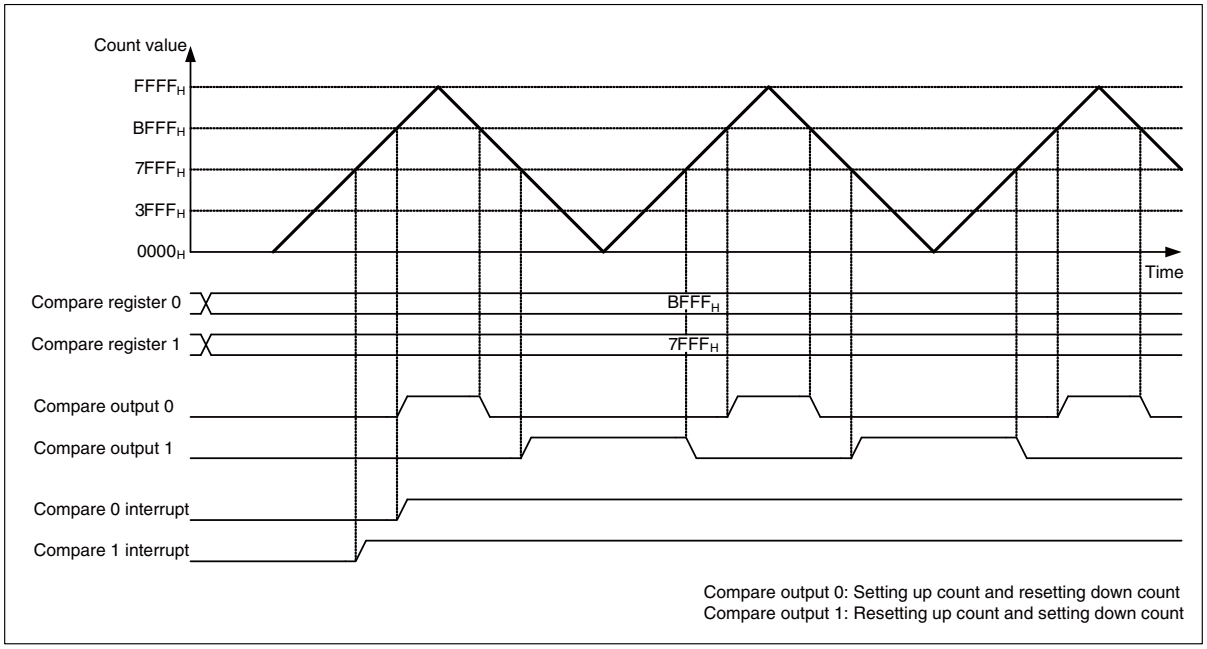
44.5.2.2 Operation of 16-bit Output Compare (Set/Reset Mode, MOD0 = 1 in OCMOD01 Register)

Figure 44.5-7 Operation of 16-bit Output Compare (Set/Reset Mode) #1



- For ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

Figure 44.5-8 Operation of 16-bit Output Compare (Set/Reset Mode) #2



- For ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11, the operation is the same as ch.0,1.

44.5.2.3 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 44.5-9 Compare Register Interrupt Timing

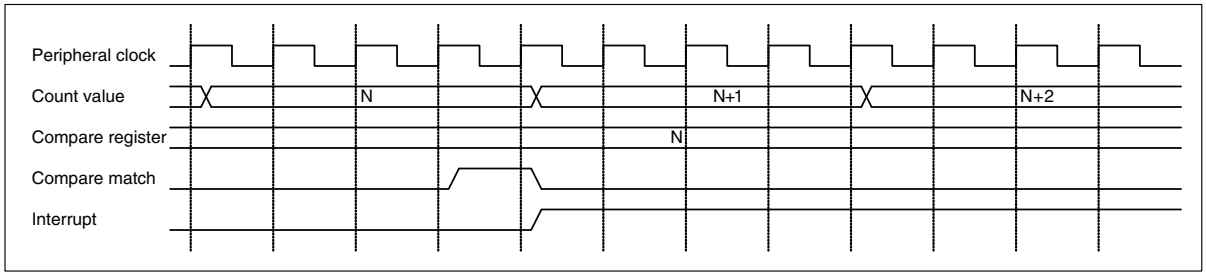
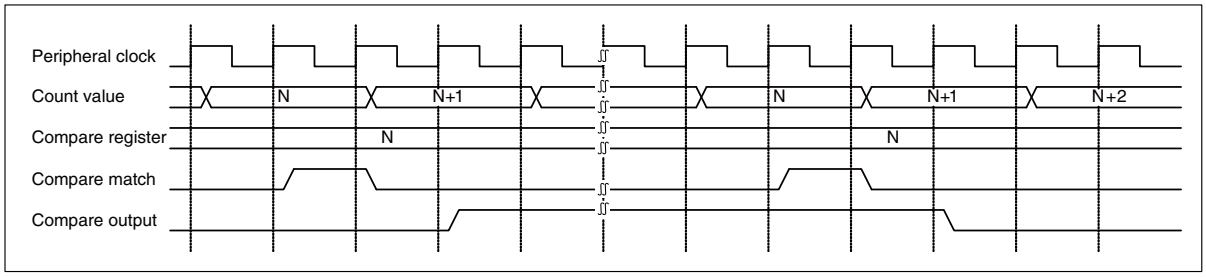


Figure 44.5-10 Pin Output Change Timing



When the free-run timer is stopped or started during the output compare is enabled(OCSxy.CSTn=1), the output compare operation is as follows;

- When the initial counter value of free-run timer is same as the compare register value, the output compare generates a compare match signal concurrently with the start of the free-run timer.
- When the free-run timer is stopped at a free-run timer match with a compare register while output compare is running(OCSxy.CSTn=1), the output compare generates a compare match signal. However, the output compare doesn't generate a compare match signal under the following conditions.
  - When any of the following applies while the free-run timer is running on up-count mode.
    - An output compare buffer register was written concurrently with a compare matched timing when the output compare buffering function is disabled(OCSxy.BUFn=1).
    - The free-run timer was stopped before 2 peripheral clock times from the output timing of the compare matched signal.
  - When any of the following applies while the free-run timer is running on up/down-count mode.
    - The output compare buffer function is disabled(OCSxy.BUFn=1).
    - Transfer is activated when 0 of the free-run timer is detected(OCSxy.BTSn=0).
    - The CMOD bit of the compare control register is set to "0"(OCSxy.CMOD=0).
    - The MOD bit of the compare mode control register is set to "0"(OCMODxy.MODn=0).

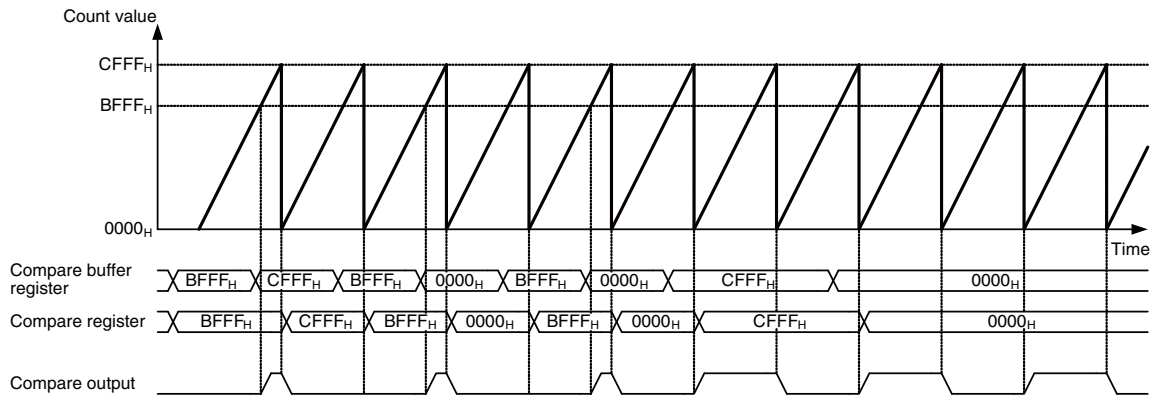
If the free-run timer is stopped at the counter value of the free-run timer matched a compare register value while the output compare is running(OCSxy.CSTn=1), after that the free-run timer is started while the free-run timer count value is matched the output compare register, the output compare may generate a compare matched signal respectively at the time of a stop and start of the free-run timer.

The output compare doesn't generate a compare match signal, when the free-run timer is stopped during the output compare is disabled(OCSxy.CSTn=0).

44.5.2.4 Operation of 16-bit Output Compare and Free-run Timer

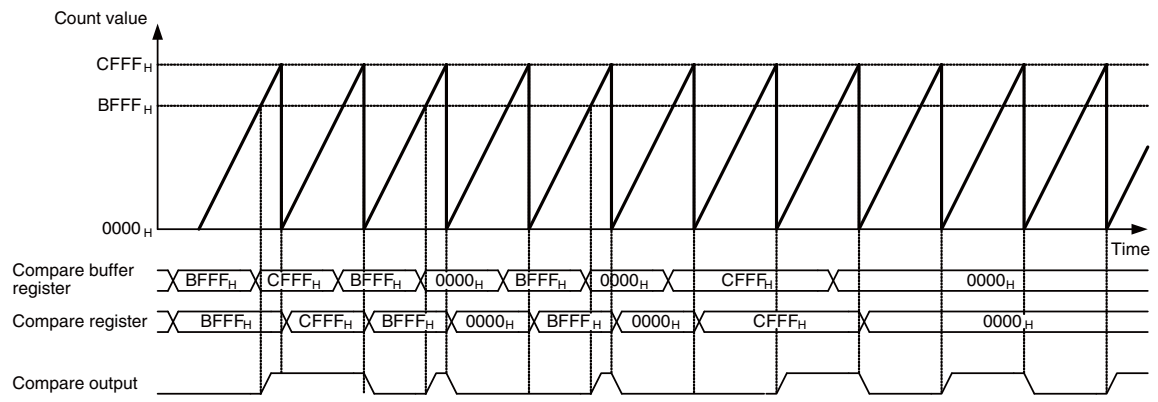
- Case #1 where the free-run timer is in up count mode

Figure 44.5-11 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 44.5-12 Case #2 Where the Free-run Timer Is in Up Count Mode



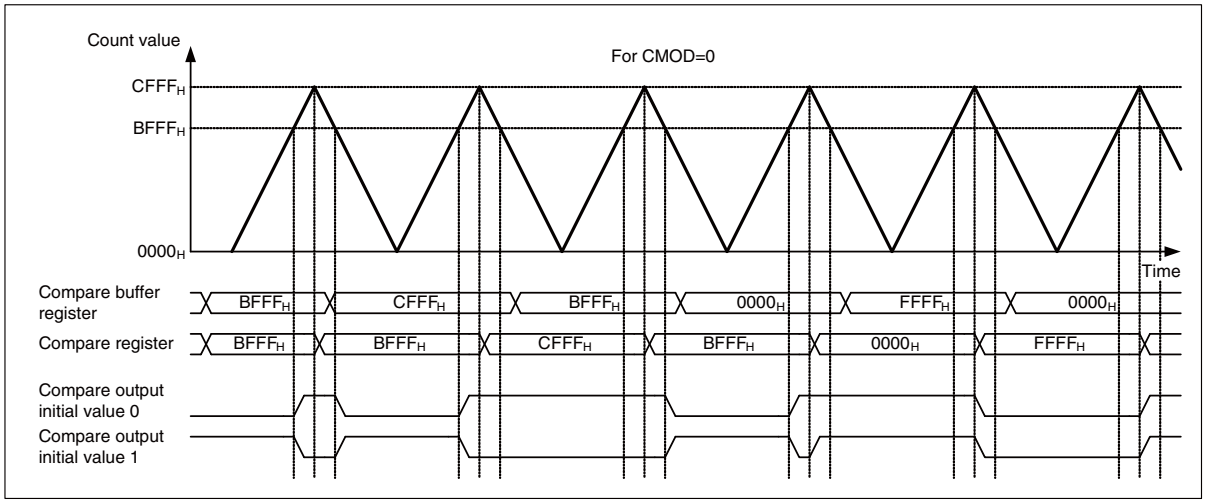
● Case #1 where the free-run timer is in up/down count mode

- 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 44.5-13 Case #1 Where the Free-run Timer Is in Up/Down Count Mode



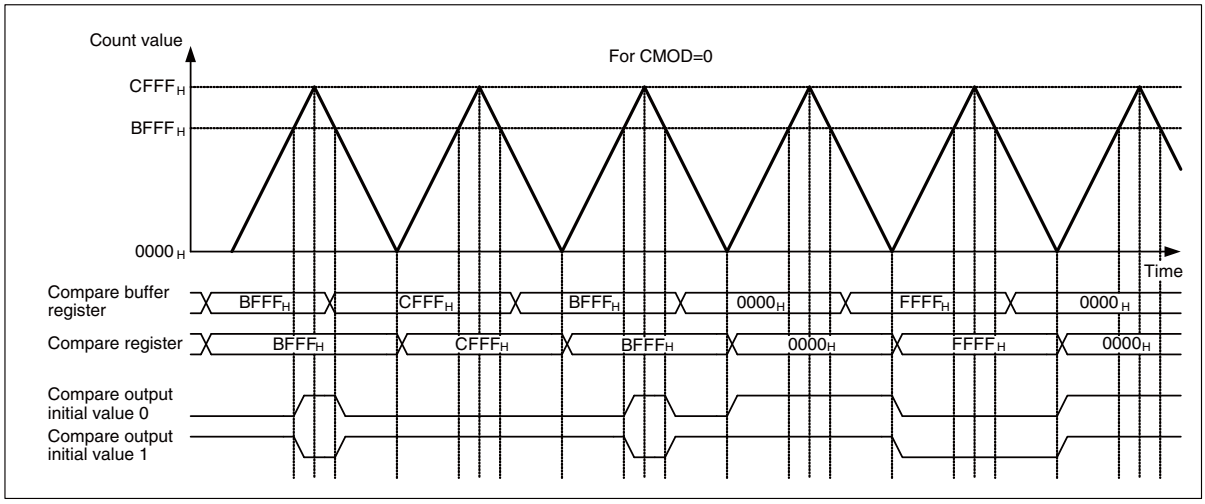
● Case #2 where the free-run timer is in up/down count mode

- 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 44.5-14 Case #2 Where the Free-run Timer Is in Up/Down Count Mode



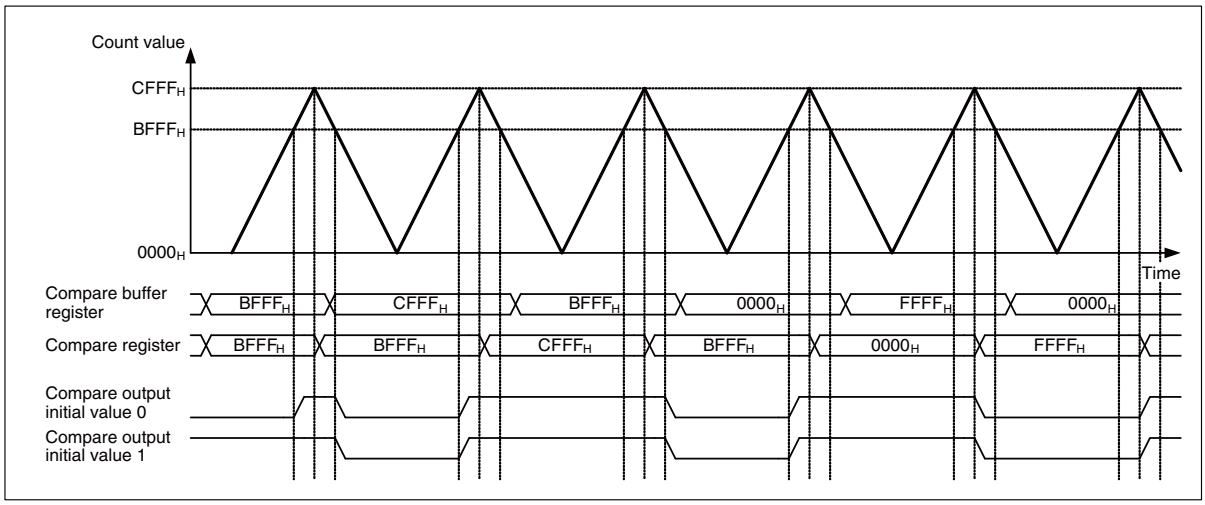
● Case #3 where the free-run timer is in up/down count mode

- 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.0,1, ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11 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 44.5-15 Case #3 Where the Free-run Timer Is in Up/Down Count Mode



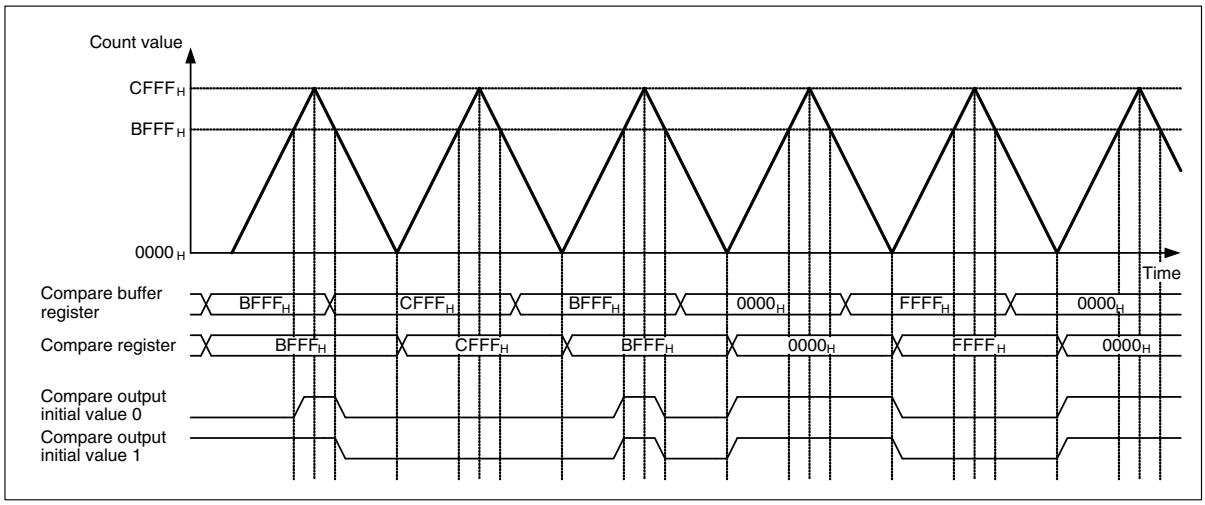
● Case #4 where the free-run timer is in up/down count mode

- 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.0,1, ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11 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 44.5-16 Case #4 Where the Free-run Timer Is in Up/Down Count Mode



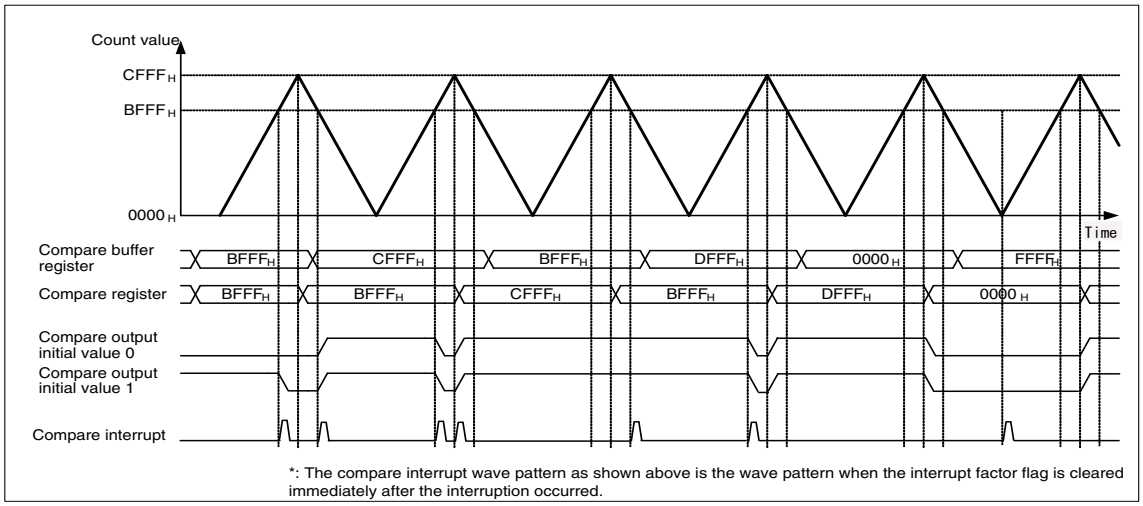
● Case #5 where the free-run timer is in up/down count mode

- 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.0,1, ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11 have the same operation.

<Notes:>

- When the compare register value is set to "0000H", 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 44.5-17 Case #5 Where the Free-run Timer Is in Up/Down Count Mode



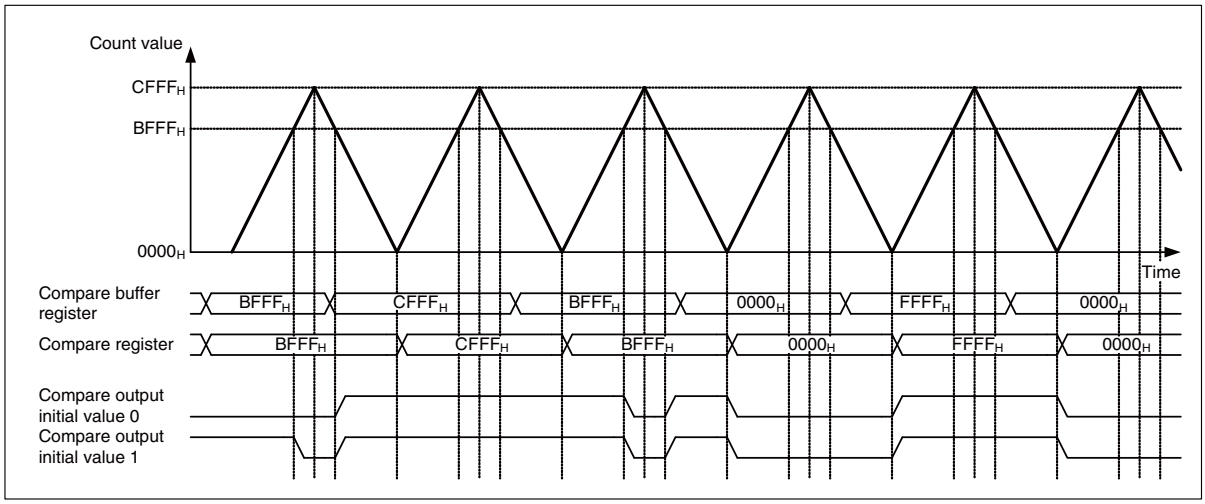
● Case #6 where the free-run timer is in up/down count mode

- 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.0,1, ch.2,3, ch.4,5, ch.6,7, ch.8,9 and ch.10,11 have the same operation.

<Notes:>

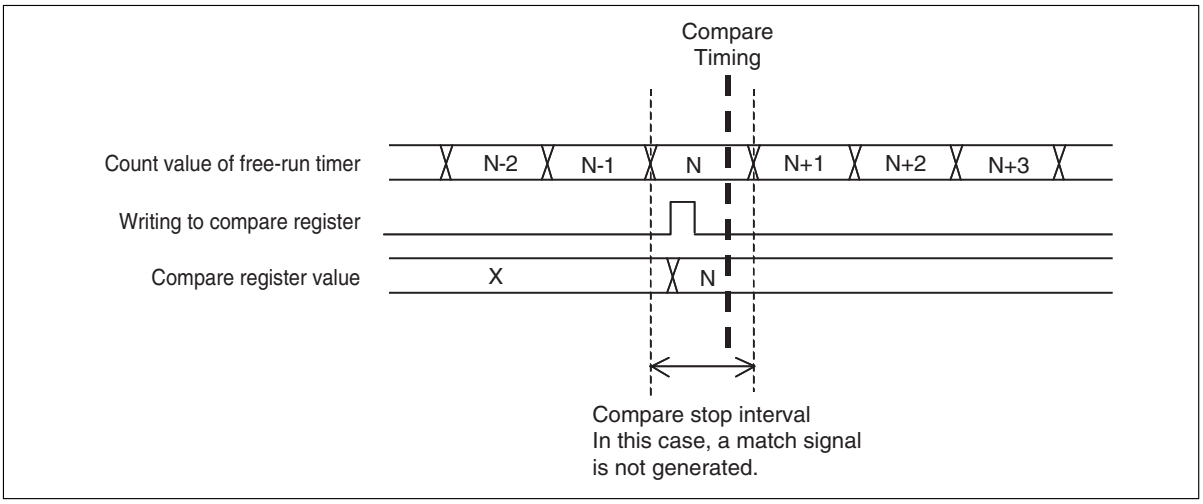
- When the compare register value is set to "0000H", the output compare output is reset to "0" regardless of the count value of the free-run timer.
- When the compare register value is set to "FFFFH", the output compare output is set to "1" regardless of the count value of the free-run timer.
- No comparison 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 44.5-18 Case #6 Where the Free-run Timer Is in Up/Down Count Mode



44.5.3 Notes on Using 16-bit Output Compare

- 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, OCCP4 = OCCP5, OCCP6 = OCCP7, OCCP8 = OCCP9, and OCCP10 = OCCP11, 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 OCU11 when the compare mode bit CMOD is set to 1.

● Read-modify-write

The interrupt request flag bits (IOP0), (IOP1), (IOP2), (IOP3), (IOP4), (IOP5), (IOP6), (IOP7), (IOP8), (IOP9), (IOP10) and (IOP11) are "1" when read using a read-modify-write instruction.

● Notes 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).

CHAPTER 45 12-BIT A/D CONVERTER

45.1 Overview

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.

The MB91F583AM/F584AM/F585AM A/D converter contains 23 channels. It does not contain channel 15. The MB91F583AS/F584AS/F585AS A/D converter contains 17 channels. It does not contain channels 15 to 18 and 21 to 23.

For the 128 MHz operation setting for and notes on the motor control function, see CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION.

45.2 Features

■ Function of A/D Activation Compare

- This function can enable/disable 23 channels of analog inputs.
- It performs operation for A/D activation request control and A/D conversion data storage.
- A certain number of A/D activation channels correspond to one 12-bit A/D converter unit. Activation channels 0 to 7 correspond to 12-bit A/D converter unit 0; activation channels 8 to 14 correspond to 12-bit A/D converter unit 1; and activation channels 16 to 23 correspond to 12-bit A/D converter unit 2.
- Each activation channel consists of a compare register, an A/D activation trigger control status register, and an A/D data register.
- Each activation channel issues an A/D activation request by one of the following methods: software, external trigger (falling), reload timer (rising), and compare match. 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, and reload timer 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 selection register (FRS4, FRS5 or FRS6). See Section "42.4.3.1 Free-run Timer Selection Register (FRS)" 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 ends.
- 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.
- When A/D conversion ends, 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.
- Each activation channel can generate an interrupt request when A/D conversion ends.
- Each A/D data register can be configured to enable/disable the data protection function. Note that the data protection function works when the activation factor is not a compare match.

Code: FIP004_FS24-2v3-91580MS-1-E

- If the protection function is enabled, 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. The current A/D conversion request or conversion operation can be forced to termination by resetting the appropriate bit to "0".

■ Function of A/D Activation Arbitration

- A/D activation arbitration is provided for each 12-bit A/D converter unit. Activation channels 0 to 7 of the A/D activation compare are handled by A/D activation arbitration 0; activation channels 8 to 14 are handled by A/D activation arbitration 1; and activation channels 16 to 23 are handled by A/D activation arbitration 2.
- The A/D activation arbitration consists of arbitration circuit, an A/D activation trigger generation section, and an analog channel number selection 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 or reload timer > software activation (priority control based on activation factors)". Activation requests that are not selected are made to wait. When the current A/D conversion ends, 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 serviced 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 ends, 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 ends, 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 ends, 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.

■ Functions of 12-bit A/D Converter Control

- There are three 12-bit A/D converters 0 to 2, to which analog input pins are assigned.

A function used to A/D-convert analog voltages (input voltages) input to the analog input pins to digital values is provided. Its features are as follows:

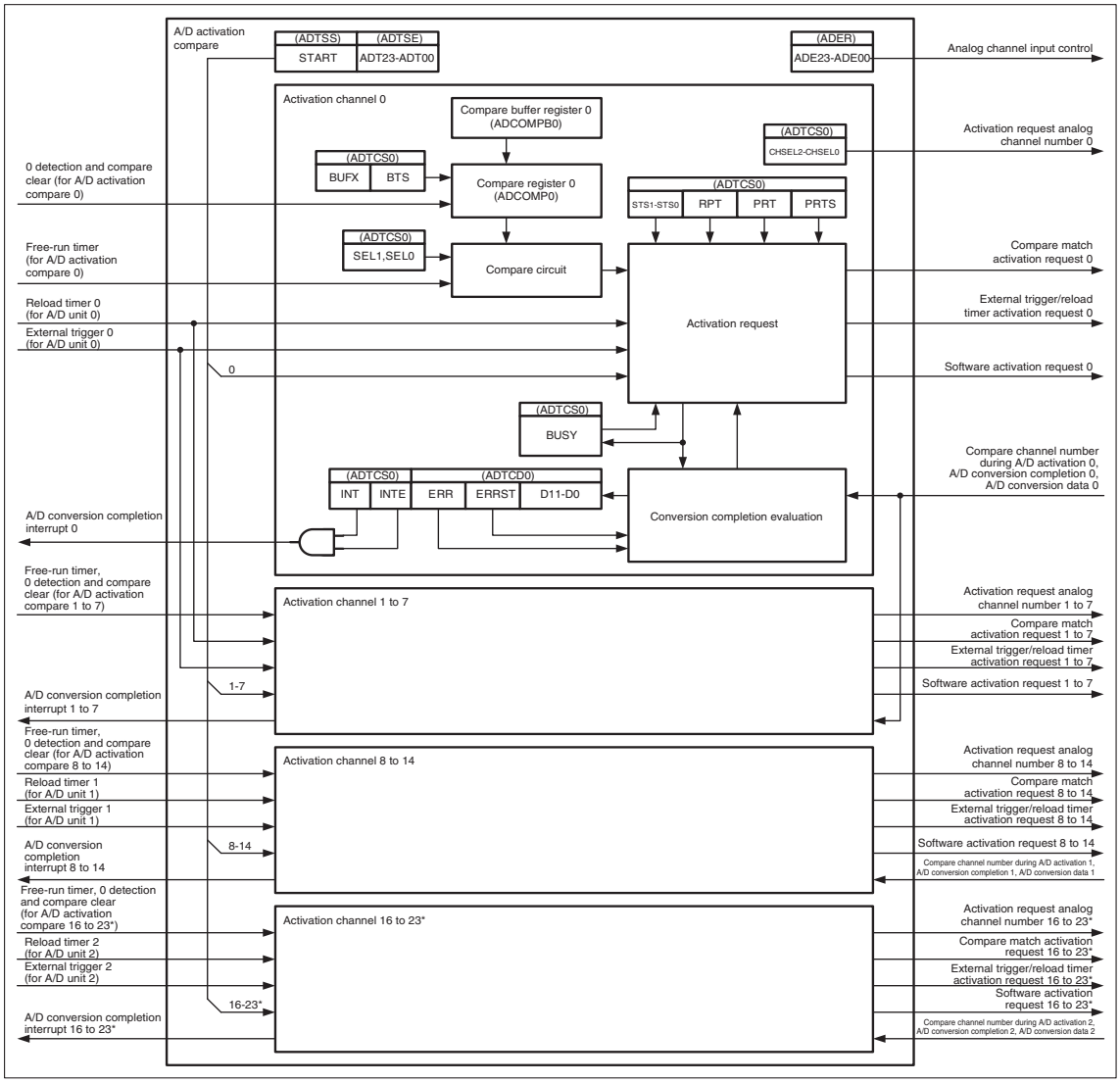
- The conversion time is at least 1.0 μs (including sampling time).
- The conversion method is the RC successive approximation conversion method with a sample hold circuit.
- The analog input pins can be selected under program control. (This program is configured in the A/D activation compare section)
- Activation signals are entered as pulse signals.
- One sequence of A/D conversion is performed for one activation factor.
- If an activation signal is entered again during A/D conversion, reactivation is performed. (Reactivation function)
- If an A/D conversion cancel signal is received during A/D conversion, the current processing is stopped and the converter is initialized. (Forced stop function)

(Note) The conversion time of repeat conversion is 1.1 μs (including sampling time).

45.3 Configuration

■ Configuration of A/D Activation Compare

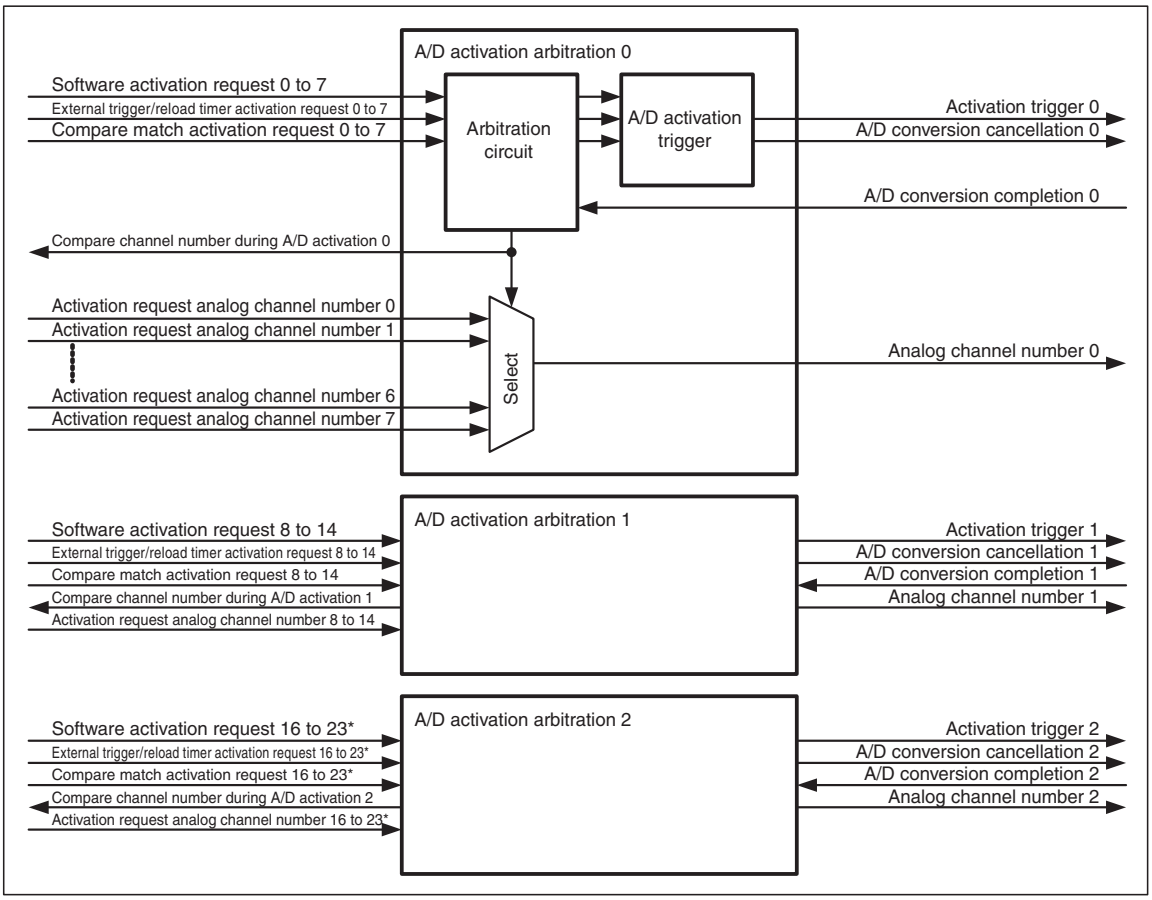
Figure 45.3-1 Configuration of A/D Activation Compare



*: Since the MB91F583AS/F584AS/F585AS A/D converter contains 17 channels, channels 19 and 20 apply for the models.

■ Configuration of A/D Activation Arbitration

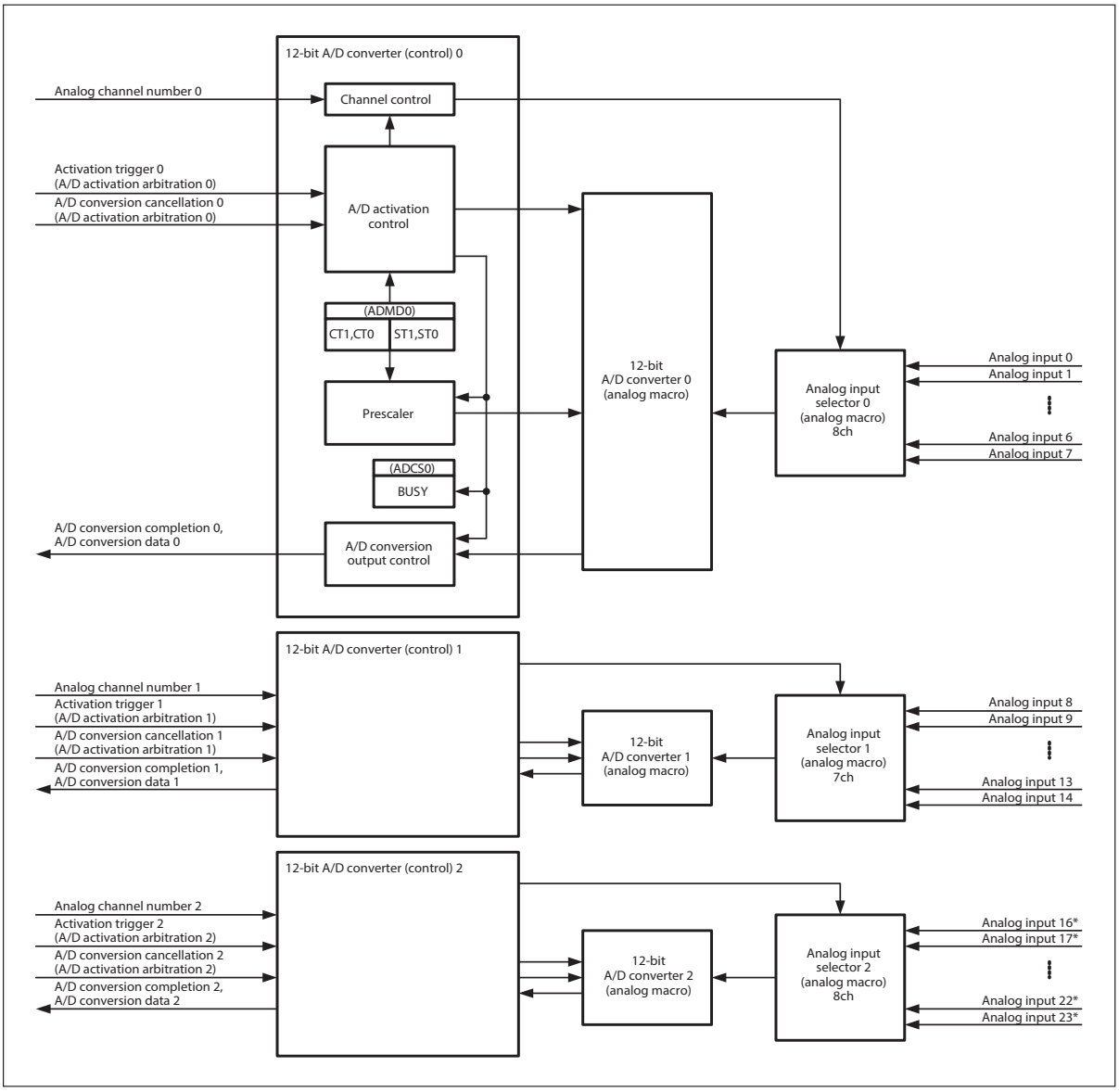
Figure 45.3-2 Configuration of A/D Activation Arbitration



*: Since the MB91F583AS/F584AS/F585AS A/D converter contains 17 channels, channels 19 and 20 apply for the models.

■ Configuration of 12-bit A/D Converter Control

Figure 45.3-3 Configuration of 12-bit A/D Converter Control



*: Since the MB91F583AS/F584AS/F585AS A/D converter contains 17 channels, only analog input 19 and analog input 20 are input in analog input selector 2.

45.4 Registers

Table 45.4-1 List of Analog Input Control Register

Address	+0	+1	+2	+3
$0F48_H$	Analog input control register high (ADERH) (key code target register)		Analog input control register low (ADERL) (key code target register)	

Table 45.4-2 List of A/D Activation Compare Registers (1 / 3)

Address	+0	+1	+2	+3
$11DC_H$	A/D software activation register (ADTSS)	Reserved	Reserved	Reserved
$11E0_H$	A/D software activation channel selection register (ADTSE)			
$11E4_H$	Compare buffer register 0 (ADCOMPB0), Compare register 0 (ADCOMP0)		Compare buffer register 1 (ADCOMPB1), Compare register 1 (ADCOMP1)	
$11E8_H$	Compare buffer register 2 (ADCOMPB2), Compare register 2 (ADCOMP2)		Compare buffer register 3 (ADCOMPB3), Compare register 3 (ADCOMP3)	
$11EC_H$	Compare buffer register 4 (ADCOMPB4), Compare register 4 (ADCOMP4)		Compare buffer register 5 (ADCOMPB5), Compare register 5 (ADCOMP5)	
$11F0_H$	Compare buffer register 6 (ADCOMPB6), Compare register 6 (ADCOMP6)		Compare buffer register 7 (ADCOMPB7), Compare register 7 (ADCOMP7)	
$11F4_H$	Compare buffer register 8 (ADCOMPB8), Compare register 8 (ADCOMP8)		Compare buffer register 9 (ADCOMPB9), Compare register 9 (ADCOMP9)	
$11F8_H$	Compare buffer register 10 (ADCOMPB10), Compare register 10 (ADCOMP10)		Compare buffer register 11 (ADCOMPB11), Compare register 11 (ADCOMP11)	
$11FC_H$	Compare buffer register 12 (ADCOMPB12), Compare register 12 (ADCOMP12)		Compare buffer register 13 (ADCOMPB13), Compare register 13 (ADCOMP13)	
1200_H	Compare buffer register 14 (ADCOMPB14), Compare register 14 (ADCOMP14)		Reserved	

Table 45.4-2 List of A/D Activation Compare Registers (2 / 3)

Address	+0	+1	+2	+3
1204_H	Compare buffer register 16 (ADCOMPB16), Compare register 16 (ADCOMP16)		Compare buffer register 17 (ADCOMPB17), Compare register 17 (ADCOMP17)	
1208_H	Compare buffer register 18 (ADCOMPB18), Compare register 18 (ADCOMP18)		Compare buffer register 19 (ADCOMPB19), Compare register 19 (ADCOMP19)	
$120C_H$	Compare buffer register 20 (ADCOMPB20), Compare register 20 (ADCOMP20)		Compare buffer register 21 (ADCOMPB21), Compare register 21 (ADCOMP21)	
1210_H	Compare buffer register 22 (ADCOMPB22), Compare register 22 (ADCOMP22)		Compare buffer register 23 (ADCOMPB23), Compare register 23 (ADCOMP23)	
1214_H | 1220_H	Reserved		Reserved	
1224_H	A/D activation trigger control status register 0 (ADTCS0)		A/D activation trigger control status register 1 (ADTCS1)	
1228_H	A/D activation trigger control status register 2 (ADTCS2)		A/D activation trigger control status register 3 (ADTCS3)	
$122C_H$	A/D activation trigger control status register 4 (ADTCS4)		A/D activation trigger control status register 5 (ADTCS5)	
1230_H	A/D activation trigger control status register 6 (ADTCS6)		A/D activation trigger control status register 7 (ADTCS7)	
1234_H	A/D activation trigger control status register 8 (ADTCS8)		A/D activation trigger control status register 9 (ADTCS9)	
1238_H	A/D activation trigger control status register 10 (ADTCS10)		A/D activation trigger control status register 11 (ADTCS11)	
$123C_H$	A/D activation trigger control status register 12 (ADTCS12)		A/D activation trigger control status register 13 (ADTCS13)	
1240_H	A/D activation trigger control status register 14 (ADTCS14)		Reserved	
1244_H	A/D activation trigger control status register 16 (ADTCS16)		A/D activation trigger control status register 17 (ADTCS17)	
1248_H	A/D activation trigger control status register 18 (ADTCS18)		A/D activation trigger control status register 19 (ADTCS19)	
$124C_H$	A/D activation trigger control status register 20 (ADTCS20)		A/D activation trigger control status register 21 (ADTCS21)	
1250_H	A/D activation trigger control status register 22 (ADTCS22)		A/D activation trigger control status register 23 (ADTCS23)	
1254_H | 1260_H	Reserved		Reserved	

Table 45.4-2 List of A/D Activation Compare Registers (3 / 3)

Address	+0	+1	+2	+3
1264_H	A/D data register 0 (ADTCD0)		A/D data register 1 (ADTCD1)	
1268_H	A/D data register 2 (ADTCD2)		A/D data register 3 (ADTCD3)	
$126C_H$	A/D data register 4 (ADTCD4)		A/D data register 5 (ADTCD5)	
1270_H	A/D data register 6 (ADTCD6)		A/D data register 7 (ADTCD7)	
1274_H	A/D data register 8 (ADTCD8)		A/D data register 9 (ADTCD9)	
1278_H	A/D data register 10 (ADTCD10)		A/D data register 11 (ADTCD11)	
$127C_H$	A/D data register 12 (ADTCD12)		A/D data register 13 (ADTCD13)	
1280_H	A/D data register 14 (ADTCD14)		Reserved	
1284_H	A/D data register 16 (ADTCD16)		A/D data register 17 (ADTCD17)	
1288_H	A/D data register 18 (ADTCD18)		A/D data register 19 (ADTCD19)	
$128C_H$	A/D data register 20 (ADTCD20)		A/D data register 21 (ADTCD21)	
1290_H	A/D data register 22 (ADTCD22)		A/D data register 23 (ADTCD23)	
1294_H | $12A0_H$	Reserved		Reserved	

Table 45.4-3 List of 12-bit A/D Converter Control Registers

Address	+0	+1	+2	+3
$12A4_H$	A/D control status register 0 (ADCS0)		A/D channel status register 0 (ADCH0)	A/D mode setting register 0 (ADMD0)
$12A8_H$	A/D control status register 1 (ADCS1)		A/D channel status register 1 (ADCH1)	A/D mode setting register 1 (ADMD1)
$12AC_H$	A/D control status register 2 (ADCS2)		A/D channel status register 2 (ADCH2)	A/D mode setting register 2 (ADMD2)

45.4.1 Analog Input Enable Registers

45.4.1.1 Analog Input Enable Register (ADER)

The analog input enable register (ADERH, ADERL) sets the pins used for analog input.

- **ADERH: Address 0F48$_H$ (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	bit
-								
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	Attribute

7	6	5	4	3	2	1	0	bit
ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16	
1	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

[bit15 to bit8] Undefined

"1" is always read. Writing has no influence on an operation.

[bit7 to bit0] ADE23 to ADE16: Analog input enable bits

ADE23 to ADE16	Function
0	Port input/output mode
1	Analog input mode

- Controls analog input pins.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

(Notes) This register is a key code target register. Key code setting is required for writing. For the setting method, refer to Sections "24.4.10 KEYCDR key code register" and "24.5.10 Key Code Register Function Settings" in "CHAPTER 24 I/O PORTS". In addition, word access to this register is disabled.

- **ADERL: Address 0F4A$_H$ (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	bit
Reserved	ADE14	ADE13	ADE12	ADE11	ADE10	ADE09	ADE08	
1	1	1	1	1	1	1	1	Initial value
R/W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
ADE07	ADE06	ADE05	ADE04	ADE03	ADE02	ADE01	ADE00	
1	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

[bit15] Reserved

Always write "1" to this bit.

[bit14 to bit0] ADE14 to ADE00: Analog input enable bits

ADE14 to ADE00	Function
0	Port input/output mode
1	Analog input mode

- Controls analog input pins.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

(Notes) This register is a key code target register. Key code setting is required for writing. For the setting method, refer to Sections "24.4.10 KEYCDR key code register" and "24.5.10 Key Code Register Function Settings" in "CHAPTER 24 I/O PORTS". In addition, word access to this register is disabled.

45.4.2 A/D Activation Compare Register

45.4.2.1 A/D Software Activation Register (ADTSS)

The A/D software activation register (ADTSS) issues a 12-bit converter A/D activation request. The activation channel is specified by the A/D software activation channel selection register (ADTSE).

- **ADTSS: Address $11DC_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved							START	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W	Attribute

[bit7 to bit1] Reserved

Always write 0 to these bits.

[bit0] START: A/D conversion activation bit (software)

START	Function
0	Does not start the A/D conversion function.
1	Starts the A/D conversion function.

- This bit starts the A/D conversion operation under software control.
- The A/D conversion starts when this bit is set to "1". The activation channel is specified by the A/D software activation channel selection register (ADTSE).
- The A/D conversion cannot be reactivated by changing this bit.

45.4.2.2 A/D Software Activation Channel Selection Register (ADTSE)

The A/D software activation channel selection register (ADTSE) is used to select an activation channel that requests A/D activation.

- **ADTSE: Address $11E0_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
-								
1	1	1	1	1	1	1	1	Initial value
R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	Attribute

23	22	21	20	19	18	17	16	bit
ADT23	ADT22	ADT21	ADT20	ADT19	ADT18	ADT17	ADT16	
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
Reserved	ADT14	ADT13	ADT12	ADT11	ADT10	ADT09	ADT08	
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

7	6	5	4	3	2	1	0	bit
ADT07	ADT06	ADT05	ADT04	ADT03	ADT02	ADT01	ADT00	
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

[bit31 to bit24] Undefined

"1" is always read. Writing has no influence on an operation.

[bit23 to bit16] ADT23 to ADT16: Software activation channel selection bits

ADT23 to ADT16	Function
0	Software activation disabled
1	Software activation enabled

- Controls 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.

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] ADT14 to ADT00: Software activation channel selection bits

ADT14 to ADT00	Function
0	Software activation disabled
1	Software activation enabled

- Controls 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.

45.4.2.3 Compare Buffer Register (ADCOMPB0 to ADCOMPB23)/ Compare Register (ADCOMP0 to ADCOMP23)

The compare buffer register (ADCOMPB) is a 16-bit buffer register for the compare register (ADCOMP).

The compare register (ADCOMP) activates the A/D converter when its value matches the free-run timer value.

The ADCOMPB and ADCOMP registers are located at the same address.

- **ADCOMPB0 to ADCOMPB23: Address $11E4_H$ to 1212_H(Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	
0	0	0	0	0	0	0	0	Initial value
W	W	W	W	W	W	W	W	Attribute

7	6	5	4	3	2	1	0	bit
CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	
0	0	0	0	0	0	0	0	Initial value
W	W	W	W	W	W	W	W	Attribute

[bit15 to bit0] CMP15 to CMP00: Compare value buffer bits

CMP15 to CMP00	Function
Compare value buffer	

- The compare buffer register is a 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 ASCOMP23: Address $11E4_H$ to 1212_H (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute

7	6	5	4	3	2	1	0	bit
CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute

[bit15 to bit0] CMP15 to CMP00: Compare value bits

	Function
CMP15 to CMP00	Compare value

- 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=11_B holds true in the A/D activation trigger control status register (ADTCS).

(Notes) 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.

45.4.2.4 A/D Activation Trigger Control Status Register (ADTCS0 to ADTCS23)

The A/D activation trigger control status register (ADTCS) is used to verify 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 ADTCS23:Address 1224_H to 1252_H(Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
BUSY	INT	INTE	STS1	STS0	RPT	PRT	PRTS	
0	0	0	0	0	0	0	0	Initial value
R(RM1),W	R (RM1), W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
SEL1	SEL0	BUFX	BTS	CHSEL3*	CHSEL2	CHSEL1	CHSEL0	
0	0	1	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W*	R/W	R/W	R/W	Attribute

*: In ADTCS0 to ADTCS14, CHSEL3 is undefined. The attribute is R0,WX.

[bit15] BUSY: A/D activation request-in-progress bit

BUSY	Function	
	Read	Write
0	A/D activation not requested	A/D activation request forced stop
1	A/D activation request or conversion in progress	No change and no influence on others

- This bit indicates an A/D activation request or conversion operation.
- In read mode, if this bit is "0", it indicates that no A/D conversion request has been issued; if it is "1", it indicates that A/D conversion request or A/D conversion is in progress.
- In write mode, the current A/D activation request or conversion operation is forced to termination by writing "0" to this bit. Writing "1" to this bit does not make any change and influence on others.

(Note) If this bit is read by a read-modify-write (RMW) instruction, "1" is read.

[bit14] INT: Interrupt request flag bit

INT	Function	
	Read	Write
0	A/D conversion not completed	Clears the bit.
1	A/D conversion completed	No change and no influence on others

- When data is set in the A/D data register (ADTCD) as a result of A/D conversion, this bit is set to "1".
- An interrupt request is generated when this bit and the interrupt request enable bit (ADTCS: INTE) 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 is cleared when the A/D conversion completion interrupt clear signal is "H".

(Notes) If this bit is read by a read-modify-write (RMW) instruction, "1" is 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.

[bit13] INTE: Interrupt request enable bit

INTE	Function
0	Interrupt request output disabled
1	Interrupt request output enabled

- 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 (ADTCS: INT) are "1".

[bit12, bit11] STS1, STS0: A/D activation factor selection bits

STS1	STS0	Function
0	0	Software Activation
0	1	External pin trigger activation (falling edge)
1	0	Reload timer activation (rising edge)
1	1	Compare Match Activation

- These bits select an A/D conversion activation factor.

(Notes) Since the activation factor changes immediately when the bits are rewritten, change these bits while the current target and target activation factor are inactive and the A/D conversion is not being requested (ADTCS:BUSY=1).
If you do not want to issue an A/D activation request, set these bits to specify software activation ("00_B") and set the pertinent bit of ADTSE to disable software activation (ADT bit = 0).

[bit10] RPT: Repeat conversion selection 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 selection bit (PRTS) occurs.

[bit8] PRTS: A/D data register protection clear selection 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) is read and the interrupt request flag bit (INT) is cleared.
- If this bit is set to "1", the A/D data register (ADTCD) is read.

[bit7, bit6] SEL1, SEL0: Counting direction selection 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" when the 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	Upon 0 detection
1	Upon compare clear

- 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.

[bit3 to bit0] CHSEL3 to CHSEL0: Analog channel selection bits

CHSEL3	CHSEL2	CHSEL1	CHSEL0	Function		
				ADTCS0 to 7	ADTCS8 to 14	ADTCS16 to 23
0	0	0	0	Channel 0	Channel 8	Channel 16
0	0	0	1	Channel 1	Channel 9	Channel 17
:				:	:	:
0	1	1	0	Channel 6	Channel 14	Channel 22
0	1	1	1	Channel 7	Setting prohibited	Channel 23
1	0	0	0	Setting prohibited		
:						
1	1	1	1			

- These bits select the analog channel with the specified value.
- Channels that are selected with ADTCS0 to 7 are Channel 0 to Channel 7.
- Channels that are selected with ADTCS8 to 14 are Channel 8 to Channel 14.
- Channels that are selected with ADTCS16 to 23 are Channel 16 to Channel 23.
- In ADTCS0 to 14, bit3 (CHSEL3) is reserved.

<Notes>

- Before setting the STS1, STS0, and/or BTS bits, be sure to confirm that the free-run timer is not running.
- Do not change the setting of the CHSEL3 to CHSEL0 bits when A/D conversion is being requested.
- 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.

45.4.2.5 A/D Data Register (ADTCD0 to ADTCD23)

The A/D data register (ADTCD) contains A/D conversion results.

- **ADTCD0 to ADTCD23: Address 1264_H to 1292_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
ERR	ERRST	Reserved		D11	D10	D9	D8	
1	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[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 results 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 (ATDTC: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 obsolete.
1	The conversion data has overwritten the new data.

- This flag indicates the content of the error in the A/D conversion data if the ERR bit is "1".
- If the ERR bit is "1" and this bit is "0", the conversion results read by the CPU are obsolete.
- 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 the 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 activation factor is not compare-match activation (ATDTC:STS1, STS0 = 11), this bit is read as "0".

[bit13, bit12] Reserved

Always write 0 to these bits.

[bit11 to 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 this bit during the A/D conversion.

45.4.3 12-bit A/D Converter Control Registers

The 12-bit A/D converter is controlled by the A/D control status register, the A/D channel status register, and the A/D mode setting register.

45.4.3.1 A/D Control Status Register (ADCS)

The A/D control status register (ADCS) can verify conversion,

- **ADCS0: Address $12A4_H$ (Access: Byte, Half-word, Word)**
- **ADCS1: Address $12A8_H$ (Access: Byte, Half-word, Word)**
- **ADCS2: Address $12AC_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
BUSY	Reserved							
0	0	0	0	0	0	0	0	Initial value
R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W0	R/W0	R0,W0	Attribute

[bit15] BUSY: A/D conversion-in-progress bit

BUSY	Function
0	A/D conversion stopped
1	A/D conversion in progress

- This bit indicates the A/D converter operation.
- In read mode, if this bit is "0", it indicates that the A/D conversion is stopped; if it is "1", it indicates that the A/D conversion is in progress.
- An attempt to write to this bit does not change the bit and has no influence on others.

[bit14 to bit0] Reserved

Always write 0 to these bits.

45.4.3.2 A/D Channel Status Register (ADCH)

The A/D channel status register (ADCH) allows the number of the analog channel that is in process of the A/D conversion to be read during the conversion.

- **ADCH0:Address $12A6_H$ (Access: Byte, Half-word, Word)**
- **ADCH1: Address $12AA_H$ (Access: Byte, Half-word, Word)**
- **ADCH2: Address $12AE_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved					CH2	CH1	CH0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	Attribute

[bit7 to bit3] Reserved

Always write 0 to these bits.

[bit2 to bit0] CH2 to CH0: Analog channel bits

CH2	CH1	CH0	Function		
			ADCH0	ADCH1	ADCH2
0	0	0	Channel 0	Channel 8	Channel 16
0	0	1	Channel 1	Channel 9	Channel 17
:			:	:	:
1	1	0	Channel 6	Channel 14	Channel 22
1	1	1	Channel 7	Setting prohibited	Channel 23

- These bits can be used to verify the number of the analog channel that is in process of the A/D conversion during the conversion.

45.4.3.3 A/D Mode Setting Register (ADMD)

The A/D mode setting register (ADMD) can set an A/D conversion compare time and a sampling time.

- **ADMD0: Address $12A7_H$ (Access: Byte, Half-word, Word)**
- **ADMD1: Address $12AB_H$ (Access: Byte, Half-word, Word)**
- **ADMD2: Address $12AF_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved				CT1	CT0	ST1	ST0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	Attribute

[bit7 to bit4] Reserved

Always write 0 to these bits.

[bit3, bit2] CT1, CT0: Compare time setting bits

CT1	CT0	Function
0	0	28 peripheral clock cycles (A/D clock output: Peripheral clock/2)
0	1	42 peripheral clock cycles (A/D clock output: Peripheral clock/3)
1	0	56 peripheral clock cycles (A/D clock output: Peripheral clock/4)
1	1	112 peripheral clock cycles (A/D clock output: Peripheral clock/8)

- These bits are used to select the A/D conversion compare time.
- Once an analog input is retrieved (sampling time elapses), the conversion result data will be confirmed after the time specified by these bits.

Recommended setting

Peripheral clock (MHz)	CT1	CT0	Compare period (ns)
40	0	0	700
32	0	0	875
24	0	0	1166.7
16	0	0	1750

(Note) Before rewriting the bits, make sure that the A/D operation prior to the conversion operation is stopped.

[bit1, bit0] ST1, ST0: Sampling time setting bits

ST1	ST0	Function
0	0	12 peripheral clock cycles (A/D clock output: Peripheral clock/2)
0	1	18 peripheral clock cycles (A/D clock output: Peripheral clock/3)
1	0	24 peripheral clock cycles (A/D clock output: Peripheral clock/4)
1	1	48 peripheral clock cycles (A/D clock output: Peripheral clock/8)

- These bits are used to select the A/D conversion sampling time.
- Once the A/D conversion is activated, an analog input will be retrieved until the time specified by these bits elapses.

Recommended setting

Peripheral clock (MHz)	ST1	ST0	Sampling period (ns)
40	0	0	300
32	0	0	375
24	0	0	500
16	0	0	750

(Note) Before rewriting the bits, make sure that the A/D operation prior to the conversion operation is stopped.

45.5 Operation

45.5.1 Interrupt for A/D Activation Compare

The following table shows the interrupt control bits and interrupt factors of the A/D activation compare.

Table 45.5-1 Interrupt Control Bits and Interrupt Factors of A/D Activation Compare

	A/D activation compare
Interrupt request flag bit	A/D activation trigger control status register (ADTCS) INT: bit14
Interrupt request enable bit	A/D activation trigger control status register (ADTCS) INTE: bit13
Interrupt factor	Write A/D conversion result to A/D data register

An interrupt request can be generated when the A/D conversion from the compare channel that has activated the A/D converter is completed.

When the A/D conversion results are set in the A/D data register (ADTCD), the INT bit of the A/D activation trigger control status register (ADTCS) is set to "1". If interrupt requests are enabled (ADTCS: INTE = 1) at this time, an interrupt request is output to the interrupt controller.

45.5.2 A/D Activation Compare Operation

An A/D activation request can be issued by any of the following: software, external trigger, reload timer, or compare match (16-bit free-run timer value becomes the specified value).

45.5.2.1 A/D Activation

One of three A/D converter units is requested to become active. Each of 23 analog channels can issue activation requests. An activation request signal to the A/D activation arbitration is generated by one of the following: software, external trigger (falling edge), reload timer (rising edge), or compare match (the free-run timer value matches the compare register value). Each activation channel can issue three types of A/D activation request signals: software activation, external trigger/reload timer activation, and compare-match activation requests. One of these requests can be active at a given time. When the A/D conversion requested by a channel is completed, the activation request is cleared and the conversion data is stored in the A/D data register. At this time, an interrupt can be generated.

Within the activation channel, when an activation factor occurs while the activation request is being serviced, the activation factor does not lead to the re-issuance of an activation request.

The 23 activation channels are assigned to three A/D converter units. Activation channels 0 to 7 are assigned to the A/D converter unit 0; activation channels 8 to 14 are assigned to the A/D converter unit 1; and activation channels 16 to 23 are assigned to the A/D converter unit 2.

45.5.2.2 A/D Activation Enabling

An A/D activation factor is selected with STS1 and STS0 of the A/D activation trigger control status register (ADTCS). One of the following is selected: software, external trigger, reload timer, and compare match. When the selected activation factor occurs, an activation request signal is issued to the A/D activation arbitration.

Activation channels that do not activate the A/D conversion can be disabled from issuing an A/D activation request by selecting software activation (STS1, STS0 = 00_B) and disabling the channel selected on the A/D software activation channel selection register (ADTSE) from software activation.

45.5.2.3 Free-run Timer Input

Each activation channel can enter data to a free-run timer for compare-match activation independently of other activation channels. If multiple activation channels need to use the same free-run timer, configure the free-run timer selector for this purpose.

45.5.2.4 Analog Channel Selection

The analog channel for the A/D conversion can be selected using the CHSEL bits on the A/D activation trigger control status register (ADTCS).

45.5.2.5 Software Activation

STS1 and STS0 of the A/D activation trigger control status register (ADTCS) can be set to "00_B" for software activation.

The channel that is to issue a software activation is enabled for activation using the A/D software activation channel selection register (ADTSE). For the channel configured in this way, an activation request can be generated at the same time.

The software activation request signal is set by writing "1" to the START bit of the A/D software activation register (ADTSS).

45.5.2.6 External Trigger Activation

STS1 and STS0 of the A/D activation trigger control status register (ADTCS) can be set to "01_B" for external trigger activation.

External triggers are entered to A/D converter units on a one-to-one basis. External trigger 0 is assigned to activation channel 0 to 7; external trigger 1 is assigned to activation channels 8 to 14; and external trigger 2 is assigned to activation channel 16 to 23.

When a falling edge of the external trigger is detected, the external trigger/reload timer activation request signal is set.

45.5.2.7 Reload Timer Activation

STS1 and STS0 of the A/D activation trigger control status register (ADTCS) can be set to "10_B" for reload timer activation.

The reload timer is loaded with data for A/D converter units on a one-to-one basis. Reload timer 0 is assigned to activation channel 0 to 7; reload timer 1 is assigned to activation channels 8 to 14; and reload timer 2 is assigned to activation channel 16 to 23.

When a rising edge of the reload timer is detected, the external trigger/reload timer activation request signal is set.

45.5.2.8 Compare Match Activation

STS1 and STS0 of the A/D activation trigger control status register (ADTCS) can be set to "11_B" for compare match activation.

Each activation channel has a compare register. When the free-run timer value matches the compare register value, a compare-match activation request is issued.

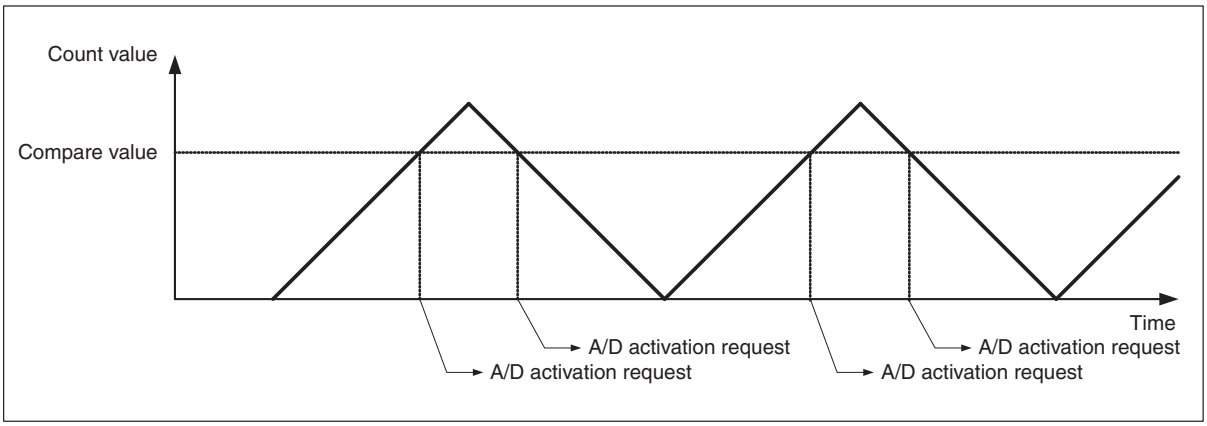
On the compare register, set the timer value to be used for activation. Set "0000_H" as the timer value if you want to issue an A/D activation request when "0" is detected on the free-run timer. If you want to issue an A/D activation request when the free-run timer value matches the compare clear register value, set the timer value equal to the free-run timer compare clear value.

45.5.2.9 A/D Compare Activation Mode

An A/D activation mode can be selected by the SEL1, SEL0 bits of the A/D activation trigger control status register (ADTCS). You can specify whether to compare the A/D activation compare register value with the free-run timer value when the counter counts both up and down, only counts up, or only counts down. If SEL1, SEL0 are set to "11_B", an activation request signal will not be issued to the A/D activation arbitration even when the free-run timer value matches the compare register value.

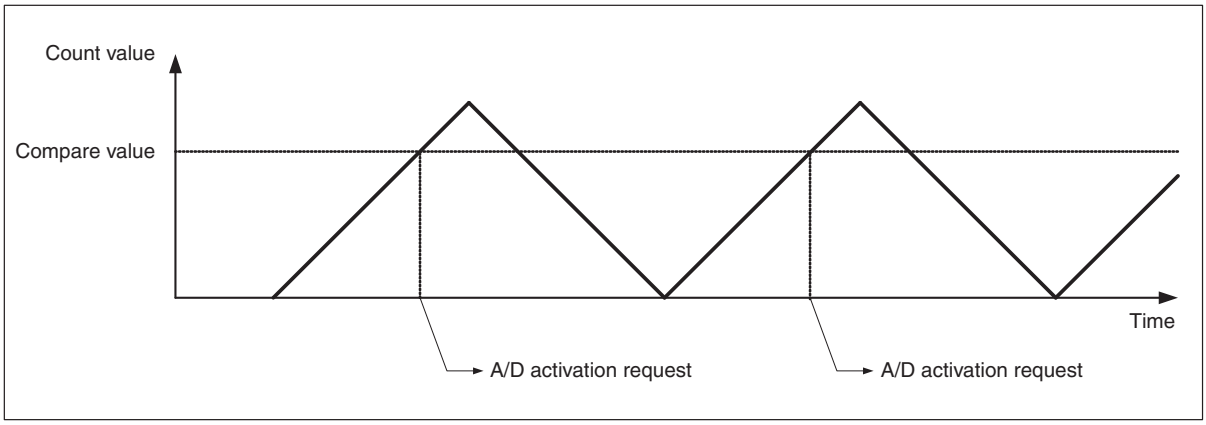
- SEL1, SEL0 = 00_B: Compare-match activation

Figure 45.5-1 SEL1, SEL0 = 00_B: Compare-match activation



- SEL1, SEL0 = 01_B: Compare-match activation only in up-count mode

Figure 45.5-2 SEL1, SEL0 = 01_B: Compare-match activation only in up-count mode



- SEL1, SEL0 = 10_B: Compare-match activation only in down-count mode

Figure 45.5-3 SEL1, SEL0 = 10_B: Compare-match activation only in down-count mode

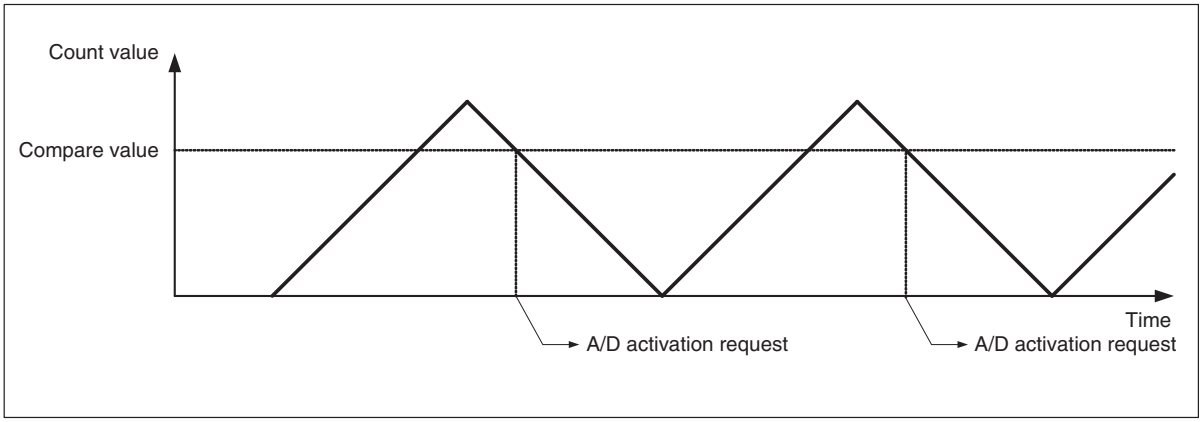
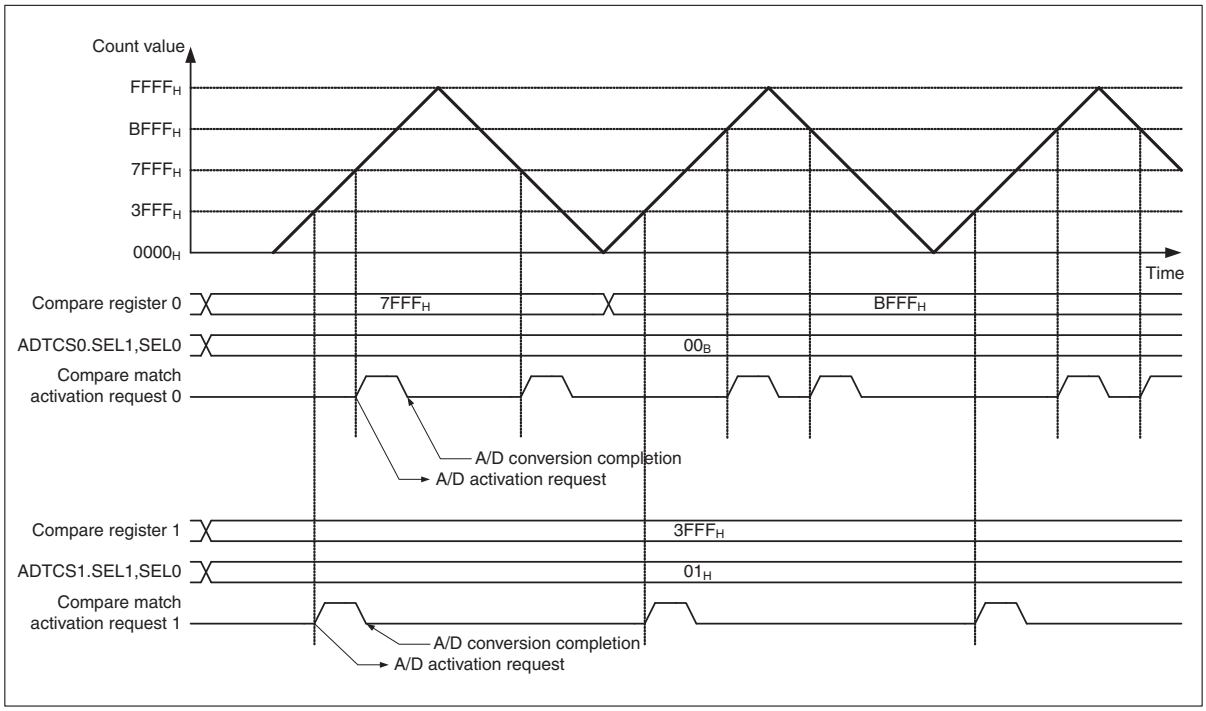


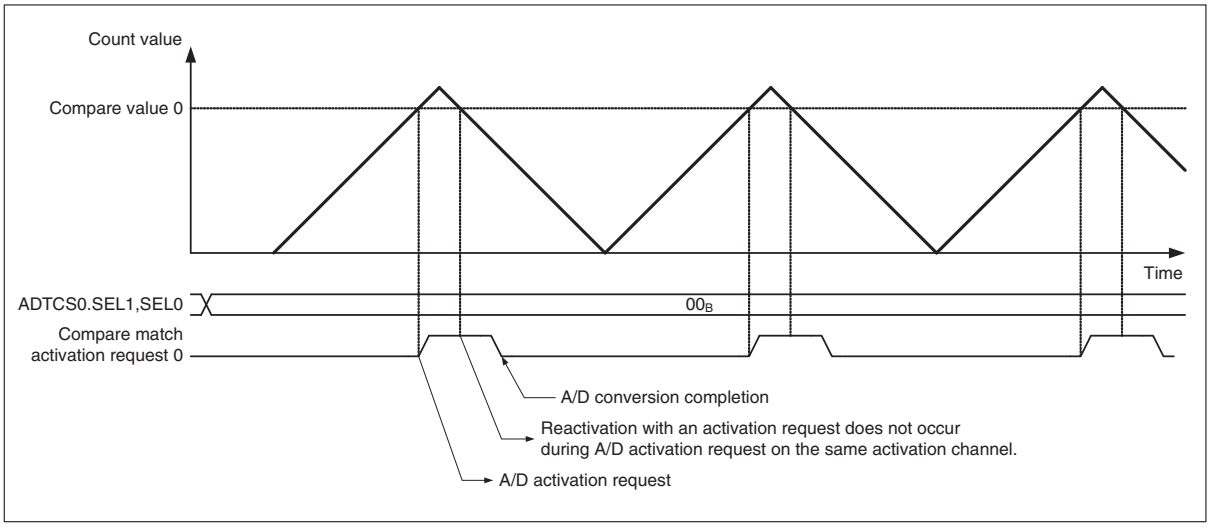
Figure 45.5-4 Activation Channel 0 Initiates A/D Activation in Up/down-count Mode, Whereas Activation Channel 1 Initiates A/D Activation in Up-count Mode.



45.5.2.10 Setting a Compare Value in A/D Compare Activation Mode

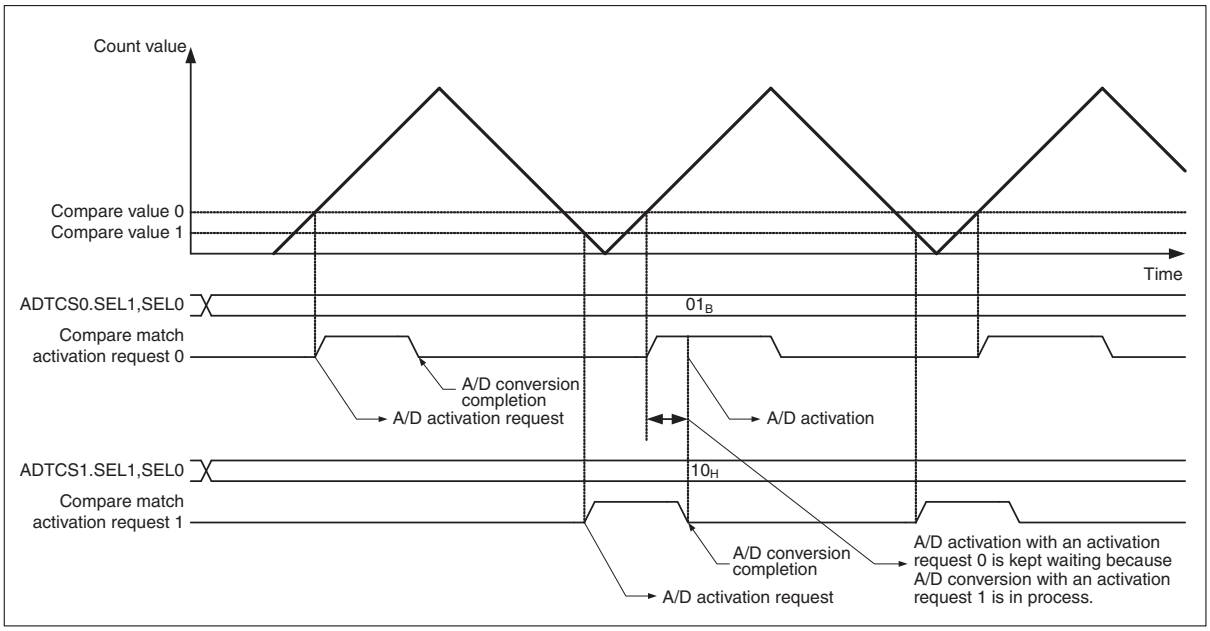
Within the same activation channel that initiates compare-match activation, if the interval between compare matches is shorter than the A/D conversion time, the compare match that has occurred during the A/D conversion is ignored.

Figure 45.5-5 If the Interval between Two Compare Matches Is Shorter than A/D Conversion Time in the Same Activation Channel



If the interval between two compare matches is shorter than the A/D conversion time in two activation channels that activate the same A/D converter, the activation request that has occurred during the A/D conversion is made to wait for the A/D conversion. The A/D conversion starts at a delayed time rather than the intended time.

Figure 45.5-6 If the Interval between Two Compare Matches Is Shorter than A/D Conversion Time in Two Activation Channels



45.5.2.11 Compare Register Buffer Function

When "0" is written to the BUFX bit of the A/D activation trigger control status register (ADTCS), the compare register buffer function is enabled. The buffering timing can be controlled by the BTS bit of the ADTCS register. The value written in the compare buffer register will be transferred to the compare register at the time of a compare clear if BTS = 1 or at the time of a 0-detection if BTS = 0.

Figure 45.5-7 Compare Register 0: Buffer Function Enabled, Compare Register 1: Buffer Function Disabled

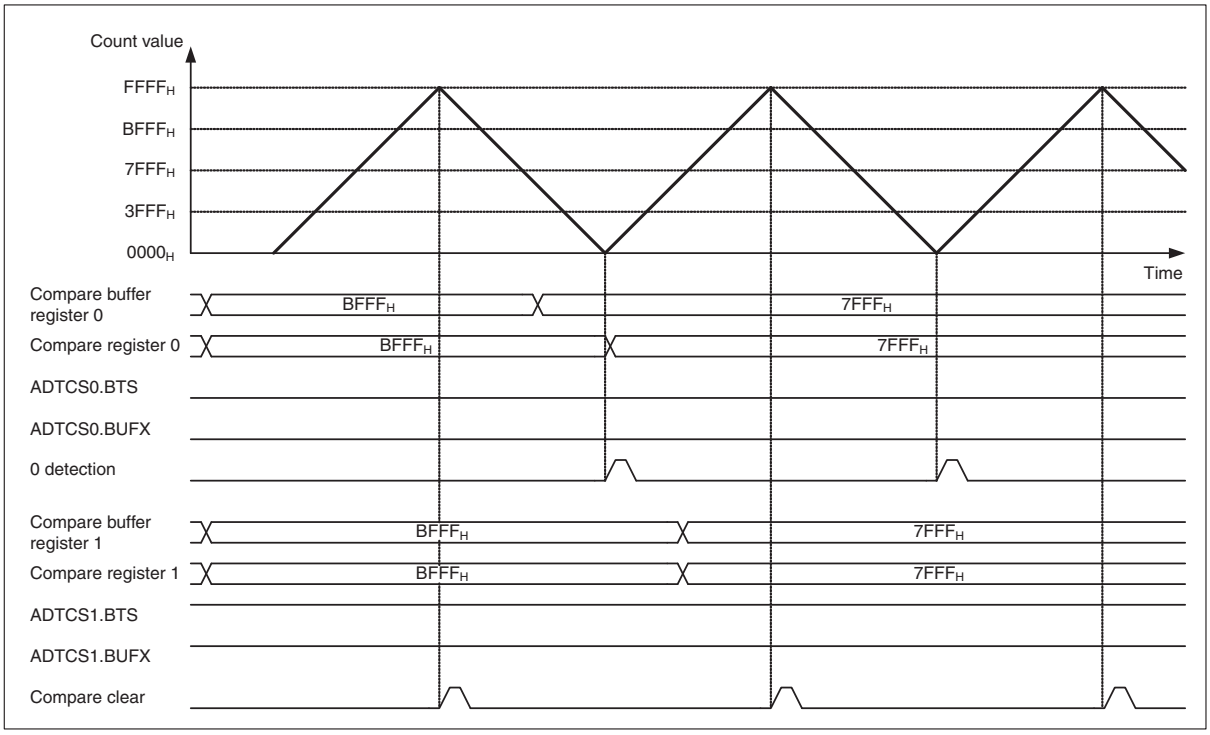


Figure 45.5-8 Timing of Compare Register Data Transfer upon Compare Match in Free-run Timer Count-up Mode

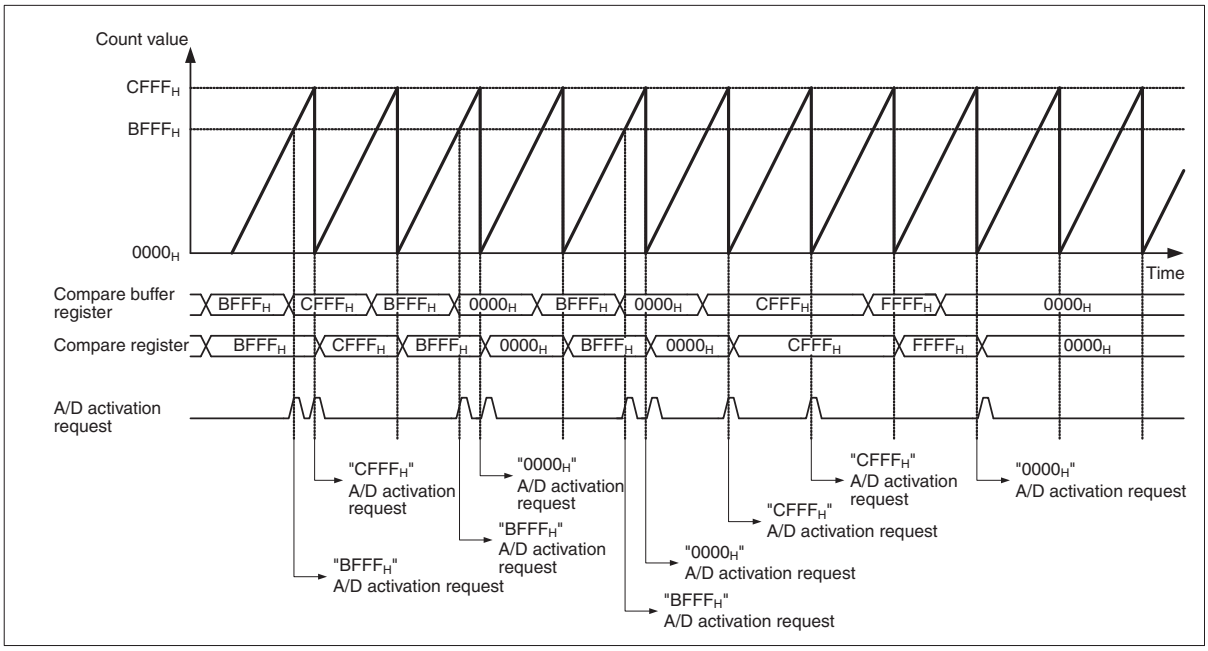


Figure 45.5-9 Timing of Compare Register Data Transfer upon 0 Detection in Free-run Timer Count-up Mode

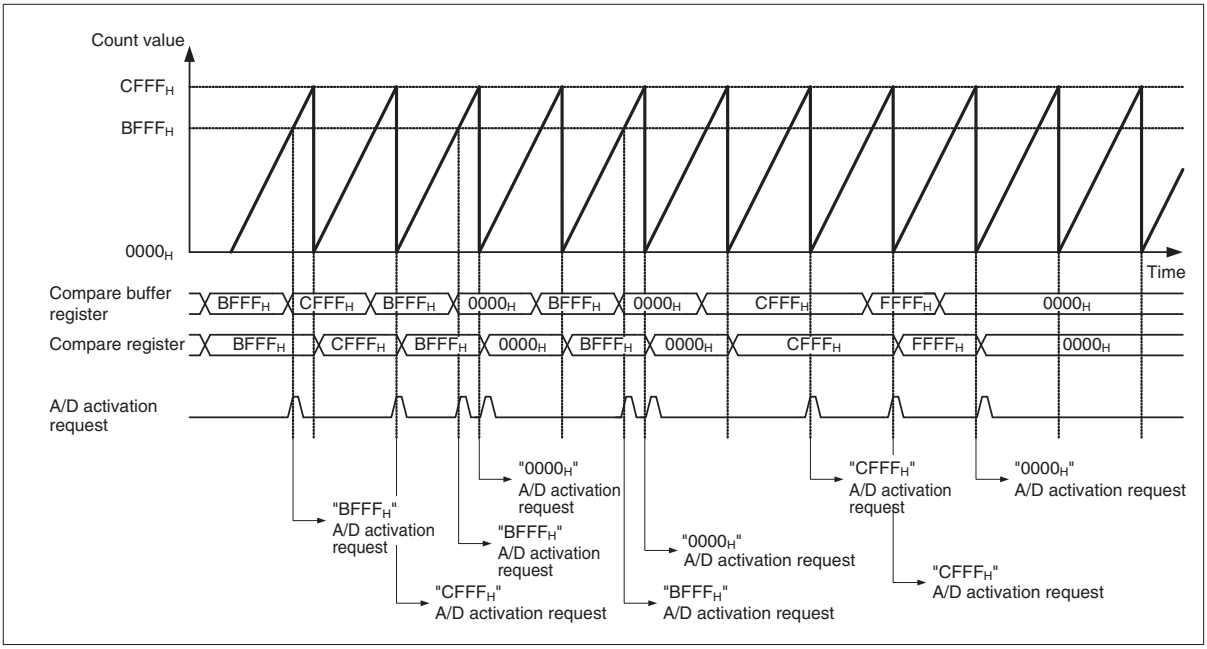


Figure 45.5-10 Timing of Compare Register Data Transfer upon Compare Match in Free-run Timer Count-up/down Mode

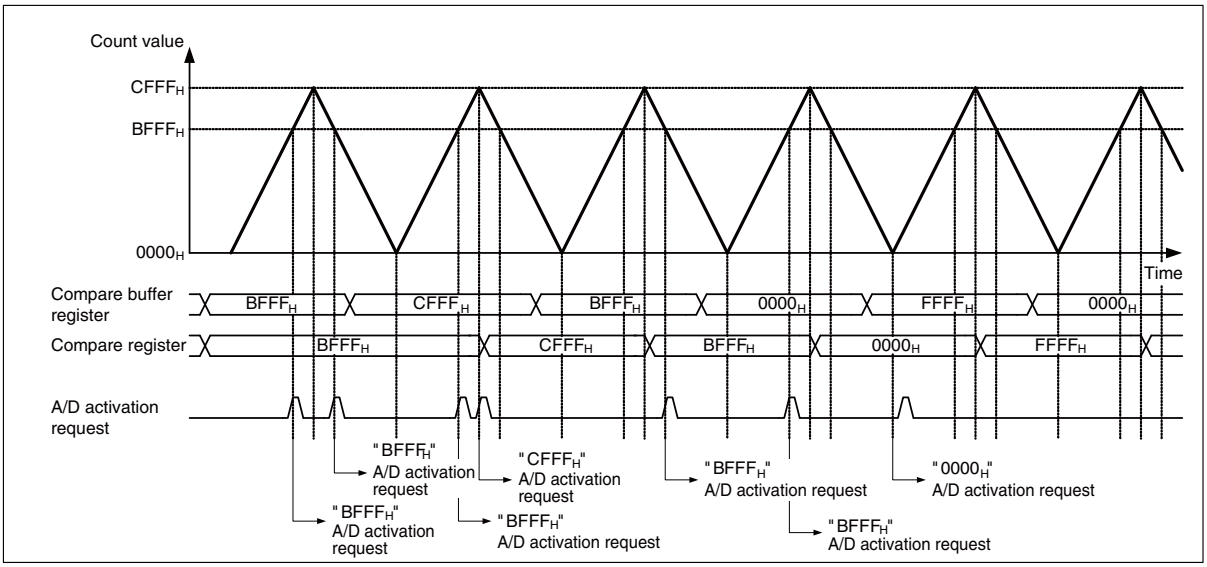
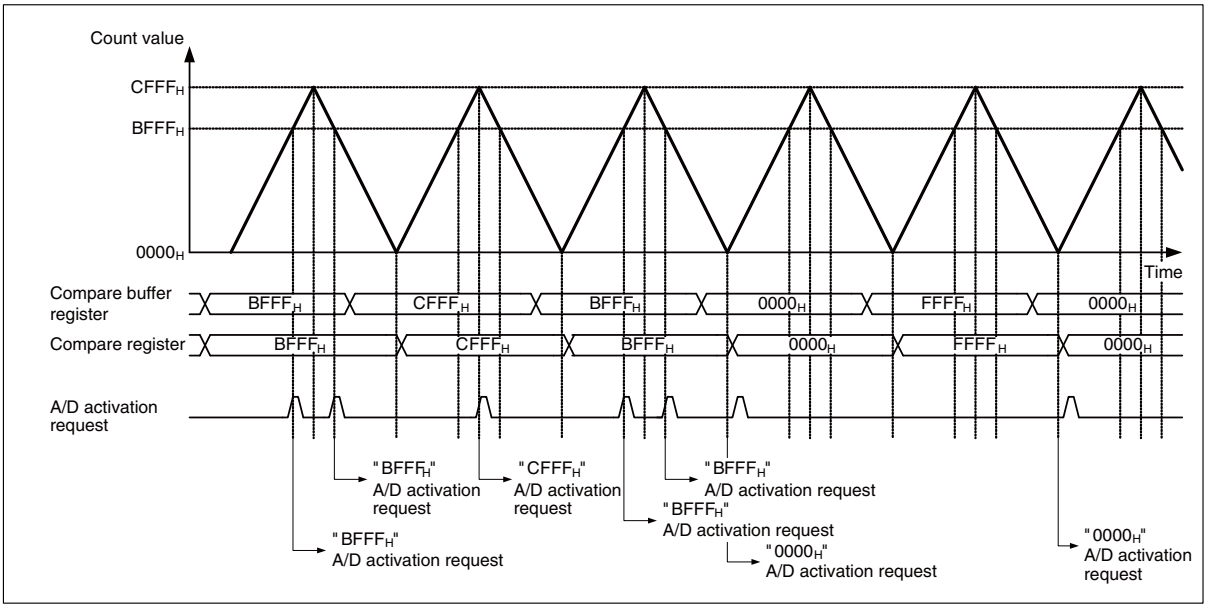


Figure 45.5-11 Timing of Compare Register Data Transfer upon 0 Detection in Free-run Timer Count-up/down Mode

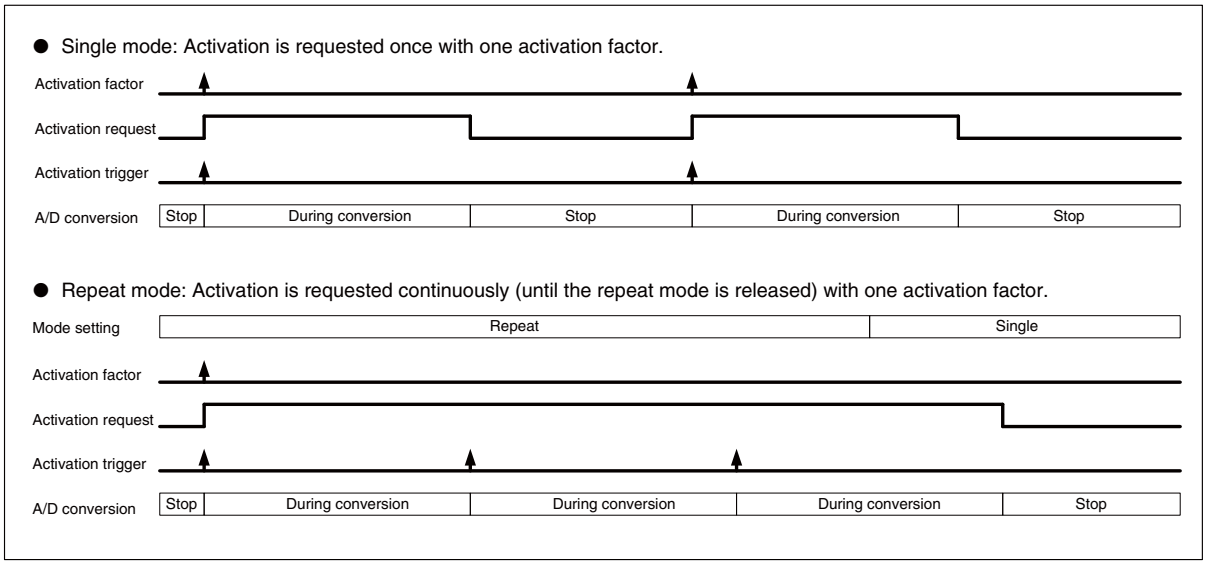


45.5.2.12 Activation Request Mode

An activation request mode can be selected for each activation channel. Two activation modes are available: single mode and repeat mode. The RPT bit of the A/D activation trigger control status register (ADTCS) is used for this setting.

- In single mode, one activation request is issued when one activation factor is encountered. One sequence of the A/D conversion is performed and the activation request is reset when the A/D conversion ends.
- In repeat mode, activation requests are issued in succession as triggered by one activation factor. The A/D conversion is performed repeatedly and the activation request continues until the repeat mode is released.

Figure 45.5-12 Activation Request Mode



45.5.2.13 Protection Function

Each A/D data register can be configured to enable/disable the data protection function. The PRT bit of the A/D activation trigger control status register (ADTCS) is used to set the protection function. The protection function works when the activation factor is not a compare match.

If the protection function is enabled, unread data in the A/D data register that has been stored in the A/D data register can be protected against being overwritten by the subsequent A/D conversion data when the next activation factor occurs, by masking (deactivating) the activation request signal based on the condition specified by the PRTS bit of the ADTCS register.

If the PRTS bit is "0", activation requests will be masked until data in the A/D data register is read and the interrupt flag is cleared. Data reading and interrupt flag clearing may be in any sequence.

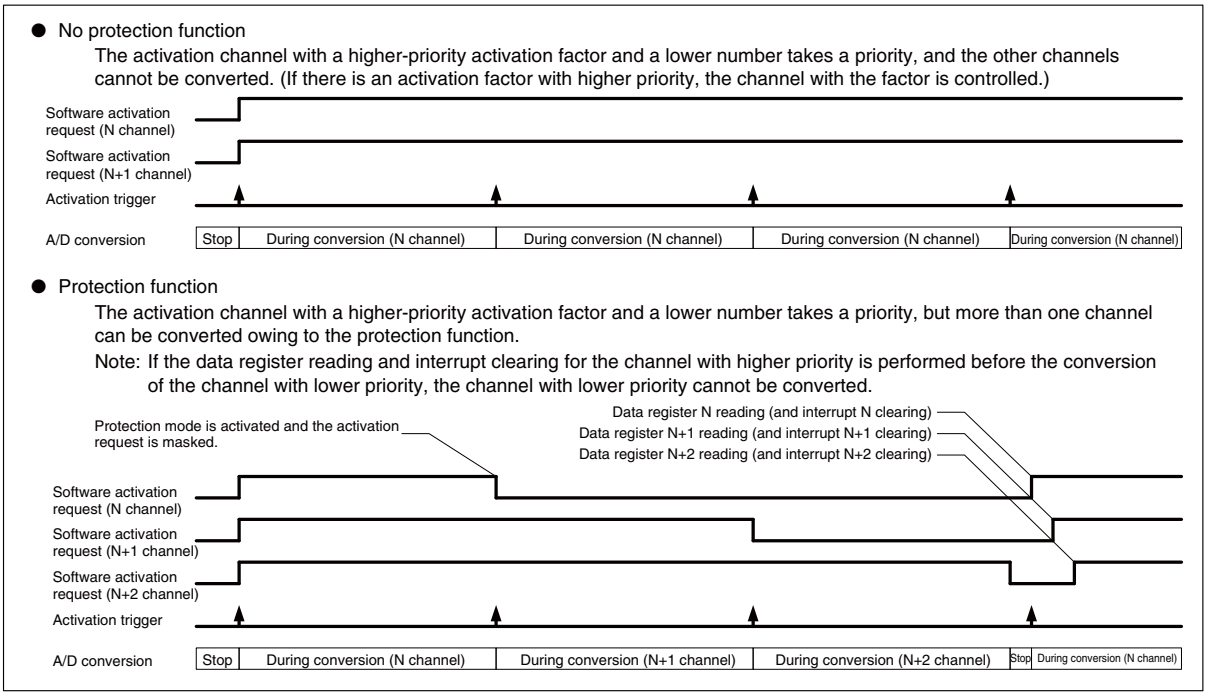
If the PRTS bit is "1", activation requests will be masked until data in the A/D data register is read.

45.5.2.14 Operation Performed When the A/D Activation Request Is in Repeat Mode

If activation channels that activate the same A/D converter are configured to use a repeat mode for multiple activation channels, only one certain channel is serviced on a priority basis during later-stage A/D activation arbitration.

In this case, it is recommended that these activation channels should be configured to enable the protection function.

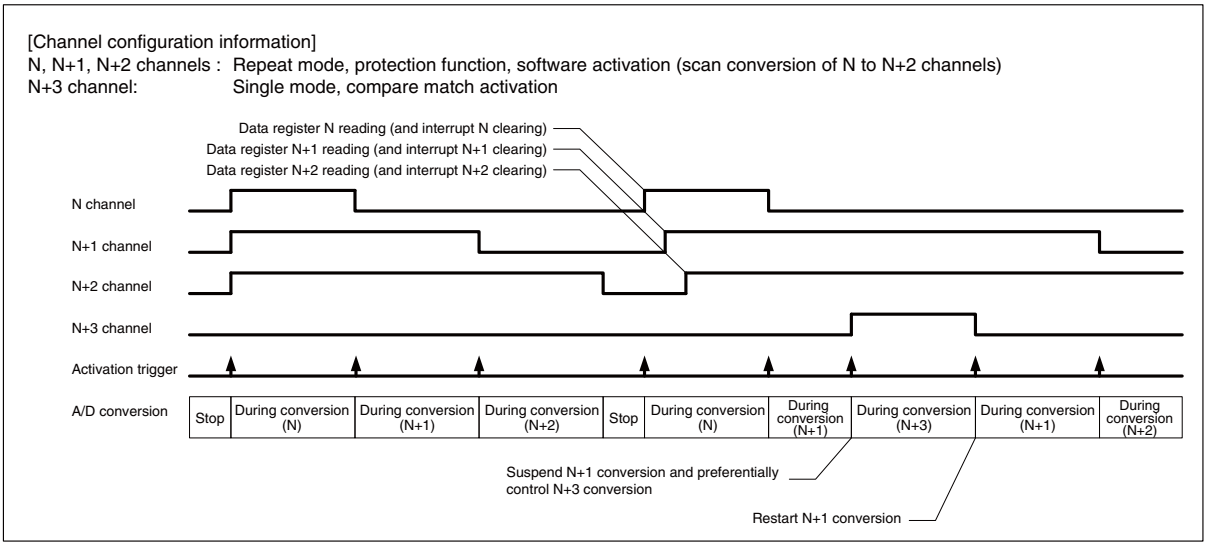
Figure 45.5-13 Operation Performed When the Activation Request Is Issued in Repeat Mode



If the A/D conversion is performed continuously by setting the repeat mode to the multiple activation channels between the activation channels corresponding to the same A/D converter (scan conversion) and an activation request is issued from other channels, priority control for the activation channel to be serviced is in effect based on information about the activation factor and the requesting activation channel. This priority control is performed during later-stage A/D activation arbitration

If an activation factor occurs from an activation channel that is assigned a higher priority than the activation channels that are the target of the scan conversion, the scan conversion is suspended and the A/D conversion is performed for the higher-priority activation channel. When this conversion ends, suspended A/D scan conversion resumes from the activation channel.

Figure 45.5-14 Operation Performed When the Activation Request Is Issued in Repeat Mode (presence of an activation request from another channel)



45.5.2.15 Forced Termination of Activation Request

Whether an A/D activation request or a conversion operation is in progress can be reported using register bits. The current A/D activation request or conversion operation can be terminated forcibly by resetting the appropriate bit to "0".

45.5.3 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.

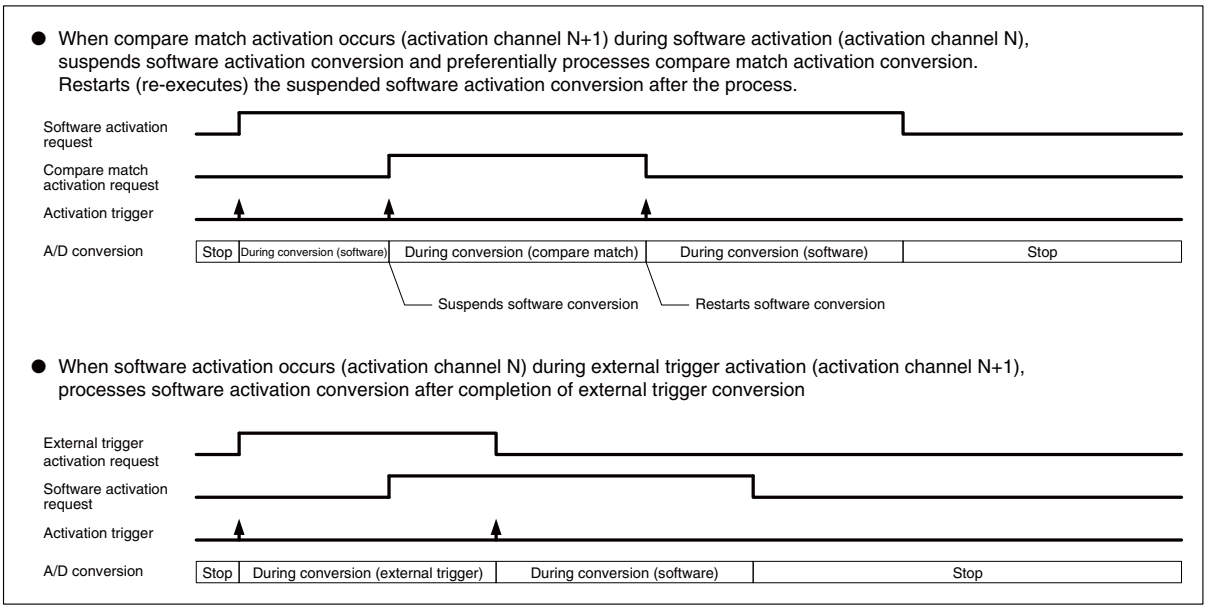
45.5.3.1 A/D Activation Trigger Arbitration

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, and compare-match activation request. A/D activation trigger signals are thus generated. If contention occurs among multiple activation requests, the activation request from the lowest compare channel number takes precedence. Activation requests that are not selected are made to wait. When the current A/D conversion ends, 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 > software activation request. If two activation requests are caused by activation factors assigned the same priority, the request from 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 ends, 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 ends, 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 ends, arbitration is performed again. The activation factor with the same priority is then processed.

Figure 45.5-15 Activation Arbitration



45.5.3.2 Analog Channel Selection

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.

45.5.3.3 A/D Conversion Cancel Function

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.

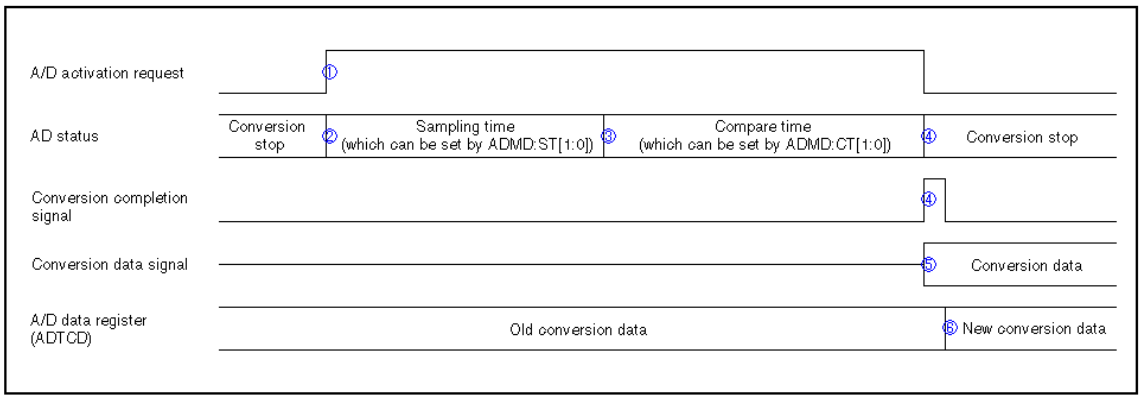
45.5.4 12-bit A/D Converter Operation

12-bit A/D converter controls the A/D conversion.

■ Operation Timing

12-bit A/D converter operation timing

Figure 45.5-16 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 end 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).

45.5.4.1 Activation Factors

AD conversion activation factors include the software activation, external trigger activation, reload timer activation, and compare match activation. It is selected with ADTCS.STS[1:0].

If you do not want to issue an A/D activation request, set these bits to specify software activation ("00B") and set the pertinent bit of ADTSE to disable software activation (ADTSE.ADT = 0).

45.5.4.2 A/D Conversion

One sequence of A/D conversion is performed in response to one activation trigger.

45.5.4.3 Re-activation

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).

45.5.4.4 A/D Conversion Cancel

If an A/D conversion cancel signal is received during the A/D conversion, the current conversion is stopped and the converter is initialized.

45.5.4.5 Analog Channel Selection Control

Information on the analog channel to be A/D-converted is entered in addition to the activation trigger.

The analog channel information that is effective when the activation trigger is active is used when selecting the analog channel.

45.5.4.6 A/D Conversion End and A/D Data Retrieval

When the A/D conversion ends normally without re-activation or cancel (the specified number of cycles have elapsed), the received conversion data is retrieved and output. At this time, an A/D conversion end signal is generated.

45.5.4.7 Power down

Power down is in effect during the standby mode.

(Note) Stop the A/D conversion before the transition to standby mode occurs.

45.6 Caution

■ Notes on Using A/D Activation Compare

- Configuration of selection

  Be sure to configure selection 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, you execute the protection cancellation which is set in the PRTS bit after the A/D conversion stops, or you 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 selection bits (SEL0 and SEL1) of the A/D activation trigger control register (ADTCS) specifies counting-up or counting-down.

- Setting of the counting direction selection bits of the A/D activation trigger control status register

  When the free-run timer is in the up count mode, the counting direction selection bits SEL0 and SEL1 of the A/D activation trigger control status register (ADTCS) must not be set respectively to "1" and "0" (only counting down).

CHAPTER 46 WAVEFORM GENERATOR

46.1 Overview

This model contains 2 waveform generator units (7 channels).

For the 128 MHz operation setting for and notes on the motor control function, see CHAPTER 47 MOTOR CONTROL EXTENSION FUNCTION.

46.2 Features

■ 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, 16-bit waveform control register, PPG output control register, and DTTI selection 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).
- 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).
- You will be able to output waveforms externally from RTO0 to RTO5 (waveform generator unit 0) and RTO6 (waveform generator unit 1).

■ Function of DTTI

- By using the DTTI pin, you can control the waveform generator output to forcibly stop it.
- By using the DTTI register, you can control the waveform generator output to forcibly stop it.
- You can forcibly disable the waveform generator output, and can handle it as a general purpose port, regardless the setting of the extended port function registers (EPFR).

Code: FS20-1v0-91580MS-1-E

46.3 Configuration

■ Configuration of the Waveform Generator

Figure 46.3-1 Configuration of the Waveform Generator (0ch)

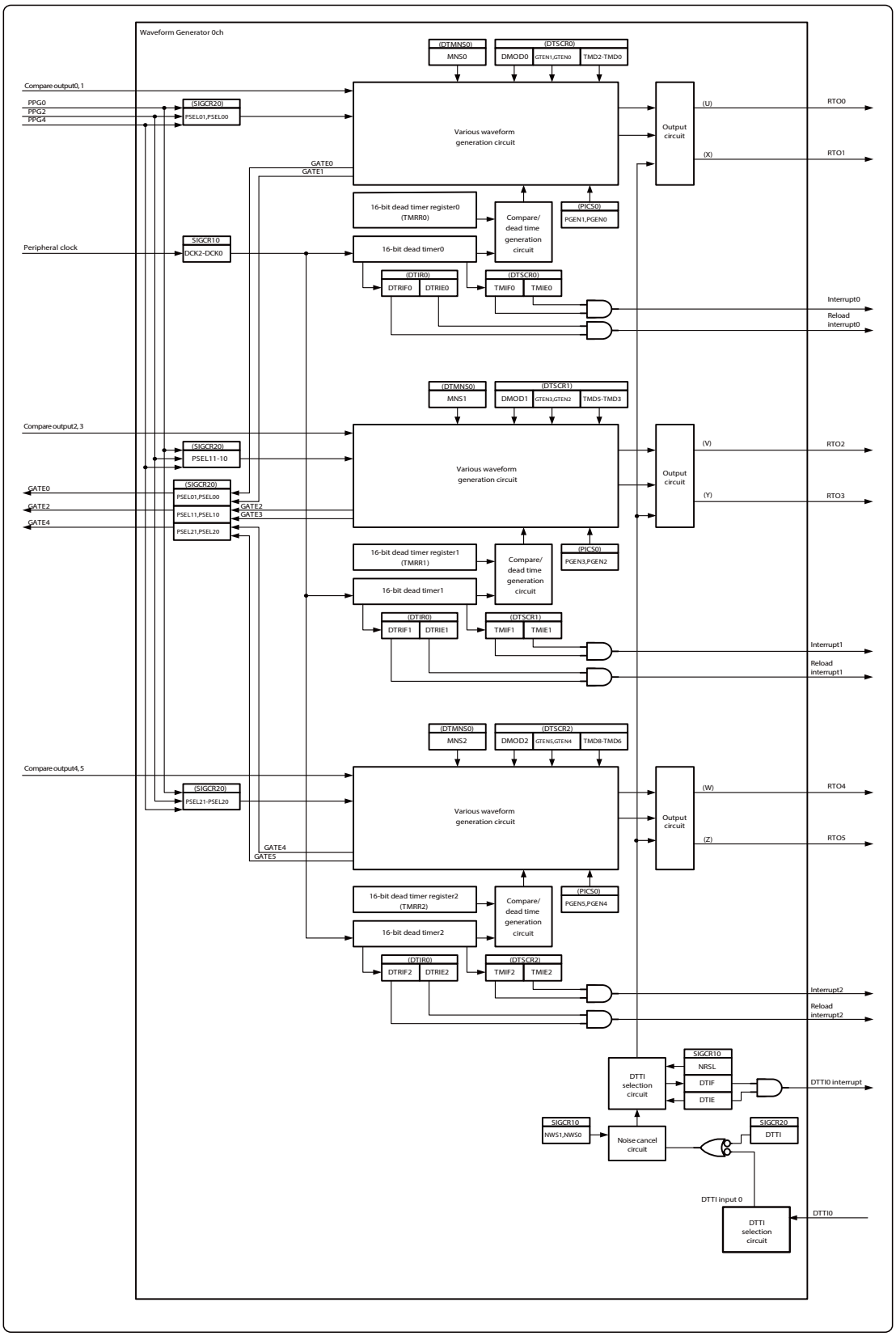
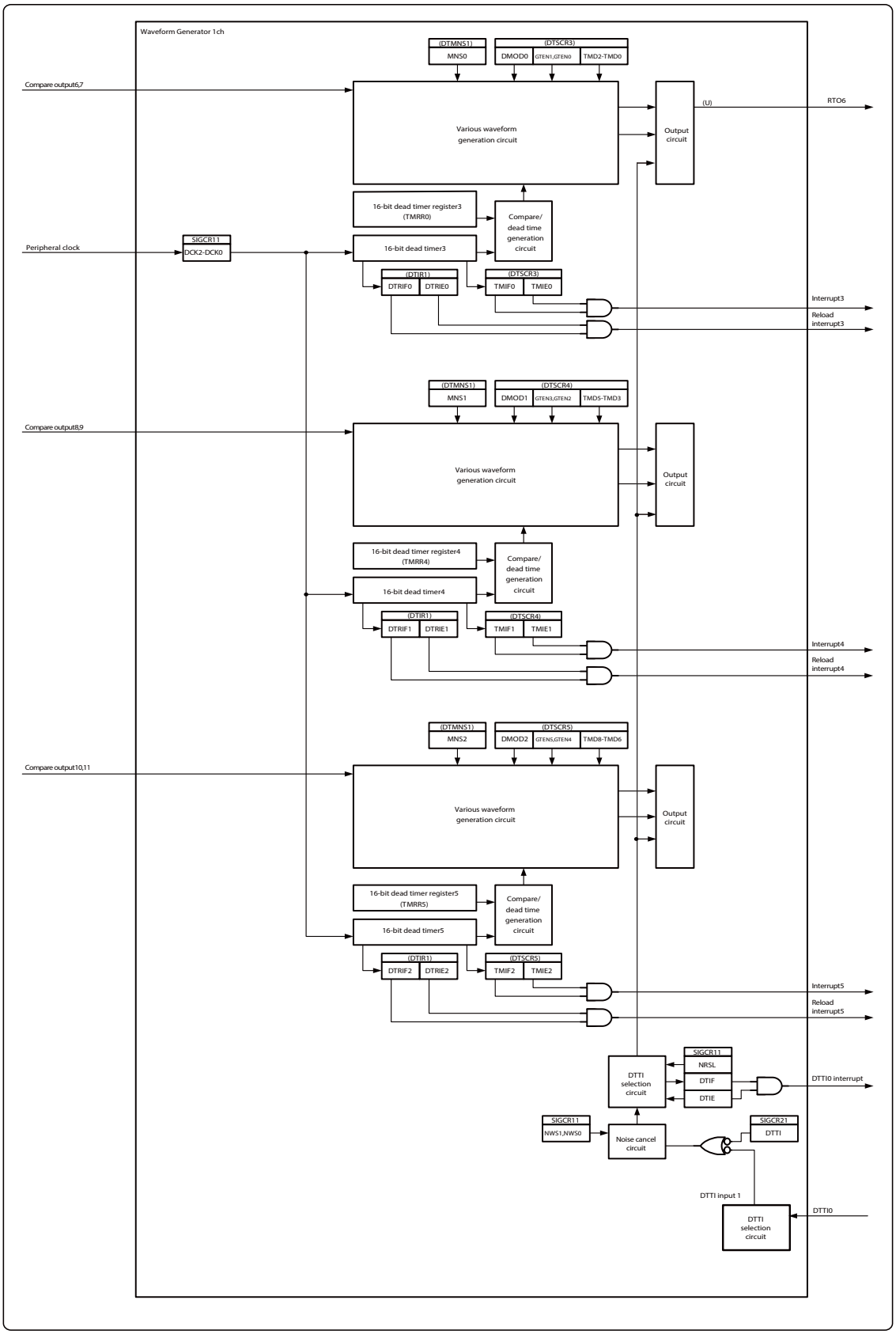


Figure 46.3-2 Configuration of the Waveform Generator (1ch)



■ Pin Application of the Waveform Generator (Unit 0 and Unit 1)

Table 46.3-1 shows the pin application of the waveform generator unit 0 and unit 1.

Table 46.3-1 Pin Application of the Waveform Generator per Ch

Waveform generator unit 0	Waveform generator unit 1
Compare output 0/1/2/3/4/5	Compare output 6/7/8/9/10/11
PPG 0/2/4	Not supported
GATE 0/2/4	Not supported
RTO 0/1/2/3/4/5	RTO 6
Interrupt 0/1/2	Interrupt 3/4/5
Interrupt reload 0/1/2	Interrupt reload 3/4/5
DTTI Interrupt 0	DTTI Interrupt 1
DTT1 0	
16-bit dead timer 0/1/2	16-bit dead timer 3/4/5

46.4 Registers

Table 46.4-1 List of Waveform Generator Register

Address	+0	+1	+2	+3
$11A4_H$	DTTI selection register (DTSR)	Reserved	Reserved	Reserved
$11A8_H$	16-bit dead timer register 0 (TMRR0)		16-bit dead timer register 1 (TMRR1)	
$11AC_H$	16-bit dead timer register 2 (TMRR2)		Reserved	Reserved
$11B0_H$	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
$11B4_H$	Reserved	16-bit dead timer reload interrupt register 0 (DTIR0)	Reserved	16-bit dead timer minus control register 0 (DTMNS0)
$11B8_H$	Reserved	waveform control register 10 (SIGCR10)	Reserved	waveform control register 20 (SIGCR20)
$11BC_H$	PPG output control register 0 (PICS0)			
$11C0_H$	16-bit dead timer register 3 (TMRR3)		16-bit dead timer register 4 (TMRR4)	
$11C4_H$	16-bit dead timer register 5 (TMRR5)		Reserved	Reserved
$11C8_H$	16-bit dead timer state control register 3 (DTSCR3)	16-bit dead timer state control register 4 (DTSCR4)	16-bit dead timer state control register 5 (DTSCR5)	Reserved
$11CC_H$	Reserved	16-bit dead timer reload interrupt register 1 (DTIR1)	Reserved	16-bit dead timer minus control register 1 (DTMNS1)
$11D0_H$	Reserved	waveform control register 11 (SIGCR11)	Reserved	waveform control register 21 (SIGCR21)
$11D4_H$	Reserved			

In the table above, registers assigned in the address range from $11A8_H$ to $11BC_H$ correspond to the unit 0 of the waveform generator and registers assigned in the address range from $11C0_H$ to $11D4_H$ correspond to the unit 1.

46.4.1 Registers for the Waveform Generator

The waveform generator consists of DTTI selection register, 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.

46.4.1.1 DTTI Selection Register (DTSR)

The DTTI selection register (DTSR) is used to configure the DTTI input for each waveform generator.

- **DTSR: Address $11A4_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved						SEL1	SEL0	
0	0	0	0	0	0	1	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	Attribute

[bit7 to bit2] Reserved

Always write 0 to these bits.

[bit1] SEL1: DTTI input selection bit for waveform generator unit 1

SEL1	Function
0	Enable
1	Disable

- This bit is used to configure the DTTI input for waveform generator unit 1.

(Note) **Before configuring this bit, make sure to verify that the waveform generator is inactive.**

[bit0] SEL0: DTTI input selection bit for waveform generator unit 0

SEL0	Function
0	Enable
1	Disable

- This bit is used to configure the DTTI input for waveform generator unit 0.

(Note) **Before configuring this bit, make sure to verify that the waveform generator is inactive.**

46.4.1.2 16-bit Dead Timer Register (TMRR)

The 16-bit dead timer register (TMRR) holds the reload value of the 16-bit dead timer.

- **TMRR0: Address $11A8_H$ (Access: Half-word, Word)**
- **TMRR1: Address $11AA_H$ (Access: Half-word, Word)**
- **TMRR2: Address $11AC_H$ (Access: Half-word, Word)**
- **TMRR3: Address $11C0_H$ (Access: Half-word, Word)**
- **TMRR4: Address $11C2_H$ (Access: Half-word, Word)**
- **TMRR5: Address $11C4_H$ (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

7	6	5	4	3	2	1	0	bit
TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	
0	0	0	0	0	0	0	1	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[bit15 to bit0] TR15 to TR00: 16-bit Dead timer reload value bits

	Function
TR15 to TR00	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

(Notes)
- When accessing these registers, use a half-word or word access instruction.
- Do not set "0000_H" to these registers.

46.4.1.3 16-bit Dead Timer State Control Register (DTSCR)

The 16-bit dead timer state control register (DTSCR) is used to control operation mode, permission of interrupt request, interrupt request flag, permission of GATE signal, and output level polarity of the waveform generator.

- **DTSCR0: Address $11B0_H$ (Access: Byte, Half-word, Word)**
- **DTSCR3: Address $11C8_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DMOD0	GTEN1	GTEN0	TMIF0	TMIE0	TMD2	TMD1	TMD0	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	Attributes

[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 bit 1

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 SIGCR20/21.

[bit5] GTEN0: GATE signal control bit 0

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 SIGCR20/21.

[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".

(Notes) 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	Disabled
Others			Disabled

- 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 0/6 and 1 signals of the output compare will be output from RTO0/RTO6 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 0/6 and 1 signals of the output compare will be output from RTO0/RTO6 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)

- **DTSCR1: Address $11B1_H$ (Access: Byte, Half-word, Word)**
- **DTSCR4: Address $11C9_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DMOD1	GTEN3	GTEN2	TMIF1	TMIE1	TMD5	TMD4	TMD3	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R (RM1), W	R/W	R/W	R/W	R/W	Attributes

[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 bit 3

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 SIGCR20/21.

[bit5] GTEN2: GATE signal control bit 2

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 SIGCR20/21.

[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".

(Notes) 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	Disabled
Others			Disabled

- 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)

- **DTSCR2: Address $11B2_H$ (Access: Byte, Half-word, Word)**
- **DTSCR5: Address $11CA_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DMOD2	GTEN5	GTEN4	TMIF2	TMIE2	TMD8	TMD7	TMD6	
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R (RM1), W	R/W	R/W	R/W	R/W	Attributes

[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 bit 5

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 SIGCR20/21.

[bit5] GTEN4: GATE signal control bit 4

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 SIGCR20/21.

[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".

(Notes) 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)
1	0	0	Non-overlap signal will be generated at OUT signal. (Dead time timer mode)
1	1	1	Disabled
Others			Disabled

- 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)

46.4.1.4 16-bit Dead Timer Reload Interrupt Register (DTIR)

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 permission.

- **DTIR0: Address $11B5_H$ (Access: Byte, Half-word, Word)**
- **DTIR1: Address $11CD_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DTRIF2	DTRIE2	DTRIF1	DTRIE1	DTRIF0	DTRIE0	Reserved		
0	0	0	0	0	0	0	0	Initial values
R (RM1), W	R/W	R (RM1), W	R/W	R (RM1), W	R/W	R0, WX	R0, WX	Attributes

[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".

(Notes) 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".

(Notes) 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".

(Notes) 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.

46.4.1.5 16-bit Dead Timer Minus Control Register (DTMNS)

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 $11B7_H$ (Access: Byte, Half-word, Word)**
- **DTMNS1: Address $11CF_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
KEY1	KEY0	Reserved			MNS2	MNS1	MNS0	
0	0	0	0	0	0	0	0	Initial values
R0,W	R0, W	R0, W0	R0, W0	R0, W0	R,W	R,W	R,W	Attributes

[bit7, bit6] KEY1, KEY0: Key enable bits

	Function
KEY1, KEY0	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.

[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.

[bit0] MNS0: Dead time function selection bit (RTO0/RTO6 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/RTO6 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.

46.4.1.6 Waveform Control Register 1/2 (SIGCR1, SIGCR2)

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 $11B9_H$ (Access: Byte, Half-word, Word)**
- **SIGCR11: Address $11D1_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0	
0	0	0	0	0	0	0	0	Initial values
R/W	R (RM1), W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

[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/RTO6 and RTO1 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".

(Notes) 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.

(Notes) 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	Disabled

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 $11BB_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
PSEL21	PSEL20	PSEL11	PSEL10	PSEL01	PSEL00	-Reserved-	DTTI	
0	0	0	0	0	0	0	1	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R0, WX	R/W	Attributes

[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 disabled (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 disabled (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 disabled (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. However, note that when the input from the DTTI pin is "0", it is not cleared.

- **SIGCR21: Address 11D3$_H$ (Access: Byte, Half-word, Word))**

7	6	5	4	3	2	1	0	bit
Reserved							DTTI	
0	0	0	0	0	0	0	1	Initial values
R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W	Attributes

[bit7 to bit1] Reserved

Always write "0" to these bits.

[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. However, note that when the input from the DTTI pin is "0", it is not cleared.

46.4.1.7 PPG Output Control Register (PICS)

The PPG output control register (PICS) is used to control the PPG output.

- **PICS0: Address 11BC$_H$ (Access: Byte, Half-word, Word)**

31	30	29	28	27	26	25	24	bit
PGEN5	PGEN4	PGEN3	PGEN2	PGEN1	PGEN0	Reserved		
0	0	0	0	0	0	0	0	Initial values
R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0	Attributes

23	22	21	20	19	18	17	16	bit
Reserved								
1	1	1	1	1	1	1	1	Initial values
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attributes

15	14	13	12	11	10	9	8	bit
Reserved								
1	1	1	1	1	1	1	1	Initial values
R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	Attributes

7	6	5	4	3	2	1	0	bit
Reserved								
0	0	0	0	0	0	0	0	Initial values
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W0	R/W0	R/W0	Attributes

[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] PGEN4: 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, bit24] Reserved

Always write "0" to these bits.

[bit23 to bit8] Reserved

Always write "1" to these bits.

[bit7 to bit0] Reserved

Always write "0" to these bits.

46.5 Operation Explanation

46.5.1 Interrupts for Waveform Generator

Table 46.5-1 and Table 46.5-2 show the interrupt control bits and interrupt factors of the waveform generator.

Table 46.5-1 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-1

	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 state control register 0/3 (DTSCR0/3) TMIF0: bit28	16-bit dead timer state control register 1/4 (DTSCR1/4) TMIF1: bit20	16-bit dead timer state control register 2/5 (DTSCR2/5) TMIF2: bit12
Interrupt request enable bit	16-bit dead timer state control register 0/3 (DTSCR0/3) TMIE0: bit27	16-bit dead timer state control register 1/4 (DTSCR1/4) TMIE1: bit19	16-bit dead timer state control register 2/5 (DTSCR2/5) TMIE2: bit11
Interrupt factor	16-bit dead timer 0/3 underflow	16-bit dead timer 1/4 underflow	16-bit dead timer 2/5 underflow

Table 46.5-2 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-2

	DTTI0	DTTI1
Interrupt request flag bit	Waveform control register 10 (SIGCR10) DTIF: bit22	Waveform control register 11 (SIGCR11) DTIF: bit22
Interrupt request enable bit	-	-
Interrupt factor	Detection of "L" level at DTTI0	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 46.5-3 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.

46.5.2 Operation of the Waveform Generator

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 46.5-4 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) *2
			0	1	Output the PPG0/PPG2/PPG4 pulses while the OUT is "H" *1	Always "0"
			1	1	Output the PPG0/PPG2/PPG4 pulses activated by GATE signals while the OUT is "H"	OUT
0	1	0	0/1	0	Activate 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*3
			0	1	Activate 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	Activate 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*3
1	0	0	X	X	Generate a non-overlap signal at OUT	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: GATE signal will be generated from the OUT of which GTEN bit is set to "1".

*3: 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 46.5-5 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

<Notes:>

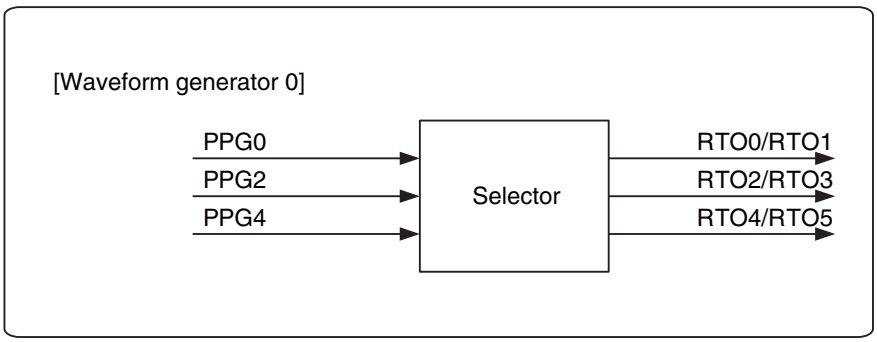
The RTO0 and RTO1 are controlled by the TMD2 to TMD0 of the 16-bit dead timer state control register (DTSCR0). The RTO2 and RTO3 are controlled by the TMD5 to TMD3 of the DTSCR1 register. The RTO4 and RTO5 are controlled by the TMD8 to TMD6 of the DTSCR2 register.

■ 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 (PICS0).

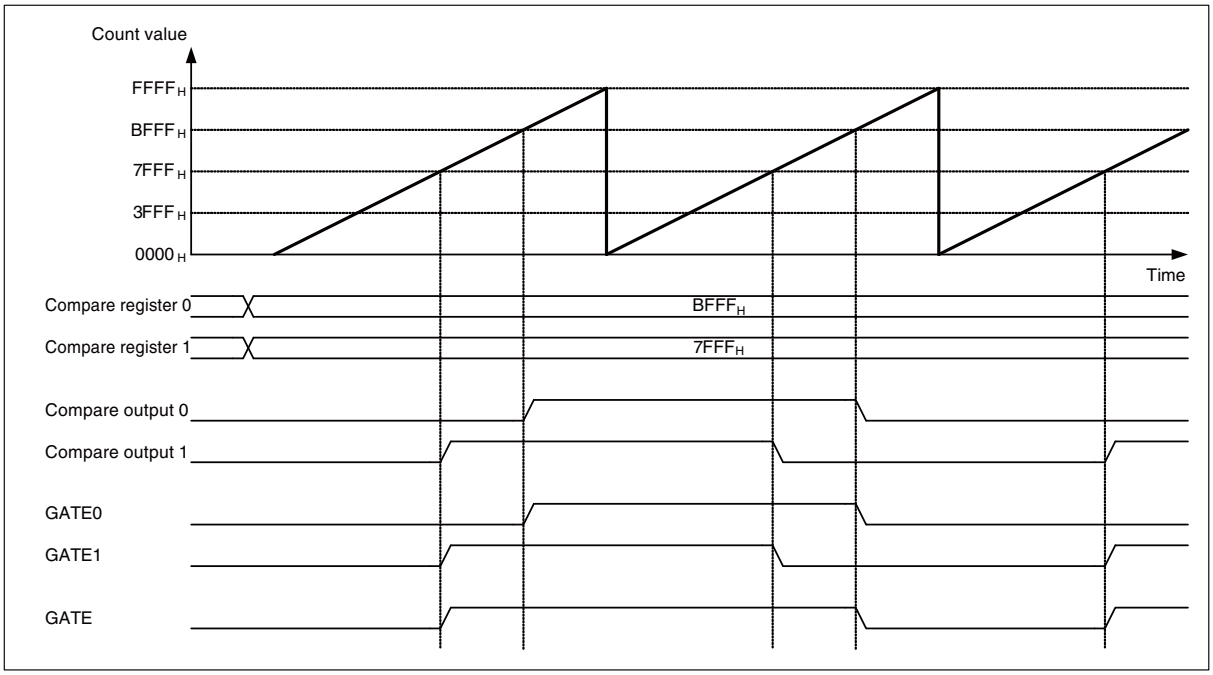
■ Gate-triggered PPG output

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 generates 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.



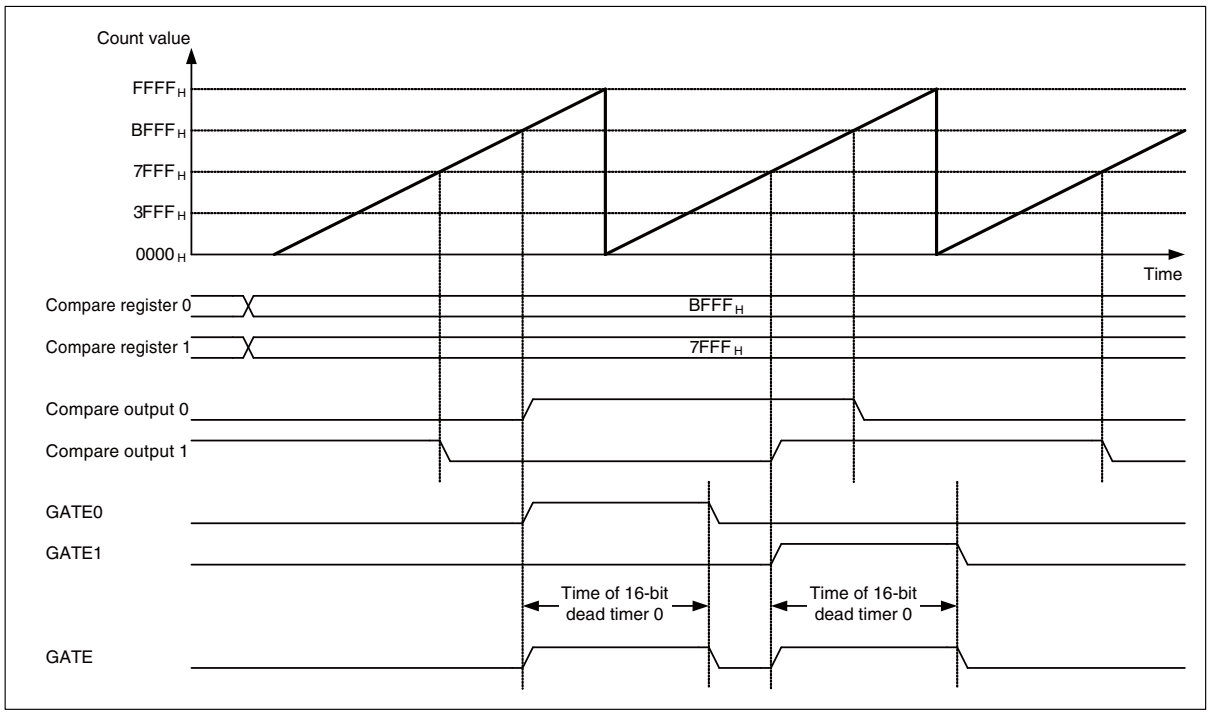
- GATE signal generation when the GATE is active and each OUT is "H" (TMD8 to TMD0 of the 16-bit dead timer state control (DTSCR0, DTSCR1, and DTSCR2) are "001_B")

Figure 46.5-1 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 46.5-2 GATE Signal Generation from the Rising Edge of the OUT through the 16-bit Dead Timer Underflow



<Notes:>

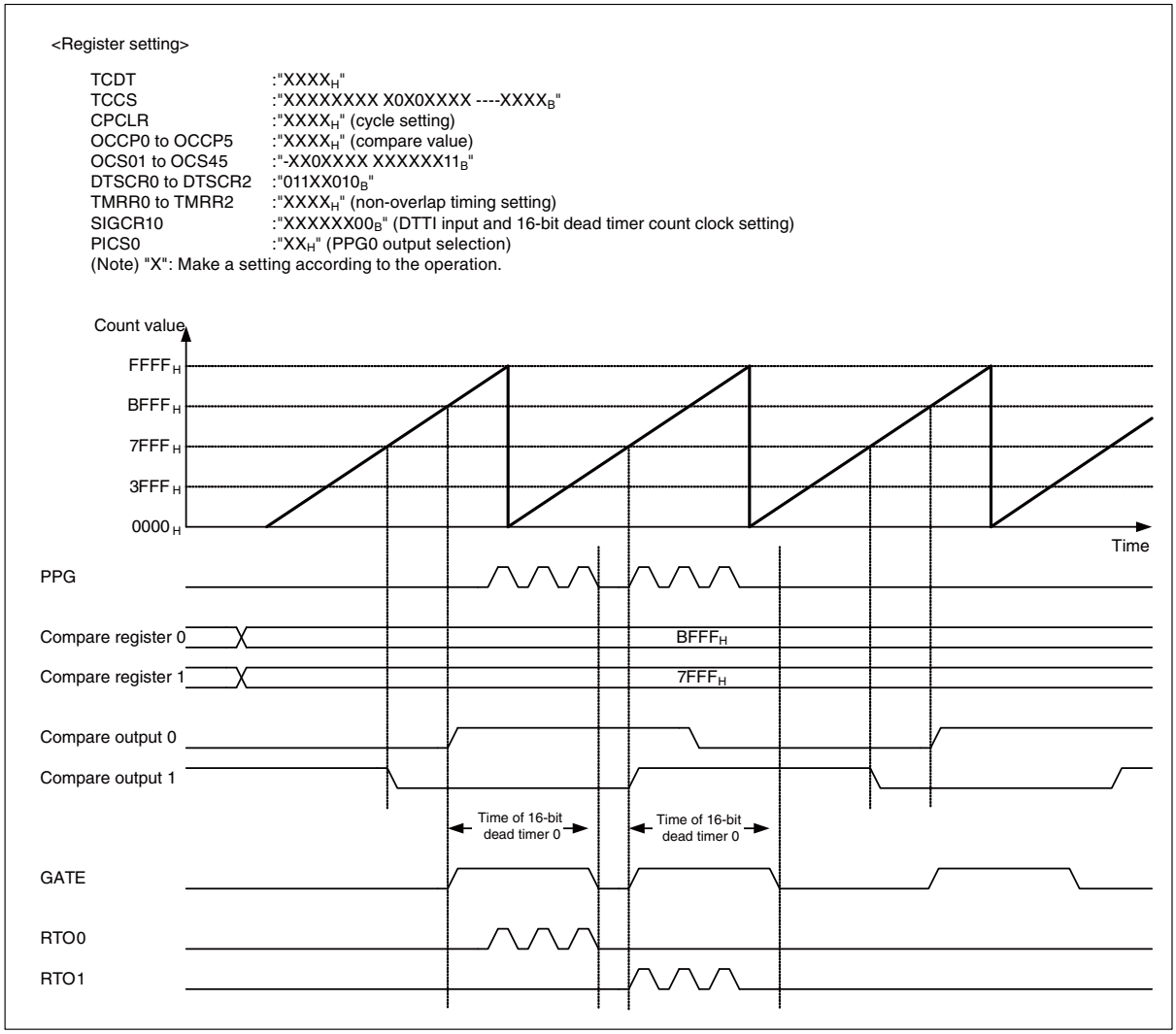
Each 16-bit dead timer will be used for two OUTs. In other words, the 16-bit dead timer 0 is used for OUT0 and OUT1, the 16-bit dead timer 1 is used for OUT2 and OUT3, and the 16-bit dead timer 2 is used for OUT4 and OUT5. Thus, do not activate the timer which is already active, using the OUT. If you 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, and DTSCR2 registers are 010_B)

Figure 46.5-3 Waveform Generated When the TMD2 to TMD0 are "010_B"



<Notes:>

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, and DTSCR2) are 100_B)

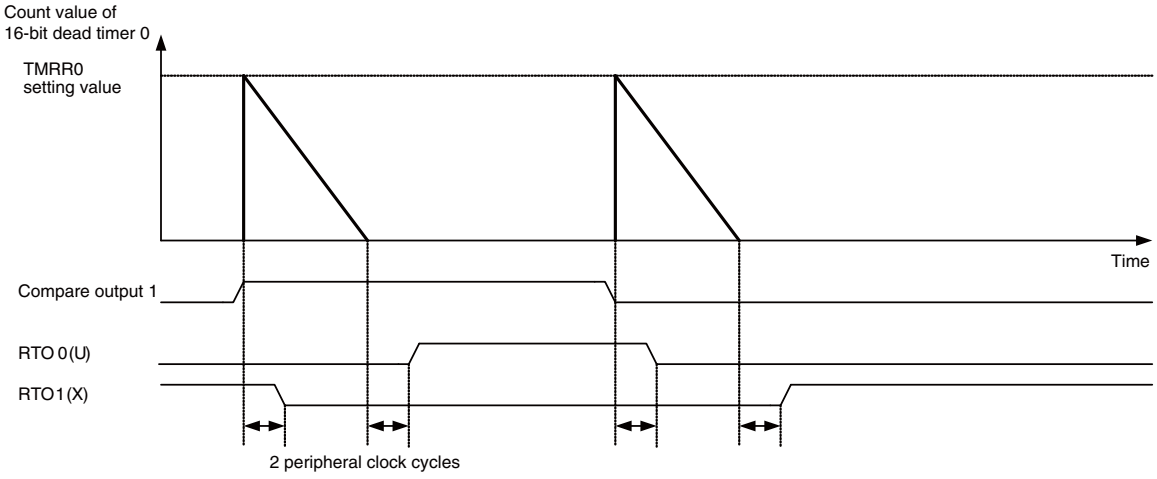
If you select the non-overlap signal of which DMOD2 to DMOD0 of the DTSCR0, DTSCR1, and DTSCR2 registers are "0" (normal polarity), the delay that corresponds to the non-overlap time configured at the 16-bit dead timer registers (TMRR0 to TMRR2) will be applied. This delay will be applied to the rising 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 46.5-4 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 $XXXXXX11_B$"
DTSCR0 to DTSCR2	:"$0XXXX100_B$"
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, and DTSCR2) are 100_B)

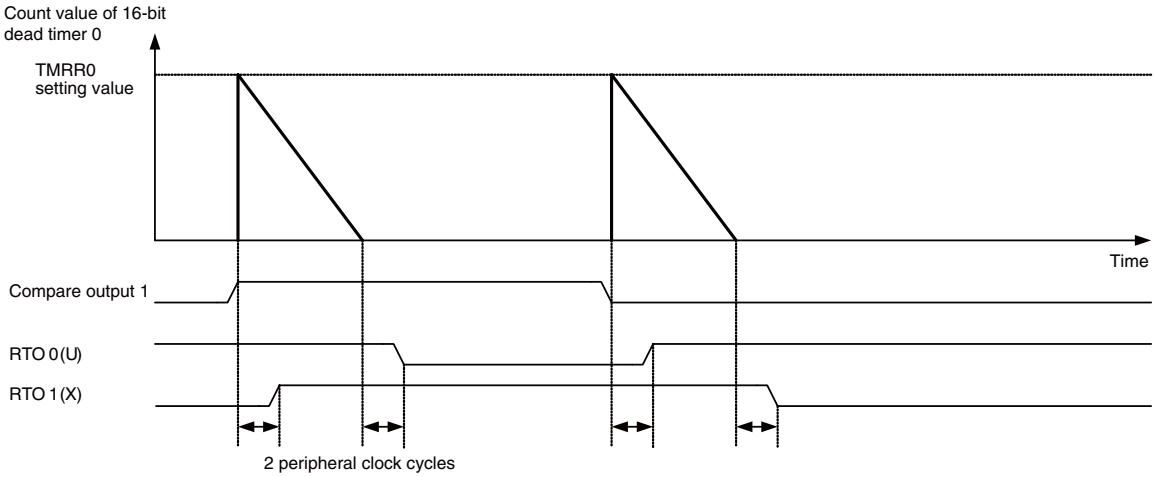
If you select the non-overlap signal of which DMOD2 to DMOD0 of the DTSCR0, DTSCR1, and DTSCR2 registers are "1" (inverted polarity), the delay that corresponds to the non-overlap time configured at the 16-bit dead timer registers (TMRR0 to TMRR2) will be applied. This delay will be applied to the rising 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 46.5-5 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 $XXXXXX11_B$"
DTSCR0 to DTSCR2	:"$1XXXX100_B$"
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
RTO 0 (U)	Delayed inverted signals are applied to the rising edge of compare output 1.
RTO 2 (V)	Delayed inverted signals are applied to the rising edge of compare output 3.
RTO 4 (W)	Delayed inverted signals are applied to the rising edge of compare output 5.
RTO 1 (X)	Delayed signals are applied to the falling edge of compare output 1.
RTO 3 (Y)	Delayed signals are applied to the falling edge of compare output 3.
RTO 5 (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, and DTSCR2) are 100_B)

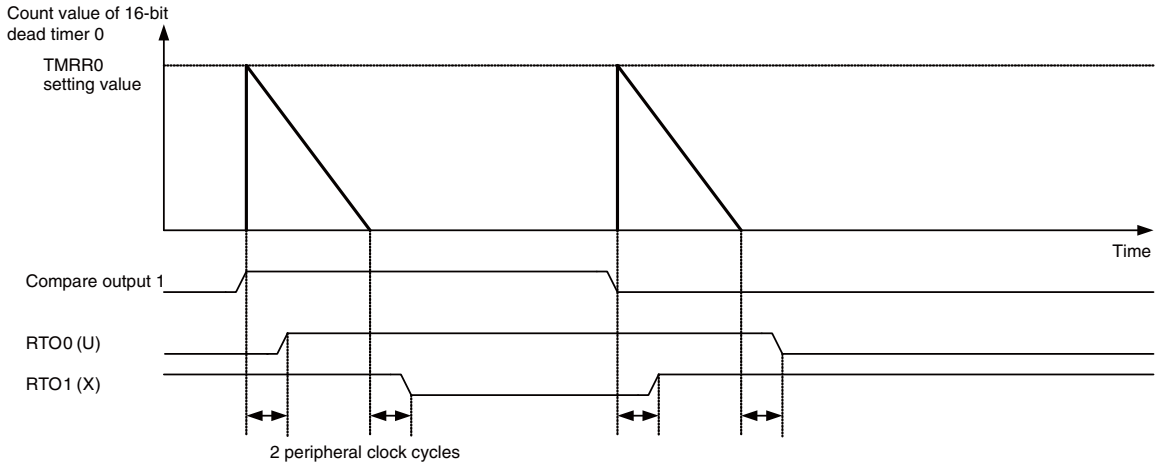
Signal generation is executed by outputting U/V/W and X/Y/Z of the inverted polarity non-overlap signal that does not operate by minus control as X/Y/Z and U/V/W.

Figure 46.5-6 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"
DTSCR0 to DTSCR2	: "1XXXX100_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 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, and DTSCR2) are 100_B)

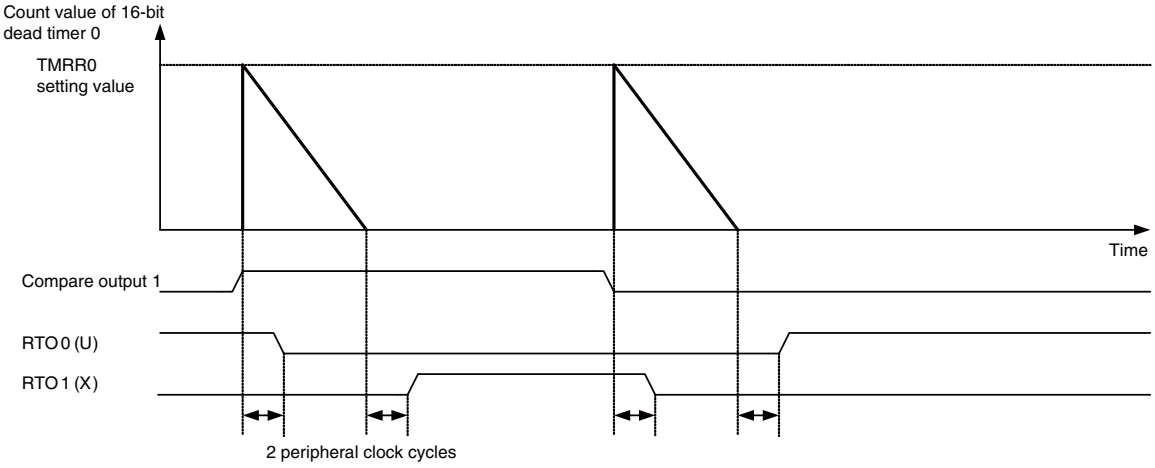
Signal generation is executed by outputting U/V/W and X/Y/Z of the normal polarity non-overlap signal that does not operate by minus control as X/Y/Z and U/V/W.

Figure 46.5-7 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 X0X0XXXX ---- XXXX_B"
CPCLR	: "XXXX_H" (cycle setting)
OCCP0 to OCCP5	: "XXXX_H" (compare value)
OCS01 to OCS45	: "-XX1XXXX XXXXXX11_B"
DTSCR0 to DTSCR2	: "0XXXX100_B"
TMRR0 to TMRR	: "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

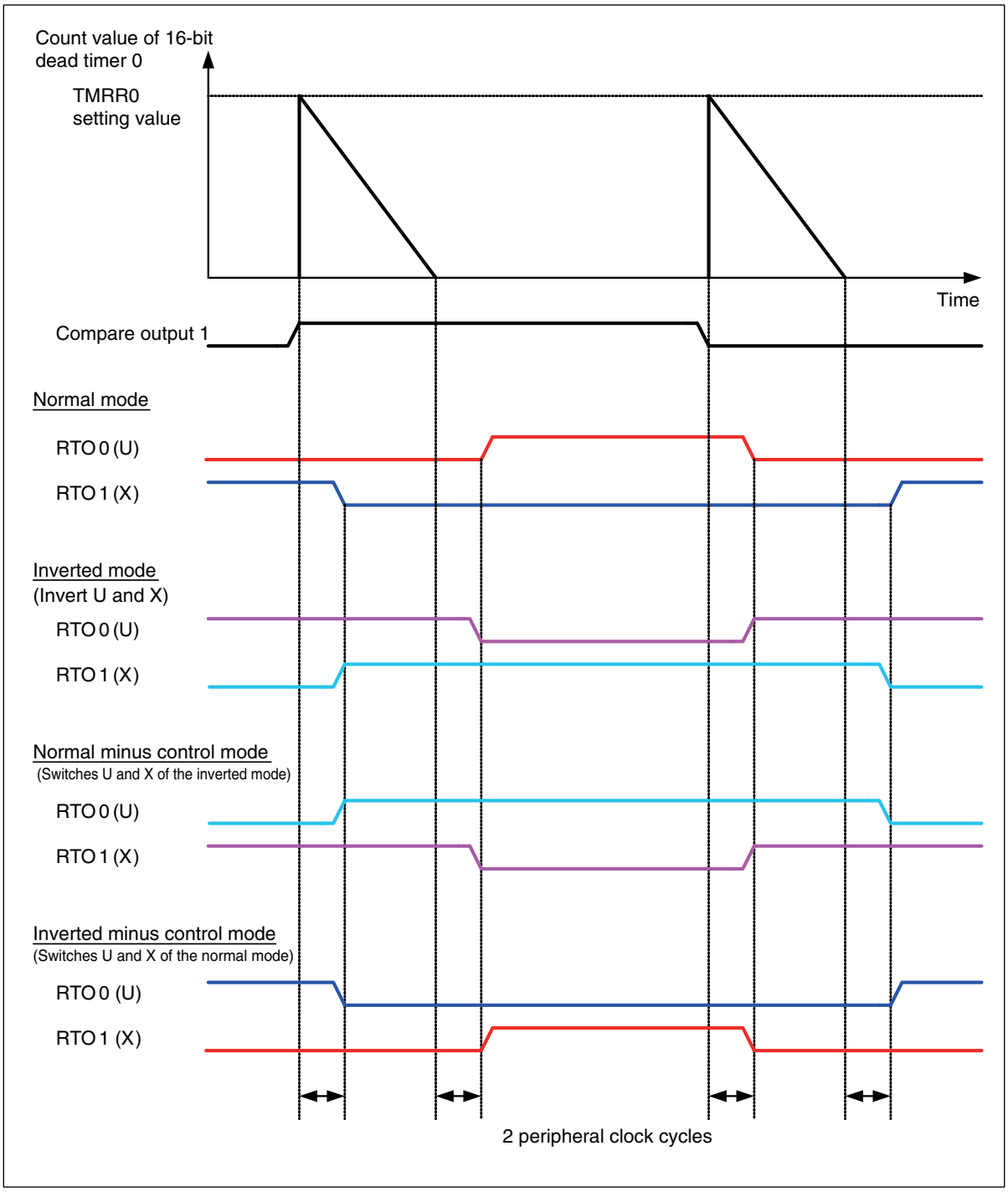
Default is the normal mode.

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 46.5-8 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 46.5-9 When the "H" Interval of the Compare Output Is Long and a Reload Occurs Before the Dead Timer Underflows

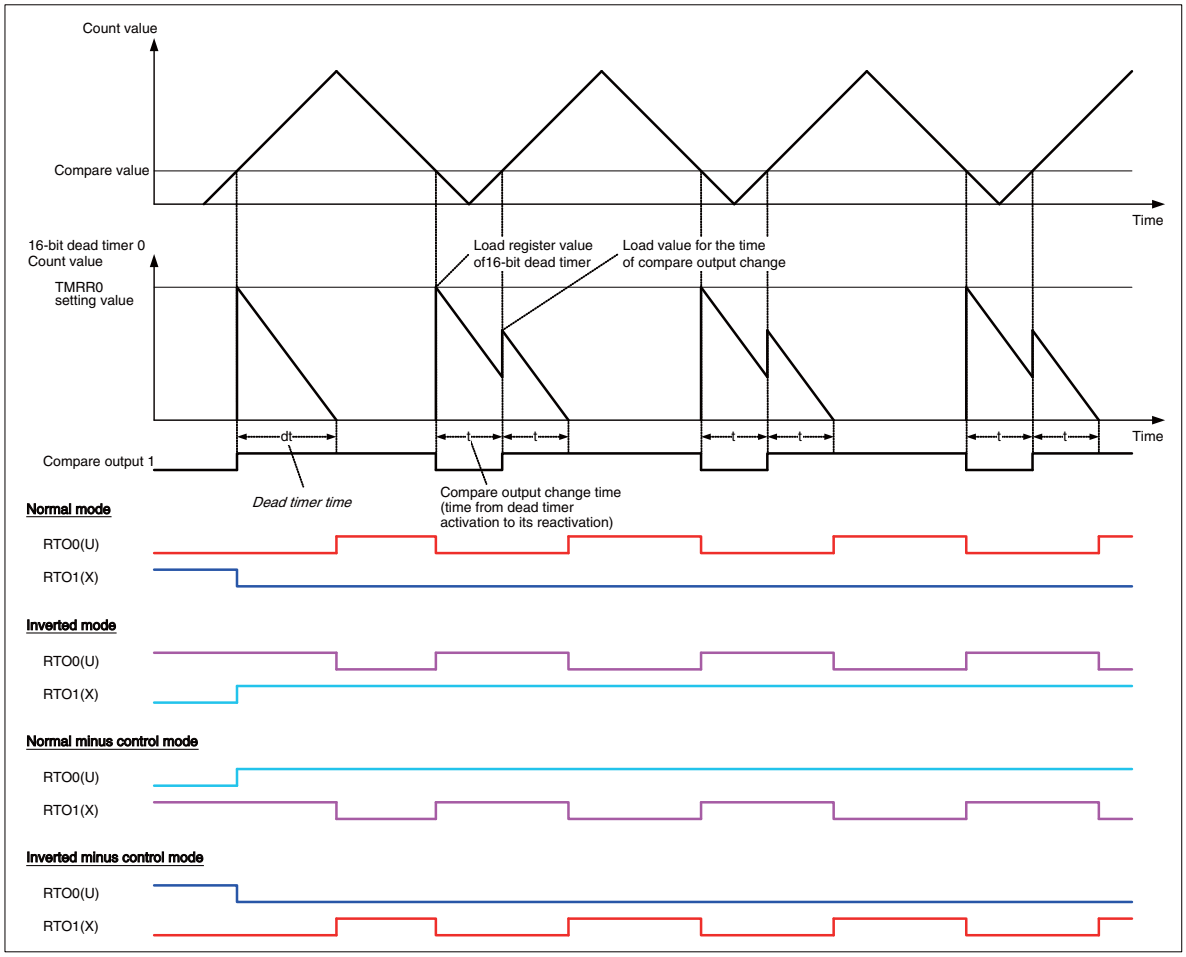
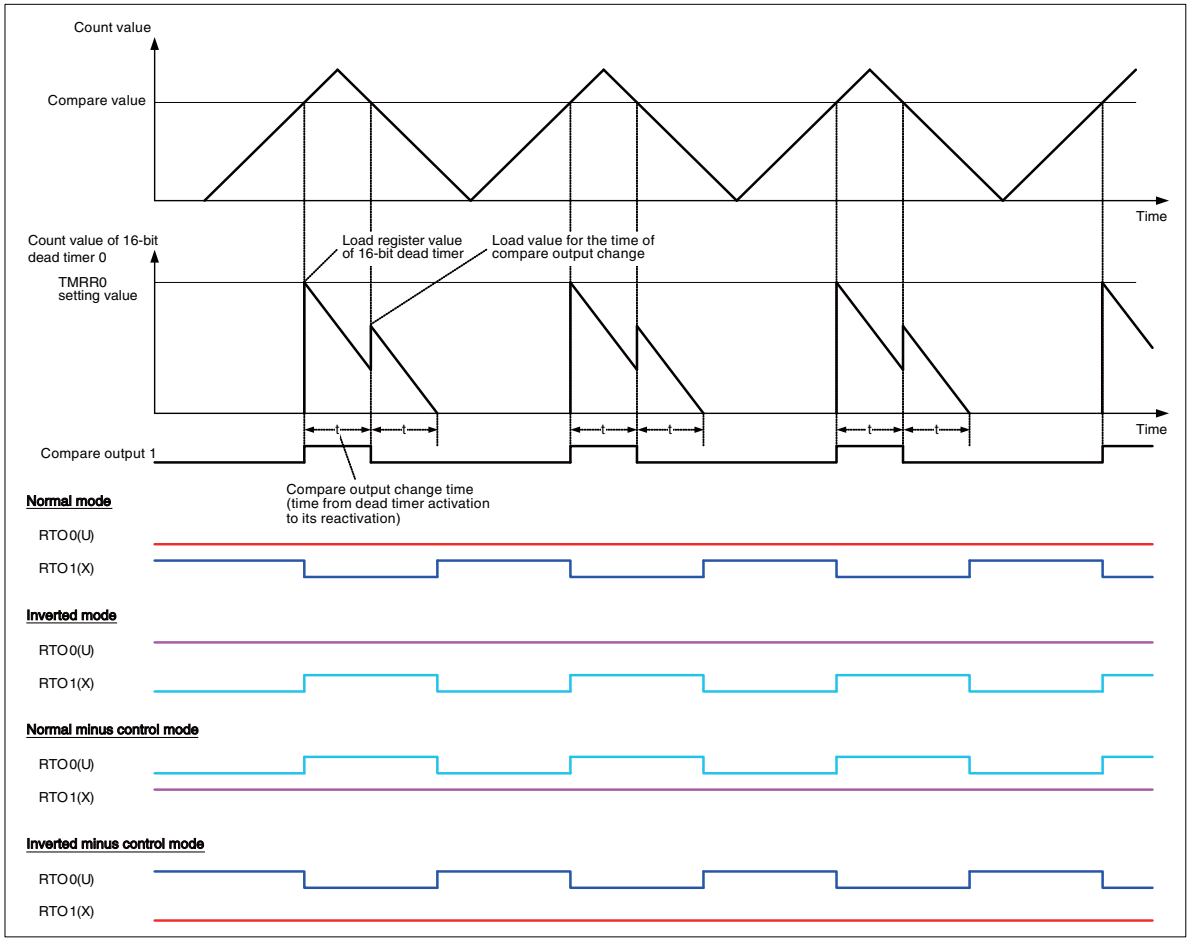


Figure 46.5-10 When the "H" Interval of the Compare Output Is Short and a Reload Occurs Before the Dead Timer Underflows



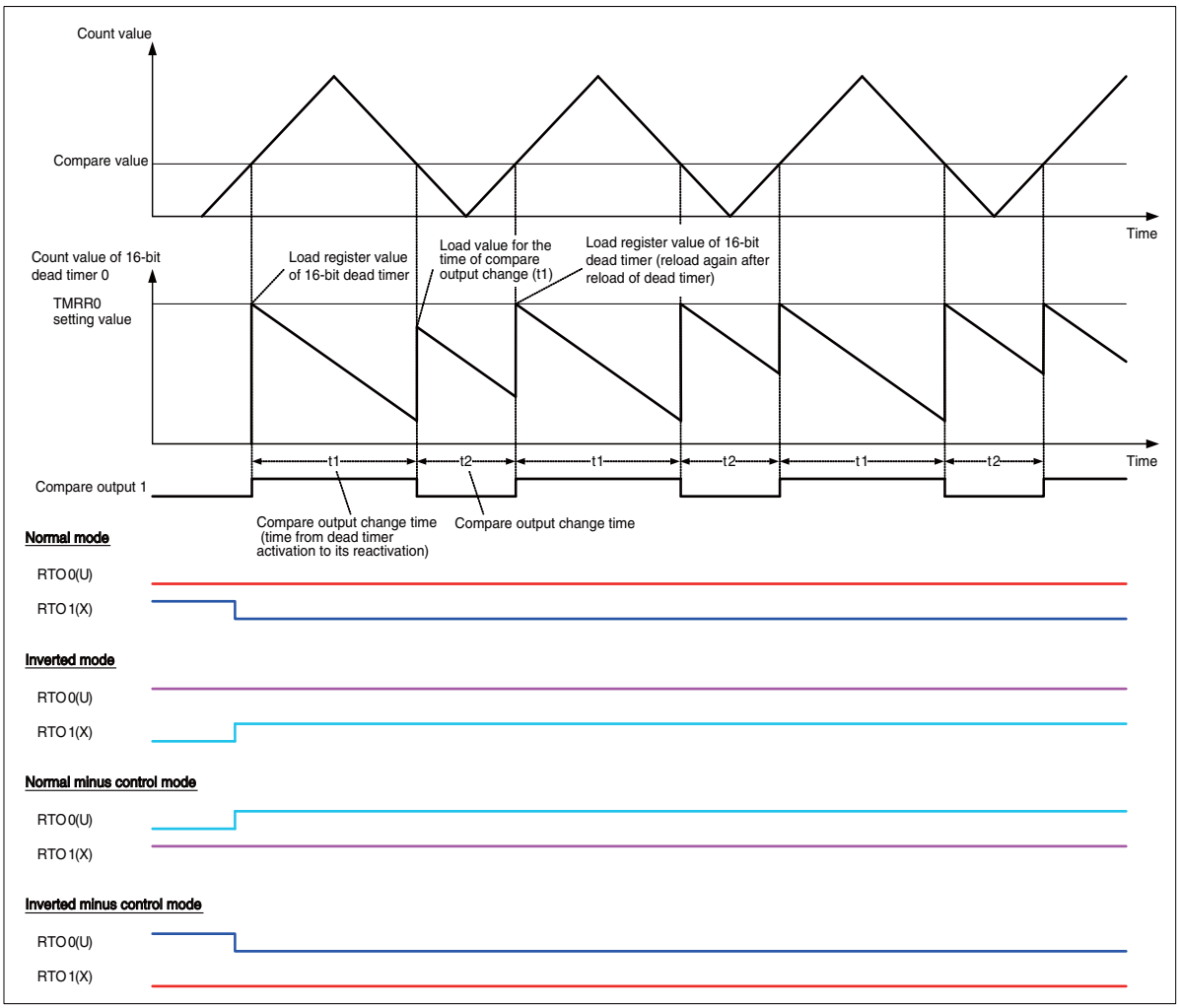
- When the compare output transition time is short and reload continues before the dead timer underflows (the reload of the dead timer continues)

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.

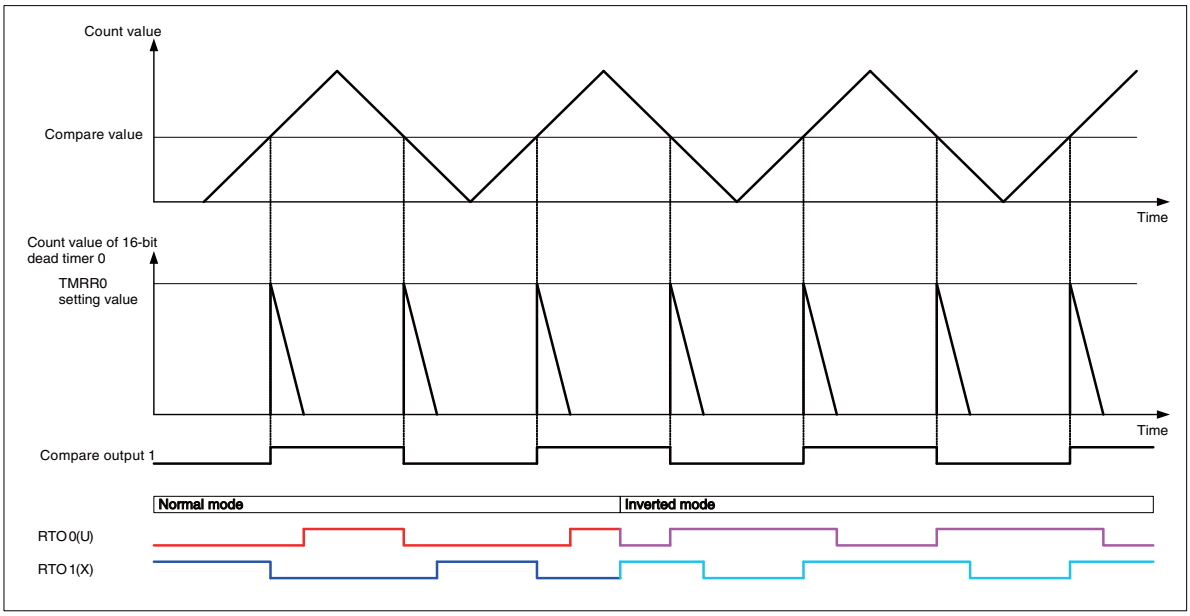
Figure 46.5-11 When the compare output transition time is short and reload continues before the dead timer underflows



● 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 46.5-12 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 46.5-13 Changing from Normal Mode to Normal Minus Control Mode While the Dead Time Timer Mode Is Active #1

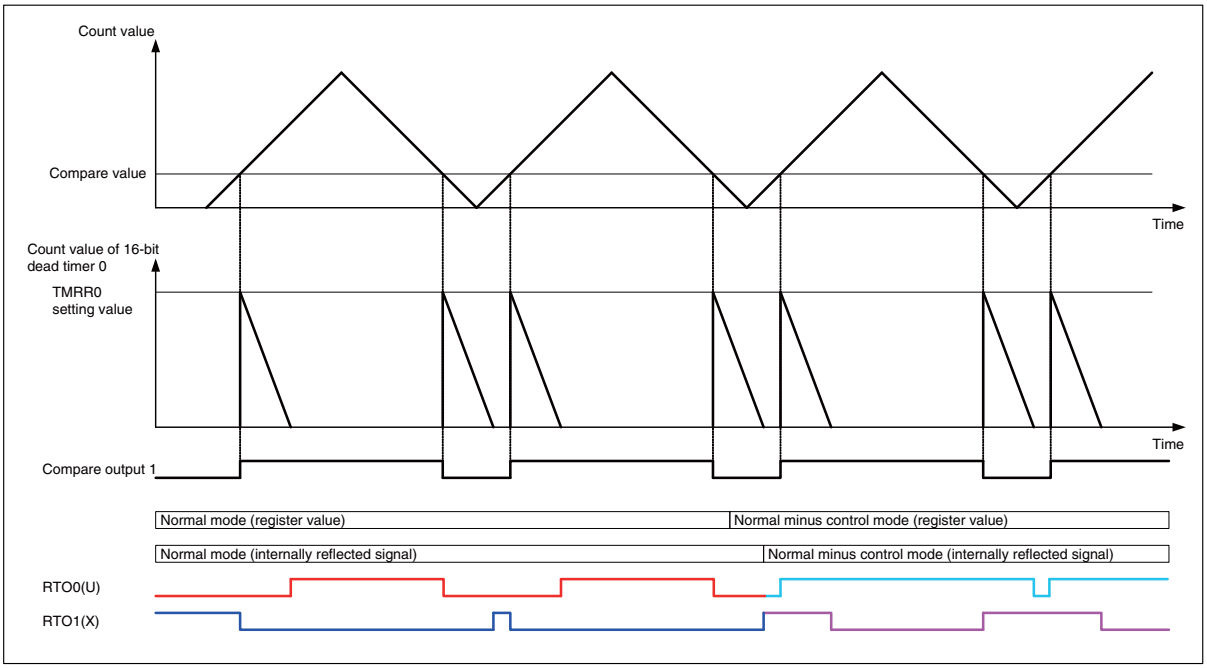
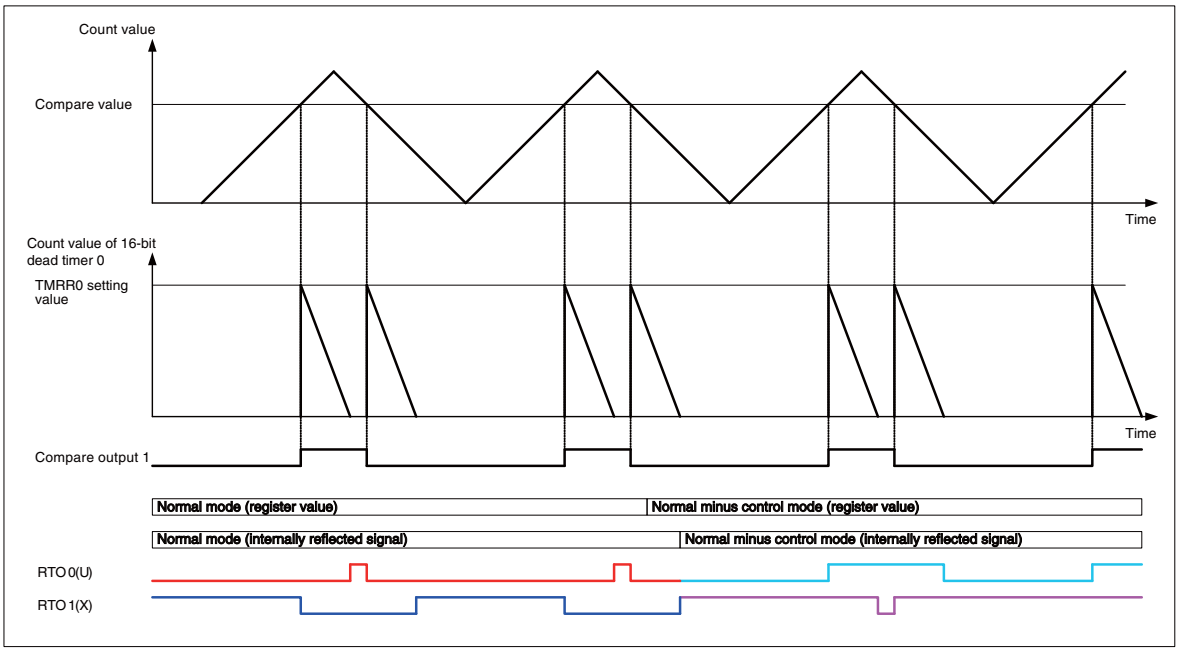


Figure 46.5-14 Changing from Normal Mode to Normal Minus Control Mode While the Dead Time Timer Mode Is Active #2



■ Operation of the DTTI pin control

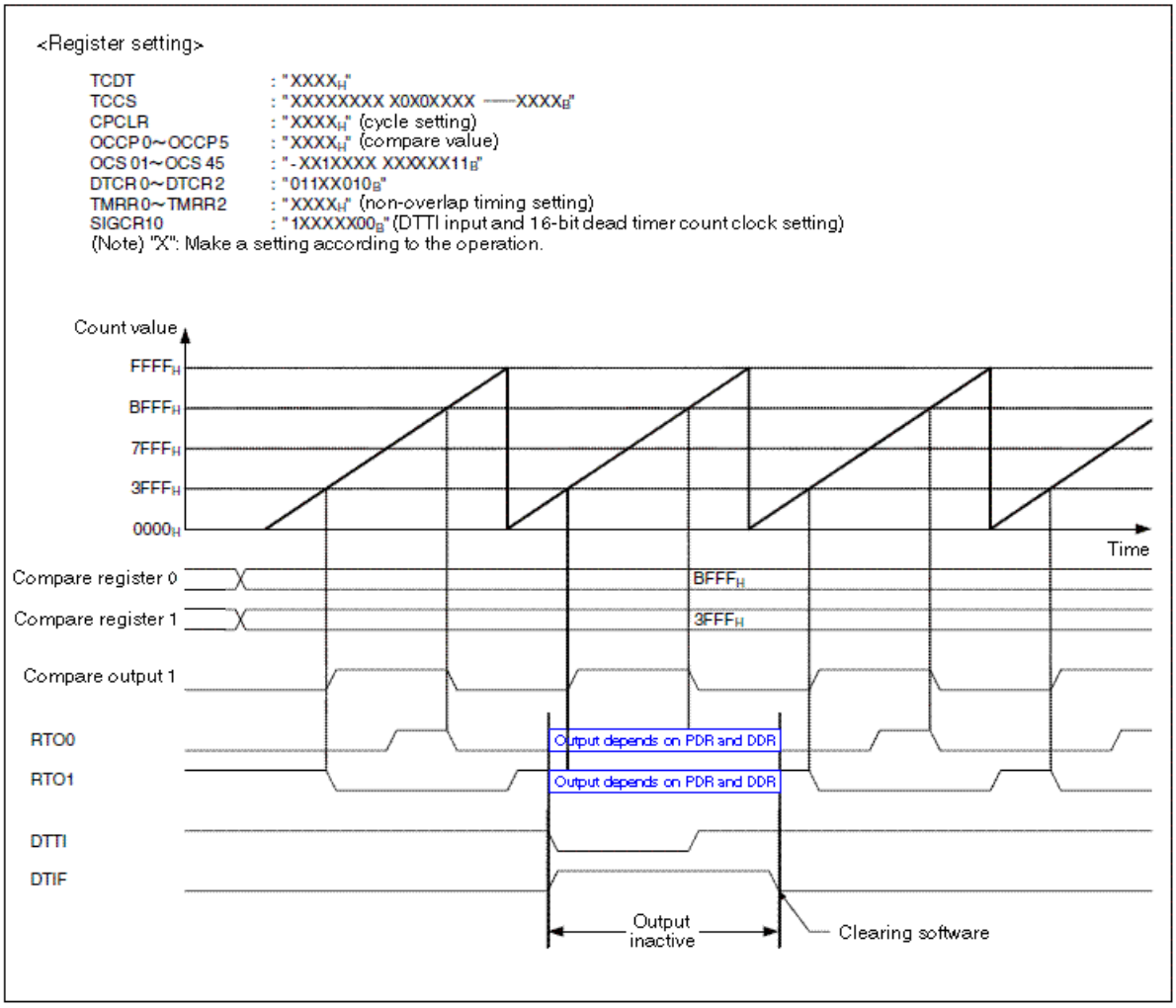
You will be able to control the RTO0 to RTO5 outputs by DTTI pins when you set the DTIE of the waveform control register 1 (SIGCR1) to "1". If "L" level of the DTTI pin is detected, RTO0 to RTO5 function as general purpose ports until the interrupt flag (DTIF of SIGCR1 register) is cleared.

The state of general purpose ports RTO0 to RTO5 during the detection of interrupt flag (DTIF) is independent of the setting of the extended port function registers (EPFR).

See "CHAPTER 24 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 46.5-15 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.

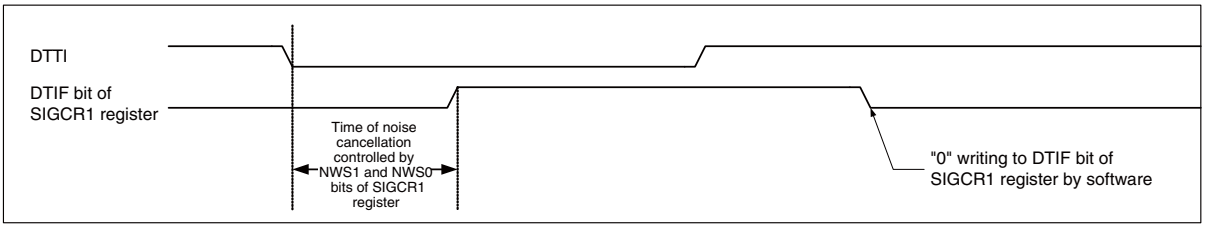
■ DTTI Pin Noise Cancellation Feature

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 46.5-16 DTTI Interrupt Timing



<Notes:>

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.

46.6 Notes:

■ 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 47 MOTOR CONTROL EXTENSION FUNCTION

47.1 Overview

The motor control extension function consists of the operating clock selection control and the waveform generator output level conversion control. These controls extend the motor control function.

Motor control function is a collective term for the free-run timer, input capture, output compare, 12-bit A/D converter, and waveform generator.

47.2 Features

■ Operating clock selection control

- With the motor control function operating clock (MTRCLK) accelerated, the waveform generator can output high-resolution control waveforms. The result is motor control with high precision.
- The motor control function operating clock (MTRCLK) can be selected from the peripheral clock (PCLK2) and on-chip bus clock (HCLK).
- The maximum frequency of the on-chip bus clock (HCLK) is 128 MHz.

■ Waveform generator output level conversion control

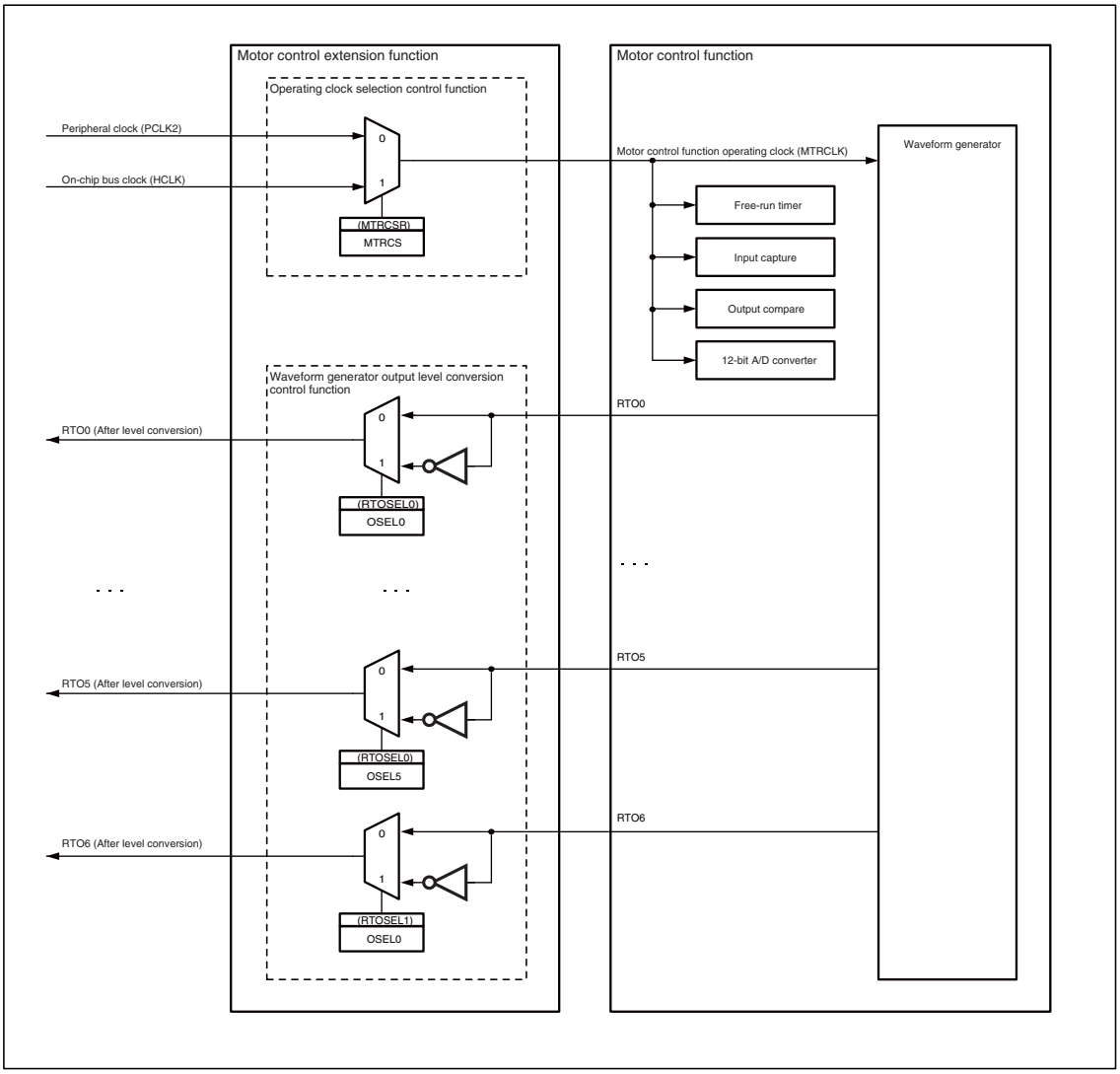
- The waveform generator output level can be set individually to correspond to the polarity of the switch connected to an external device.

Code: BZASBB_MTR-1v0-91580MS-1-E

47.3 Configuration

■ Configuration of the motor control extension function

Figure 47.3-1 Configuration of the Motor Control Extension Function



47.4 Registers

Table 47.4-1 Motor Control Extension Function Register List

Address	+0	+1	+2	+3
$12B0_H$	Motor control operating clock selection register (MTRCSR)	Reserved	Reserved	Reserved
$12B4_H$	Waveform generator output level conversion setting register 0 (RTOSEL0)	Waveform generator output level conversion setting register 1 (RTOSEL1)	Reserved	Reserved

47.4.1 Motor Control Function Operating Clock Selection Register: MTRCSR

This register selects either the peripheral clock (PCLK2) or on-chip bus clock (HCLK) as the motor control function operating clock (MTRCLK). To change the value of this register, stop all of the motor control function (including even the LIN sync field that uses the input capture, PPG-GATE, and AD activation trigger used for a reload timer). For the clock setting procedure, see "CHAPTER 6 CLOCK".

- **MTRCSR: Address $12B0_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
Reserved							MTRCS	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	Attribute

[bit7 to bit1] Reserved

Always write "0" to these bits.

[bit1] MTRCS: Motor control function operating clock selection bit

MTRCS	MTRCLK Output Selection
0	Peripheral clock (PCLK2)
1	On-chip bus clock (HCLK)

•This bit selects the motor control function operating clock (MTRCLK).

(Note) Interrupt/interrupt clear of the motor control function, the LIN sync field that uses the input capture, the PPG GATE function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer cannot normally be passed at the PCLK2-HCLK switching time. Therefore, for PCLK2-HCLK switching, stop all of the motor control function (including even the LIN sync field that uses the input capture, the PPG GATE function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer).

47.4.2 Waveform Generator Output Level Conversion Setting Register: RTOSEL0, RTOSEL1

This register enables the RTO6 to RTO0 outputs of the waveform generator to be individually inverted for output to external pins.

- **RTOSEL0: Address 12B4$_H$ (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
-		OSEL5	OSEL4	OSEL3	OSEL2	OSEL1	OSEL0	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] Undefined

These bits read out "0". Writing has no effect on operation.

[bit5] OSEL5 (Output SELect 5): Waveform generator output level conversion setting bit 5

OSEL5	Function
0	Outputs the RTO5 output of the waveform generator as is to an external pin.
1	Inverts the RTO5 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO5).

[bit4] OSEL4 (Output SELect 4): Waveform generator output level conversion setting bit 4

OSEL4	Function
0	Outputs the RTO4 output of the waveform generator as is to an external pin.
1	Inverts the RTO4 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO4).

[bit3] OSEL3 (Output SELect 3): Waveform generator output level conversion setting bit 3

OSEL3	Function
0	Outputs the RTO3 output of the waveform generator as is to an external pin.
1	Inverts the RTO3 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO3).

[bit2] OSEL2 (Output SELect 2): Waveform generator output level conversion setting bit 2

OSEL2	Function
0	Outputs the RTO2 output of the waveform generator as is to an external pin.
1	Inverts the RTO2 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO2).

[bit1] OSEL1 (Output SELect 1): Waveform generator output level conversion setting bit 1

OSEL1	Function
0	Outputs the RTO1 output of the waveform generator as is to an external pin.
1	Inverts the RTO1 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO1).

[bit0] OSEL0 (Output SELect 0): Waveform generator output level conversion setting bit 0

OSEL0	Function
0	Outputs the RTO0 output of the waveform generator as is to an external pin.
1	Inverts the RTO0 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 0 (RTO0).

• RTOSEL1: Address $12B5_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
-		Reserved					OSEL0	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R/W0	R/W0	R/W0	R/W0	R/W0	R/W	Attribute

[bit7 to bit6] Undefined

These bits read out "0". Writing has no effect on operation.

[bit5 to bit1] Reserved

Always write "0" to these bits.

[bit0] OSEL0 (Output SELect 0): Waveform generator output level conversion setting bit 0

OSEL0	Function
0	Outputs the RTO6 output of the waveform generator as is to an external pin.
1	Inverts the RTO6 output of the waveform generator and outputs it to an external pin.

•This bit enables the selection of level conversion for output from waveform generator unit 1 (RTO6).

47.5 Operation Explanation

47.5.1 Operating Clock Selection Control

This function selects either the peripheral clock (PCLK2) or on-chip bus clock (HCLK) as the motor control function operating clock (MTRCLK). MTRCSR.MTRCS is used to select the clock.

For details on the on-chip bus clock (HCLK) settings, see "CHAPTER 6 CLOCK".

(Notes)

- If the on-chip bus clock (HCLK) is selected as the motor control function operating clock (MTRCLK), asynchronous clock transfer is generated between the peripheral and motor control functions. In this case, if the clock mode is RUN1 or RUN2, a penalty of MTRCLK x (5 to 8) cycles is added to the access cycle. If the clock mode is RUN3, the penalty is PCLK2 x (5 to 8) cycles + MTRCLK x (5 to 8) cycles.
- If the on-chip bus clock (HCLK) is selected as the motor control function operating clock (MTRCLK), asynchronous clock transfer is generated between the multi-function serial interface and motor control function. This results in a malfunction of the LIN sync field function that uses the input capture.
- If the on-chip bus clock (HCLK) is selected as the motor control function operating clock (MTRCLK), asynchronous clock transfer is generated between individual functions. These functions are interrupt/interrupt clear of the motor control function, the PPG GATE function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer. This results in a transmission delay. The delay caused between an interrupt of the motor control function and the PPG GATE function is PCLK2 x 2 cycles. The delay caused between any of an interrupt clear of the motor control function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer is MTRCLK x 2 cycles.
- Interrupt/interrupt clear of the motor control function, the LIN sync field that uses the input capture, the PPG GATE function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer cannot normally be passed at the PCLK2-HCLK switching time. Therefore, for PCLK2-HCLK switching, stop all of the motor control function (including even the LIN sync field that uses the input capture, the PPG GATE function, PPG output of the waveform generator, and the AD activation trigger used for a reload timer).

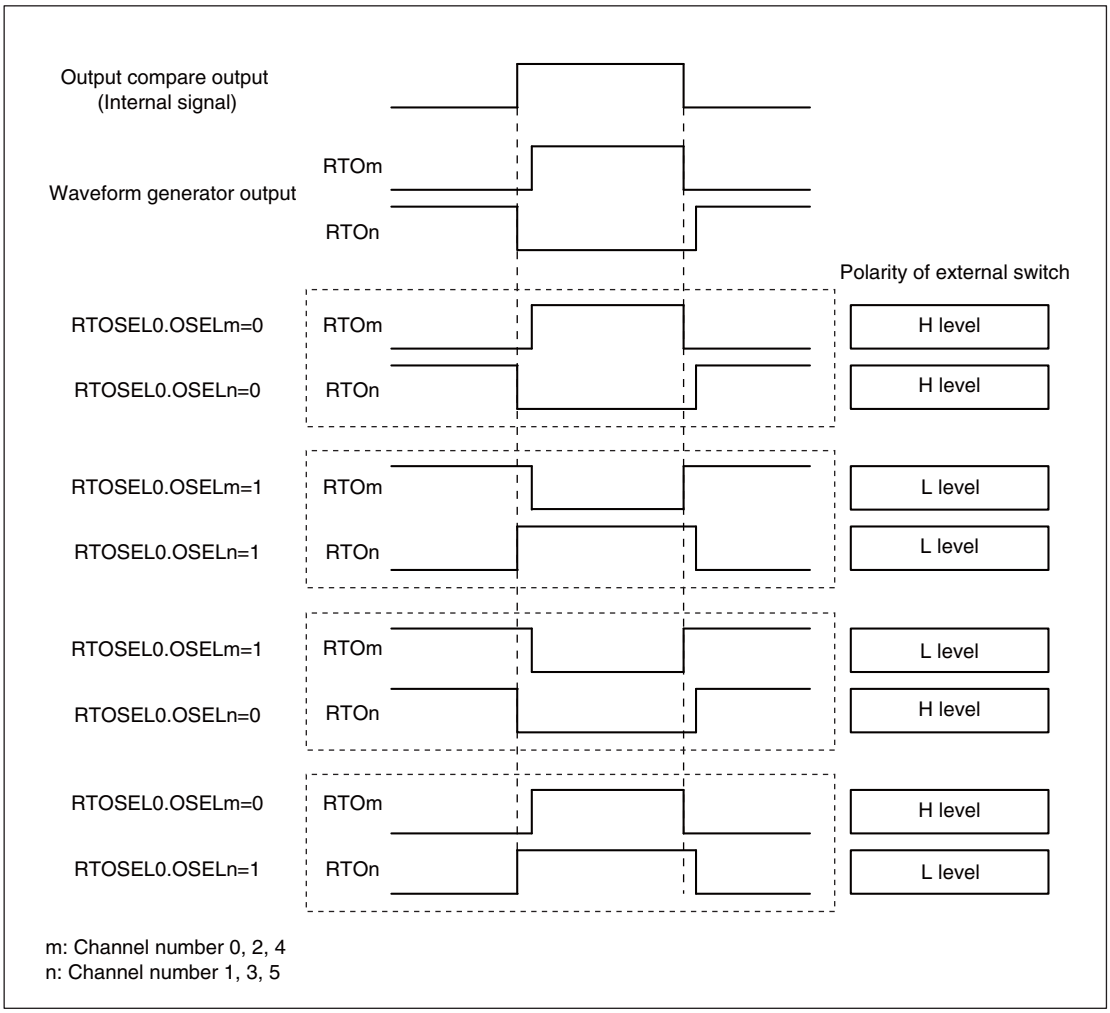
47.5.2 Waveform Generator Output Level Conversion Control

In motor control, control waveforms must be output adjusted to correspond to the polarity of a switch such as a pre-driver mounted externally.

Normally, the waveform generator outputs complementary waveforms with dead time (non-overlap time) as real-time outputs. In contrast, the waveform generator output level conversion control can individually set the following level conversion control on each output channel.

The following figure shows the waveform outputs of waveform generator unit 0 as an example.

Figure 47.5-1 Waveform Output with Output Level Conversion Control of Waveform Generator Unit 0

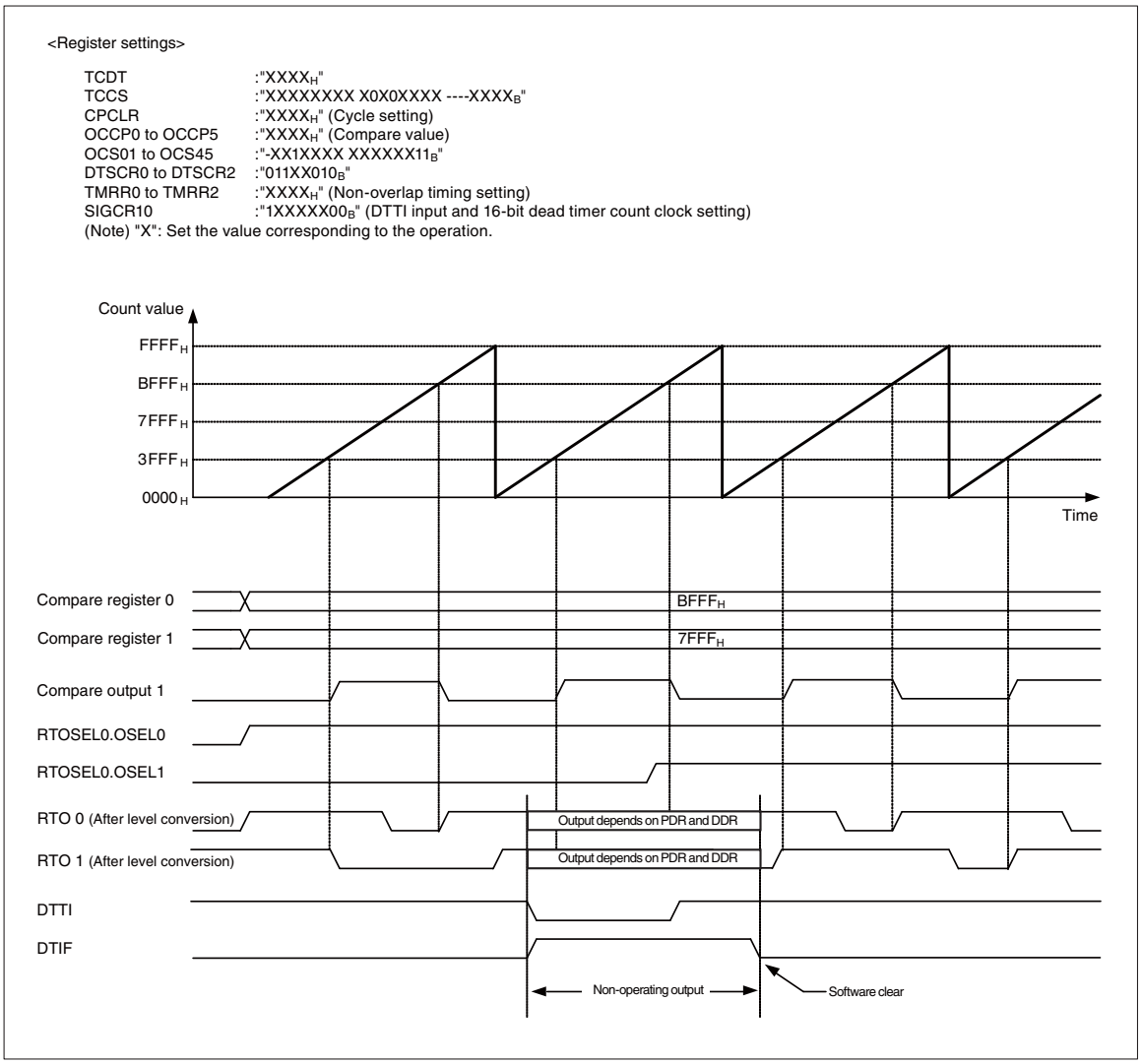


Before starting waveform output with the waveform generator, configure the conversion settings. Otherwise, the intended output may not be generated because settings made while waveforms are output are immediately reflected in the conversion settings.

For details on the waveform generator settings, see "CHAPTER 46 WAVEFORM GENERATOR".

If the DTTI register forcibly stops waveform generator output, the function registers serve as general-purpose ports. If the data direction register is set to output, the value of the port data register is output regardless of the RTOSEL0/1.OSELx settings.

Figure 47.5-2 Operation with DTTI Input Enabled in RTOSEL0/1.OSELx Settings



47.5.3 A/D Mode Setting Register of the A/D Converter (ADMD)

The compare time must be set to at least 700 ns, and the sampling time to at least 300 ns.

If the on-chip bus clock (HCLK) is selected as the motor control function operating clock (MTRCLK), the compare time and sampling time of A/D conversion must be set appropriate to the frequency with the A/D mode setting register (ADMD).

The following tables list the recommended settings (settings for the minimum compare time and sampling time) in cases where the set frequency is higher than the conventional frequency allowed to be set.

Recommended settings

Motor control function operating clock (MHz)	CT1	CT0	Compare time (ns)
128	1	1	875
120	1	1	933.3
80	1	0	700
64	1	0	875
40	0	0	700
32	0	0	875
24	0	0	1166.7
16	0	0	1750

Motor control function operating clock (MHz)	ST1	ST0	Sampling time (ns)
128	1	1	375
120	1	1	400
80	1	0	300
64	1	0	375
40	0	0	300
32	0	0	375
24	0	0	500
16	0	0	750

(Note) When rewriting bits, be sure that the A/D operation before the conversion operation has stopped.

47.6 Notes

Note the following when using the motor control extension function.

■ Timing of the waveform generator output level conversion control function

When an output level conversion setting is changed by a waveform generator output level conversion setting register, the new setting is reflected in operation immediately.

Therefore, any change in the output level conversion settings while the waveform generator output operation is in progress may cause the output of a glitch or other unintended waveform. For this reason, we recommend that the waveform generator output level conversion setting registers be set prior to the waveform generator output operation.

CHAPTER 48 D/A CONVERTER

48.1 Overview

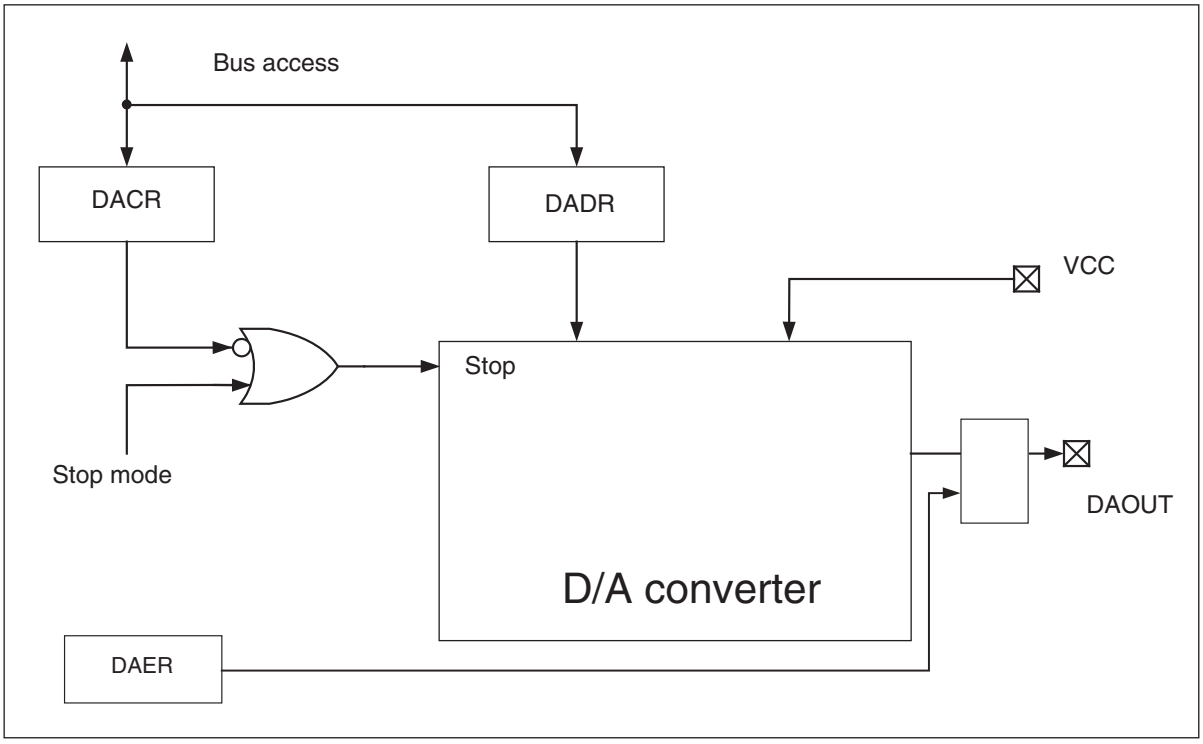
The D/A converter is a peripheral function to convert digital signals to analog signals. This product incorporates 1 channel of the 10-bit D/A converter.

48.2 Features

- Power Down Function
  Power down function which turns the power off when the outputs are disabled by the D/A converter is incorporated.
- See the data sheets for the output voltage range.

48.3 Configuration

Figure 48.3-1 Block Diagram of D/A Converter



Code: ZRDC-0v5-91580MS-1-E

48.4 Registers

Table 48.4-1 D/A Converter Register List

Address	Register				Register function
	+0	+1	+2	+3	
$0F4C_H$	DAER	Reserved	Reserved	Reserved	Analog output control register (key code target register)
1400_H	DACR	Reserved	DADR		D/A control register D/A data register

48.4.1 DACR (D/A Control Register)

This register enables the output from the D/A pins.

- **DACR: Address 1400_H (Access: Byte, Half-word, Word)**

Bit	15	14	13	12	11	10	9	8
	-							DAE
	0	0	0	0	0	0	0	0
	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit15 to bit9] Undefined bits

The read value is always "0". Write does not affect the operation.

[bit8] DAE:D/A output enable bit

DAE	DA output enable
0	Output disabled
1	Output enabled

48.4.2 DADR (D/A Data Register)

This register is used to set the output voltage from the DAOUT pins. The output voltage from the D/A converter is calculated based on the values stored in this register.

- **DADR: Address 1402_H (Access: Half-word)**

Bit	15	14	13	12	11	10	9	8
	-						DA[9:8]	
	0	0	0	0	0	0	X	X
	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	DA[7:0]							
	X	X	X	X	X	X	X	X
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] Undefined bits

The read value is always "0". Write does not affect the operation.

[bit9 to bit0] DA9 to DA0: D/A output value

DA[9:0]	DA output voltage
00 0000 0000	0/1024 × VCC
00 0000 0001	1/1024 × VCC
00 0000 0010	2/1024 × VCC
~	
11 1111 1101	1021/1024 × VCC
11 1111 1110	1022/1024 × VCC
11 1111 1111	1023/1024 × VCC

48.4.3 DAER (D/A Enable Register) Analog Output control Register

This register specifies whether the port function or the DA output pin is set to P030.

- **DAER: Address 0F4C$_H$ (Access: Byte, Half-word)**

Bit	15	14	13	12	11	10	9	8
	-							DAS
	0	0	0	0	0	0	0	0
	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit15 to bit9] Undefined bits

The read value is always "0". Write does not affect the operation.

[bit8] DAS: Analog Output Control Bit

DAS	Analog output control
0	Port function (Initial)
1	D/A output pin

(Note) This register is a key code target register. It is necessary to set the key code in order to execute writing. For information on the setting method, see Sections "24.4.10 KEYCDR key code register" and "24.5.10 Key Code Register Function Settings" in "CHAPTER 24 I/O PORTS". Word access for this register is prohibited.

(Note) If the DA output is disabled, the internal amplifier is powered down. In this state, if the DA output is enabled by controlling the DAS bit of DAER, note that a certain time frame (10μs) is required for power-on of the amplifier and for output stabilization.

48.5 Operating Explanation

The D/A converter outputs analog voltage from the DAOUT pins by calculating the output voltage based on the values written in the D/A data register (DADR).

If values are written to the DA9 to DA0 bits of the D/A data register (DADR) and "1" is written to the DAE bit of the D/A control register (DACR), analog signals will be output from the D/A converter.

In addition, if "0" is written to the DAE bit of the D/A control register (DACR), the output pin DAOUT from the D/A converter becomes Hi-Z state. In addition, even if "1" is written to the DAE bit, the output pin DAOUT from the D/A converter also becomes Hi-Z state when the CPU is in stop mode.

The voltage, which can be output when the output from the D/A converter is enabled by writing "1" to the DAE bit of the D/A control register (DACR), is from 0.0 V to 1023/1024 × VCC.

■ Setting Procedure/Methods

Set the D/A converter with following procedures and methods.

● Output from DAOUT pins (the setting must be made in the order (1) to (4)):

(1) Set P030 to port function side.(DAER setting)
(2) Set the output voltage of the D/A converter. (DADR setting)
(3) Enable the output of the D/A converter. (DADR DAE="1" setting)
(4) Set P030 to D/A output side.(DAER setting)
The output of the D/A converted values will be started from the P030 pin.

● Disabling the output from the DAOUT pins

- Set the D/A output pin to port function side.(DAER=0 setting)
- Set the D/A converter output disabled. (DACR DAE="0" setting)

CHAPTER 49 ON-CHIP DEBUGGER (OCD)

49.1 Overview

This chapter explains an overview of the on-chip debugger (OCD) in this product 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, and the like), and security function.

49.2 Features

- One-wire debug tool I/F
- Debug security function
- Debug mode control function
- Execution control function
  - Status display functions (chip status, CPU status, and the like)
  - Debug command execution control function
  - Small debug main memory (8 bytes = 4 instructions)
  - CPU register save register (PC/PS)
  - PC monitor function
  - Reset function
    - Chip reset (INIT)
    - CPU reset (RST)
- Break function
  - Step execution break
  - Event trigger break
  - Forced break
  - Guarded access break
  - Trace end break
  - Control on interrupt acceptance immediately after the execution start address
- Debug DMA function (DDMA function)
  - Support of transfer modes (address mode, verify mode, burst transfer of DEBUG I/F)
- Event function
  - Code event (including conditional code event)
  - Data event
  - Interrupt event
  - User event
  - Event sequencer
- Execution time measurement timer function
  - Go-Break measurement
  - Inter-trigger measurement (single measurement/cumulative measurement)

Code: FR81S09_OCD-1v0-91580MS-1-E

- Trace function
  - Special state trace
  - Branch trace
  - Data trace
  - Trace delay

49.3 Configuration

Figure 49.3-1 Block Diagram of OCDU

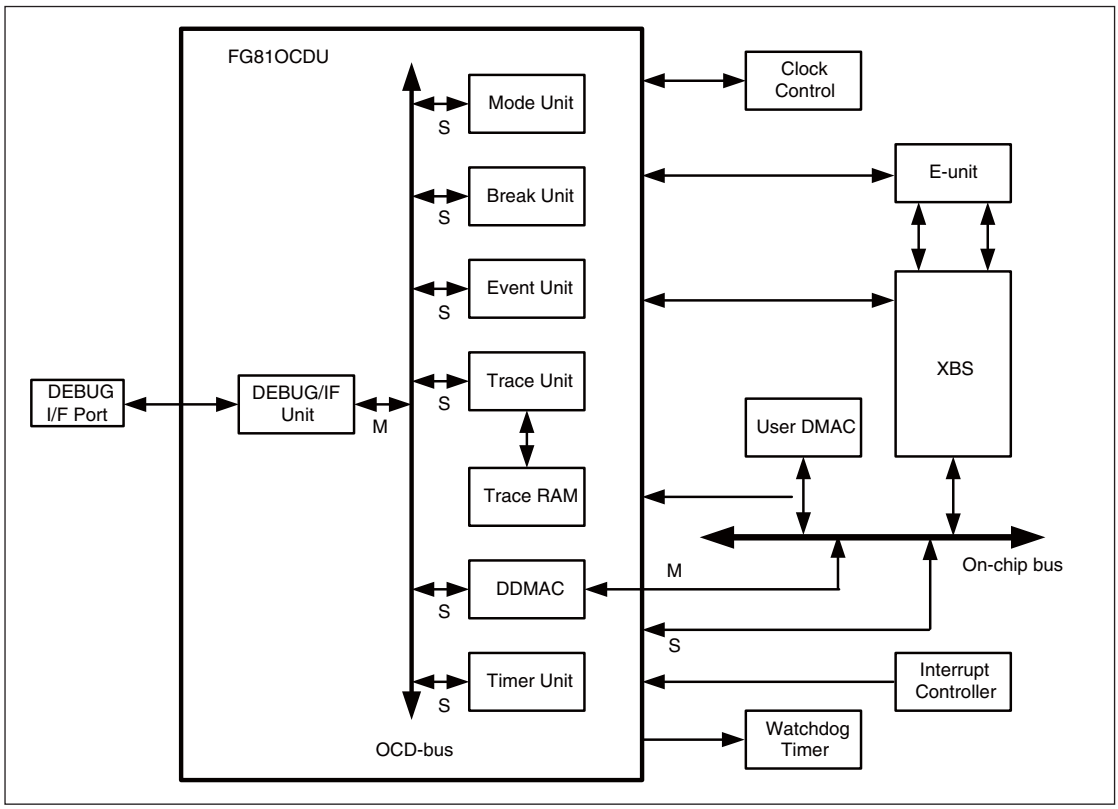
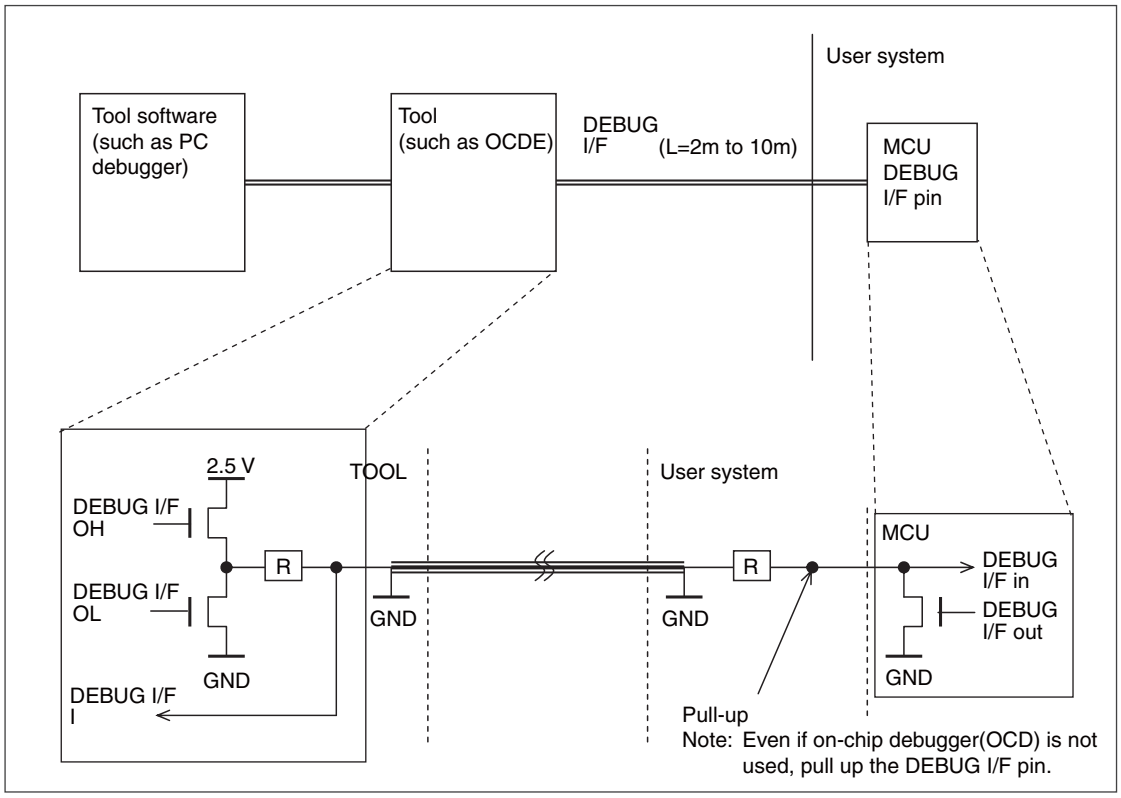


Figure 49.3-2 OCD Connection Diagram



49.3.1 Clock for DEBUG I/F

See "CHAPTER 6 CLOCK" for the clock connection composition of the clock for DEBUG I/F.

49.3.1.1 Main clock for DEBUG I/F (M_MCLK)

When the OCD tool is connected, the main clock (MCLK) is supplied to the main clock (M_MCLK) for DEBUG
I/F.

The main clock (M_MCLK) for DEBUG I/F stops when the OCD tool is unconnected.

49.3.1.2 PLL clock for DEBUG I/F (M_PCLK)

When High-speed UART mode and phase modulation UART mode are selected at connected to the OCD tool, the PLL clock (PLLCLK) is supplied to the PLL clock (M_PCLK) for DEBUG I/F.

The PLL clock (M_PCLK) for DEBUG I/F stops when the OCD tool is unconnected.

49.4 Registers

49.4.1 DBG Registers

Table 49.4-1 Registers Map (DBG Registers)

Address	Register				Registers function
	+0	+1	+2	+3	
0xFF00	DSUCR		Reserved		DSU control register

49.4.1.1 DSU Control Registers (DSUCR)

This register is used to control DSU in the free-run mode.
For details, contact a Spansion support representative.

- **DSUCR: Address $FF00_H$(Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved								
X	X	X	X	X	X	X	X	Initial value
RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	Attribute

7	6	5	4	3	2	1	0	bit
Reserved							DSU	
X	X	X	X	X	X	X	0	Initial values
RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R,W	Attribute

49.4.2 User IO Registers

Table 49.4-2 Register Map (User IO Registers)

Address	Register				Register function
	+0	+1	+2	+3	
0x0BFC	Reserved		UER		User event register

49.4.2.1 User Event Register (UER)

This register is used to detect a user event.
For details, contact a Spansion support representative.

- **UER: Address $0BFE_H$ (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
Reserved								
X	X	X	X	X	X	X	X	Initial values
RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	Attributes

7	6	5	4	3	2	1	0	bit
Reserved							UEVT	
X	X	X	X	X	X	X	0	Initial values
RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W	Attribute

49.5 Operation

49.5.1 OCDU Operating Mode

49.5.1.1 About Operating Modes

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

49.5.1.2 Operating Mode State Transition

At INIT releasing (including RST accompanied by INIT), control transits to the debug state of the emulator mode or to the free-run mode according to the mode command from DEBUG I/F in the chip reset sequence.

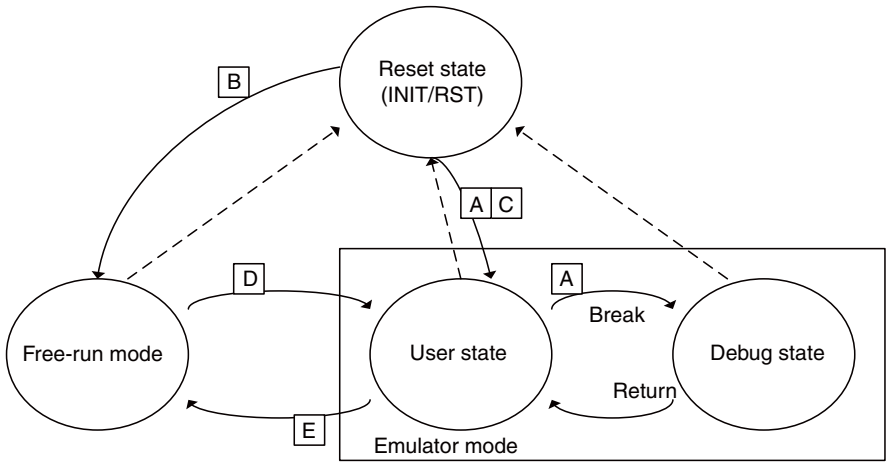
At RST releasing (not accompanied by INIT), control transits to the operating mode occurring before RST generation. However, if a forced break request is issued after RST occurs in the user state, control transits to the debug state of the emulator mode at RST releasing.

Moreover, transition between the free-run mode and user state of emulator mode is enabled by OCD register control.

At transition from the reset status to the debug state, control first transits to the user state. In this case, requesting a break by OCDU makes the following transition: reset status → user state → (break) → debug state.

The transition conditions are shown below.

Figure 49.5-1 OCDU Operating Mode Diagram



[Transition indicated by broken lines]
Transition when a reset (INT, RST) occurs. All states transit to the reset state.
[Transition indicated by solid lines]
A: (1) Transition when a mode command is received successfully in the DEBUG I/F chip reset sequence at INIT release after the reset with INIT occurs
(2) Transition after RST release after the reset without INIT occurs in the debug state
(3) Transition at RST release when a forced break is requested before RST release after the reset without INIT occurs in the user state or pseudo on-the-fly break is requested
B: (1) Transition when a mode command is not received successfully in the DEBUG I/F chip reset sequence at INIT release after the reset with INIT occurs
(2) Transition at RST release after the reset without INIT occurs in the free-run mode
C: Transition at RST release after the reset without INIT occurs in the user state
D: Transition if "1" is written to the E_MSTSR.DMODE bit when OCDU operation is enabled in the free-run mode
E: Transition if "0" is written to the E_MSTSR.DMODE bit in the user state.

49.5.2 Overview of DEBUG I/F

DEBUG I/F is a single-wire debug interface that connects between MCU and 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 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. The chip reset sequence consists of the function that enables MCU to provide notification of INIT generation and the function that detects the debug mode to be activated after INIIT releasing. The stall function provides communication stall and forced break requests from the tool, and communication error notification from MCU.

The main DEBUG I/F functions are shown below.

- Chip reset sequence function (INIT notification, mode command)
- UART function (normal UART, high-speed UART, phase modulation UART)
- Stall request (communication stall request, forced break request, communication error notification)

The two-way pin of DEBUG I/F is accomplished by 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 49.3-2 OCD Connection Diagram".

49.5.2.1 Chip reset sequence

When INIT is generated, OCDU executes the chip reset sequence according to the specification of DEBUG I/F. A standard clock that executes the chip reset sequence is usually a sampling clock of UART (clock of main clock (M_MCLK) for DEBUG I/F eight dividing frequency).

The chip reset sequence has the following five phases.

Beginning phase, INIT phase, Level sense phase, Mode entry phase, End phase

- Beginning phase
  The period from point to which generated INIT is released to sampling clock 32 cycles of normal UART is the beginning phase. OCDU does not do special operation by this phase.

- INIT notification phase
  The period from beginning phase end to sampling clock 480 cycles of normal UART is the INIT notification phase. OCDU outputs L to DEBUG I/F while it is in this phase, and notifies the tool the generation of INIT.

- Level sense phase
  The period from INIT notification phase end to sampling clock 256 cycles of normal UART is the level sense phase. OCDU does not do special operation by this phase.

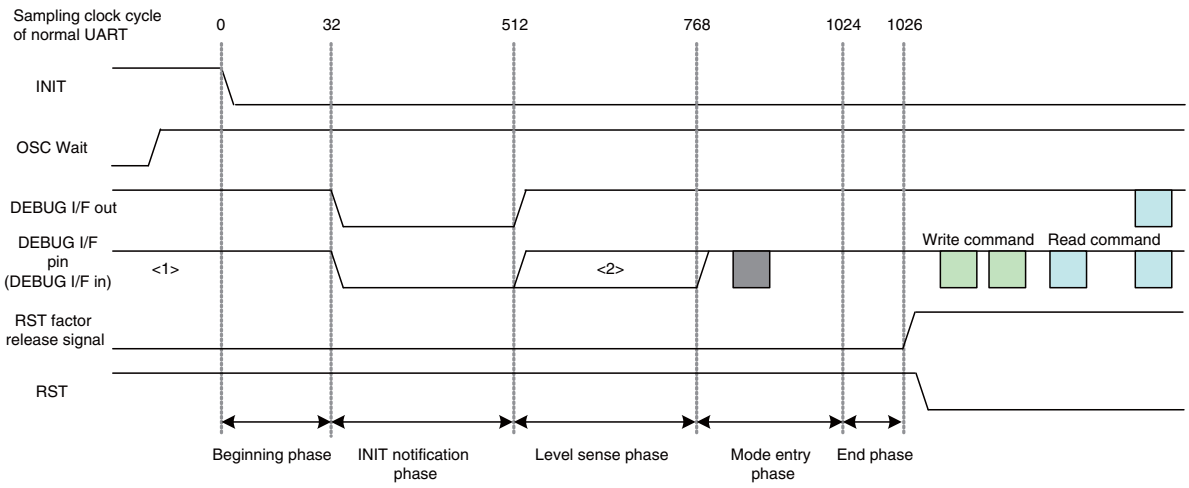
- Mode entry phase
  The period from level sense phase end to sampling clock 256 cycles of normal UART is the mode entry phase. OCDU begins the reception of the mode command from the tool from this phase.
  OCDU starts in the emulator mode (debugging state) when the reception beginning the mode command is detected by this phase (The start bit is detected by the UART reception). Afterwards, if normal mode command (no reception error and mode code agreement) is received, the acceptance of the register access command described later becomes possible after this as for OCDU. OCDU generates the INIT demand, and after releasing INIT, executes the chip reset sequence again when normal mode command (reception error or mode code disagreement) is not received.
  OCDU starts in the free orchis mode when the reception beginning the mode command is not detected by this phase (The start bit is detected by the UART reception).
  When you receive the mode command immediately after beginning of the mode entry phase, it is necessary to wait for that H is input more than the width of one cycle DEBUG I/F with the UART reception sampling clock, and to receive the mode command. When this is not satisfied, the start bit of the mode command reception cannot be normally detected. Therefore, there is a possibility that the mode cannot be correctly entered.

- End phase
  The period from mode entry phase end to sampling clock two cycles of normal UART is the end phase. OCDU does not do special operation by this phase. OCDU executes the reset publishing sequence described in "■ Reset (RST)" of the end of the end phase. The RST factor is released.

The relation between cycle number and the phase of the sampling clock of UART is usually as follows of the chip reset sequence.

Phase of chip reset sequence	Beginning phase	INIT notification phase	Level sense phase	Mode entry phase	End phase
Sampling clock cycle of normal UART from INIT release	1 - 32	33 - 512	513 - 768	769 - 1024	1025, 1026

The appearance of the chip reset sequence is shown as follows.



OSC Wait : Oscillation stability of the main clock. After the oscillation stability is confirmed, INIT is released.
<1> : DEBUG I/F becomes the H level by the pull-up of the tool.
<2> : DEBUG I/F becomes the level of the pull-up in the user system.

49.5.2.2 Security function

The security function is installed in OCDU. OCDU can make the security function effective by setting security information memorized in the debugging security area that exists in the memory space of CPU. It enters the state of the security lock when the security function is made effective. Security can be unlocked by writing the password set by security information in the specified length frequency and the E_SLPR register to release this.

Security information

The debugging security area is arranged in 30 bytes of first address +4 to +33 of installed Flash. Refer to this area by the security sequence for OCDU.

In the debugging security area, there is the following security information.

- Security password length (PW length)
  It is data of the length of 16bit at the first address in the debugging security area. It is length of PW to which lower 4bit is effective. Upper 12bit does not have the effect in operation. When the length of PW is 0x0 or 0xF, security is invalid. When the length of PW is 0x1 to 0xE (1 to 14), security is effective.

- Security password (PW)
  It is data of the length of 16bit in the debugging security area. There are 1 to 14 values in this data. It is sequentially arranged from the following address (PW1, PW2, ...PW14) of the address of the length of PW (Refer to the figure below). When security is effective (PW length: 1 to 14), PW to which the value of the length of PW is effective is shown.

  (For example, when the length of PW is eight, PW1 to PW8 are valid and PW9 to PW14 are invalid.

Address	15 ... 0
ROM/Flash first address +4	PW length
ROM/Flash first address +6	PW1
ROM/Flash first address +8	PW2
...	...
ROM/Flash first address +32	PW14

(Note) Do not write anything in this area, and keep the initial state immediately after the flash deletion (all bits=1) when you do not use the security function of on chip debugger (OCD).

49.5.3 Specification Restrictions at Connection to OCD Tool of This Product

The following restrictions are placed at OCD tool connection:

49.5.3.1 Clock Setting

- The main clock oscillation does not stop when the OCD tool is connected. However, reading to CMONR.MCRDY is available in the same way as when the OCD tool is not connected.
- A PLL oscillation does not stop in OCD high-speed UART and phase modulation UART communication. However, writing to CSELR.PCEN and reading to CMONR.PCRDY are available in the same way as when the OCD tool is not connected.
  PLL oscillation continues in OCD high-speed UART and phase modulation UART communication. For this reason, changes in the following PLL setting registers do not become effective. However, writing and reading to the registers are available in the same way as when the OCD tool is not connected.
  - PLLCR.PDS
  - CCPSDIVR.PODS
  - CCPLLFBR.IDIV

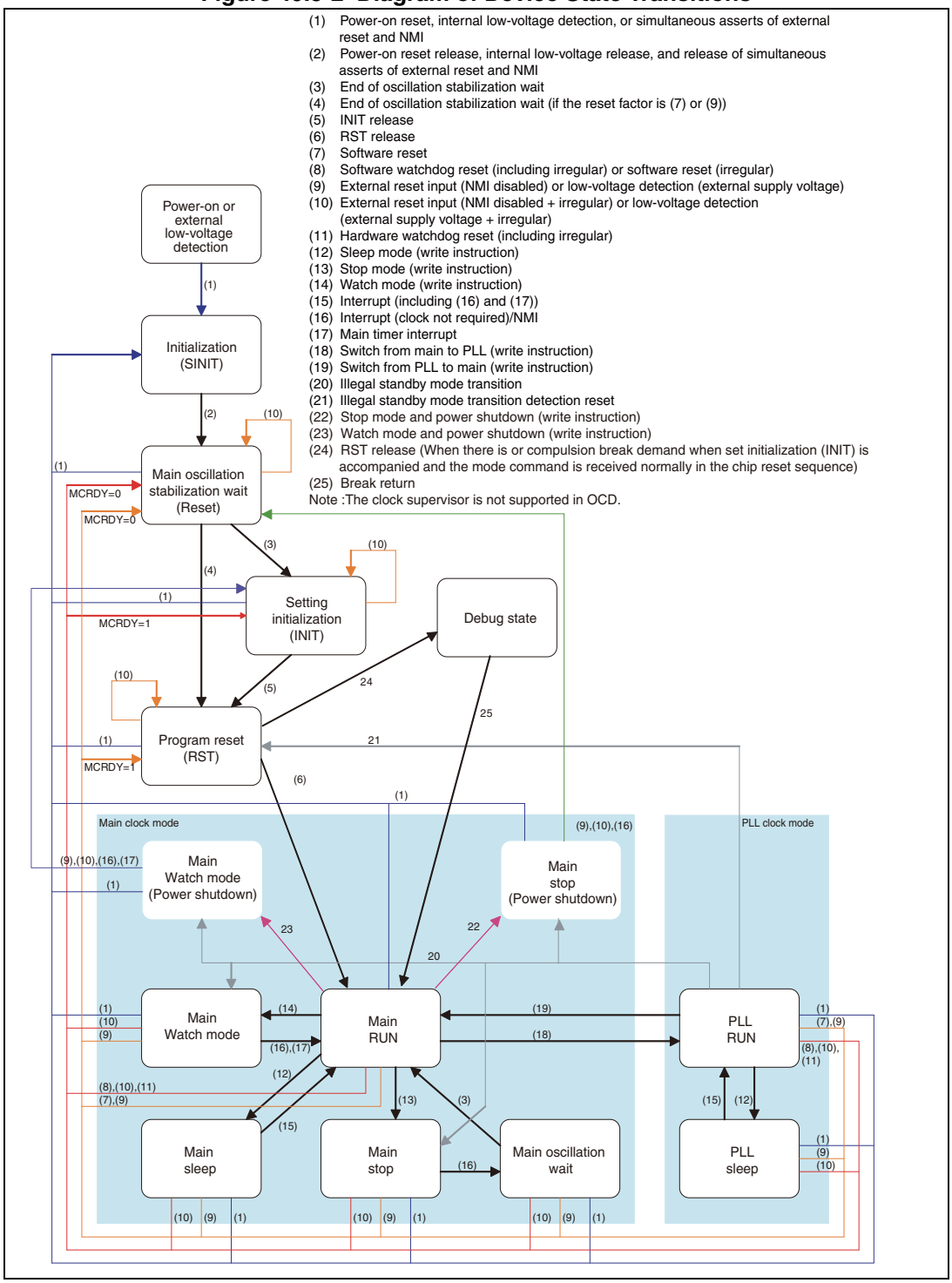
49.5.3.2 Standby Mode

- Even if the watch mode is entered, the PLL oscillation does not stop when the OCD high-speed UART and the phase modulation UART communication are enabled.
- The main clock oscillation does not stop even if the stop mode is entered. The PLL oscillation does not stop when the OCD high-speed UART and the phase modulation UART communication are enabled.
- The following shows the functions that differ in operation when the OCD tool is not connected, according to the above restrictions:
  - The CAN operation continues in the watch mode and stop mode when PLL is stopped down (CSELR. PCEN=0) or the OCD high-speed UART and the phase modulation UART communication are enabled. (This operation is performed within the range in which no CPU processing occurs.)
  - The counter operation for the WDT1 calibration continues in the stop mode.
- The following functions perform the same operations as those when the OCD tool is not connected, with the above restrictions not placed:
  - The main timer does not run in the stop mode because it is cleared in that mode.
- The power consumption in the watch mode becomes greater than that when the OCD tool is not connected because the PLL clock oscillation continues.
- The power consumption in the stop mode becomes greater than that when the OCD tool is not connected because the PLL clock and the main clock oscillation continue.

49.5.3.3 Clock Reset State Transitions

The device state transition when the OCD tool is connected is shown. The device state transition is shown as follows. This is the figure where the emulator mode of "Figure 49.5-1 OCDU Operating Mode Diagram" was made an aspect.

Figure 49.5-2 Diagram of Device State Transitions



(Note) At recovery from clock mode (power shutdown) and recovery from stop mode (power shutdown), some registers are not reset. For details, see "Restrictions on Power Shutdown/Normal Standby Control" in "CHAPTER 9 POWER CONSUMPTION CONTROL".

49.5.4 Summary of Restrictions

■ Communication mode*1: Normal UART

(Note) When debugging in standby mode (power shutdown), make sure that the OCD tool is not connected.

<table>
<tr><th rowspan="2">Reset reason</th><th colspan="2">Difference between OCD tool connected state and unconnected state</th><th rowspan="2">Remarks</th></tr>
<tr><th>Scope of initializa-tion</th><th>Processing time</th></tr>
<tr><td>Power-on reset</td><td rowspan="17">No</td><td rowspan="13">Yes</td><td rowspan="2">Transition to emulator mode (debug state) after reset cancellation</td></tr>
<tr><td>RSTX pin input (irregular)</td></tr>
<tr><td>RSTX pin input</td><td>No wait time for voltage reduction circuit switching stabilization*2
Note: Only for return from stop mode or watch mode
No wait time for main oscillation stabilization
Note: Only for return from stop mode</td></tr>
<tr><td>Watchdog reset 0 (irregular)</td><td rowspan="6">Transition to emulator mode (debug state) after reset cancellation</td></tr>
<tr><td>Watchdog reset 0</td></tr>
<tr><td>Watchdog reset 1 (irregular)</td></tr>
<tr><td>Watchdog reset 1</td></tr>
<tr><td>External low-voltage detection reset (irregular)</td></tr>
<tr><td>External low-voltage detection reset</td><td>No wait time for voltage reduction circuit switching stabilization*2
Note: Only for return from stop mode or watch mode</td></tr>
<tr><td>Reset by detection of transition to wrong standby mode (irregular)</td><td>Transition to emulator mode (debug state) after reset cancellation</td></tr>
<tr><td>Reset by detection of transition to wrong standby mode</td><td>No</td><td></td></tr>
<tr><td>Internal low-voltage detection reset</td><td rowspan="2">Yes</td><td rowspan="2">Transition to emulator mode (debug state) after reset cancellation</td></tr>
<tr><td>Flash security violation reset (irregular)</td></tr>
<tr><td>Flash security violation reset</td><td>No</td><td></td></tr>
<tr><td>Software reset (irregular)</td><td>Yes</td><td>Transition to emulator mode (debug state) after reset cancellation</td></tr>
<tr><td>Software reset</td><td>No</td><td></td></tr>
</table>

*1: For information on communication mode settings, see the SOFTUNE Workbench Operation Manual.

*2: The wait time for voltage reduction circuit stabilization is about 6 μs.

Interrupt cause	Difference in processing time between OCD tool connected state and unconnected state	Remarks
All interrupts	Yes	No wait time for voltage reduction circuit switching stabilization* Note: Only for return from stop mode or watch mode

*: The wait time for voltage reduction circuit stabilization is about 6 μs.

Device state unrelated to reset	Difference in operation between OCD tool connected state and unconnected state	Remarks
Main Run/Main Sleep mode	No	
PLL Run/PLL Sleep mode		
Stop mode	Yes	Voltage reduction circuit fixed Main oscillation continued Operation (WDT1 calibration counter operation) continued
Watch mode		Voltage reduction circuit fixed Main oscillation continued

■ Communication mode*1: High-speed UART/phase modulation UART

Reset reason	Difference between OCD tool connected state and unconnected state		Remarks
	Scope of initialization	Processing time	
Power-on reset	No	Yes	Transition to emulator mode (debug state) after reset cancellation
RSTX pin input (irregular)			
RSTX pin input			No wait time for voltage reduction circuit switching stabilization*2 Note: Only for return from stop mode or watch mode
Watchdog reset 0 (irregular)			Transition to emulator mode (debug state) after reset cancellation
Watchdog reset 0			
Watchdog reset 1 (irregular)			
Watchdog reset 1			
External low-voltage detection reset (irregular)			
External low-voltage detection reset			No wait time for voltage reduction circuit switching stabilization*2 Note: Only for return from stop mode or watch mode
Reset by detection of transition to wrong standby mode (irregular)		Yes	Transition to emulator mode (debug state) after reset cancellation
Reset by detection of transition to wrong standby mode		No	
Internal low-voltage detection reset		Yes	Transition to emulator mode (debug state) after reset cancellation
Flash security violation reset (irregular)		Yes	
Flash security violation reset		No	
Software reset (irregular)		Yes	Transition to emulator mode (debug state) after reset cancellation
Software reset		No	

*1: For information on communication mode settings, see the SOFTUNE Workbench Operation Manual.
*2: The wait time for voltage reduction circuit stabilization is about 6 μs.

Interrupt cause	Difference in processing time between OCD tool connected state and unconnected state	Remarks
All interrupts	Yes	No wait time for voltage reduction circuit switching stabilization* Note: Only for return from stop mode or watch mode

*: The wait time for voltage reduction circuit stabilization is about 6 μs.

Device state unrelated to reset	Difference in operation between OCD tool connected state and unconnected state	Remarks
Main Run/Main Sleep mode	Yes	PLL oscillation continued*
PLL Run/PLL Sleep mode	No	
Stop mode	Yes	Voltage reduction circuit fixed Main oscillation continued Operation (CAN and WDT1 calibration counter operation) continued
Watch mode		Voltage reduction circuit fixed Main oscillation continued PLL oscillation continued (detection of transition to wrong standby mode is disabled) Operation (CAN operation) continued

49.5.5 OCD-DSU ID Code of This Product Type

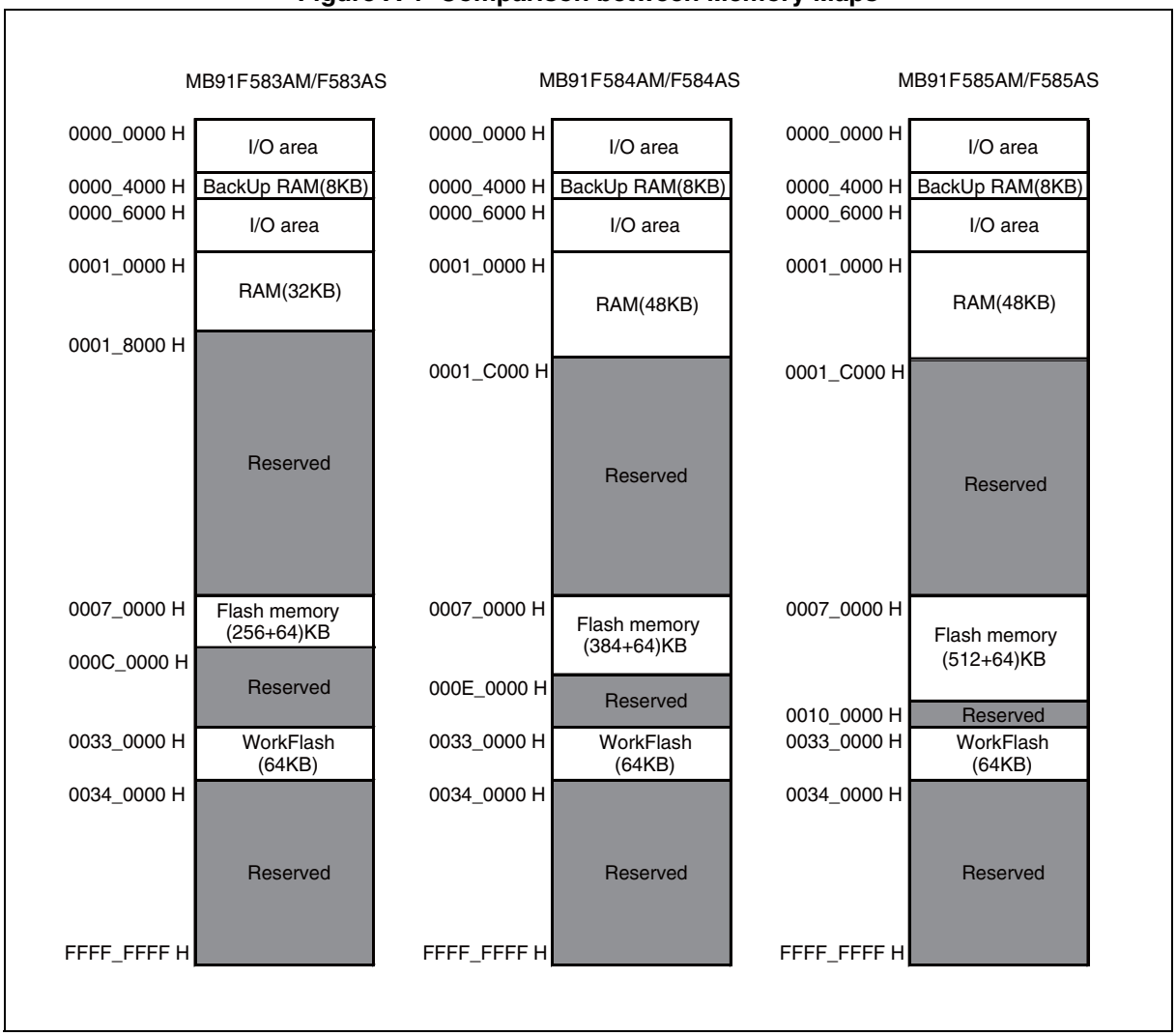
Table 49.5-1 OCD-DSU ID Code of This Product Type

ID name	Bit width	Associated ID register name	Address in the OCD space	Value	Remarks
Manufacturer ID	16	E_IDMCR	0x000	0x0400	
CPU family ID	16	E_IDFCR	0x001	0x0200	FR81E/FR81S
DSU type ID	8	E_IDVCR	0x003	0x06	
DSU version ID	4	E_IDVCR	0x003	0x2	
Device ID	16	E_IDDCR	0x002	0x001e	
Device version ID	4	E_IDVCR	0x003	0x1	

APPENDIX

APPENDIX A. Memory Map

Figure A-1 Comparison between Memory Maps



APPENDIX B. I/O Map

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (1 / 35)

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 -XXXXXXX	PDR06[R/W] B,H,W -XXXXXXX	PDR07[R/W] B,H,W -----XXX	
000008_H	PDR08[R/W] B,H,W XXXXXXXX	PDR09[R/W] B,H,W XXXXXXXX	PDR10[R/W] B,H,W -----XXX	-	
$00000C_H$	-	-	-	-	
000010_H | 000038_H	-	-	-	-	Reserved
$00003C_H$	WDTCR0[R/W] B,H,W -0--0000	WDTCPR0[W] B,H,W 00000000	WDTCR1[R] B,H,W ----0010	WDTCPR1[W] B,H,W 00000000	Watchdog timer [S]
000040_H	-	-	-	-	Reserved
000044_H	DICR[R/W] B -------0	-	-	-	Delay interrupt
000048_H | $00005C_H$	-	-	-	-	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 | $00007C_H$	-	-	-	-	Reserved
000080_H	BT0TMR[R] H 00000000 00000000		BT0TMCR[R/W] H -0000000 00000000		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		
$00008C_H$	-	-	-	-	
000090_H	BT1TMR[R] H 00000000 00000000		BT1TMCR[R/W] H -0000000 00000000		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
$0000A0_H$ | $0000FC_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (2 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000100_H	TMRLRA1[R/W] H XXXXXXXX XXXXXXXX		TMR1[R] H XXXXXXXX XXXXXXXX		Reload timer 1
000104_H	TMRLRB1[R/W] H XXXXXXXX XXXXXXXX		TMCSR1[R/W] B,H,W 00000000 0-000000		
000108_H	TMRLRA2[R/W] H XXXXXXXX XXXXXXXX		TMR2[R] H XXXXXXXX XXXXXXXX		Reload timer 2
$00010C_H$	TMRLRB2[R/W] H XXXXXXXX XXXXXXXX		TMCSR2[R/W] B,H,W 00000000 0-000000		
000110_H	TMRLRA3[R/W] H XXXXXXXX XXXXXXXX		TMR3[R] H XXXXXXXX XXXXXXXX		Reload timer 3
000114_H	TMRLRB3[R/W] H XXXXXXXX XXXXXXXX		TMCSR3[R/W] B,H,W 00000000 0-000000		
000118_H | $0011C_H$	-	-	-	-	Reserved
000120_H	IRPR0H[R] B,H,W 00------	IRPR0L[R] B,H,W 00------	IRPR1H[R] B,H,W 00------	IRPR1L[R] B,H,W --------	Interrupt request batch read register
000124_H	IRPR2H[R] B,H,W --------	IRPR2L[R] B,H,W *5 0000----	IRPR3H[R] B,H,W 00------	IRPR3L[R] B,H,W 00------	
000128_H	IRPR4H[R] B,H,W 00------	IRPR4L[R] B,H,W 000000--	IRPR5H[R] B,H,W 00------	IRPR5L[R] B,H,W 00------	
$00012C_H$	IRPR6H[R] B,H,W 0000----	IRPR6L[R] B,H,W 00------	IRPR7H[R] B,H,W 00------	IRPR7L[R] B,H,W --------	
000130_H	IRPR8H[R] B,H,W --------	IRPR8L[R] B,H,W 00------	IRPR9H[R] B,H,W 00------	IRPR9L[R] B,H,W 00------	
000134_H	IRPR10H[R] B,H,W 00------	IRPR10L[R] B,H,W 00------	IRPR11H[R] B,H,W 00------	IRPR11L[R] B,H,W 0000000-	
000138_H	IRPR12H[R] B,H,W 0000000-	IRPR12L[R] B,H,W 00000000	IRPR13H[R] B,H,W 0000000-	IRPR13L[R] B,H,W 00000000	
$00013C_H$	IRPR14H[R] B,H,W 00------	IRPR14L[R] B,H,W 00------	IRPR15H[R] B,H,W 00000000	IRPR15L[R] B,H,W 0000----	
000140_H	IRPR16H[R] B,H,W 00------	IRPR16L[R] B,H,W 00------	IRPR17H[R] B,H,W 00------	IRPR17L[R]B,H,W --------	
000144_H	IRPR18H[R]B,H,W --------	IRPR18L[R]B,H,W 000000--	-	-	
000148_H | $0001FC_H$	-	-	-	-	Reserved
000200_H	PCN0[R/W] B,H,W 00000000 000000-0		PCSR0[W] H,W XXXXXXXX XXXXXXXX		PPG0
000204_H	PDUT0[W] H,W XXXXXXXX XXXXXXXX		PTMR0[R] H,W 11111111 11111111		
000208_H	PCN1[R/W] B,H,W 00000000 000000-0		PCSR1[W] H,W XXXXXXXX XXXXXXXX		PPG1
$00020C_H$	PDUT1[W] H,W XXXXXXXX XXXXXXXX		PTMR1[R] H,W 11111111 11111111		

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (3 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000210_H	PCN2[R/W] B,H,W 00000000 000000-0		PCSR2[W] H,W XXXXXXXX XXXXXXXX		PPG2
000214_H	PDUT2[W] H,W XXXXXXXX XXXXXXXX		PTMR2[R] H,W 11111111 11111111		
000218_H	PCN3[R/W] B,H,W 00000000 000000-0		PCSR3[W] H,W XXXXXXXX XXXXXXXX		PPG3
$00021C_H$	PDUT3[W] H,W XXXXXXXX XXXXXXXX		PTMR3[R] H,W 11111111 11111111		
000220_H	PCN4[R/W] B,H,W 00000000 000000-0		PCSR4[W] H,W XXXXXXXX XXXXXXXX		PPG4
000224_H	PDUT4[W] H,W XXXXXXXX XXXXXXXX		PTMR4[R] H,W 11111111 11111111		
000228_H	PCN5[R/W] B,H,W 00000000 000000-0		PCSR5[W] H,W XXXXXXXX XXXXXXXX		PPG5
$00022C_H$	PDUT5[W] H,W XXXXXXXX XXXXXXXX		PTMR5[R] H,W 11111111 11111111		
000230_H | $0002BC_H$	-	-	-	-	Reserved
$0002C0_H$	GTRS0[R/W] B,H,W -0000000 -0000000		GTRS1[R/W] B,H,W -0000000 -0000000		PPG Control
$0002C4_H$	GTRS2[R/W] B,H,W -0000000 -0000000		-		
$0002C8_H$	-		-		
$0002CC_H$	-		-		
$0002D0_H$	-		-		
$0002D4_H$	-		-		
$0002D8_H$	GTREN0[R/W] H,W -------- --000000		-		
$0002DC_H$	-		-		Reserved
$0002E0_H$	-	GATEC0[R/W] B,H,W ------00	-	GATEC2[R/W] B,H,W ------00	PPG GATE Control
$0002E4_H$	-	GATEC4[R/W] B,H,W ------00	-	-	
$0002E8_H$	-	-	-	-	
$0002EC_H$	-	-	-	-	Reserved
$0002F0_H$	RCRH0[W] H,W 00000000	RCRL0[W] B,H,W 00000000	UDCRH0[R] H,W 00000000	UDCRL0[R] B,H,W 00000000	U/D counter 0
$0002F4_H$	CCR0[R/W] B,H 00000000 -0001000		-	CSR0[R] B 00000000	
$0002F8_H$	RCRH1[W] H,W 00000000	RCRL1[W] B,H,W 00000000	UDCRH1[R] H,W 00000000	UDCRL1[R] B,H,W 00000000	U/D counter 1
$0002FC_H$	CCR1[R/W] B,H 00000000 -0001000		-	CSR1[R] B 00000000	
000300_H	-				Reserved
000304_H	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (4 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000308_H	-				Reserved
$00030C_H$	-	-	-	-	
000310_H	-	-	MPUCR[R/W] H 000000-0 ----0100		MPU [S] (Only the CPU can access this area)
000314_H	-	-	-	-	
000318_H	-				
$00031C_H$	-	-	-	-	
000320_H	DPVAR[R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000324_H	-	-	DPVSR[R/W] H -------- 00000--0		
000328_H	DEAR[R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00032C_H$	-	-	DESR[R/W] H -------- 00000--0		
000330_H	PABR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000334_H	-	-	PACR0[R/W] H 000000-0 00000--0		
000338_H	PABR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00033C_H$	-	-	PACR1[R/W] H 000000-0 00000--0		
000340_H	PABR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000344_H	-	-	PACR2[R/W] H 000000-0 00000--0		
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		

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (5 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000370_H	-				Reserved [S]
000374_H	-	-	-		
000378_H	-				
$00037C_H$	-	-	-		
000380_H	-				
000384_H	-	-	-		
000388_H	-				
$00038C_H$	-	-	-		
000390_H	-				
000394_H	-	-	-		Reserved [S]
000398_H	-				
$00039C_H$	-	-	-		
$0003A0_H$	-				
$0003A4_H$	-	-	-		
$0003A8_H$	-				
$0003AC_H$	-	-	-		
$0003B0_H$ | $0003CC_H$	-	-	-	-	Reserved [S]
$0003D0_H$	-				Reserved [S]
$0003D4_H$	-				
$0003D8_H$	-				
$0003DC_H$	-				
$0003E0_H$ | $0003FC_H$	-	-	-	-	Reserved [S]

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (6 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000400_H	ICSEL0[R/W] B,H,W -----000	ICSEL1[R/W] B,H,W -------0	ICSEL2[R/W] B,H,W -------0	ICSEL3[R/W] B,H,W -------0	Generation and clear of DMA transfer request
000404_H	ICSEL4[R/W] B,H,W -------0	ICSEL5[R/W] B,H,W -------0	ICSEL6[R/W] B,H,W -------0	ICSEL7[R/W] B,H,W -----000	
000408_H	ICSEL8[R/W] B,H,W -------0	ICSEL9[R/W] B,H,W -------0	ICSEL10[R/W] B,H,W ------00	ICSEL11[R/W] B,H,W -------0	
$00040C_H$	ICSEL12[R/W] B,H,W -------0	ICSEL13[R/W] B,H,W -------0	ICSEL14[R/W] B,H,W -------0	ICSEL15[R/W] B,H,W -------0	
000410_H	ICSEL16[R/W] B,H,W -------0	ICSEL17[R/W] B,H,W -------0	ICSEL18[R/W] B,H,W -------0	ICSEL19[R/W] B,H,W -------0	
000414_H	ICSEL20[R/W] B,H,W -------0	ICSEL21[R/W] B,H,W -----000	ICSEL22[R/W] B,H,W -----000	ICSEL23[R/W] B,H,W -----000	
000418_H	ICSEL24[R/W] B,H,W -----000	ICSEL25[R/W] B,H,W -----000	ICSEL26[R/W] B,H,W -------0	ICSEL27[R/W] B,H,W -------0	
$00041C_H$	-	-	-	-	
000420_H	-	-	-	-	
000424_H | $00043C_H$	-	-	-	-	Reserved
000440_H	ICR00[R/W] B,H,W ---11111	ICR01[R/W] B,H,W ---11111	ICR02[R/W] B,H,W ---11111	ICR03[R/W] B,H,W ---11111	Interrupt controller [S]
000444_H	ICR04[R/W] B,H,W ---11111	ICR05[R/W] B,H,W ---11111	ICR06[R/W] B,H,W ---11111	ICR07[R/W] B,H,W ---11111	
000448_H	ICR08[R/W] B,H,W ---11111	ICR09[R/W] B,H,W ---11111	ICR10[R/W] B,H,W ---11111	ICR11[R/W] B,H,W ---11111	
$00044C_H$	ICR12[R/W] B,H,W ---11111	ICR13[R/W] B,H,W ---11111	ICR14[R/W] B,H,W ---11111	ICR15[R/W] B,H,W ---11111	
000450_H	ICR16[R/W] B,H,W ---11111	ICR17[R/W] B,H,W ---11111	ICR18[R/W] B,H,W ---11111	ICR19[R/W] B,H,W ---11111	
000454_H	ICR20[R/W] B,H,W ---11111	ICR21[R/W] B,H,W ---11111	ICR22[R/W] B,H,W ---11111	ICR23[R/W] B,H,W ---11111	
000458_H	ICR24[R/W] B,H,W ---11111	ICR25[R/W] B,H,W ---11111	ICR26[R/W] B,H,W ---11111	ICR27[R/W] B,H,W ---11111	
$00045C_H$	ICR28[R/W] B,H,W ---11111	ICR29[R/W] B,H,W ---11111	ICR30[R/W] B,H,W ---11111	ICR31[R/W] B,H,W ---11111	
000460_H	ICR32[R/W] B,H,W ---11111	ICR33[R/W] B,H,W ---11111	ICR34[R/W] B,H,W ---11111	ICR35[R/W] B,H,W ---11111	
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	
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 | $00047C_H$	-	-	-	-	Reserved [S]

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (7 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
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 consumption control [S] * Writing to STBCR by DMA is disabled.
000484_H	-	-	-	-	Reserved [S]
000488_H	DIVR0[R/W] B,H,W 000-----	-	DIVR2[R/W] B,H,W 0011----	-	Clock control [S]
$00048C_H$	-	-	-	-	Reserved [S]
000490_H	IORR0[R/W] B,H,W -0000000	IORR1[R/W] B,H,W -0000000	IORR2[R/W] B,H,W -0000000	IORR3[R/W] B,H,W -0000000	DMA transfer request from a peripheral [S]
000494_H	IORR4[R/W] B,H,W -0000000	IORR5[R/W] B,H,W -0000000	IORR6[R/W] B,H,W -0000000	IORR7[R/W] B,H,W -0000000	
000498_H	-	-	-	-	
$00049C_H$	-	-	-	-	
$0004A0_H$	-	-	-	-	Reserved
$0004A4_H$	CANPRE[R/W] B,H,W ---00000	-	-	-	CAN prescaler
$0004A8_H$ | $0004AC_H$	-	-	-	-	Reserved
$0004B0_H$	-	-	-	-	Reserved
$0004B4_H$ | $0004C0_H$	-	-	-	-	Reserved
$0004C4_H$	CUCR1[R/W] B,H,W -------- ---0--00		CUTD1[R/W] B,H,W 11000011 01010000		WDT1 calibration
$0004C8_H$	CUTR1[R] B,H,W -------- 00000000 00000000 00000000				
$0004CC_H$ | $0004DC_H$	-	-	-	-	Reserved
$0004E0_H$	-	-	CSCFG[R/W] B,H,W ---0----	CMCFG[R/W] B,H,W 00000000	Clock monitor
$0004E4_H$	-	-	-	-	
$0004E8_H$	PLL2DIVM[R/W] B,H,W ----0000	PLL2DIVN[R/W] B,H,W -0000000	PLL2DIVG[R/W] B,H,W ----0000	PLL2MULG[R/W] B,H,W 00000000	FlexRay clock control *5
$0004EC_H$	PLL2CTRL[R/W] B,H,W ----0000	PLL2DIVK[R/W] B,H,W -------0	CLKR2[R/W] B,H,W 000--000	-	
$0004F0_H$ | $0004FC_H$	-	-	-	-	Reserved
000500_H	-				Reserved
000504_H	-				Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (8 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000508_H | $00050C_H$	-	-	-	-	Reserved
000510_H	CSELR[R/W] B,H,W -0----00	CMONR[R] B,H,W -01---00	MTMCR[R/W] B,H,W 00001111	-	Clock control [S]
000514_H	PLLCR[R/W] B,H,W 00-00000 11110000		CSTBR[R/W] B,H,W ----0000	PTMCR[R/W] B,H,W 00------	
000518_H	-	-	CPUAR[R/W] B,H,W 0---XXXX	-	Reset [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
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] B,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	-	-	CCPMUCR0[R/W] B,H,W 0-----00	CCPMUCR1[R/W] B,H,W 0--00000	
000534_H	-	-	-	-	
000538_H	-	-	-	-	
$00053C_H$	-	-	-	-	
000540_H | $00054C_H$	-	-	-	-	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 | 000568_H	-	-	-	-	Reserved
$00056C_H$	-	CSVCR[R/W] B -0--1--0	-	-	CSV
000570_H	CRTR[R/W] B,H,W 01111111	-	-	-	WDT1 calibration (trimming)
000574_H | $00057C_H$	-	-	-	-	Reserved
000580_H	REGSEL[R/W] B,H,W 01--110-	-	-	-	Regulator control
000584_H	LVD5R[R/W] B,H,W -------1	LVD5F[R/W] B,H,W 001100-1	LVD[R/W] B,H,W 01000--0	-	Low-voltage detection

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (9 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000588_H | $00058C_H$	-	-	-	-	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	-	PMUINTF1[R/W] B,H,W 00000000	PMUINTF2[R/W] B,H,W -00-----	-	
000598_H	-	-	-	-	
$00059C_H$	-	-	-	-	
$0005A0_H$ | $0005FC_H$	-	-	-	-	Reserved
000600_H | $00060C_H$	-	-	-	-	Reserved [S]
000610_H | $00063C_H$	-	-	-	-	Reserved [S]
000640_H | $00064C_H$	-	-	-	-	Reserved [S]
000650_H | $00067C_H$	-	-	-	-	Reserved [S]
000680_H | $00068C_H$	-	-	-	-	Reserved [S]
000690_H | $0006BC_H$	-	-	-	-	Reserved [S]
$0006C0_H$ | $0006CC_H$	-	-	-	-	Reserved [S]
$0006D0_H$ | $0006F0_H$	-	-	-	-	Reserved
$0006F4_H$	-				Reserved
$0006F8_H$ | $0006FC_H$	-	-	-	-	Reserved
000700_H	-				Reserved
000704_H | $00070C_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (10 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
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 | $0007F8_H$	-	-	-	-	Reserved
$0007FC_H$	BMODR[R] B,H,W XXXXXXXX	-	-	-	Operation mode
000800_H | $00083C_H$	-	-	-	-	Reserved [S]
000840_H	FCTLR[R/W] H -0--1000 0--0-----		-	FSTR[R/W] B -----001	Flash memory register [S]
000844_H	-	-	-	-	Reserved [S]
000848_H | 000854_H	-	-	-	-	Reserved [S]
000858_H	-	-	WREN[R/W] H 00000000 00000000		Wild register [S]
$00085C_H$ | $00087C_H$	-	-	-	-	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				
000898_H	WRAR03[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$00089C_H$	WRDR03[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008A0_H$	WRAR04[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008A4_H$	WRDR04[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (11 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0008A8_H$	WRAR05[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register [S]
$0008AC_H$	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				
$0008F0_H$	WRAR14[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008F4_H$	WRDR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008F8_H$	WRAR15[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008FC_H$	WRDR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000900_H | $000BF8_H$	-	-	-	-	Reserved
$000BFC_H$	-		UER[W] B,H,W -------- -------X		OCDU

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (12 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000C00_H$	DCCR0[R/W] W 0----000 --00--00 00000000 0-000000				DMA controller [S]
$000C04_H$	DCSR0[R/W] H 0------- -----000		DTCR0[R/W] H 00000000 00000000		
$000C08_H$	DSAR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C0C_H$	DDAR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C10_H$	DCCR1[R/W] W 0----000 --00--00 00000000 0-000000				
$000C14_H$	DCSR1[R/W] H 0------- -----000		DTCR1[R/W] H 00000000 00000000		
$000C18_H$	DSAR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C1C_H$	DDAR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C20_H$	DCCR2[R/W] W 0----000 --00--00 00000000 0-000000				
$000C24_H$	DCSR2[R/W] H 0------- -----000		DTCR2[R/W] H 00000000 00000000		
$000C28_H$	DSAR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C2C_H$	DDAR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C30_H$	DCCR3[R/W] W 0----000 --00--00 00000000 0-000000				
$000C34_H$	DCSR3[R/W] H 0------- -----000		DTCR3[R/W] H 00000000 00000000		
$000C38_H$	DSAR3[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C3C_H$	DDAR3[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C40_H$	DCCR4[R/W] W 0----000 --00--00 00000000 0-000000				
$000C44_H$	DCSR4[R/W] H 0------- -----000		DTCR4[R/W] H 00000000 00000000		
$000C48_H$	DSAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C4C_H$	DDAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C50_H$	DCCR5[R/W] W 0----000 --00--00 00000000 0-000000				
$000C54_H$	DCSR5[R/W] H 0------- -----000		DTCR5[R/W] H 00000000 00000000		
$000C58_H$	DSAR5[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (13 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000C5C_H$	DDAR5[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA controller [S]
$000C60_H$	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				
$000C80_H$ | $000DF0_H$	-	-	-	-	
$000DF4_H$	-	-	DNMIR[R/W] B 0------0	DILVR[R/W] B ---11111	
$000DF8_H$	DMACR[R/W] W 0------- -------- 0------- --------				
$000DFC_H$	-	-	-	-	Reserved [S]
$000E00_H$	DDR00[R/W] B,H 00000000	DDR01[R/W] B,H 00000000	DDR02[R/W] B,H 00000000	DDR03[R/W] B,H 00000000	Data direction register
$000E04_H$	DDR04[R/W] B,H 00000000	DDR05[R/W] B,H -0000000	DDR06[R/W] B,H -0000000	DDR07[R/W] B,H -----000	
$000E08_H$	DDR08[R/W] B,H 00000000	DDR09[R/W] B,H 00000000	DDR10[R/W] B,H -----000	-	
$000E0C_H$	-	-	-	-	
$000E10_H$ | $000E1C_H$	-	-	-	-	Reserved
$000E20_H$	PFR00[R/W] B,H 00000000	PFR01[R/W] B,H 00000000	PFR02[R/W] B,H 00000000	PFR03[R/W] B,H 00000000	Port function register
$000E24_H$	PFR04[R/W] B,H 00000000	PFR05[R/W] B,H -0000000	PFR06[R/W] B,H -0000000	PFR07[R/W] B,H -----000	
$000E28_H$	PFR08[R/W] B,H 00000000	PFR09[R/W] B,H 00000000	PFR10[R/W] B,H -----000	-	
$000E2C_H$	-	-	-	-	
$000E30_H$ | $000E3C_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (14 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000E40_H$	PDDR00[R] B,H,W XXXXXXXX	PDDR01[R] B,H,W XXXXXXXX	PDDR02[R] B,H,W XXXXXXXX	PDDR03[R] B,H,W XXXXXXXX	Input data direct read register
$000E44_H$	PDDR04[R] B,H,W XXXXXXXX	PDDR05[R] B,H,W -XXXXXXX	PDDR06[R] B,H,W -XXXXXXX	PDDR07[R] B,H,W -----XXX	
$000E48_H$	PDDR08[R] B,H,W XXXXXXXX	PDDR09[R] B,H,W XXXXXXXX	PDDR10[R] B,H,W -----XXX	-	
$000E4C_H$	-	-	-	-	
$000E50_H$ | $000E5C_H$	-	-	-	-	Reserved
$000E60_H$	EPFR00[R/W] B,H ------00	EPFR01[R/W] B,H ---00000	EPFR02[R/W] B,H --000000	EPFR03[R/W] B,H --000000	Extended port function register
$000E64_H$	-	-	EPFR06[R/W] B,H ------00	EPFR07[R/W] B,H ----0000	
$000E68_H$	EPFR08[R/W] B,H *5 ----0000	EPFR09[R/W] B,H -------0	EPFR10[R/W] B,H -0000000	-	
$000E6C_H$	-	-	EPFR14[R/W] B,H --0-0-0-	-	
$000E70_H$	-	-	-	-	
$000E74_H$	-	-	-	-	
$000E78_H$	-	-	-	-	
$000E7C_H$	-	-	-	-	
$000E80_H$	-	-	-	-	
$000E84_H$ | $000EBC_H$	-	-	-	-	Reserved
$000EC0_H$	PPER00[R/W] B,H 00000000	PPER01[R/W] B,H 00000000	PPER02[R/W] B,H 00000000	PPER03[R/W] B,H 00000000	Port pull-up/down enable register
$000EC4_H$	PPER04[R/W] B,H 00000000	PPER05[R/W] B,H -0000000	PPER06[R/W] B,H -0000000	PPER07[R/W] B,H -----000	
$000EC8_H$	PPER08[R/W] B,H 00000000	PPER09[R/W] B,H 00000000	PPER10[R/W] B,H -----000	-	
$000ECC_H$	-	-	-	-	
$000ED0_H$ | $000EDC_H$	-	-	-	-	Reserved
$000EE0_H$	PILR00[R/W] B,H 11111111	PILR01[R/W] B,H 11111111	PILR02[R/W] B,H 11111111	PILR03[R/W] B,H 11111111	Port input level selection register
$000EE4_H$	PILR04[R/W] B,H 11111111	PILR05[R/W] B,H -1111111	PILR06[R/W] B,H -1111111	PILR07[R/W] B,H -----111	
$000EE8_H$	PILR08[R/W] B,H 11111111	PILR09[R/W] B,H 11111111	PILR10[R/W] B,H -----111	-	
$000EEC_H$	-	-	-	-	
$000EF0_H$ | $000EFC_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (15 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000F00_H$ | $000F1C_H$	-	-	-	-	Reserved
$000F20_H$	PODR00[R/W] B,H 00000000	PODR01[R/W] B,H 00000000	PODR02[R/W] B,H 00000000	PODR03[R/W] B,H 00000000	Port output drive register
$000F24_H$	PODR04[R/W] B,H 00000000	PODR05[R/W] B,H -0000000	PODR06[R/W] B,H -0000000	PODR07[R/W] B,H -----000	
$000F28_H$	PODR08[R/W] B,H 00000000	PODR09[R/W] B,H 00000000	PODR10[R/W] B,H -----000	-	
$000F2C_H$	-	-	-	-	
$000F30_H$ | $000F3C_H$	-	-	-	-	Reserved
$000F40_H$	PORTEN[R/W] B,H,W ------00	-	-	-	Port input enable register
$000F44_H$	KEYCDR[R/W] H 00000000 00000000		-	-	Port key code
$000F48_H$	ADERH[R/W] B,H -------- 11111111		ADERL[R/W] B,H -1111111 11111111		Analog input enable register
$000F4C_H$	DAER[R/W] B,H -------0	-	-	-	Analog output enable register
$000F50_H$ | $000FFF_H$	-	-	-	-	Reserved
001000_H	SACR[R/W] B,H,W -------0	PICD[R/W] B,H,W ----0011	-	-	Synchronous/ asynchronous switch control
001004_H | $0010BC_H$	-	-	-	-	Reserved
$0010C0_H$	-	-	-	CRCCR[R/W] B,H,W -0000000	CRC arithmetic operation 0
$0010C4_H$	CRCINIT[R/W] B,H,W 11111111 11111111 11111111 11111111				
$0010C8_H$	CRCIN[R/W] B,H,W 00000000 00000000 00000000 00000000				
$0010CC_H$	CRCR[R] B,H,W 11111111 11111111 11111111 11111111				
$0010D0_H$	-	-	-	CRCCR1[R/W] B,H,W -0000000	CRC arithmetic operation 1
$0010D4_H$	CRCINIT1[R/W] B,H,W 11111111 11111111 11111111 11111111				
$0010D8_H$	CRCIN1[R/W] B,H,W 00000000 00000000 00000000 00000000				
$0010DC_H$	CRCR1[R] B,H,W 11111111 11111111 11111111 11111111				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (16 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0010E0_H$ | $0010FC_H$	-	-	-	-	Reserved
001100_H	TCGS[R/W] B,H,W ------00	-	-	TCGSE[R/W] B,H,W --000000	Free-run timer simultaneous activation
001104_H	CPCLRB0/CPCLR0[R/W] H,W 11111111 11111111		TCDT0[R/W] H,W 00000000 00000000		Free-run timer 0
001108_H	TCCS0[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00110C_H$	CPCLRB1/CPCLR1[R/W] H,W 11111111 11111111		TCDT1[R/W] H,W 00000000 00000000		Free-run timer 1
001110_H	TCCS1[R/W] B,H,W 00000000 01000000 ----0000 --------				
001114_H	CPCLRB2/CPCLR2[R/W] H,W 11111111 11111111		TCDT2[R/W] H,W 00000000 00000000		Free-run timer 2
001118_H	TCCS2[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00111C_H$	CPCLRB3/CPCLR3[R/W] H,W 11111111 11111111		TCDT3[R/W] H,W 00000000 00000000		Free-run timer 3
001120_H	TCCS3[R/W] B,H,W 00000000 01000000 ----0000 --------				
001124_H	CPCLRB4/CPCLR4[R/W] H,W 11111111 11111111		TCDT4[R/W] H,W 00000000 00000000		Free-run timer 4
001128_H	TCCS4[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00112C_H$	CPCLRB5/CPCLR5[R/W] H,W 11111111 11111111		TCDT5[R/W] H,W 00000000 00000000		Free-run timer 5
001130_H	TCCS5[R/W] B,H,W 00000000 01000000 ----0000 --------				
001134_H	FRS0[R/W] B,H,W -------- -000-000 -000-000 -000-000				Free-run timer selection
001138_H	FRS1[R/W] B,H,W -------- -------- -000-000 -000-000				
$00113C_H$	FRS2[R/W] B,H,W -------- -000-000 -000-000 -000-000				
001140_H	-				
001144_H	FRS4[R/W] B,H,W -000-000 -000-000 -000-000 -000-000				
001148_H	FRS5[R/W] B,H,W -----000 -000-000 -000-000 -000-000				
$00114C_H$	FRS6[R/W] B,H,W -000-000 -000-000 -000-000 -000-000				
001150_H	-				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (17 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001154_H	OCCPB0/OCCP0[R/W] H,W 00000000 00000000		OCCPB1/OCCP1[R/W] H,W 00000000 00000000		Output compare 0/1
001158_H	OCS01[R/W] B,H,W -110--00 00001100		-	OCMOD01[R/W] B,H,W ------00	
$00115C_H$	OCCPB2/OCCP2[R/W] H,W 00000000 00000000		OCCPB3/OCCP3[R/W] H,W 00000000 00000000		Output compare 2/3
001160_H	OCS23[R/W] B,H,W -110--00 00001100		-	OCMOD23[R/W] B,H,W ------00	
001164_H	OCCPB4/OCCP4[R/W] H,W 00000000 00000000		OCCPB5/OCCP5[R/W] H,W 00000000 00000000		Output compare 4/5
001168_H	OCS45[R/W] B,H,W -110--00 00001100		-	OCMOD45[R/W] B,H,W ------00	
$00116C_H$	OCCPB6/OCCP6[R/W] H,W 00000000 00000000		OCCPB7/OCCP7[R/W] H,W 00000000 00000000		Output compare 6/7
001170_H	OCS67[R/W] B,H,W -110--00 00001100		-	OCMOD67[R/W] B,H,W ------00	
001174_H	OCCPB8/OCCP8[R/W] H,W 00000000 00000000		OCCPB9/OCCP9[R/W] H,W 00000000 00000000		Output compare 8/9
001178_H	OCS89[R/W] B,H,W -110--00 00001100		-	OCMOD89[R/W] B,H,W ------00	
$00117C_H$	OCCPB10/OCCP10[R/W] H,W 00000000 00000000		OCCPB11/OCCP11[R/W] H,W 00000000 00000000		Output compare 10/11
001180_H	OCS1011[R/W] B,H,W -110--00 00001100		-	OCMOD1011[R/W] B,H,W ------00	
001184_H	IPCP0[R] H,W 00000000 00000000		IPCP1[R] H,W 00000000 00000000		Input capture 0/1
001188_H	ICS01[R/W] B,H,W ------00 00000000		-	LSYNS[R/W] B,H,W ----0000	
$00118C_H$	IPCP2[R] H,W 00000000 00000000		IPCP3[R] H,W 00000000 00000000		Input capture 2/3
001190_H	ICS23[R/W] B,H,W ------00 00000000		-	-	
001194_H	-		-		Reserved
001198_H	-		-	-	
$00119C_H$	-		-		Reserved
$0011A0_H$	-		-	-	
$0011A4_H$	DTSR[R/W] B,H,W ------10	-	-	-	DTTI selection

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (18 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0011A8_H$	TMRR0[R/W] H,W 00000000 00000001		TMRR1[R/W] H,W 00000000 00000001		Waveform generator 0/1/2
$0011AC_H$	TMRR2[R/W] H,W 00000000 00000001		-	-	
$0011B0_H$	DTSCR0[R/W] B,H,W 00000000	DTSCR1[R/W] B,H,W 00000000	DTSCR2[R/W] B,H,W 00000000	-	
$0011B4_H$	-	DTIR0[R/W] B,H,W 000000--	-	DTMNS0[R/W] B,H,W 00---000	
$0011B8_H$	-	SIGCR10[R/W] B,H,W 00000000	-	SIGCR20[R/W] B,H,W 000000-1	
$0011BC_H$	PICS0[R/W] B,H,W 000000-- -------- -------- --------				
$0011C0_H$	TMRR3[R/W] H,W 00000000 00000001		TMRR4[R/W] H,W 00000000 00000001		Waveform generator 3/4/5
$0011C4_H$	TMRR5[R/W] H,W 00000000 00000001		-	-	
$0011C8_H$	DTSCR3[R/W] B,H,W 00000000	DTSCR4[R/W] B,H,W 00000000	DTSCR5[R/W] B,H,W 00000000	-	
$0011CC_H$	-	DTIR1[R/W] B,H,W 000000--	-	DTMNS1[R/W] B,H,W 00---000	
$0011D0_H$	-	SIGCR11[R/W] B,H,W 00000000	-	SIGCR21[R/W] B,H,W -------1	
$0011D4_H$	-				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (19 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0011D8_H$	-	-	-	-	12-bit A/D converter
$0011DC_H$	ADTSS[R/W] B,H,W -------0	-	-	-	
$0011E0_H$	ADTSE[R/W] B,H,W -------- 00000000 -0000000 00000000				
$0011E4_H$	ADCOMP0/ADCOMPB0[R/W] H,W 00000000 00000000		ADCOMP1/ADCOMPB1[R/W] H,W 00000000 00000000		
$0011E8_H$	ADCOMP2/ADCOMPB2[R/W] H,W 00000000 00000000		ADCOMP3/ADCOMPB3[R/W] H,W 00000000 00000000		
$0011EC_H$	ADCOMP4/ADCOMPB4[R/W] H,W 00000000 00000000		ADCOMP5/ADCOMPB5[R/W] H,W 00000000 00000000		
$0011F0_H$	ADCOMP6/ADCOMPB6[R/W] H,W 00000000 00000000		ADCOMP7/ADCOMPB7[R/W] H,W 00000000 00000000		
$0011F4_H$	ADCOMP8/ADCOMPB8[R/W] H,W 00000000 00000000		ADCOMP9/ADCOMPB9[R/W] H,W 00000000 00000000		
$0011F8_H$	ADCOMP10/ADCOMPB10[R/W] H,W 00000000 00000000		ADCOMP11/ADCOMPB11[R/W] H,W 00000000 00000000		
$0011FC_H$	ADCOMP12/ADCOMPB12[R/W] H,W 00000000 00000000		ADCOMP13/ADCOMPB13[R/W] H,W 00000000 00000000		
001200_H	ADCOMP14/ADCOMPB14[R/W] H,W 00000000 00000000		-		
001204_H	ADCOMP16/ADCOMPB16[R/W] H,W 00000000 00000000		ADCOMP17/ADCOMPB17[R/W] H,W 00000000 00000000		
001208_H	ADCOMP18/ADCOMPB18[R/W] H,W 00000000 00000000		ADCOMP19/ADCOMPB19[R/W] H,W 00000000 00000000		
$00120C_H$	ADCOMP20/ADCOMPB20[R/W] H,W 00000000 00000000		ADCOMP21/ADCOMPB21[R/W] H,W 00000000 00000000		
001210_H	ADCOMP22/ADCOMPB22[R/W] H,W 00000000 00000000		ADCOMP23/ADCOMPB23[R/W] H,W 00000000 00000000		
001214_H	-	-	-	-	
001218_H	-	-	-	-	
$00121C_H$	-	-	-	-	
001220_H	-	-	-	-	
001224_H	ADTCS0[R/W] B,H,W 00000000 0010-000		ADTCS1[R/W] B,H,W 00000000 0010-000		
001228_H	ADTCS2[R/W] B,H,W 00000000 0010-000		ADTCS3[R/W] B,H,W 00000000 0010-000		
$00122C_H$	ADTCS4[R/W] B,H,W 00000000 0010-000		ADTCS5[R/W] B,H,W 00000000 0010-000		
001230_H	ADTCS6[R/W] B,H,W 00000000 0010-000		ADTCS7[R/W] B,H,W 00000000 0010-000		
001234_H	ADTCS8[R/W] B,H,W 00000000 0010-000		ADTCS9[R/W] B,H,W 00000000 0010-000		
001238_H	ADTCS10[R/W] B,H,W 00000000 0010-000		ADTCS11[R/W] B,H,W 00000000 0010-000		
$00123C_H$	ADTCS12[R/W] B,H,W 00000000 0010-000		ADTCS13[R/W] B,H,W 00000000 0010-000		

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (20 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001240_H	ADTCS14[R/W] B,H,W 00000000 0010-000		-		12-bit A/D converter
001244_H	ADTCS16[R/W] B,H,W 00000000 00100000		ADTCS17[R/W] B,H,W 00000000 00100000		
001248_H	ADTCS18[R/W] B,H,W 00000000 00100000		ADTCS19[R/W] B,H,W 00000000 00100000		
$00124C_H$	ADTCS20[R/W] B,H,W 00000000 00100000		ADTCS21[R/W] B,H,W 00000000 00100000		
001250_H	ADTCS22[R/W] B,H,W 00000000 00100000		ADTCS23[R/W] B,H,W 00000000 00100000		
001254_H	-	-	-	-	
001258_H	-	-	-	-	
$00125C_H$	-	-	-	-	
001260_H	-	-	-	-	
001264_H	ADTCD0[R] B,H,W 10--0000 00000000		ADTCD1[R] B,H,W 10--0000 00000000		
001268_H	ADTCD2[R] B,H,W 10--0000 00000000		ADTCD3[R] B,H,W 10--0000 00000000		
$00126C_H$	ADTCD4[R] B,H,W 10--0000 00000000		ADTCD5[R] B,H,W 10--0000 00000000		
001270_H	ADTCD6[R] B,H,W 10--0000 00000000		ADTCD7[R] B,H,W 10--0000 00000000		
001274_H	ADTCD8[R] B,H,W 10--0000 00000000		ADTCD9[R] B,H,W 10--0000 00000000		
001278_H	ADTCD10[R] B,H,W 10--0000 00000000		ADTCD11[R] B,H,W 10--0000 00000000		
$00127C_H$	ADTCD12[R] B,H,W 10--0000 00000000		ADTCD13[R] B,H,W 10--0000 00000000		
001280_H	ADTCD14[R] B,H,W 10--0000 00000000		-		
001284_H	ADTCD16[R] B,H,W 10--0000 00000000		ADTCD17[R] B,H,W 10--0000 00000000		
001288_H	ADTCD18[R] B,H,W 10--0000 00000000		ADTCD19[R] B,H,W 10--0000 00000000		
$00128C_H$	ADTCD20[R] B,H,W 10--0000 00000000		ADTCD21[R] B,H,W 10--0000 00000000		
001290_H	ADTCD22[R] B,H,W 10--0000 00000000		ADTCD23[R] B,H,W 10--0000 00000000		
001294_H	-	-	-	-	
001298_H	-	-	-	-	
$00129C_H$	-	-	-	-	
$0012A0_H$	-	-	-	-	
$0012A4_H$	ADCS0[R/W] B,H,W 0------- --------		ADCH0[R] B,H,W -----000	ADMD0[R/W] B,H,W ----0000	
$0012A8_H$	ADCS1[R/W] B,H,W 0------- --------		ADCH1[R] B,H,W -----000	ADMD1[R/W] B,H,W ----0000	

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (21 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0012AC_H$	ADCS2[R/W] B,H,W 0------- --------		ADCH2[R] B,H,W -----000	ADMD2[R/W] B,H,W ----0000	12-bit A/D converter
$0012B0_H$	MTRCSR[R/W] B,H,W -------0				Motor control extension function
$0012B4_H$	RTOSEL0[R/W] B,H,W --000000	RTOSEL1[R/W] B,H,W -------0			
$0012B8_H$ | $0012FC_H$	-	-	-	-	Reserved
001300_H	-	-	-	-	Reserved
001304_H	-	-	-	-	
001308_H	-	-	-	-	
$00130C_H$	-	-	-	-	
001310_H	-	-	-	-	
001314_H	-	-	-	-	
001318_H	-	-	-	-	
$00131C_H$	-	-	-	-	
001320_H	-	-	-	-	
001324_H	-		-		
001328_H | $00132C_H$	-	-	-	-	
001330_H	-		-		
001334_H | $0013FC_H$	-	-	-	-	Reserved
001400_H	DACR[R/W] B,H,W -------0	-	DADR[R/W] H,W ------XX XXXXXXXX		DAC
001404_H | $0014FC_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (22 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001500_H	SCR0/(IBCR0)[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000000-0	SSR0[R/W] B,H,W 0--00011	ESCR0/(IBSR0)[R/W] B,H,W 00000000	Multi Function Serial I/F 0 *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
001504_H	-/(RDR10/(TDR10))[R/W] H,W -------- -------- *3		RDR00/(TDR00)[R/W] B,H,W -------0 00000000 *1		
001508_H	SACSR0[R/W] B,H,W 0----000 00000000		STMR0[R] B,H,W 00000000 00000000		
$00150C_H$	STMCR0[R/W] B,H,W 00000000 00000000		-/(SFUR0) [R/W] B,H,W -------- -------- *3 *4		
001510_H	-	-	-/(SFLR10) [R/W] B,H,W -------- *3 *4	-/(SFLR00) [R/W] B,H,W -------- *3 *4	
001514_H	-	-	-	-	
001518_H	-	-	-	-	
$00151C_H$	BGR0[R/W] H,W 00000000 00000000		-/(ISMK0)[R/W] B,H,W -------- *2	-/(ISBA0)[R/W] B,H,W -------- *2	
001520_H	FCR10[R/W] B,H,W 00-00100	FCR00[R/W] B,H,W -0000000	FBYTE20[R/W] B,H,W 00000000	FBYTE10[R/W] B,H,W 00000000	
001524_H	SCR1[R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000000-0	SSR1[R/W] B,H,W 0--00011	ESCR1[R/W] B,H,W 00000000	Multi Function Serial I/F 1 *1 Byte access is possible only for access to lower 8 bits. *3 Reserved because CSIO mode is not set immediately after reset *4 Reserved because LIN2.1 mode is not set immediately after reset
001528_H	-/(RDR11/(TDR11))[R/W] H,W -------- -------- *3		RDR01/(TDR01)[R/W] B,H,W -------0 00000000 *1		
$00152C_H$	SACSR1[R/W] B,H,W 0----000 00000000		STMR1[R] B,H,W 00000000 00000000		
001530_H	STMCR1[R/W] B,H,W 00000000 00000000		-/(SCSCR1/SFUR1) [R/W] B,H,W -------- -------- *3 *4		
001534_H	-/(SCSTR31) [R/W] B,H,W -------- *3	-/(SCSTR21) [R/W] B,H,W -------- *3	-/SCSTR11/SFLR11) [R/W] B,H,W -------- *3 *4	-/(SCSTR01/SFLR01) [R/W] B,H,W -------- *3 *4	
001538_H	-	-	-	-	
$00153C_H$	-	-	-	TBYTE01[R/W] B,H,W 00000000	
001540_H	BGR1[R/W] H,W 00000000 00000000		-	-	
001544_H	FCR11[R/W] B,H,W 00-00100	FCR01[R/W] B,H,W -0000000	FBYTE21[R/W] B,H,W 00000000	FBYTE11[R/W] B,H,W 00000000	

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (23 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001548_H	SCR2/(IBCR2)[R/W] B,H,W 0--00000	SMR2[R/W] B,H,W 000000-0	SSR2[R/W] B,H,W 0--00011	ESCR2/(IBSR2)[R/W] B,H,W 00000000	Multi Function Serial I/F 2 *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
$00154C_H$	-/(RDR12/(TDR12))[R/W] H,W -------- -------- *3		RDR02/(TDR02)[R/W] B,H,W -------0 00000000 *1		
001550_H	SACSR2[R/W] B,H,W 0----000 00000000		STMR2[R] B,H,W 00000000 00000000		
001554_H	STMCR2[R/W] B,H,W 00000000 00000000		-/(SCSCR2/SFUR2) [R/W] B,H,W -------- -------- *3 *4		
001558_H	-/(SCSTR32) [R/W] B,H,W -------- *3	-/(SCSTR22) [R/W] B,H,W -------- *3	-/(SCSTR12/SFLR12) [R/W] B,H,W -------- *3 *4	-/(SCSTR02/SFLR02) [R/W] B,H,W -------- *3 *4	
$00155C_H$	-	-	-	-	
001560_H	-	-	-	TBYTE02[R/W] B,H,W 00000000	
001564_H	BGR2[R/W] H,W 00000000 00000000		-/(ISMK2)[R/W] B,H,W -------- *2	-/(ISBA2)[R/W] B,H,W -------- *2	
001568_H	FCR12[R/W] B,H,W 00-00100	FCR02[R/W] B,H,W -0000000	FBYTE22[R/W] B,H,W 00000000	FBYTE12[R/W] B,H,W 00000000	
$00156C_H$	SCR3/(IBCR3)[R/W] B,H,W 0--00000	SMR3[R/W] B,H,W 000000-0	SSR3[R/W] B,H,W 0--00011	ESCR3/(IBSR3)[R/W] B,H,W 00000000	Multi Function Serial I/F 3 *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
001570_H	-/(RDR13/(TDR13))[R/W] H,W -------- -------- *3		RDR03/(TDR03)[R/W] B,H,W -------0 00000000 *1		
001574_H	SACSR3[R/W] B,H,W 0----000 00000000		STMR3[R] B,H,W 00000000 00000000		
001578_H	STMCR3[R/W] B,H,W 00000000 00000000		-/(SCSCR3/SFUR3) [R/W] B,H,W -------- -------- *3 *4		
$00157C_H$	-/(SCSTR33) [R/W] B,H,W -------- *3	-/(SCSTR23) [R/W] B,H,W -------- *3	-/(SCSTR13/SFLR13) [R/W] B,H,W -------- *3 *4	-/(SCSTR03/SFLR03) [R/W] B,H,W -------- *3 *4	
001580_H	-	-	-	-	
001584_H	-	-	-	TBYTE03[R/W] B,H,W 00000000	
001588_H	BGR3[R/W] H,W 00000000 00000000		-/(ISMK3)[R/W] B,H,W -------- *2	-/(ISBA3)[R/W] B,H,W -------- *2	
$00158C_H$	FCR13[R/W] B,H,W 00-00100	FCR03[R/W] B,H,W -0000000	FBYTE23[R/W] B,H,W 00000000	FBYTE13[R/W] B,H,W 00000000	
001590_H | $001FFC_H$	-	-	-	-	Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (24 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002000_H	CTRLR0[R/W] B,H,W -------- 000-0001		STATR0[R/W] B,H,W -------- 00000000		CAN 0 64msb
002004_H	ERRCNT0 [R] B,H,W 00000000 00000000		BTR0[R/W] B,H,W -0100011 00000001		
002008_H	INTR0[R] B,H,W 00000000 00000000		TESTR0[R/W] B,H,W -------- X00000--		
$00200C_H$	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)				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (25 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002068_H | $00207C_H$	-		-		CAN 0 64msb
002080_H	TREQR20[R] B,H,W 00000000 00000000		TREQR10[R] B,H,W 00000000 00000000		
002084_H	TREQR40[R] B,H,W 00000000 00000000		TREQR30[R] B,H,W 00000000 00000000		
002088_H	-		-		
$00208C_H$	-		-		
002090_H	NEWDT20[R] B,H,W 00000000 00000000		NEWDT10[R] B,H,W 00000000 00000000		
002094_H	NEWDT40[R] B,H,W 00000000 00000000		NEWDT30[R] B,H,W 00000000 00000000		
002098_H	-		-		
$00209C_H$	-		-		
$0020A0_H$	INTPND20[R] B,H,W 00000000 00000000		INTPND10[R] B,H,W 00000000 00000000		
$0020A4_H$	INTPND40[R] B,H,W 00000000 00000000		INTPND30[R] B,H,W 00000000 00000000		
$0020A8_H$	-		-		
$0020AC_H$	-		-		
$0020B0_H$	MSGVAL20[R] B,H,W 00000000 00000000		MSGVAL10[R] B,H,W 00000000 00000000		
$0020B4_H$	MSGVAL40[R] B,H,W 00000000 00000000		MSGVAL30[R] B,H,W 00000000 00000000		
$0020B8_H$	-		-		
$0020BC_H$	-		-		
$0020C0_H$ | $0020FC_H$	-		-		

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (26 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002100_H	CTRLR1[R/W] B,H,W -------- 000-0001		STATR1[R/W] B,H,W -------- 00000000		CAN 1 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		
002114_H	IF1MSK21[R/W] B,H,W 11-11111 11111111		IF1MSK11[R/W] B,H,W 11111111 11111111		
002118_H	IF1ARB21[R/W] B,H,W 00000000 00000000		IF1ARB11[R/W] B,H,W 00000000 00000000		
$00211C_H$	IF1MCTR1[R/W] B,H,W 00000000 0---0000		-		
002120_H	IF1DTA11[R/W] B,H,W 00000000 00000000		IF1DTA21[R/W] B,H,W 00000000 00000000		
002124_H	IF1DTB11[R/W] B,H,W 00000000 00000000		IF1DTB21[R/W] B,H,W 00000000 00000000		
002128_H, $00212C_H$	-		-		CAN 1 64msb
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 | $00217C_H$	-		-		
002180_H	TREQR21[R] B,H,W 00000000 00000000		TREQR11[R] B,H,W 00000000 00000000		

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (27 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002184_H	TREQR41[R] B,H,W 00000000 00000000		TREQR31[R] B,H,W 00000000 00000000		CAN 1 64msb
002188_H	-		-		
$00218C_H$	-		-		
002190_H	NEWDT21[R] B,H,W 00000000 00000000		NEWDT11[R] B,H,W 00000000 00000000		
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		
$0021B8_H$	-		-		
$0021BC_H$	-		-		
$0021C0_H$ | $0021FC_H$	-		-		
002200_H | $0022FC_H$	-		-		Reserved
002300_H	DFCTLR[R/W] B,H,W -0------ --------		-	DFSTR[R/W] B,H,W -----001	WorkFlash
002304_H	-	-	-	-	
002308_H	FLIFCTLR[R/W] B,H,W ---0--00	-	FLIFFER1[R/W] B,H,W --------	FLIFFER2[R/W] B,H,W --------	
$00230C_H$ | $002FFC_H$	-	-	-	-	Reserved
003000_H	SEEARX[R] B,H,W --000000 00000000		DEEARX[R] B,H,W --000000 00000000		XBS RAM ECC control register
003004_H	EECSRX[R/W] B,H,W ----00-0	-	EFEARX[R/W] B,H,W --000000 00000000		
003008_H	-	EFECRX[R/W] B,H,W -------0 00000000 00000000			

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (28 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00300C_H$	TEAR0X[R] B,H,W 000----- -------- --000000 00000000				XBS RAM diagnosis register
003010_H	TEAR1X[R] B,H,W 000----- -------- --000000 00000000				
003014_H	TEAR2X[R] B,H,W 000----- -------- --000000 00000000				
003018_H	TAEARX[R/W] B,H,W --101111 11111111		TASARX[R/W] B,H,W --000000 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[R/W] B,H,W 0-------	-	-	TKCCRX[R/W] B,H,W 00----00	
003024_H	SEEARA[R] B,H,W --000000 00000000		DEEARA[R] B,H,W --000000 00000000		Backup RAM ECC control register
003028_H	EECSRA[R/W] B,H,W ----00-0	-	EFEARA[R/W] B,H,W --000000 00000000		
$00302C_H$	-	EFECRA[R/W] B,H,W -------0 00000000 00000000			
003030_H	TEAR0A[R] B,H,W 000----- -------- -----000 00000000				Backup RAM diagnosis register
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		
003044_H	TSRCRA[R/W] B,H,W 0-------	-	-	TKCCRA[R/W] B,H,W 00----00	
003048_H | $0030FC_H$	-	-	-	-	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				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (29 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
003110_H	BUSADR2[R] W 00000000 00000000 00000000 00000000				Bus diagnosis
003114_H	-		BUSDIGSR3[R/W] H,W 00000000 0-----00		
003118_H	BUSDIGSR4[R/W] H,W 00000000 0-----00		BUSTSTR1[R/W] H,W 00--0000 00000000		
$00311C_H$	-				
003120_H	BUSADR3[R] W 00000000 00000000 00000000 00000000				
003124_H	BUSADR4[R] W 00000000 00000000 00000000 00000000				
003128_H | $003FFC_H$	-	-	-	-	Reserved
004000_H | $005FFC_H$	Backup RAM				Backup RAM area
006000_H | $00CFFC_H$	-	-	-	-	Reserved
$00D000_H$	CIF0[R] W 00000100 11111111 01011011 11111111				FlexRay CIF *5
$00D004_H$	CIF1[R/W] W 00000000 -------0 -0000000 --------				
$00D008_H$ | $00D00C_H$	-	-	-	-	Reserved
$00D010_H$	-				FlexRay GIF *5
$00D014_H$	-				
$00D018_H$	-	-	-	-	
$00D01C_H$	LCK[R/W] W -------- -------- -------- 00000000				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (30 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D020_H$	EIR[R/W] W -----000 -----000 ----0000 00000000				FlexRay INT *5
$00D024_H$	SIR[R/W] W ------00 ------00 00000000 00000000				
$00D028_H$	EILS[R/W] W -----000 -----000 ----0000 00000000				
$00D02C_H$	SILS[R/W] W ------11 ------11 11111111 11111111				
$00D030_H$	EIES[R/W] W -----000 -----000 ----0000 00000000				
$00D034_H$	EIER[R/W] W -----000 -----000 ----0000 00000000				
$00D038_H$	SIES[R/W] W ------00 ------00 00000000 00000000				
$00D03C_H$	SIER[R/W] W ------00 ------00 00000000 00000000				
$00D040_H$	ILE[R/W] W -------- -------- -------- ------00				
$00D044_H$	T0C[R/W] W --000000 00000000 -0000000 ------00				
$00D048_H$	T1C[R/W] W --000000 00000010 -------- ------00				
$00D04C_H$	STPW1[R/W] W --000000 00000000 --000000 -0000000				
$00D050_H$	STPW2[R] W -----000 00000000 -----000 00000000				
$00D054_H$ | $00D07C_H$	-	-	-	-	Reserved
$00D080_H$	SUCC1[R/W] W ----1100 01000000 00010-00 1---0000				FlexRay SUC *5
$00D084_H$	SUCC2[R/W] W ----0001 ---00000 00000101 00000100				
$00D088_H$	SUCC3[R/W] W -------- -------- -------- 00010001				
$00D08C_H$	NEMC[R/W] W -------- -------- -------- ----0000				FlexRay NEM *5
$00D090_H$	PRTC1[R/W] W 000010-0 01001100 0000-110 00110011				FlexRay PRT *5
$00D094_H$	PRTC2[R/W] W --001111 00101101 --001010 --001110				
$00D098_H$	MHDC[R/W] W ---00000 00000000 -------- -0000000				FlexRay MHD *5
$00D09C_H$	-				Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (31 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D0A0_H$	GTUC1[R/W] W -------- ----0000 00000010 10000000				FlexRay GTU *5
$00D0A4_H$	GTUC2[R/W] W -------- ----0010 --000000 00001010				
$00D0A8_H$	GTUC3[R/W] W -0000010 -0000010 00000000 00000000				
$00D0AC_H$	GTUC4[R/W] W --000000 00001000 --000000 00000111				
$00D0B0_H$	GTUC5[R/W] W 00001110 ---00000 00000000 00000000				
$00D0B4_H$	GTUC6[R/W] W -----000 00000010 -----000 00000000				
$00D0B8_H$	GTUC7[R/W] W ------00 00000010 ------00 00000100				
$00D0BC_H$	GTUC8[R/W] W ---00000 00000000 -------- --000010				
$00D0C0_H$	GTUC9[R/W] W -------- ------00 ---00001 --000001				
$00D0C4_H$	GTUC10[R/W] W -----000 00000010 --000000 00000101				
$00D0C8_H$	GTUC11[R/W] W -----000 -----000 ------00 ------00				
$00D0CC_H$ | $00D0FC_H$	-				Reserved
$00D100_H$	CCSV[R] W --000000 00010000 -100--00 00000000				FlexRay SUC *5
$00D104_H$	CCEV[R] W -------- -------- ---00000 00--0000				
$00D108_H$ $00D10C_H$	-				Reserved
$00D110_H$	SCV[R] W -----000 00000000 -----000 00000000				FlexRay GTU *5
$00D114_H$	MTCCV[R] W -------- --000000 --000000 00000000				
$00D118_H$	RCV[R] W -------- -------- ----0000 00000000				
$00D11C_H$	OCV[R] W -------- -----000 00000000 00000000				
$00D120_H$	SFS[R] W -------- ----0000 00000000 00000000				
$00D124_H$	SWNIT[R] W -------- -------- ----0000 00000000				
$00D128_H$	ACS[R/W] W -------- -------- ---00000 ---00000				
$00D12C_H$	-				
$00D130_H$	ESID1[R] W -------- -------- 00----00 00000000				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (32 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D134_H$	ESID2[R] W -------- -------- 00----00 00000000				FlexRay GTU *5
$00D138_H$	ESID3[R] W -------- -------- 00----00 00000000				
$00D13C_H$	ESID4[R] W -------- -------- 00----00 00000000				
$00D140_H$	ESID5[R] W -------- -------- 00----00 00000000				
$00D144_H$	ESID6[R] W -------- -------- 00----00 00000000				
$00D148_H$	ESID7[R] W -------- -------- 00----00 00000000				
$00D14C_H$	ESID8[R] W -------- -------- 00----00 00000000				
$00D150_H$	ESID9[R] W -------- -------- 00----00 00000000				
$00D154_H$	ESID10[R] W -------- -------- 00----00 00000000				
$00D158_H$	ESID11[R] W -------- -------- 00----00 00000000				
$00D15C_H$	ESID12[R] W -------- -------- 00----00 00000000				
$00D160_H$	ESID13[R] W -------- -------- 00----00 00000000				
$00D164_H$	ESID14[R] W -------- -------- 00----00 00000000				
$00D168_H$	ESID15[R] W -------- -------- 00----00 00000000				
$00D16C_H$	-				
$00D170_H$	OSID1[R] W -------- -------- 00----00 00000000				
$00D174_H$	OSID2[R] W -------- -------- 00----00 00000000				
$00D178_H$	OSID3[R] W -------- -------- 00----00 00000000				
$00D17C_H$	OSID4[R] W -------- -------- 00----00 00000000				
$00D180_H$	OSID5[R] W -------- -------- 00----00 00000000				
$00D184_H$	OSID6[R] W -------- -------- 00----00 00000000				
$00D188_H$	OSID7[R] W -------- -------- 00----00 00000000				
$00D18C_H$	OSID8[R] W -------- -------- 00----00 00000000				
$00D190_H$	OSID9[R] W -------- -------- 00----00 00000000				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (33 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D194_H$	OSID10[R] W -------- -------- 00----00 00000000				FlexRay GTU *5
$00D198_H$	OSID11[R] W -------- -------- 00----00 00000000				
$00D19C_H$	OSID12[R] W -------- -------- 00----00 00000000				
$00D1A0_H$	OSID13[R] W -------- -------- 00----00 00000000				
$00D1A4_H$	OSID14[R] W -------- -------- 00----00 00000000				
$00D1A8_H$	OSID15[R] W -------- -------- 00----00 00000000				
$00D1AC_H$	-				Reserved
$00D1B0_H$	NMV1[R] W 00000000 00000000 00000000 00000000				FlexRay NEM *5
$00D1B4_H$	NMV2[R] W 00000000 00000000 00000000 00000000				
$00D1B8_H$	NMV3[R] W 00000000 00000000 00000000 00000000				
$00D1BC_H$ | $00D2FC_H$	-				Reserved
$00D300_H$	MRC[R/W] W -----001 10000000 00000000 00000000				FlexRay MHD *5
$00D304_H$	FRF[R/W] W -------1 10000000 ---00000 00000000				
$00D308_H$	FRFM[R/W] W -------- -------- ---00000 000000--				
$00D30C_H$	FCL[R/W] W -------- -------- -------- 10000000				
$00D310_H$	MHDS[R/W] W -0000000 -0000000 -0000000 00000000				
$00D314_H$	LDTS[R] W -----000 00000000 -----000 00000000				
$00D318_H$	FSR[R] W -------- -------- 00000000 -----000				
$00D31C_H$	MHDF[R/W] W -------- -------- -------0 00000000				
$00D320_H$	TXRQ1[R] W 00000000 00000000 00000000 00000000				
$00D324_H$	TXRQ2[R] W 00000000 00000000 00000000 00000000				
$00D328_H$	TXRQ3[R] W 00000000 00000000 00000000 00000000				
$00D32C_H$	TXRQ4[R] W 00000000 00000000 00000000 00000000				
$00D330_H$	NDAT1[R] W 00000000 00000000 00000000 00000000				

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (34 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D334_H$	NDAT2[R] W 00000000 00000000 00000000 00000000				FlexRay MHD *5
$00D338_H$	NDAT3[R] W 00000000 00000000 00000000 00000000				
$00D33C_H$	NDAT4[R] W 00000000 00000000 00000000 00000000				
$00D340_H$	MBSC1[R] W 00000000 00000000 00000000 00000000				
$00D344_H$	MBSC2[R] W 00000000 00000000 00000000 00000000				
$00D348_H$	MBSC3[R] W 00000000 00000000 00000000 00000000				
$00D34C_H$	MBSC4[R] W 00000000 00000000 00000000 00000000				
$00D350_H$ | $00D3EC_H$	-				Reserved
$00D3F0_H$	CREL[R] W 00010000 00111001 00000010 00000110				FlexRay GIF *5
$00D3F4_H$	ENDN[R] W 10000111 01100101 01000011 00100001				
$00D3F8_H$ | $00D3FC_H$	-				Reserved
$00D400_H$ | $00D4FC_H$	WRDSn[1-64][R/W] W 00000000 00000000 00000000 00000000				FlexRay IBF *5
$00D500_H$	WRHS1[R/W] W --000000 -0000000 -----000 00000000				
$00D504_H$	WRHS2[R/W] W -------- -0000000 -----000 00000000				
$00D508_H$	WRHS3[R/W] W -------- -------- -----000 00000000				
$00D50C_H$	-				
$00D510_H$	IBCM[R/W] W -------- ------00 -------- -----000				
$00D514_H$	IBCR[R/W] W 0------- -0000000 0------- -0000000				
$00D518_H$ | $00D5FC_H$	-				Reserved

Table B-1 I/O Map (MB91F583AM/F584AM/F585AM) (35 / 35)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D600_H$ | $00D6FC_H$	RDDSn[1-64][R] W 00000000 00000000 00000000 00000000				FlexRay OBF *5
$00D700_H$	RDHS1[R] W --000000 -0000000 -----000 00000000				
$00D704_H$	RDHS2[R] W -0000000 -0000000 -----000 00000000				
$00D708_H$	RDHS3[R] W --000000 --000000 -----000 00000000				
$00D70C_H$	MBS[R] W --000000 --000000 00-00000 00000000				
$00D710_H$	OBCM[R/W] W -------- ------00 -------- ------00				
$00D714_H$	OBCR[R/W] W -------- -0000000 0-----00 -0000000				
$00D718_H$ | $00D7FC_H$	-				Reserved
$00D800_H$ | $00EFFC_H$	-				Reserved
$00F000_H$ | $00FEFC_H$	-				Reserved [S]
$00FF00_H$	DSUCR[R/W] B,H,W -------- -------0		-	-	OCDU [S]
$00FF04_H$ | $00FF0C_H$	-	-	-	-	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				
$00FF18_H$ | $00FFF4_H$	-	-	-	-	Reserved [S]
$00FFF8_H$	EDIR1[R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FFFC_H$	EDIR0[R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

[S]: It is a system register. The illegal instruction exception (data access error) is generated when reading and writing these registers in the user mode.

*5: For FlexRay, the MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ has corresponding functions.

The following registers are reserved registers for models without the FlexRay function.

000125_H IRPR2L[5:4], $000E68_H$, $0004E8$-$0004EF_H$, $00D000$-$00D717_H$

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (1 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000000_H	-	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 -XXXXXXX	-	PDR07[R/W] B,H,W -----XXX	
000008_H	-	PDR09[R/W] B,H,W ---XX---	-	-	
$00000C_H$	-	-	-	-	
000010_H | 000038_H	-	-	-	-	Reserved
$00003C_H$	WDTCR0[R/W] B,H,W -0--0000	WDTCPR0[W] B,H,W 00000000	WDTCR1[R] B,H,W ----0010	WDTCPR1[W] B,H,W 00000000	Watchdog timer [S]
000040_H	-	-	-		Reserved
000044_H	DICR[R/W] B -------0	-	-	-	Delay interrupt
000048_H | $00005C_H$	-		-		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 | $00007C_H$	-	-	-	-	Reserved
000080_H	BT0TMR[R] H 00000000 00000000		BT0TMCR[R/W] H -0000000 00000000		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		
$00008C_H$	-	-	-	-	
000090_H	BT1TMR[R] H 00000000 00000000		BT1TMCR[R/W] H -0000000 00000000		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
$0000A0_H$ | $0000FC_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (2 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000100_H	TMRLRA1[R/W] H XXXXXXXX XXXXXXXX		TMR1[R] H XXXXXXXX XXXXXXXX		Reload timer 1
000104_H	TMRLRB1[R/W] H XXXXXXXX XXXXXXXX		TMCSR1[R/W] B,H,W 00000000 0-000000		
000108_H	TMRLRA2[R/W] H XXXXXXXX XXXXXXXX		TMR2[R] H XXXXXXXX XXXXXXXX		Reload timer 2
$00010C_H$	TMRLRB2[R/W] H XXXXXXXX XXXXXXXX		TMCSR2[R/W] B,H,W 00000000 0-000000		
000110_H	TMRLRA3[R/W] H XXXXXXXX XXXXXXXX		TMR3[R] H XXXXXXXX XXXXXXXX		Reload timer 3
000114_H	TMRLRB3[R/W] H XXXXXXXX XXXXXXXX		TMCSR3[R/W] B,H,W 00000000 0-000000		
000118_H | $0011C_H$	-	-	-	-	Reserved
000120_H	IRPR0H[R] B,H,W 00------	IRPR0L[R] B,H,W 00------	IRPR1H[R] B,H,W 00------	IRPR1L[R] B,H,W --------	Interrupt request batch read register
000124_H	IRPR2H[R] B,H,W --------	IRPR2L[R] B,H,W Å¶5 0000----	IRPR3H[R] B,H,W 00------	IRPR3L[R] B,H,W 00------	
000128_H	IRPR4H[R] B,H,W 00------	IRPR4L[R] B,H,W 000000--	IRPR5H[R] B,H,W 00------	IRPR5L[R] B,H,W 00------	
$00012C_H$	IRPR6H[R] B,H,W 0000----	IRPR6L[R] B,H,W 00------	IRPR7H[R] B,H,W 00------	IRPR7L[R] B,H,W --------	
000130_H	IRPR8H[R] B,H,W --------	IRPR8L[R] B,H,W 00------	IRPR9H[R] B,H,W 00------	IRPR9L[R] B,H,W 00------	
000134_H	IRPR10H[R] B,H,W 00------	IRPR10L[R] B,H,W 00------	IRPR11H[R] B,H,W 00------	IRPR11L[R] B,H,W 0000000-	
000138_H	IRPR12H[R] B,H,W 0000000-	IRPR12L[R] B,H,W 00000000	IRPR13H[R] B,H,W 0000000-	IRPR13L[R] B,H,W ---00---	
$00013C_H$	IRPR14H[R] B,H,W 00------	IRPR14L[R] B,H,W 00------	IRPR15H[R] B,H,W 00000000	IRPR15L[R] B,H,W 0000----	
000140_H	IRPR16H[R] B,H,W 00------	IRPR16L[R] B,H,W --------	IRPR17H[R] B,H,W --------	IRPR17L[R]B,H,W --------	
000144_H	IRPR18H[R]B,H,W --------	IRPR18L[R]B,H,W 000000--	-	-	
000148_H | $0001FC_H$	-	-	-	-	Reserved
000200_H	PCN0[R/W] B,H,W 00000000 000000-0		PCSR0[W] H,W XXXXXXXX XXXXXXXX		PPG0
000204_H	PDUT0[W] H,W XXXXXXXX XXXXXXXX		PTMR0[R] H,W 11111111 11111111		
000208_H	PCN1[R/W] B,H,W 00000000 000000-0		PCSR1[W] H,W XXXXXXXX XXXXXXXX		PPG1
$00020C_H$	PDUT1[W] H,W XXXXXXXX XXXXXXXX		PTMR1[R] H,W 11111111 11111111		

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (3 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000210_H	PCN2[R/W] B,H,W 00000000 000000-0		PCSR2[W] H,W XXXXXXXX XXXXXXXX		PPG2
000214_H	PDUT2[W] H,W XXXXXXXX XXXXXXXX		PTMR2[R] H,W 11111111 11111111		
000218_H	PCN3[R/W] B,H,W 00000000 000000-0		PCSR3[W] H,W XXXXXXXX XXXXXXXX		PPG3
$00021C_H$	PDUT3[W] H,W XXXXXXXX XXXXXXXX		PTMR3[R] H,W 11111111 11111111		
000220_H	PCN4[R/W] B,H,W 00000000 000000-0		PCSR4[W] H,W XXXXXXXX XXXXXXXX		PPG4
000224_H	PDUT4[W] H,W XXXXXXXX XXXXXXXX		PTMR4[R] H,W 11111111 11111111		
000228_H	PCN5[R/W] B,H,W 00000000 000000-0		PCSR5[W] H,W XXXXXXXX XXXXXXXX		PPG5
$00022C_H$	PDUT5[W] H,W XXXXXXXX XXXXXXXX		PTMR5[R] H,W 11111111 11111111		
000230_H | $0002BC_H$	-		-		Reserved
$0002C0_H$	GTRS0[R/W] B,H,W -0000000 -0000000		GTRS1[R/W] B,H,W -0000000 -0000000		PPG Control
$0002C4_H$	GTRS2[R/W] B,H,W -0000000 -0000000		-		
$0002C8_H$	-		-		
$0002CC_H$	-		-		
$0002D0_H$	-		-		
$0002D4_H$	-		-		
$0002D8_H$	GTREN0[R/W] H,W -------- --000000		-		
$0002DC_H$	-		-		Reserved
$0002E0_H$	-	GATEC0[R/W] B,H,W ------00	-	GATEC2[R/W] B,H,W ------00	PPG GATE Control
$0002E4_H$	-	GATEC4[R/W] B,H,W ------00	-	-	
$0002E8_H$	-	-	-	-	
$0002EC_H$	-	-	-	-	Reserved
$0002F0_H$	RCRH0[W] H,W 00000000	RCRL0[W] B,H,W 00000000	UDCRH0[R] H,W 00000000	UDCRL0[R] B,H,W 00000000	U/D counter 0
$0002F4_H$	CCR0[R/W] B,H 00000000 -0001000		-	CSR0[R] B 00000000	
$0002F8_H$	RCRH1[W] H,W 00000000	RCRL1[W] B,H,W 00000000	UDCRH1[R] H,W 00000000	UDCRL1[R] B,H,W 00000000	U/D counter 1
$0002FC_H$	CCR1[R/W] B,H 00000000 -0001000		-	CSR1[R] B 00000000	
000300_H	-				Reserved
000304_H	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (4 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000308_H	-				Reserved
$00030C_H$	-	-	-	-	
000310_H	-	-	MPUCR[R/W] H 000000-0 ----0100		MPU [S] (Only the CPU can access this area)
000314_H	-	-	-	-	
000318_H	-				
$00031C_H$	-	-	-		
000320_H	DPVAR[R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000324_H	-	-	DPVSR[R/W] H -------- 00000--0		
000328_H	DEAR[R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00032C_H$	-	-	DESR[R/W] H -------- 00000--0		
000330_H	PABR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000334_H	-	-	PACR0[R/W] H 000000-0 00000--0		
000338_H	PABR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00033C_H$	-	-	PACR1[R/W] H 000000-0 00000--0		
000340_H	PABR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000344_H	-	-	PACR2[R/W] H 000000-0 00000--0		
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				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (5 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00036C_H$	-	-	PACR7[R/W] H 000000-0 00000--0		MPU [S] (Only the CPU can access this area)
000370_H	-				Reserved [S]
000374_H	-	-	-		
000378_H	-				
$00037C_H$	-	-	-		
000380_H	-				
000384_H	-	-	-		
000388_H	-				
$00038C_H$	-	-	-		
000390_H	-				
000394_H	-	-	-		Reserved [S]
000398_H	-				
$00039C_H$	-	-	-		
$0003A0_H$	-				
$0003A4_H$	-	-	-		
$0003A8_H$	-				
$0003AC_H$	-	-	-		
$0003B0_H$ | $0003CC_H$	-	-	-	-	Reserved [S]
$0003D0_H$	-				Reserved [S]
$0003D4_H$	-				
$0003D8_H$	-				
$0003DC_H$	-				
$0003E0_H$ | $0003FC_H$	-	-	-	-	Reserved [S]

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (6 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000400_H	ICSEL0[R/W] B,H,W -----000	ICSEL1[R/W] B,H,W -------0	ICSEL2[R/W] B,H,W -------0	ICSEL3[R/W] B,H,W -------0	Generation and clear of DMA transfer request
000404_H	ICSEL4[R/W] B,H,W -------0	ICSEL5[R/W] B,H,W -------0	ICSEL6[R/W] B,H,W -------0	ICSEL7[R/W] B,H,W -----000	
000408_H	ICSEL8[R/W] B,H,W -------0	ICSEL9[R/W] B,H,W -------0	ICSEL10[R/W] B,H,W ------00	ICSEL11[R/W] B,H,W -------0	
$00040C_H$	ICSEL12[R/W] B,H,W -------0	ICSEL13[R/W] B,H,W -------0	ICSEL14[R/W] B,H,W -------0	ICSEL15[R/W] B,H,W -------0	
000410_H	ICSEL16[R/W] B,H,W -------0	ICSEL17[R/W] B,H,W -------0	ICSEL18[R/W] B,H,W -------0	ICSEL19[R/W] B,H,W -------0	
000414_H	ICSEL20[R/W] B,H,W -------0	ICSEL21[R/W] B,H,W -----000	ICSEL22[R/W] B,H,W -----000	ICSEL23[R/W] B,H,W -----000	
000418_H	ICSEL24[R/W] B,H,W -----000	ICSEL25[R/W] B,H,W -----000	ICSEL26[R/W] B,H,W -------0	ICSEL27[R/W] B,H,W -------0	
$00041C_H$	-	-	-	-	
000420_H	-	-	-	-	
000424_H | $00043C_H$	-	-	-	-	Reserved
000440_H	ICR00[R/W] B,H,W ---11111	ICR01[R/W] B,H,W ---11111	ICR02[R/W] B,H,W ---11111	ICR03[R/W] B,H,W ---11111	Interrupt controller [S]
000444_H	ICR04[R/W] B,H,W ---11111	ICR05[R/W] B,H,W ---11111	ICR06[R/W] B,H,W ---11111	ICR07[R/W] B,H,W ---11111	
000448_H	ICR08[R/W] B,H,W ---11111	ICR09[R/W] B,H,W ---11111	ICR10[R/W] B,H,W ---11111	ICR11[R/W] B,H,W ---11111	
$00044C_H$	ICR12[R/W] B,H,W ---11111	ICR13[R/W] B,H,W ---11111	ICR14[R/W] B,H,W ---11111	ICR15[R/W] B,H,W ---11111	
000450_H	ICR16[R/W] B,H,W ---11111	ICR17[R/W] B,H,W ---11111	ICR18[R/W] B,H,W ---11111	ICR19[R/W] B,H,W ---11111	
000454_H	ICR20[R/W] B,H,W ---11111	ICR21[R/W] B,H,W ---11111	ICR22[R/W] B,H,W ---11111	ICR23[R/W] B,H,W ---11111	
000458_H	ICR24[R/W] B,H,W ---11111	ICR25[R/W] B,H,W ---11111	ICR26[R/W] B,H,W ---11111	ICR27[R/W] B,H,W ---11111	
$00045C_H$	ICR28[R/W] B,H,W ---11111	ICR29[R/W] B,H,W ---11111	ICR30[R/W] B,H,W ---11111	ICR31[R/W] B,H,W ---11111	
000460_H	ICR32[R/W] B,H,W ---11111	ICR33[R/W] B,H,W ---11111	ICR34[R/W] B,H,W ---11111	ICR35[R/W] B,H,W ---11111	
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	
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 | $00047C_H$	-	-	-	-	Reserved [S]

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (7 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
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 consumption control [S] * Writing to STBCR by DMA is disabled.
000484_H	-	-	-	-	Reserved [S]
000488_H	DIVR0[R/W] B,H,W 000-----	-	DIVR2[R/W] B,H,W 0011----	-	Clock control [S]
$00048C_H$	-	-	-	-	Reserved [S]
000490_H	IORR0[R/W] B,H,W -0000000	IORR1[R/W] B,H,W -0000000	IORR2[R/W] B,H,W -0000000	IORR3[R/W] B,H,W -0000000	DMA transfer request from a peripheral [S]
000494_H	IORR4[R/W] B,H,W -0000000	IORR5[R/W] B,H,W -0000000	IORR6[R/W] B,H,W -0000000	IORR7[R/W] B,H,W -0000000	
000498_H	-	-	-	-	
$00049C_H$	-	-	-	-	
$0004A0_H$	-	-	-	-	Reserved
$0004A4_H$	CANPRE[R/W] B,H,W ---00000	-	-	-	CAN prescaler
$0004A8_H$ | $0004AC_H$	-	-	-	-	Reserved
$0004B0_H$	-	-	-	-	Reserved
$0004B4_H$ | $0004C0_H$	-	-	-	-	Reserved
$0004C4_H$	CUCR1[R/W] B,H,W -------- ---0--00		CUTD1[R/W] B,H,W 11000011 01010000		WDT1 calibration
$0004C8_H$	CUTR1[R] B,H,W -------- 00000000 00000000 00000000				
$0004CC_H$ | $0004DC_H$	-	-	-	-	Reserved
$0004E0_H$	-	-	CSCFG[R/W] B,H,W ---0----	CMCFG[R/W] B,H,W 00000000	Clock monitor
$0004E4_H$	-	-	-	-	
$0004E8_H$	PLL2DIVM[R/W] B,H,W ----0000	PLL2DIVN[R/W] B,H,W -0000000	PLL2DIVG[R/W] B,H,W ----0000	PLL2MULG[R/W] B,H,W 00000000	FlexRay clock control *5
$0004EC_H$	PLL2CTRL[R/W] B,H,W ----0000	PLL2DIVK[R/W] B,H,W -------0	CLKR2[R/W] B,H,W 000--000	-	
$0004F0_H$ | $0004FC_H$	-	-	-	-	Reserved
000500_H	-				Reserved
000504_H	-				Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (8 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000508_H | $00050C_H$	-	-	-	-	Reserved
000510_H	CSELR[R/W] B,H,W -0----00	CMONR[R] B,H,W -01---00	MTMCR[R/W] B,H,W 00001111	-	Clock control [S]
000514_H	PLLCR[R/W] B,H,W 00-00000 11110000		CSTBR[R/W] B,H,W ----0000	PTMCR[R/W] B,H,W 00------	
000518_H	-	-	CPUAR[R/W] B,H,W 0---XXXX	-	Reset [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
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] B,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	-	-	CCPMUCR0[R/W] B,H,W 0-----00	CCPMUCR1[R/W] B,H,W 0--00000	
000534_H	-	-	-	-	
000538_H	-	-	-	-	
$00053C_H$	-	-	-	-	
000540_H | $00054C_H$	-	-	-	-	Reserved
000550_H	EIRR0[R/W] B,H,W -XXXXXXX	ENIR0[R/W] B,H,W -0000000	ELVR0[R/W] B,H,W --000000 00000000		External interrupt (INT0 to 6)
000554_H | 000568_H	-	-	-	-	Reserved
$00056C_H$	-	CSVCR[R/W] B -0--1--0	-	-	CSV
000570_H	CRTR[R/W] B,H,W 01111111	-	-	-	WDT1 calibration (trimming)
000574_H | $00057C_H$	-	-	-	-	Reserved
000580_H	REGSEL[R/W] B,H,W 01--110-	-	-	-	Regulator control
000584_H	LVD5R[R/W] B,H,W -------1	LVD5F[R/W] B,H,W 001100-1	LVD[R/W] B,H,W 01000--0	-	Low-voltage detection

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (9 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
000588_H | $00058C_H$	-	-	-	-	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	-	PMUINTF1[R/W] B,H,W 00000000	PMUINTF2[R/W] B,H,W -00-----	-	
000598_H	-	-	-	-	
$00059C_H$	-	-	-	-	
$0005A0_H$ | $0005FC_H$	-	-	-	-	Reserved
000600_H | $00060C_H$	-				Reserved [S]
000610_H | $00063C_H$	-	-	-	-	Reserved [S]
000640_H | $00064C_H$	-				Reserved [S]
000650_H | $00067C_H$	-	-	-	-	Reserved [S]
000680_H | $00068C_H$	-				Reserved [S]
000690_H | $0006BC_H$	-	-	-	-	Reserved [S]
$0006C0_H$ | $0006CC_H$	-				Reserved [S]
$0006D0_H$ | $0006F0_H$	-	-	-	-	Reserved
$0006F4_H$	-				Reserved
$0006F8_H$ | $0006FC_H$	-	-	-	-	Reserved
000700_H	-				Reserved
000704_H | $00070C_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (10 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
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 | $0007F8_H$	-	-	-	-	Reserved
$0007FC_H$	BMODR[R] B,H,W XXXXXXXX	-	-	-	Operation mode
000800_H | $00083C_H$	-	-	-	-	Reserved [S]
000840_H	FCTLR[R/W] H -0--1000 0--0-----		-	FSTR[R/W] B -----001	Flash memory register [S]
000844_H	-	-	-	-	Reserved [S]
000848_H | 000854_H	-	-	-	-	Reserved [S]
000858_H	-	-	WREN[R/W] H 00000000 00000000		Wild register [S]
$00085C_H$ | $00087C_H$	-	-	-	-	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				
000898_H	WRAR03[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$00089C_H$	WRDR03[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008A0_H$	WRAR04[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008A4_H$	WRDR04[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (11 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0008A8_H$	WRAR05[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register [S]
$0008AC_H$	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				
$0008F0_H$	WRAR14[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008F4_H$	WRDR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008F8_H$	WRAR15[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008FC_H$	WRDR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000900_H | $000BF8_H$	-	-	-	-	Reserved
$000BFC_H$	-		UER[W] B,H,W -------- -------X		OCDU

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (12 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000C00_H$	DCCR0[R/W] W 0----000 --00--00 00000000 0-000000				DMA controller [S]
$000C04_H$	DCSR0[R/W] H 0------- -----000		DTCR0[R/W] H 00000000 00000000		
$000C08_H$	DSAR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C0C_H$	DDAR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C10_H$	DCCR1[R/W] W 0----000 --00--00 00000000 0-000000				
$000C14_H$	DCSR1[R/W] H 0------- -----000		DTCR1[R/W] H 00000000 00000000		
$000C18_H$	DSAR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C1C_H$	DDAR1[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C20_H$	DCCR2[R/W] W 0----000 --00--00 00000000 0-000000				
$000C24_H$	DCSR2[R/W] H 0------- -----000		DTCR2[R/W] H 00000000 00000000		
$000C28_H$	DSAR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C2C_H$	DDAR2[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C30_H$	DCCR3[R/W] W 0----000 --00--00 00000000 0-000000				
$000C34_H$	DCSR3[R/W] H 0------- -----000		DTCR3[R/W] H 00000000 00000000		
$000C38_H$	DSAR3[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C3C_H$	DDAR3[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C40_H$	DCCR4[R/W] W 0----000 --00--00 00000000 0-000000				
$000C44_H$	DCSR4[R/W] H 0------- -----000		DTCR4[R/W] H 00000000 00000000		
$000C48_H$	DSAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C4C_H$	DDAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C50_H$	DCCR5[R/W] W 0----000 --00--00 00000000 0-000000				
$000C54_H$	DCSR5[R/W] H 0------- -----000		DTCR5[R/W] H 00000000 00000000		
$000C58_H$	DSAR5[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (13 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000C5C_H$	DDAR5[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA controller [S]
$000C60_H$	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				
$000C80_H$ | $000DF0_H$	-	-	-	-	
$000DF4_H$	-	-	DNMIR[R/W] B 0------0	DILVR[R/W] B ---11111	
$000DF8_H$	DMACR[R/W] W 0------- -------- 0------- --------				
$000DFC_H$	-	-	-	-	Reserved [S]
$000E00_H$	-	DDR01[R/W] B,H 00000000	DDR02[R/W] B,H 00000000	DDR03[R/W] B,H 00000000	Data direction register
$000E04_H$	DDR04[R/W] B,H 00000000	DDR05[R/W] B,H -0000000	-	DDR07[R/W] B,H -----000	
$000E08_H$	-	DDR09[R/W] B,H ---00---	-	-	
$000E0C_H$	-	-	-	-	
$000E10_H$ | $000E1C_H$	-	-	-	-	Reserved
$000E20_H$	-	PFR01[R/W] B,H 00000000	PFR02[R/W] B,H 00000000	PFR03[R/W] B,H 00000000	Port function register
$000E24_H$	PFR04[R/W] B,H 00000000	PFR05[R/W] B,H -0000000	-	PFR07[R/W] B,H -----000	
$000E28_H$	-	PFR09[R/W] B,H ---00---	-	-	
$000E2C_H$	-	-	-	-	
$000E30_H$ | $000E3C_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (14 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000E40_H$	-	PDDR01[R] B,H,W XXXXXXXX	PDDR02[R] B,H,W XXXXXXXX	PDDR03[R] B,H,W XXXXXXXX	Input data direct read register
$000E44_H$	PDDR04[R] B,H,W XXXXXXXX	PDDR05[R] B,H,W -XXXXXXX	-	PDDR07[R] B,H,W -----XXX	
$000E48_H$	-	PDDR09[R] B,H,W ---XX---	-	-	
$000E4C_H$	-	-	-	-	
$000E50_H$ | $000E5C_H$	-	-	-	-	Reserved
$000E60_H$	EPFR00[R/W] B,H -------0	EPFR01[R/W] B,H ------00	EPFR02[R/W] B,H -----000	EPFR03[R/W] B,H --000000	Extended port function register
$000E64_H$	-	-	EPFR06[R/W] B,H ------00	EPFR07[R/W] B,H ----0000	
$000E68_H$	EPFR08[R/W] B,H *5 ----0000	EPFR09[R/W] B,H -------0	EPFR10[R/W] B,H -0000000	-	
$000E6C_H$	-	-	-	-	
$000E70_H$	-	-	-	-	
$000E74_H$	-	-	-	-	
$000E78_H$	-	-	-	-	
$000E7C_H$	-	-	-	-	
$000E80_H$	-	-	-	-	
$000E84_H$ | $000EBC_H$	-	-	-	-	Reserved
$000EC0_H$	-	PPER01[R/W] B,H 00000000	PPER02[R/W] B,H 00000000	PPER03[R/W] B,H 00000000	Port pull-up/down enable register
$000EC4_H$	PPER04[R/W] B,H 00000000	PPER05[R/W] B,H -0000000	-	PPER07[R/W] B,H -----000	
$000EC8_H$	-	PPER09[R/W] B,H ---00---	-	-	
$000ECC_H$	-	-	-	-	
$000ED0_H$ | $000EDC_H$	-	-	-	-	Reserved
$000EE0_H$	-	PILR01[R/W] B,H 11111111	PILR02[R/W] B,H 11111111	PILR03[R/W] B,H 11111111	Port input level selection register
$000EE4_H$	PILR04[R/W] B,H 11111111	PILR05[R/W] B,H -1111111	-	PILR07[R/W] B,H -----111	
$000EE8_H$	-	PILR09[R/W] B,H ---11---	-	-	
$000EEC_H$	-	-	-	-	
$000EF0_H$ | $000EFC_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (15 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$000F00_H$ | $000F1C_H$	-	-	-	-	Reserved
$000F20_H$	-	PODR01[R/W] B,H 00000000	PODR02[R/W] B,H 00000000	PODR03[R/W] B,H 00000000	Port output drive register
$000F24_H$	PODR04[R/W] B,H 00000000	PODR05[R/W] B,H -0000000	-	PODR07[R/W] B,H -----000	
$000F28_H$	-	PODR09[R/W] B,H ---00---	-	-	
$000F2C_H$	-	-	-	-	
$000F30_H$ | $000F3C_H$	-	-	-	-	Reserved
$000F40_H$	PORTEN[R/W] B,H,W ------00	-	-	-	Port input enable register
$000F44_H$	KEYCDR[R/W] H 00000000 00000000		-	-	Port key code
$000F48_H$	ADERH[R/W] B,H -------- ---11---		ADERL[R/W] B,H -1111111 11111111		Analog input enable register
$000F4C_H$	DAER[R/W] B,H -------0	-	-	-	Analog output enable register
$000F50_H$ | $000FFF_H$	-	-	-	-	Reserved
001000_H	SACR[R/W] B,H,W -------0	PICD[R/W] B,H,W ----0011	-	-	Synchronous/ asynchronous switch control
001004_H | $0010BC_H$	-	-	-	-	Reserved
$0010C0_H$	-	-	-	CRCCR[R/W] B,H,W -0000000	CRC arithmetic operation 0
$0010C4_H$	CRCINIT[R/W] B,H,W 11111111 11111111 11111111 11111111				
$0010C8_H$	CRCIN[R/W] B,H,W 00000000 00000000 00000000 00000000				
$0010CC_H$	CRCR[R] B,H,W 11111111 11111111 11111111 11111111				
$0010D0_H$	-	-	-	CRCCR1[R/W] B,H,W -0000000	CRC arithmetic operation 1
$0010D4_H$	CRCINIT1[R/W] B,H,W 11111111 11111111 11111111 11111111				
$0010D8_H$	CRCIN1[R/W] B,H,W 00000000 00000000 00000000 00000000				
$0010DC_H$	CRCR1[R] B,H,W 11111111 11111111 11111111 11111111				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (16 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0010E0_H$ | $0010FC_H$	-	-	-	-	Reserved
001100_H	TCGS[R/W] B,H,W ------00	-	-	TCGSE[R/W] B,H,W --000000	Free-run timer simultaneous activation
001104_H	CPCLRB0/CPCLR0[R/W] H,W 11111111 11111111		TCDT0[R/W] H,W 00000000 00000000		Free-run timer 0
001108_H	TCCS0[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00110C_H$	CPCLRB1/CPCLR1[R/W] H,W 11111111 11111111		TCDT1[R/W] H,W 00000000 00000000		Free-run timer 1
001110_H	TCCS1[R/W] B,H,W 00000000 01000000 ----0000 --------				
001114_H	CPCLRB2/CPCLR2[R/W] H,W 11111111 11111111		TCDT2[R/W] H,W 00000000 00000000		Free-run timer 2
001118_H	TCCS2[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00111C_H$	CPCLRB3/CPCLR3[R/W] H,W 11111111 11111111		TCDT3[R/W] H,W 00000000 00000000		Free-run timer 3
001120_H	TCCS3[R/W] B,H,W 00000000 01000000 ----0000 --------				
001124_H	CPCLRB4/CPCLR4[R/W] H,W 11111111 11111111		TCDT4[R/W] H,W 00000000 00000000		Free-run timer 4
001128_H	TCCS4[R/W] B,H,W 00000000 01000000 ----0000 --------				
$00112C_H$	CPCLRB5/CPCLR5[R/W] H,W 11111111 11111111		TCDT5[R/W] H,W 00000000 00000000		Free-run timer 5
001130_H	TCCS5[R/W] B,H,W 00000000 01000000 ----0000 --------				
001134_H	FRS0[R/W] B,H,W -------- -000-000 -000-000 -000-000				Free-run timer selection
001138_H	FRS1[R/W] B,H,W -------- -------- -000-000 -000-000				
$00113C_H$	FRS2[R/W] B,H,W -------- -000-000 -000-000 -000-000				
001140_H	-				
001144_H	FRS4[R/W] B,H,W -000-000 -000-000 -000-000 -000-000				
001148_H	FRS5[R/W] B,H,W -----000 -000-000 -000-000 -000-000				
$00114C_H$	FRS6[R/W] B,H,W -------- -----000 -000---- --------				
001150_H	-				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (17 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001154_H	OCCPB0/OCCP0[R/W] H,W 00000000 00000000		OCCPB1/OCCP1[R/W] H,W 00000000 00000000		Output compare 0/1
001158_H	OCS01[R/W] B,H,W -110--00 00001100		-	OCMOD01[R/W] B,H,W ------00	
$00115C_H$	OCCPB2/OCCP2[R/W] H,W 00000000 00000000		OCCPB3/OCCP3[R/W] H,W 00000000 00000000		Output compare 2/3
001160_H	OCS23[R/W] B,H,W -110--00 00001100		-	OCMOD23[R/W] B,H,W ------00	
001164_H	OCCPB4/OCCP4[R/W] H,W 00000000 00000000		OCCPB5/OCCP5[R/W] H,W 00000000 00000000		Output compare 4/5
001168_H	OCS45[R/W] B,H,W -110--00 00001100		-	OCMOD45[R/W] B,H,W ------00	
$00116C_H$	OCCPB6/OCCP6[R/W] H,W 00000000 00000000		OCCPB7/OCCP7[R/W] H,W 00000000 00000000		Output compare 6/7
001170_H	OCS67[R/W] B,H,W -110--00 00001100		-	OCMOD67[R/W] B,H,W ------00	
001174_H	OCCPB8/OCCP8[R/W] H,W 00000000 00000000		OCCPB9/OCCP9[R/W] H,W 00000000 00000000		Output compare 8/9
001178_H	OCS89[R/W] B,H,W -110--00 00001100		-	OCMOD89[R/W] B,H,W ------00	
$00117C_H$	OCCPB10/OCCP10[R/W] H,W 00000000 00000000		OCCPB11/OCCP11[R/W] H,W 00000000 00000000		Output compare 10/11
001180_H	OCS1011[R/W] B,H,W -110--00 00001100		-	OCMOD1011[R/W] B,H,W ------00	
001184_H	IPCP0[R] H,W 00000000 00000000		IPCP1[R] H,W 00000000 00000000		Input capture 0/1
001188_H	ICS01[R/W] B,H,W ------00 00000000		-	LSYNS[R/W] B,H,W ------00	
$00118C_H$	IPCP2[R] H,W 00000000 00000000		IPCP3[R] H,W 00000000 00000000		Input capture 2/3
001190_H	ICS23[R/W] B,H,W ------00 00000000		-	-	
001194_H	-		-		Reserved
001198_H	-		-	-	
$00119C_H$	-		-		Reserved
$0011A0_H$	-		-	-	
$0011A4_H$	DTSR[R/W] B,H,W ------10	-	-	-	DTTI selection

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (18 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0011A8_H$	TMRR0[R/W] H,W 00000000 00000001		TMRR1[R/W] H,W 00000000 00000001		Waveform generator 0/1/2
$0011AC_H$	TMRR2[R/W] H,W 00000000 00000001		-	-	
$0011B0_H$	DTSCR0[R/W] B,H,W 00000000	DTSCR1[R/W] B,H,W 00000000	DTSCR2[R/W] B,H,W 00000000	-	
$0011B4_H$	-	DTIR0[R/W] B,H,W 000000--	-	DTMNS0[R/W] B,H,W 00---000	
$0011B8_H$	-	SIGCR10[R/W] B,H,W 00000000	-	SIGCR20[R/W] B,H,W 000000-1	
$0011BC_H$	PICS0[R/W] B,H,W 000000-- -------- -------- --------				
$0011C0_H$	TMRR3[R/W] H,W 00000000 00000001		TMRR4[R/W] H,W 00000000 00000001		Waveform generator 3/4/5
$0011C4_H$	TMRR5[R/W] H,W 00000000 00000001		-	-	
$0011C8_H$	DTSCR3[R/W] B,H,W 00000000	DTSCR4[R/W] B,H,W 00000000	DTSCR5[R/W] B,H,W 00000000	-	
$0011CC_H$	-	DTIR1[R/W] B,H,W 000000--	-	DTMNS1[R/W] B,H,W 00---000	
$0011D0_H$	-	SIGCR11[R/W] B,H,W 00000000	-	SIGCR21[R/W] B,H,W -------1	
$0011D4_H$	-				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (19 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0011D8_H$	-	-	-	-	12-bit A/D converter
$0011DC_H$	ADTSS[R/W] B,H,W -------0	-	-	-	
$0011E0_H$	ADTSE[R/W] B,H,W -------- ---00--- -0000000 00000000				
$0011E4_H$	ADCOMP0/ADCOMPB0[R/W] H,W 00000000 00000000		ADCOMP1/ADCOMPB1[R/W] H,W 00000000 00000000		
$0011E8_H$	ADCOMP2/ADCOMPB2[R/W] H,W 00000000 00000000		ADCOMP3/ADCOMPB3[R/W] H,W 00000000 00000000		
$0011EC_H$	ADCOMP4/ADCOMPB4[R/W] H,W 00000000 00000000		ADCOMP5/ADCOMPB5[R/W] H,W 00000000 00000000		
$0011F0_H$	ADCOMP6/ADCOMPB6[R/W] H,W 00000000 00000000		ADCOMP7/ADCOMPB7[R/W] H,W 00000000 00000000		
$0011F4_H$	ADCOMP8/ADCOMPB8[R/W] H,W 00000000 00000000		ADCOMP9/ADCOMPB9[R/W] H,W 00000000 00000000		
$0011F8_H$	ADCOMP10/ADCOMPB10[R/W] H,W 00000000 00000000		ADCOMP11/ADCOMPB11[R/W] H,W 00000000 00000000		
$0011FC_H$	ADCOMP12/ADCOMPB12[R/W] H,W 00000000 00000000		ADCOMP13/ADCOMPB13[R/W] H,W 00000000 00000000		
001200_H	ADCOMP14/ADCOMPB14[R/W] H,W 00000000 00000000		-		
001204_H	-		-		
001208_H	-		ADCOMP19/ADCOMPB19[R/W] H,W 00000000 00000000		
$00120C_H$	ADCOMP20/ADCOMPB20[R/W] H,W 00000000 00000000		-		
001210_H	-		-		
001214_H	-	-	-	-	
001218_H	-	-	-	-	
$00121C_H$	-	-	-	-	
001220_H	-	-	-	-	
001224_H	ADTCS0[R/W] B,H,W 00000000 0010-000		ADTCS1[R/W] B,H,W 00000000 0010-000		
001228_H	ADTCS2[R/W] B,H,W 00000000 0010-000		ADTCS3[R/W] B,H,W 00000000 0010-000		
$00122C_H$	ADTCS4[R/W] B,H,W 00000000 0010-000		ADTCS5[R/W] B,H,W 00000000 0010-000		
001230_H	ADTCS6[R/W] B,H,W 00000000 0010-000		ADTCS7[R/W] B,H,W 00000000 0010-000		
001234_H	ADTCS8[R/W] B,H,W 00000000 0010-000		ADTCS9[R/W] B,H,W 00000000 0010-000		
001238_H	ADTCS10[R/W] B,H,W 00000000 0010-000		ADTCS11[R/W] B,H,W 00000000 0010-000		
$00123C_H$	ADTCS12[R/W] B,H,W 00000000 0010-000		ADTCS13[R/W] B,H,W 00000000 0010-000		

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (20 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001240_H	ADTCS14[R/W] B,H,W 00000000 0010-000		-		12-bit A/D converter
001244_H	-		-		
001248_H	-		ADTCS19[R/W] B,H,W 00000000 00100000		
$00124C_H$	ADTCS20[R/W] B,H,W 00000000 00100000		-		
001250_H	-		-		
001254_H	-	-	-	-	
001258_H	-	-	-	-	
$00125C_H$	-	-	-	-	
001260_H	-	-	-	-	
001264_H	ADTCD0[R] B,H,W 10--0000 00000000		ADTCD1[R] B,H,W 10--0000 00000000		
001268_H	ADTCD2[R] B,H,W 10--0000 00000000		ADTCD3[R] B,H,W 10--0000 00000000		
$00126C_H$	ADTCD4[R] B,H,W 10--0000 00000000		ADTCD5[R] B,H,W 10--0000 00000000		
001270_H	ADTCD6[R] B,H,W 10--0000 00000000		ADTCD7[R] B,H,W 10--0000 00000000		
001274_H	ADTCD8[R] B,H,W 10--0000 00000000		ADTCD9[R] B,H,W 10--0000 00000000		
001278_H	ADTCD10[R] B,H,W 10--0000 00000000		ADTCD11[R] B,H,W 10--0000 00000000		
$00127C_H$	ADTCD12[R] B,H,W 10--0000 00000000		ADTCD13[R] B,H,W 10--0000 00000000		
001280_H	ADTCD14[R] B,H,W 10--0000 00000000		-		
001284_H	-		-		
001288_H	-		ADTCD19[R] B,H,W 10--0000 00000000		
$00128C_H$	ADTCD20[R] B,H,W 10--0000 00000000		-		
001290_H	-		-		
001294_H	-	-	-	-	
001298_H	-	-	-	-	
$00129C_H$	-	-	-	-	
$0012A0_H$	-	-	-	-	
$0012A4_H$	ADCS0[R/W] B,H,W 0------- --------		ADCH0[R] B,H,W -----000	ADMD0[R/W] B,H,W ----0000	
$0012A8_H$	ADCS1[R/W] B,H,W 0------- --------		ADCH1[R] B,H,W -----000	ADMD1[R/W] B,H,W ----0000	
$0012AC_H$	ADCS2[R/W] B,H,W 0------- --------		ADCH2[R] B,H,W -----000	ADMD2[R/W] B,H,W ----0000	

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (21 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$0012B0_H$	MTRCSR[R/W] B,H,W -------0				Motor control extension function
$0012B4_H$	RTOSEL0[R/W] B,H,W --000000	RTOSEL1[R/W] B,H,W -------0			
$0012B8_H$ | $0012FC_H$	-	-	-	-	Reserved
001300_H	-	-	-	-	Reserved
001304_H	-	-	-	-	
001308_H	-	-	-	-	
$00130C_H$	-	-	-	-	
001310_H	-	-	-	-	
001314_H	-	-	-	-	
001318_H	-	-	-	-	
$00131C_H$	-	-	-	-	
001320_H	-	-	-	-	
001324_H	-		-		
001328_H | $00132C_H$	-	-	-	-	
001330_H	-		-		
001334_H | $0013FC_H$	-	-	-	-	Reserved
001400_H	DACR[R/W] B,H,W -------0	-	DADR[R/W] H,W ------XX XXXXXXXX		DAC
001404_H | $0014FC_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (22 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
001500_H	SCR0/(IBCR0)[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000000-0	SSR0[R/W] B,H,W 0--00011	ESCR0/(IBSR0)[R/W] B,H,W 00000000	Multi Function Serial I/F 0 *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
001504_H	-/(RDR10/(TDR10))[R/W] H,W -------- -------- *3		RDR00/(TDR00)[R/W] B,H,W -------0 00000000 *1		
001508_H	SACSR0[R/W] B,H,W 0----000 00000000		STMR0[R] B,H,W 00000000 00000000		
$00150C_H$	STMCR0[R/W] B,H,W 00000000 00000000		-/(SFUR0) [R/W] B,H,W -------- -------- *3 *4		
001510_H	-	-	-/(SFLR10) [R/W] B,H,W -------- *3 *4	-/(SFLR00) [R/W] B,H,W -------- *3 *4	
001514_H	-	-	-	-	
001518_H	-	-	-	-	
$00151C_H$	BGR0[R/W] H,W 00000000 00000000		-/(ISMK0)[R/W] B,H,W -------- *2	-/(ISBA0)[R/W] B,H,W -------- *2	
001520_H	FCR10[R/W] B,H,W 00-00100	FCR00[R/W] B,H,W -0000000	FBYTE20[R/W] B,H,W 00000000	FBYTE10[R/W] B,H,W 00000000	
001524_H	SCR1[R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000000-0	SSR1[R/W] B,H,W 0--00011	ESCR1[R/W] B,H,W 00000000	Multi Function Serial I/F 1 *1 Byte access is possible only for access to lower 8 bits. *3 Reserved because CSIO mode is not set immediately after reset *4 Reserved because LIN2.1 mode is not set immediately after reset
001528_H	-/(RDR11/(TDR11))[R/W] H,W -------- -------- *3		RDR01/(TDR01)[R/W] B,H,W -------0 00000000 *1		
$00152C_H$	SACSR1[R/W] B,H,W 0----000 00000000		STMR1[R] B,H,W 00000000 00000000		
001530_H	STMCR1[R/W] B,H,W 00000000 00000000		-/(SCSCR1/SFUR1) [R/W] B,H,W -------- -------- *3 *4		
001534_H	-/(SCSTR31) [R/W] B,H,W -------- *3	-/(SCSTR21) [R/W] B,H,W -------- *3	-/(SCSTR11/SFLR11) [R/W] B,H,W -------- *3 *4	-/(SCSTR01/SFLR01) [R/W] B,H,W -------- *3 *4	
001538_H	-	-	-	-	
$00153C_H$	-	-	-	TBYTE01[R/W] B,H,W 00000000	
001540_H	BGR1[R/W] H,W 00000000 00000000		-	-	
001544_H	FCR11[R/W] B,H,W 00-00100	FCR01[R/W] B,H,W -0000000	FBYTE21[R/W] B,H,W 00000000	FBYTE11[R/W] B,H,W 00000000	
001548_H | $001FFC_H$	-	-	-	-	Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (23 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002000_H	CTRLR0[R/W] B,H,W -------- 000-0001		STATR0[R/W] B,H,W -------- 00000000		CAN 0 64msb
002004_H	ERRCNT0 [R] B,H,W 00000000 00000000		BTR0[R/W] B,H,W -0100011 00000001		
002008_H	INTR0[R] B,H,W 00000000 00000000		TESTR0[R/W] B,H,W -------- X00000--		
$00200C_H$	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 | $00207C_H$	-		-		
002080_H	TREQR20[R] B,H,W 00000000 00000000		TREQR10[R] B,H,W 00000000 00000000		

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (24 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
002084_H	TREQR40[R] B,H,W 00000000 00000000		TREQR30[R] B,H,W 00000000 00000000		CAN 0 64msb
002088_H	-		-		
$00208C_H$	-		-		
002090_H	NEWDT20[R] B,H,W 00000000 00000000		NEWDT10[R] B,H,W 00000000 00000000		
002094_H	NEWDT40[R] B,H,W 00000000 00000000		NEWDT30[R] B,H,W 00000000 00000000		
002098_H	-		-		
$00209C_H$	-		-		
$0020A0_H$	INTPND20[R] B,H,W 00000000 00000000		INTPND10[R] B,H,W 00000000 00000000		
$0020A4_H$	INTPND40[R] B,H,W 00000000 00000000		INTPND30[R] B,H,W 00000000 00000000		
$0020A8_H$	-		-		
$0020AC_H$	-		-		
$0020B0_H$	MSGVAL20[R] B,H,W 00000000 00000000		MSGVAL10[R] B,H,W 00000000 00000000		
$0020B4_H$	MSGVAL40[R] B,H,W 00000000 00000000		MSGVAL30[R] B,H,W 00000000 00000000		
$0020B8_H$	-		-		
$0020BC_H$	-		-		
$0020C0_H$ | $0020FC_H$	-		-		
002100_H | $0022FC_H$	-		-		Reserved
002300_H	DFCTLR[R/W] B,H,W -0------ --------		-	DFSTR[R/W] B,H,W -----001	WorkFlash
002304_H	-	-	-	-	
002308_H	FLIFCTLR[R/W] B,H,W ---0--00	-	FLIFFER1[R/W] B,H,W --------	FLIFFER2[R/W] B,H,W --------	
$00230C_H$ | $002FFC_H$	-	-	-	-	Reserved
003000_H	SEEARX[R] B,H,W --000000 00000000		DEEARX[R] B,H,W --000000 00000000		XBS RAM ECC control register
003004_H	EECSRX[R/W] B,H,W ----00-0	-	EFEARX[R/W] B,H,W --000000 00000000		
003008_H	-	EFECRX[R/W] B,H,W -------0 00000000 00000000			

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (25 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00300C_H$	TEAR0X[R] B,H,W 000----- -------- --000000 00000000				XBS RAM diagnosis register
003010_H	TEAR1X[R] B,H,W 000----- -------- --000000 00000000				
003014_H	TEAR2X[R] B,H,W 000----- -------- --000000 00000000				
003018_H	TAEARX[R/W] B,H,W --101111 11111111		TASARX[R/W] B,H,W --000000 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[R/W] B,H,W 0-------	-	-	TKCCRX[R/W] B,H,W 00----00	
003024_H	SEEARA[R] B,H,W --000000 00000000		DEEARA[R] B,H,W --000000 00000000		Backup RAM ECC control register
003028_H	EECSRA[R/W] B,H,W ----00-0	-	EFEARA[R/W] B,H,W --000000 00000000		
$00302C_H$	-	EFECRA[R/W] B,H,W -------0 00000000 00000000			
003030_H	TEAR0A[R] B,H,W 000----- -------- -----000 00000000				Backup RAM diagnosis register
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		
003044_H	TSRCRA[R/W] B,H,W 0-------	-	-	TKCCRA[R/W] B,H,W 00----00	
003048_H | $0030FC_H$	-	-	-	-	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				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (26 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
003110_H	BUSADR2[R] W 00000000 00000000 00000000 00000000				Bus diagnosis
003114_H	-		BUSDIGSR3[R/W] H,W 00000000 0-----00		
003118_H	BUSDIGSR4[R/W] H,W 00000000 0-----00		BUSTSTR1[R/W] H,W 00--0000 00000000		
$00311C_H$	-				
003120_H	BUSADR3[R] W 00000000 00000000 00000000 00000000				
003124_H	BUSADR4[R] W 00000000 00000000 00000000 00000000				
003128_H | $003FFC_H$	-	-	-	-	Reserved
004000_H | $005FFC_H$	Backup RAM				Backup RAM area
006000_H | $00CFFC_H$	-	-	-	-	Reserved
$00D000_H$	CIF0[R] W 00000100 11111111 01011011 11111111				FlexRay CIF *5
$00D004_H$	CIF1[R/W] W 00000000 -------0 -0000000 --------				
$00D008_H$ | $00D00C_H$	-	-	-	-	Reserved
$00D010_H$	-				FlexRay GIF *5
$00D014_H$	-				
$00D018_H$	-	-	-	-	
$00D01C_H$	LCK[R/W] W -------- -------- -------- 00000000				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (27 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D020_H$	EIR[R/W] W -----000 -----000 ----0000 00000000				FlexRay INT *5
$00D024_H$	SIR[R/W] W ------00 ------00 00000000 00000000				
$00D028_H$	EILS[R/W] W -----000 -----000 ----0000 00000000				
$00D02C_H$	SILS[R/W] W ------11 ------11 11111111 11111111				
$00D030_H$	EIES[R/W] W -----000 -----000 ----0000 00000000				
$00D034_H$	EIER[R/W] W -----000 -----000 ----0000 00000000				
$00D038_H$	SIES[R/W] W ------00 ------00 00000000 00000000				
$00D03C_H$	SIER[R/W] W ------00 ------00 00000000 00000000				
$00D040_H$	ILE[R/W] W -------- -------- -------- ------00				
$00D044_H$	T0C[R/W] W --000000 00000000 -0000000 ------00				
$00D048_H$	T1C[R/W] W --000000 00000010 -------- ------00				
$00D04C_H$	STPW1[R/W] W --000000 00000000 --000000 -0000000				
$00D050_H$	STPW2[R] W -----000 00000000 -----000 00000000				
$00D054_H$ | $00D07C_H$	-	-	-	-	Reserved
$00D080_H$	SUCC1[R/W] W ----1100 01000000 00010-00 1---0000				FlexRay SUC *5
$00D084_H$	SUCC2[R/W] W ----0001 ---00000 00000101 00000100				
$00D088_H$	SUCC3[R/W] W -------- -------- -------- 00010001				
$00D08C_H$	NEMC[R/W] W -------- -------- -------- ----0000				FlexRay NEM *5
$00D090_H$	PRTC1[R/W] W 000010-0 01001100 0000-110 00110011				FlexRay PRT *5
$00D094_H$	PRTC2[R/W] W --001111 00101101 --001010 --001110				
$00D098_H$	MHDC[R/W] W ---00000 00000000 -------- -0000000				FlexRay MHD *5
$00D09C_H$	-				Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (28 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D0A0_H$	GTUC1[R/W] W -------- ----0000 00000010 10000000				FlexRay GTU *5
$00D0A4_H$	GTUC2[R/W] W -------- ----0010 --000000 00001010				
$00D0A8_H$	GTUC3[R/W] W -0000010 -0000010 00000000 00000000				
$00D0AC_H$	GTUC4[R/W] W --000000 00001000 --000000 00000111				
$00D0B0_H$	GTUC5[R/W] W 00001110 ---00000 00000000 00000000				
$00D0B4_H$	GTUC6[R/W] W -----000 00000010 -----000 00000000				
$00D0B8_H$	GTUC7[R/W] W ------00 00000010 ------00 00000100				
$00D0BC_H$	GTUC8[R/W] W ---00000 00000000 -------- --000010				
$00D0C0_H$	GTUC9[R/W] W -------- ------00 ---00001 --000001				
$00D0C4_H$	GTUC10[R/W] W -----000 00000010 --000000 00000101				
$00D0C8_H$	GTUC11[R/W] W -----000 -----000 ------00 ------00				
$00D0CC_H$ | $00D0FC_H$	-				Reserved
$00D100_H$	CCSV[R] W --000000 00010000 -100--00 00000000				FlexRay SUC *5
$00D104_H$	CCEV[R] W -------- -------- ---00000 00--0000				
$00D108_H$ $00D10C_H$	-				Reserved
$00D110_H$	SCV[R] W -----000 00000000 -----000 00000000				FlexRay GTU *5
$00D114_H$	MTCCV[R] W -------- --000000 --000000 00000000				
$00D118_H$	RCV[R] W -------- -------- ----0000 00000000				
$00D11C_H$	OCV[R] W -------- -----000 00000000 00000000				
$00D120_H$	SFS[R] W -------- ----0000 00000000 00000000				
$00D124_H$	SWNIT[R] W -------- -------- ----0000 00000000				
$00D128_H$	ACS[R/W] W -------- -------- ---00000 ---00000				
$00D12C_H$	-				
$00D130_H$	ESID1[R] W -------- -------- 00----00 00000000				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (29 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D134_H$	ESID2[R] W -------- -------- 00----00 00000000				FlexRay GTU *5
$00D138_H$	ESID3[R] W -------- -------- 00----00 00000000				
$00D13C_H$	ESID4[R] W -------- -------- 00----00 00000000				
$00D140_H$	ESID5[R] W -------- -------- 00----00 00000000				
$00D144_H$	ESID6[R] W -------- -------- 00----00 00000000				
$00D148_H$	ESID7[R] W -------- -------- 00----00 00000000				
$00D14C_H$	ESID8[R] W -------- -------- 00----00 00000000				
$00D150_H$	ESID9[R] W -------- -------- 00----00 00000000				
$00D154_H$	ESID10[R] W -------- -------- 00----00 00000000				
$00D158_H$	ESID11[R] W -------- -------- 00----00 00000000				
$00D15C_H$	ESID12[R] W -------- -------- 00----00 00000000				
$00D160_H$	ESID13[R] W -------- -------- 00----00 00000000				
$00D164_H$	ESID14[R] W -------- -------- 00----00 00000000				
$00D168_H$	ESID15[R] W -------- -------- 00----00 00000000				
$00D16C_H$	-				
$00D170_H$	OSID1[R] W -------- -------- 00----00 00000000				
$00D174_H$	OSID2[R] W -------- -------- 00----00 00000000				
$00D178_H$	OSID3[R] W -------- -------- 00----00 00000000				
$00D17C_H$	OSID4[R] W -------- -------- 00----00 00000000				
$00D180_H$	OSID5[R] W -------- -------- 00----00 00000000				
$00D184_H$	OSID6[R] W -------- -------- 00----00 00000000				
$00D188_H$	OSID7[R] W -------- -------- 00----00 00000000				
$00D18C_H$	OSID8[R] W -------- -------- 00----00 00000000				
$00D190_H$	OSID9[R] W -------- -------- 00----00 00000000				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (30 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D194_H$	OSID10[R] W -------- -------- 00----00 00000000				FlexRay GTU *5
$00D198_H$	OSID11[R] W -------- -------- 00----00 00000000				
$00D19C_H$	OSID12[R] W -------- -------- 00----00 00000000				
$00D1A0_H$	OSID13[R] W -------- -------- 00----00 00000000				
$00D1A4_H$	OSID14[R] W -------- -------- 00----00 00000000				
$00D1A8_H$	OSID15[R] W -------- -------- 00----00 00000000				
$00D1AC_H$	-				Reserved
$00D1B0_H$	NMV1[R] W 00000000 00000000 00000000 00000000				FlexRay NEM *5
$00D1B4_H$	NMV2[R] W 00000000 00000000 00000000 00000000				
$00D1B8_H$	NMV3[R] W 00000000 00000000 00000000 00000000				
$00D1BC_H$ | $00D2FC_H$	-				Reserved
$00D300_H$	MRC[R/W] W -----001 10000000 00000000 00000000				FlexRay MHD *5
$00D304_H$	FRF[R/W] W -------1 10000000 ---00000 00000000				
$00D308_H$	FRFM[R/W] W -------- -------- ---00000 000000--				
$00D30C_H$	FCL[R/W] W -------- -------- -------- 10000000				
$00D310_H$	MHDS[R/W] W -0000000 -0000000 -0000000 00000000				
$00D314_H$	LDTS[R] W -----000 00000000 -----000 00000000				
$00D318_H$	FSR[R] W -------- -------- 00000000 -----000				
$00D31C_H$	MHDF[R/W] W -------- -------- -------0 00000000				
$00D320_H$	TXRQ1[R] W 00000000 00000000 00000000 00000000				
$00D324_H$	TXRQ2[R] W 00000000 00000000 00000000 00000000				
$00D328_H$	TXRQ3[R] W 00000000 00000000 00000000 00000000				
$00D32C_H$	TXRQ4[R] W 00000000 00000000 00000000 00000000				
$00D330_H$	NDAT1[R] W 00000000 00000000 00000000 00000000				

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (31 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D334_H$	NDAT2[R] W 00000000 00000000 00000000 00000000				FlexRay MHD *5
$00D338_H$	NDAT3[R] W 00000000 00000000 00000000 00000000				
$00D33C_H$	NDAT4[R] W 00000000 00000000 00000000 00000000				
$00D340_H$	MBSC1[R] W 00000000 00000000 00000000 00000000				
$00D344_H$	MBSC2[R] W 00000000 00000000 00000000 00000000				
$00D348_H$	MBSC3[R] W 00000000 00000000 00000000 00000000				
$00D34C_H$	MBSC4[R] W 00000000 00000000 00000000 00000000				
$00D350_H$ | $00D3EC_H$	-				Reserved
$00D3F0_H$	CREL[R] W 00010000 00111001 00000010 00000110				FlexRay GIF *5
$00D3F4_H$	ENDN[R] W 10000111 01100101 01000011 00100001				
$00D3F8_H$ | $00D3FC_H$	-				Reserved
$00D400_H$ | $00D4FC_H$	WRDSn[1-64][R/W] W 00000000 00000000 00000000 00000000				FlexRay IBF *5
$00D500_H$	WRHS1[R/W] W --000000 -0000000 -----000 00000000				
$00D504_H$	WRHS2[R/W] W -------- -0000000 -----000 00000000				
$00D508_H$	WRHS3[R/W] W -------- -------- -----000 00000000				
$00D50C_H$	-				
$00D510_H$	IBCM[R/W] W -------- ------00 -------- -----000				
$00D514_H$	IBCR[R/W] W 0------- -0000000 0------- -0000000				
$00D518_H$ | $00D5FC_H$	-				Reserved

Table B-2 I/O Map (MB91F583AS/F584AS/F585AS) (32 / 32)

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
$00D600_H$ | $00D6FC_H$	RDDSn[1-64][R] W 00000000 00000000 00000000 00000000				FlexRay OBF *5
$00D700_H$	RDHS1[R] W --000000 -0000000 -----000 00000000				
$00D704_H$	RDHS2[R] W -0000000 -0000000 -----000 00000000				
$00D708_H$	RDHS3[R] W --000000 --000000 -----000 00000000				
$00D70C_H$	MBS[R] W --000000 --000000 00-00000 00000000				
$00D710_H$	OBCM[R/W] W -------- ------00 -------- ------00				
$00D714_H$	OBCR[R/W] W -------- -0000000 0-----00 -0000000				
$00D718_H$ | $00D7FC_H$	-				Reserved
$00D800_H$ | $00EFFC_H$	-				Reserved
$00F000_H$ | $00FEFC_H$	-				Reserved [S]
$00FF00_H$	DSUCR[R/W] B,H,W -------- -------0		-	-	OCDU [S]
$00FF04_H$ | $00FF0C_H$	-	-	-	-	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				
$00FF18_H$ | $00FFF4_H$	-	-	-	-	Reserved [S]
$00FFF8_H$	EDIR1[R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FFFC_H$	EDIR0[R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

[S]: It is a system register. The illegal instruction exception (data access error) is generated when reading and writing these registers in the user mode.

*5: For FlexRay, the MB91F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ has corresponding functions.

The following registers are reserved registers for models without the FlexRay function.

000125_H IRPR2L[5:4], $000E68_H$, $0004E8$-$0004EF_H$, $00D000$-$00D717_H$

APPENDIX C. Interrupt Vector

This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.

Table C-1 Interrupt Vectors (MB91F583AM/F584AM/F585AM) (1 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
Reset	0	00	-	$3FC_H$	000FFFFCH	-	
System reserved	1	01	-	3F8H	000FFFF8H	-	
System reserved	2	02	-	3F4H	000FFFF4H	-	
System reserved	3	03	-	3F0H	000FFFF0H	-	
System reserved	4	04	-	3ECH	000FFFECH	-	
FPU exception	5	05	-	3E8H	000FFFE8H	-	
Instruction access protection violation exception	6	06	-	3E4H	000FFFE4H	-	
Data access protection violation exception	7	07	-	3E0H	000FFFE0H	-	
Data access error interrupts	8	08	-	3DCH	000FFFDCH	-	
INTE instruction	9	09	-	3D8H	000FFFD8H	-	
Instruction break	10	0A	-	3D4H	000FFFD4H	-	
System reserved	11	0B	-	3D0H	000FFFD0H	-	
System reserved	12	0C	-	3CCH	000FFFCCH	-	
System reserved	13	0D	-	3C8H	000FFFC8H	-	
Exception of invalid instruction	14	0E	-	3C4H	000FFFC4H	-	
NMI request Error generation at internal bus diagnosis RAM double-bit error Backup RAM double-bit error	15	0F	15(FH) Fixed	3C0H	000FFFC0H	-	○
External interrupt 0-7	16	10	ICR00	3BCH	000FFFBCH	0	
Reload timer 0 / 1	17	11	ICR01	3B8H	000FFFB8H	1	○
Reload timer 2 / 3	18	12	ICR02	3B4H	000FFFB4H	2	○
Multifunction serial interface ch0 (reception completed) Multifunction serial interface ch0 (status)	19	13	ICR03	3B0H	000FFFB0H	3 (*2)	○
Multifunction serial interface ch0 (transmission completed)	20	14	ICR04	3ACH	000FFFACH	4	
Multifunction serial interface ch1 (reception completed) Multifunction serial interface ch1 (status)	21	15	ICR05	3A8H	000FFFA8H	5 (*2)	○
Multifunction serial interface ch1 (transmission completed)	22	16	ICR06	3A4H	000FFFA4H	6	
Multifunction serial interface ch2 (reception completed) Multifunction serial interface ch2 (status)	23	17	ICR07	3A0H	000FFFA0H	7 (*2)	○
Multifunction serial interface ch2 (transmission completed)	24	18	ICR08	39CH	000FFF9CH	8	
Multifunction serial interface ch3 (reception completed) Multifunction serial interface ch3 (status)	25	19	ICR09	398H	000FFF98H	9 (*2)	○

Table C-1 Interrupt Vectors (MB91F583AM/F584AM/F585AM) (2 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
Multifunction serial interface ch3 (transmission completed)	26	1A	ICR10	394H	000FFF94H	10	
*5	27	1B	ICR11	390H	000FFF90H	-	
*5	28	1C	ICR12	38CH	000FFF8CH	-	
CAN 0	29	1D	ICR13	388H	000FFF88H	-	
CAN 1	30	1E	ICR14	384H	000FFF84H	-	
FlexRay 0 *6	31	1F	ICR15	380H	000FFF80H	-	
FlexRay 1 *6	32	20	ICR16	37CH	000FFF7CH	-	
FlexRay timer 0 *6	33	21	ICR17	378H	000FFF78H	-	
FlexRay timer 1 *6	34	22	ICR18	374H	000FFF74H	-	
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	35	23	ICR19	370H	000FFF70H	-	○
Main timer/PLL timer/ PLL gear for FlexRay*6/ PLL alarm for FlexRay*6	36	24	ICR20	36CH	000FFF6CH	20 (*3)	○
Clock calibration unit (CR oscillation)	37	25	ICR21	368H	000FFF68H	-	
U/D counter 0 / 1	38	26	ICR22	364H	000FFF64H	22	○
Free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360H	000FFF60H	23	○
Free-run timer 1 (0 detection) / (compare clear)	40	28	ICR24	35CH	000FFF5CH	24	○
Free-run timer 2 (0 detection) / (compare clear) PPG 0 / 1 / 2 / 3	41	29	ICR25	358H	000FFF58H	25	○
Free-run timer 3 (0 detection) / (compare clear))	42	2A	ICR26	354H	000FFF54H	26	○
Free-run timer 4 (0 detection) / (compare clear)	43	2B	ICR27	350H	000FFF50H	27	○
Free-run timer 5 (0 detection) / (compare clear) PPG 4 / 5	44	2C	ICR28	34CH	000FFF4CH	28	○
ICU 0 (fetching) / ICU 1 (fetching)	45	2D	ICR29	348H	000FFF48H	29	○
ICU 2 (fetching) / ICU 3 (fetching)	46	2E	ICR30	344H	000FFF44H	30	○
*5	47	2F	ICR31	340H	000FFF40H	-	
*5	48	30	ICR32	33CH	000FFF3CH	-	
OCU 0 (match) / OCU 1 (match)	49	31	ICR33	338H	000FFF38H	33	○
OCU 2 (match) / OCU 3 (match)	50	32	ICR34	334H	000FFF34H	34	○
OCU 4 (match) / OCU 5 (match)	51	33	ICR35	330H	000FFF30H	35	○
OCU 6 (match) / OCU 7 (match)	52	34	ICR36	32CH	000FFF2CH	36	○
OCU 8 (match) / OCU 9 (match)	53	35	ICR37	328H	000FFF28H	37	○
OCU 10 (match) / OCU 11 (match)	54	36	ICR38	324H	000FFF24H	38	○

Table C-1 Interrupt Vectors (MB91F583AM/F584AM/F585AM) (3 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320H	000FFF20H	39	○
WG dead timer underflow 3 / 4 / 5 WG dead timer reload 3 / 4 / 5 WG DTTI 1	56	38	ICR40	31CH	000FFF1CH	40	○
AD converter 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7	57	39	ICR41	318H	000FFF18H	41	○
AD converter 8 / 9 / 10 / 11 / 12 / 13 / 14	58	3A	ICR42	314H	000FFF14H	42	○
AD converter 16 / 17 / 18 / 19 / 20 / 21 / 22 / 23	59	3B	ICR43	310H	000FFF10H	43	○
Base timer 0 IRQ 0/ base timer 0 IRQ 1	60	3C	ICR44	30CH	000FFF0CH	44	○
Base timer 1 IRQ 0/ base timer 1 IRQ 1	61	3D	ICR45	308H	000FFF08H	45	○
DMAC 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7	62	3E	ICR46	304H	000FFF04H	-	○
Delay interrupt	63	3F	ICR47	300H	000FFF00H	-	-
System reserved (Used for REALOS (*4))	64	40	-	2FCH	000FFEFCH	-	-
System reserved (Used for REALOS (*4))	65	41	-	2F8H	000FFEF8H	-	-
Used with the INTE instruction.	66 | 255	42 | FF	-	2F4H | 000H	000FFEF4H | 000FFC00H	-	-

*1 : Does not support a DMA transfer request caused by an interrupt generated from a peripheral to which no RN (resource number) is assigned.

*2 : The multi-function serial interface status does not support DMA transfer caused by I^2C reception.

*3 : "PLL gear for FlexRay" and "PLL alarm for FlexRay" do not support DMA transfer.

*4 : REALOS is a registered trademark of Spansion LLC.

*5 : For MB91F583AM/F584AM/F585AM, the interrupt vectors are unused.

*6 : For FlexRay, the MB91F583AMG/F584AMG/F585AMG/F583AMJ/F584AMJ/F585AMJ has corresponding functions.

Table C-2 Interrupt Vectors (MB91F583AS/F584AS/F585AS) (1 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
Reset	0	00	-	$3FC_H$	000FFFFCH	-	
System reserved	1	01	-	3F8H	000FFFF8H	-	
System reserved	2	02	-	3F4H	000FFFF4H	-	
System reserved	3	03	-	3F0H	000FFFF0H	-	
System reserved	4	04	-	3ECH	000FFFECH	-	
FPU exception	5	05	-	3E8H	000FFFE8H	-	
Instruction access protection violation exception	6	06	-	3E4H	000FFFE4H	-	
Data access protection violation exception	7	07	-	3E0H	000FFFE0H	-	
Data access error interrupts	8	08	-	3DCH	000FFFDCH	-	
INTE instruction	9	09	-	3D8H	000FFFD8H	-	
Instruction break	10	0A	-	3D4H	000FFFD4H	-	
System reserved	11	0B	-	3D0H	000FFFD0H	-	
System reserved	12	0C	-	3CCH	000FFFCCH	-	
System reserved	13	0D	-	3C8H	000FFFC8H	-	
Exception of invalid instruction	14	0E	-	3C4H	000FFFC4H	-	
NMI request Error generation at internal bus diagnosis RAM double-bit error Backup RAM double-bit error	15	0F	15(FH) Fixed	3C0H	000FFFC0H	-	○
External interrupt 0-6	16	10	ICR00	3BCH	000FFFBCH	0	
Reload timer 0 / 1	17	11	ICR01	3B8H	000FFFB8H	1	○
Reload timer 2 / 3	18	12	ICR02	3B4H	000FFFB4H	2	○
Multifunction serial interface ch0 (reception completed) Multifunction serial interface ch0 (status)	19	13	ICR03	3B0H	000FFFB0H	3 (*2)	○
Multifunction serial interface ch0 (transmission completed)	20	14	ICR04	3ACH	000FFFACH	4	
Multifunction serial interface ch1 (reception completed) Multifunction serial interface ch1 (status)	21	15	ICR05	3A8H	000FFFA8H	5 (*2)	○
Multifunction serial interface ch1 (transmission completed)	22	16	ICR06	3A4H	000FFFA4H	6	
*5	23	17	ICR07	3A0H	000FFFA0H	-	
*5	24	18	ICR08	39CH	000FFF9CH	-	
*5	25	19	ICR09	398H	000FFF98H	-	
*5	26	1A	ICR10	394H	000FFF94H	-	
*5	27	1B	ICR11	390H	000FFF90H	-	
*5	28	1C	ICR12	38CH	000FFF8CH	-	
CAN 0	29	1D	ICR13	388H	000FFF88H	-	
*5	30	1E	ICR14	384H	000FFF84H	-	
FlexRay 0 *6	31	1F	ICR15	380H	000FFF80H	-	
FlexRay 1 *6	32	20	ICR16	37CH	000FFF7CH	-	

Table C-2 Interrupt Vectors (MB91F583AS/F584AS/F585AS) (2 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
FlexRay timer 0 *6	33	21	ICR17	378H	000FFF78H	-	
FlexRay timer 1 *6	34	22	ICR18	374H	000FFF74H	-	
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	35	23	ICR19	370H	000FFF70H	-	○
Main timer/PLL timer/ PLL gear for FlexRay*6/ PLL alarm for FlexRay*6	36	24	ICR20	36CH	000FFF6CH	20 (*3)	○
Clock calibration unit (CR oscillation)	37	25	ICR21	368H	000FFF68H	-	
U/D counter 0 / 1	38	26	ICR22	364H	000FFF64H	22	○
Free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360H	000FFF60H	23	○
Free-run timer 1 (0 detection) / (compare clear)	40	28	ICR24	35CH	000FFF5CH	24	○
Free-run timer 2 (0 detection) / (compare clear) PPG 0 / 1 / 2 / 3	41	29	ICR25	358H	000FFF58H	25	○
Free-run timer 3 (0 detection) / (compare clear))	42	2A	ICR26	354H	000FFF54H	26	○
Free-run timer 4 (0 detection) / (compare clear)	43	2B	ICR27	350H	000FFF50H	27	○
Free-run timer 5 (0 detection) / (compare clear) PPG 4 / 5	44	2C	ICR28	34CH	000FFF4CH	28	○
ICU 0 (fetching) / ICU 1 (fetching)	45	2D	ICR29	348H	000FFF48H	29	○
ICU 2 (fetching) / ICU 3 (fetching)	46	2E	ICR30	344H	000FFF44H	30	○
*5	47	2F	ICR31	340H	000FFF40H	-	
*5	48	30	ICR32	33CH	000FFF3CH	-	
OCU 0 (match) / OCU 1 (match)	49	31	ICR33	338H	000FFF38H	33	○
OCU 2 (match) / OCU 3 (match)	50	32	ICR34	334H	000FFF34H	34	○
OCU 4 (match) / OCU 5 (match)	51	33	ICR35	330H	000FFF30H	35	○
OCU 6 (match) / OCU 7 (match)	52	34	ICR36	32CH	000FFF2CH	36	○
OCU 8 (match) / OCU 9 (match)	53	35	ICR37	328H	000FFF28H	37	○
OCU 10 (match) / OCU 11 (match)	54	36	ICR38	324H	000FFF24H	38	○
WG dead timer underflow 0 / 1/ 2 WG dead timer reload 0 / 1/ 2 WG DTTI 0	55	37	ICR39	320H	000FFF20H	39	○
WG dead timer underflow 3 / 4 / 5 WG dead timer reload 3 / 4 / 5 WG DTTI 1	56	38	ICR40	31CH	000FFF1CH	40	○
AD converter 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7	57	39	ICR41	318H	000FFF18H	41	○
AD converter 8 / 9 / 10 / 11 / 12 / 13 / 14	58	3A	ICR42	314H	000FFF14H	42	○
AD converter 19 / 20	59	3B	ICR43	310H	000FFF10H	43	○
Base timer 0 IRQ 0/ base timer 0 IRQ 1	60	3C	ICR44	30CH	000FFF0CH	44	○

Table C-2 Interrupt Vectors (MB91F583AS/F584AS/F585AS) (3 / 3)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address	RN *1	Interrupt request batch read target
	Decimal	Hexadecimal					
Base timer 1 IRQ 0/ base timer 1 IRQ 1	61	3D	ICR45	308H	000FFF08H	45	○
DMAC 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7	62	3E	ICR46	304H	000FFF04H	-	○
Delay interrupt	63	3F	ICR47	300H	000FFF00H	-	-
System reserved (Used for REALOS (*4))	64	40	-	2FCH	000FFEFCH	-	-
System reserved (Used for REALOS (*4))	65	41	-	2F8H	000FFEF8H	-	-
Used with the INTE instruction.	66 | 255	42 | FF	-	2F4H | 000H	000FFEF4H | 000FFC00H	-	-

*1 : Does not support a DMA transfer request caused by an interrupt generated from a peripheral to which no RN (resource number) is assigned.

*2 : The multi-function serial interface status does not support DMA transfer caused by I^2C reception.

*3 : "PLL gear for FlexRay" and "PLL alarm for FlexRay" do not support DMA transfer.

*4 : REALOS is a registered trademark of Spansion LLC.

*5 : For MB91F583AS/F584AS/F585AS, the interrupt vectors are unused.

*6 : For FlexRay, the MB91F583ASG/F584ASG/F585ASG/F583ASJ/F584ASJ/F585ASJ has corresponding functions.

APPENDIX D. Pin Statuses in CPU Status

Table D-1 Pin Statuses (MB91F583AM/F584AM/F585AM) (1 / 3)

Pin No.	Pin name	Port number	Specified function name			Pin attribute					Status																						
											External reset factor(*1)			External reset factor(*2)				Internal reset factor(*4)		Sleep mode		Stop mode		Power Shut-off Stop Mode				Power Shut-off Stop Mode (With external reset factor 2 generated)	Watch mode		Power Shut-off Watch Mode				Power Shut-off Watch Mode (With external reset factor 2 generated)
											External factor generation in progress	After external factor releasing		External factor generation in progres		After external factor releasing		-						SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x				SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	
			Input	Output	I/O	I/O circuit type	Input level	Active Level	PullUp/Down	GPORTEN Control	Internal reset issuance in progress	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress		After CPU reset releasing(*13)		Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress		After CPU reset releasing(*13)		Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing
2	P000	P000				D	CS/AM	-	U	With control	Hi-Z/Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input blocked		Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked
4	P001	P001				D																											
7	P002	P002				D																											
9	P003	P003				D																											
11	P004	P004				D																											
13	P005	P005				D																											
15	P006	P006				D																											
17	P007	P007				D																											
19	P010/AN0/IN0	P010	AN0/IN0			F																											
20	P011/AN1/IN1	P011	AN1/IN1			F																											
21	P012/AN2/IN2	P012	AN2/IN2			F																											
22	P013/AN3/IN3	P013	AN3/IN3			F																											
23	P014/AN4/TRG1	P014	AN4/TRG1			F																											
24	P015/AN5	P015	AN5			F																											
27	P016/AN6/INT6	P016	AN6/INT6			G																											
29	P017/AN7/INT5	P017	AN7/INT5			G																											
35	P020/TRG0/SCK1	P020	TRG0		SCK1	D																											
36	P021/SIN1/TXENB/INT0	P021	SIN1/INT0	TXENB		L																											
37	P022/SOT1/RXDB	P022	RXDB	SOT1		K																											
38	P023/SCS1/TXDB	P023		TXDB	SCS1	K																											
39	P024/PPG0/STOPWT	P024	STOPWT	PPG0		D																											
40	P025/PPG1/TXENA	P025		PPG1/TXENA		K																											
41	P026/RXDA/PPG2	P026	RXDA	PPG2		K																											
42	P027/PPG3/TXDA	P027		PPG3/TXDA		K																											
53	P030/DAOUT	P030		DAOUT		M																											
54	P031/AN14	P031	AN14			F																											
55	P032/AN13	P032	AN13			F																											

Table D-1 Pin Statuses (MB91F583AM/F584AM/F585AM) (2 / 3)

Pin No.	Pin name	Port number	Specified function name			Pin attribute					Status																						
											External reset factor(*1)			External reset factor(*2)				Internal reset factor(*4)				Sleep mode		Stop mode		Power Shut-off Stop Mode				Power Shut-off Stop Mode (With external reset factor 2 generated)	Watch mode		Power Shut-off Watch Mode				Power Shut-off Watch Mode (With external reset factor 2 generated)
											External factor generation in progress	After external factor releasing		External factor generation in progres	After external factor releasing			-								SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x				SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	
			Input	Output	I/O	I/O circuit type	Input level	Active Level	PullUp/Down	GPORTEN Control	Internal reset issuance in progress	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)			Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)			Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing
61	P033/AN12/ TIOB1/ TOT3	P033	AN12/ TIOB1	TOT3		F	CS/ AM	-	U	With control	Hi-Z/ Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/ Input blocked	Hi-Z/ Input blocked			Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/ Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/ Input blocked
62	P034/AN11/ TIOA1/ TIN3	P034	AN11/ TIN3		TIOA1	F																											
63	P035/AN10/ TIOB0/ TOT2	P035	AN10/ TIOB0	TOT2		F																											
64	P036/AN9/ TIOA0/ TIN2	P036	AN9/TIN2	TIOA0		F																											
68	P037/AN8	P037	AN8			F																											
71	P040/ SOT0_0	P040		SOT0_0		H																											
72	P041/ SCK0_0	P041			SCK0_0	H																											
74	P042/ SIN0_0	P042	SIN0_0			D																											
78	P043/ SIN0_1/ ADTG1/ MONCLK	P043	SIN0_1/ ADTG1	MONCL K		D																											
86	P044/TX0	P044		TX0		D																											
87	P045/RX0/ INT1	P045	RX0/INT1			E																											
97	P046/ ADTG0/ MM	P046	ADTG0	MM		D																											
98	P047/INT2/ TOT0/ ADTG2	P047	INT2/ ADTG2	TOT0		E																											
6	P050/ZIN0/ RTO5	P050	ZIN0	RTO5		D																											
8	P051/ FRCK5/ AIN1/RTO4	P051	FRCK5/ AIN1	RTO4		D																											
10	P052/ FRCK4/ BIN1/RTO3	P052	FRCK4/ BIN1	RTO3		D																											
12	P053/ FRCK3/ ZIN1/RTO2	P053	FRCK3/ ZIN1	RTO2		D																											
14	P054/ FRCK2/ RTO1	P054	FRCK2	RTO1		D																											
16	P055/ FRCK1/ RTO0	P055	FRCK1	RTO0		D																											
18	P056/ FRCK0/ DTTI0	P056	FRCK0/ DTTI0			D																											
59	P060/RX1/ INT7	P060	RX1/INT7			E																											
60	P061/TX1	P061		TX1		D																											

Table D-1 Pin Statuses (MB91F583AM/F584AM/F585AM) (3 / 3)

Pin No.	Pin name	Port number	Specified function name			Pin attribute					Status																						
											External reset factor(*1)			External reset factor(*2)				Internal reset factor(*4)						Power Shut-off Stop Mode				Power Shut-off Stop Mode (With external reset factor 2 generated)	Watch mode		Power Shut-off Watch Mode				Power Shut-off Watch Mode (With external reset factor 2 generated)
											External factor generation in progress	After external factor releasing		External factor generation in progres	After external factor releasing			-		Sleep mode		Stop mode		SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x				SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	
			Input	Output	I/O	I/O circuit type	Input level	Active Level	PullUp/Down	GPORTEN Control	Internal reset issuance in progress	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress		After CPU reset releasing(*13)		Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress		After CPU reset releasing(*13)		Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing
65	P062/SIN2	P062	SIN2			D	CS/AM	-	U	With control	Hi-Z/Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input blocked		Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked
66	P063/SOT2	P063		SOT2		H																											
69	P064/SCK2	P064			SCK2	H																											
70	P065/SCS2	P065			SCS2	D																											
73	P066	P066				D																											
99	P070/TIN0/INT3	P070	TIN0/INT3			E																											
3	P071/TIN1/AIN0/INT4	P071	TIN1/AIN0/INT4			E																											
5	P072/BIN0/TOT1/RTO6	P072	BIN0	TOT1/RTO6		D																											
79	P080/SOT0_1	P080		SOT0_1		H																											
80	P081/SCK0_1	P081			SCK0_1	H																											
89	P082/SIN3	P082	SIN3			D																											
90	P083/SOT3	P083		SOT3		H																											
91	P084/SCK3	P084			SCK3	H																											
94	P085/SCS3	P085			SCS3	D																											
95	P086	P086				D																											
96	P087	P087				D																											
32	P090/AN16	P090	AN16			F																											
33	P091/AN17	P091	AN17			F																											
34	P092/AN18	P092	AN18			F																											
43	P093/AN19/PPG4	P093	AN19	PPG4		F																											
44	P094/AN20/PPG5	P094	AN20	PPG5		F																											
45	P095/AN21	P095	AN21			F																											
46	P096//AN22	P096	AN22			F																											
52	P097/AN23	P097	AN23			F																											
47	P100	P100	IN1			D																											
48	P101	P101				D																											
49	P102	P102				D																											
83	X0		-	-	-	A	-	-	-	Without control	-			-				-		-				-									
84	X1		-	-	-	A																											
67	NMIX		-	-	-	B	CS	L	U		Input enabled			Input enabled				Input enabled		Input enabled													
88	RSTX		-	-	-	B	CS	L	U																								
77	DEBUGIF		-	-	-	I	TTL	-	-		Hi-Z/Input enabled	Input enabled	Input disabled (*9)	Input disabled	Input enabled (*10)	Input disabled	Input enabled (*10)	Input disabled (*9)		Input enabled (*10)	Input disabled (*9)	Input disabled (*9)											
81	MD0		-	-	-	C	CS	-	-		Input enabled			Input enabled				Input enabled		Input enabled													
82	MD1		-	-	-	C	CS	-	-																								

(*1) Power-on reset, low-voltage detection (internal low-voltage detection), and NMIX+RSTX are factors.
(*2) Low-voltage detection (external low-voltage detection) and external reset are factors.
(*4) Software reset and software/hardware watchdog reset are factors. (This also includes the timeout time.)
(*5) Input cut-off of the following pins is disabled when external interrupt is enabled.
P017, P021, P023, P045, P047, P060, P070, P071
(*7) Operation is continued according to the peripheral function.
(*9) Input is possible in emulation mode, and input is not possible in free run mode.
(*10) Input is possible only when a reset of the INIT level is issued. Otherwise, input is disabled.
(*13) After recovery from a power shut-off, a pin will enter the internal reset factor status. However, if I/O latch control is switched to software processing control, a pin will retain the standby status until it is released by the register.

Table D-2 Pin Statuses (MB91F583AS/F584AS/F585AS) (1 / 3)

Pin No.	Pin name	Port number	Specified function name			Pin attribute					Status																							
											External reset factor(*1)			External reset factor(*2)				Internal reset factor(*4)				Sleep mode		Stop mode		Power Shut-off Stop Mode				Power Shut-off Stop Mode (With external reset factor 2 generated)	Watch mode		Power Shut-off Watch Mode				Power Shut-off Watch Mode (With external reset factor 2 generated)
											External factor generation in progress	After external factor releasing		External factor generation in progres	After external factor releasing			-								SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x				SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	
			Input	Output	I/O	I/O circuit type	Input level	Active Level	PullUp/Down	GPORTEN Control	Internal reset issuance in progress	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)			Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)			Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasin
11	P010/AN0/IN0	P010	AN0/IN0			F	CS/AM	-	U	With control	Hi-Z/Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input blocked			Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked
12	P011/AN1/IN1	P011	AN1/IN1			F																												
13	P012/AN2/IN2	P012	AN2/IN2			F																												
14	P013/AN3/IN3	P013	AN3/IN3			F																												
15	P014/AN4/TRG1	P014	AN4/TRG1			F																												
16	P015/AN5	P015	AN5			F																												
17	P016/AN6/INT6	P016	AN6/INT6			G																												
19	P017/AN7/INT5	P017	AN7/INT5			G																												
22	P020/TRG0/SCK1	P020	TRG0		SCK1	D																												
23	P021/SIN1/TXENB/INT0	P021	SIN1/INT0	TXENB		L																												
24	P022/SOT1/RXDB	P022	RXDB	SOT1		K																												
25	P023/SCS1/TXDB	P023		TXDB	SCS1	K																												
26	P024/PPG0/STOPWT	P024	STOPWT	PPG0		D																												
27	P025/PPG1/TXENA	P025		PPG1/TXENA		K																												
28	P026/RXDA/PPG2	P026	RXDA	PPG2		K																												
29	P027/PPG3/TXDA	P027		PPG3/TXDA		K																												
34	P030/DAOUT	P030		DAOUT		M																												
35	P031/AN14	P031	AN14			F																												
36	P032/AN13	P032	AN13			F																												
40	P033/AN12/TIOB1/TOT3	P033	AN12/TIOB1	TOT3		F																												
41	P034/AN11/TIOA1/TIN3	P034	AN11/TIN3		TIOA1	F																												

Table D-2 Pin Statuses (MB91F583AS/F584AS/F585AS) (2 / 3)

Pin No.	Pin name	Port number	Specified function name			Pin attribute					Status																								
											External reset factor(*1)			External reset factor(*2)				Internal reset factor(*4)		Sleep mode		Stop mode		Power Shut-off Stop Mode				Power Shut-off Stop Mode (With external reset factor 2 generated)	Watch mode		Power Shut-off Watch Mode				Power Shut-off Watch Mode (With external reset factor 2 generated)
											External factor generation in progress	After external factor releasing		External factor generation in progres		After external factor releasing		-						SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x				SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	
			Input	Output	I/O	I/O circuit type	Input level	Active Level	PullUp/Down	GPORTEN Control	Internal reset issuance in progress	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Before internal reset issuance	Internal reset issuance in progress	Before internal reset issuance	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	After internal reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	SLVL[1:0] = 2'b0x	SLVL[1:0] = 2'b1x	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	After CPU reset releasing(*13)	Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasin
42	P035/AN10/ TIOB0/ TOT2	P035	AN10/ TIOB0	TOT2		F	CS/ AM	-	U	With control	Hi-Z/ Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/ Input blocked	Hi-Z/ Input blocked			Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/ Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/ Input blocked
43	P036/AN9/ TIOA0/ TIN2	P036	AN9/TIN2	TIOA0		F																													
45	P037/AN8	P037	AN8			F																													
46	P040/ SOT0_0	P040		SOT0_0		H																													
47	P041/ SCK0_0	P041			SCK0_0	H																													
48	P042/ SIN0_0	P042	SIN0_0			D																													
50	P043/ ADTG1/ MONCLK	P043	ADTG1	MONCL K		D																													
56	P044/TX0	P044		TX0		D																													
57	P045/RX0/ INT1	P045	RX0/INT1			E																													
61	P046/ ADTG0/ MM	P046	ADTG0	MM		D																													
62	P047/INT2/ TOT0/ ADTG2	P047	INT2/ ADTG2	TOT0		E																													
4	P050/ZIN0/ RTO5	P050	ZIN0	RTO5		D																													
5	P051/ FRCK5/ AIN1/RTO4	P051	FRCK5/ AIN1	RTO4		D																													
6	P052/ FRCK4/ BIN1/RTO3	P052	FRCK4/ BIN1	RTO3		D																													
7	P053/ FRCK3/ ZIN1/RTO2	P053	FRCK3/ ZIN1	RTO2		D																													
8	P054/ FRCK2/ RTO1	P054	FRCK2	RTO1		D																													
9	P055/ FRCK1/ RTO0	P055	FRCK1	RTO0		D																													
10	P056/ FRCK0/ DTTI0	P056	FRCK0/ DTTI0			D																													
63	P070/TIN0/ INT3	P070	TIN0/INT3			E																													
2	P071/TIN1/ AIN0/INT4	P071	TIN1/ AIN0/INT4			E																													

Table D-2 Pin Statuses (MB91F583AS/F584AS/F585AS) (3 / 3)

Pin No.	Pin name	Port number	Specified function name: Input	Specified function name: Output	Specified function name: I/O	Pin attribute: I/O circuit type	Pin attribute: Input level	Pin attribute: Active Level	Pin attribute: PullUp/Down	Pin attribute: GPORTEN Control	Status: External reset factor(*1) – External factor generation in progress: Internal reset issuance in progress	Status: External reset factor(*1) – After external factor releasing: Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	Status: External reset factor(*1) – After external factor releasing: After internal reset releasing	Status: External reset factor(*2) – External factor generation in progres: Before internal reset issuance	Status: External reset factor(*2) – External factor generation in progres: Internal reset issuance in progress	Status: External reset factor(*2) – After external factor releasing: Before internal reset issuance	Status: External reset factor(*2) – After external factor releasing: Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	Status: External reset factor(*2) – After external factor releasing: After internal reset releasing	Status: Internal reset factor(*4) – -: Internal reset issuance in progress to after internal reset releasing (Oscillation stabilization wait in progress)	Status: Internal reset factor(*4) – -: After internal reset releasing	Status: Sleep mode: SLVL[1:0] = 2'b0x	Status: Sleep mode: SLVL[1:0] = 2'b1x	Status: Stop mode: SLVL[1:0] = 2'b0x	Status: Stop mode: SLVL[1:0] = 2'b1x	Status: Power Shut-off Stop Mode – SLVL[1:0] = 2'b0x: Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	Status: Power Shut-off Stop Mode – SLVL[1:0] = 2'b1x: Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	Status: Power Shut-off Stop Mode – SLVL[1:0] = 2'b0x: After CPU reset releasing(*13)	Status: Power Shut-off Stop Mode – SLVL[1:0] = 2'b1x: After CPU reset releasing(*13)	Status: Power Shut-off Stop Mode (With external reset factor 2 generated): Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasing	Status: Watch mode: SLVL[1:0] = 2'b0x	Status: Watch mode: SLVL[1:0] = 2'b1x	Status: Power Shut-off Watch Mode – SLVL[1:0] = 2'b0x: Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	Status: Power Shut-off Watch Mode – SLVL[1:0] = 2'b1x: Transiting to power shut-off in progress to recovering from power shut-off (before isolator releasing) to CPU reset in progress	Status: Power Shut-off Watch Mode – SLVL[1:0] = 2'b0x: After CPU reset releasing(*13)	Status: Power Shut-off Watch Mode – SLVL[1:0] = 2'b1x: After CPU reset releasing(*13)	Status: Power Shut-off Watch Mode (With external reset factor 2 generated): Transiting to power shut-off to power shut-off in progress to recovering from power shut-off to CPU reset in progress to after CPU reset releasin
3	P072/BIN0/TOT1/RTO6	P072	BIN0	TOT1/RTO6		D	CS/AM -	- L	U -	With control	Hi-Z/Input blocked			Hi-Z/Input enabled	Hi-Z/Input blocked	Hi-Z/Input enabled	Hi-Z/Input blocked		Hi-Z/Input blocked		Last status retained(*7)	Last status retained	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Last status retained	Hi-Z/Input blocked(*5)	Hi-Z/Input blocked
30	P093/AN19/PPG4	P093	AN19	PPG4		F																														
31	P094/AN20/PPG5	P094	AN20	PPG5		F																														
53	X0		-	-	-	A	-	-	-	Without control	-			-					-		-															
54	X1		-	-	-	A																														
44	NMIX		-	-	-	B	CS	L	U		Input enabled			Input enabled					Input enabled		Input enabled															
58	RSTX		-	-	-	B	CS	L	U																											
49	DEBUGIF		-	-	-	I	TTL	-	-		Hi-Z/Input enabled	Input enabled	Input disabled (*9)	Input disabled	Input enabled (*10)	Input disabled	Input enabled (*10)	Input disabled (*9)	Input enabled (*10)	Input disabled (*9)	Input disabled (*9)															
51	MD0		-	-	-	C	CS	-	-		Input enabled			Input enabled					Input enabled		Input enabled															
52	MD1		-	-	-	C	CS	-	-																											

(*1) Power-on reset, low-voltage detection (internal low-voltage detection), and NMIX+RSTX are factors.
(*2) Low-voltage detection (external low-voltage detection) and external reset are factors.
(*4) Software reset and software/hardware watchdog reset are factors. (This also includes the timeout time.)
(*5) Input cut-off of the following pins is disabled when external interrupt is enabled.
P017, P021, P023, P045, P047, P070, P071
(*7) Operation is continued according to the peripheral function.
(*9) Input is possible in emulation mode, and input is not possible in free run mode.
(*10) Input is possible only when a reset of the INIT level is issued. Otherwise, input is disabled.
(*13) After recovery from a power shut-off, a pin will enter the internal reset factor status. However, if I/O latch control is switched to software processing control, a pin will retain the standby status until it is released by the register.

Major Changes

Page	Section	Change Results
Revision 0v01		
		Initial release
Revision 1v0		
		No document.
Revision 2v0		
-	-	Preliminary -> Full Production
-	-	Company name and layout design change. Page number update.
14-17 20-24 28 30 33-38 43 94 139 183 206-210 227 228 234 291 315 329 347 348 443 523 527 529 1249 1377 1378-1412 1445-1447 1451-1453	2.2.2 2.3 2.4 2.5 2.7 2.8.1 3.1 3.2 6.3 7.1 9.5.2 11.4 11.4.1 11.4.2 11.4.4 11.4.5 11.4.6 11.4.7 16.2 16.3 16.5.2 19.3 21.2 23.4.6 24.1 24.4 28.5 35.2 35.4 45.1 APPENDIX A APPENDIX B APPENDIX C APPENDIX D	The product series name should be corrected as indicated by the shaded below. (Error) (Correct) MB91F585MG MB91F585AMG MB91F585MH MB91F585AMH MB91F585MJ MB91F585AMJ MB91F585MK MB91F585AMK MB91F584MG MB91F584AMG MB91F584MH MB91F584AMH MB91F584MJ MB91F584AMJ MB91F584MK MB91F584AMK MB91F583MG MB91F583AMG MB91F583MH MB91F583AMH MB91F583MJ MB91F583AMJ MB91F583MK MB91F583AMK

Page	Item	Change Results
14-16	2.2.2	The product series name should be corrected as indicated by the shaded below.
18-22	2.3	
25	2.4	(Error) (Correct)
29	2.5	MB91F585SG MB91F585ASG
31	2.7	MB91F585SH MB91F585ASH
39-42	2.8.2	MB91F585SJ MB91F585ASJ
50	3.1	MB91F585SK MB91F585ASK
95	3.3	MB91F584SG MB91F584ASG
139	6.3	MB91F584SH MB91F584ASH
181	7.1	MB91F584SJ MB91F584ASJ
183	9.4.4	MB91F584SK MB91F584ASK
206-210	9.5.2	MB91F583SG MB91F583ASG
227	11.4	MB91F583SH MB91F583ASH
228	11.4.1	MB91F583SJ MB91F583ASJ
234	11.4.2	MB91F583SK MB91F583ASK
292	11.4.4	
315-317	11.4.5	
329	11.4.6	
340	11.4.7	
343	16.2	
344	16.3	
347-349	16.5.2	
357	19.3	
359	21.1	
367	21.2	
428	21.4	
440	23.4.6	
523	23.4.28	
527	23.4.34	
534	23.4.35	
789	24.1	
1160	24.4	
1192	24.4.8	
1193	24.5.1	
1195	28.4.1	
1196	28.4.26	
1207	35.2	
1208	35.4.1.1	
1217	36.1	
1249	42.3	
1252-1254	42.4.3.1	
1377	42.5.3	
1413-1444	43.4.1.3	
1448-1450	45.1	
1455-1457	45.3	
	APPENDIX A	
	APPENDIX B	
	APPENDIX C	
	APPENDIX D	

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>14</td><td>2.2.2</td><td>The description of "2.2.2 Peripheral Functions" should be corrected as indicated by the shaded below.

(Error)
• CR oscillation
- Oscillation frequency: 100kHz, with frequency precision ± 10%
- Trimming is enabled

(Correct)
• CR oscillation
- Oscillation frequency: 100kHz, with frequency accuracy ± 50%(pre-trimming)
- Trimming is enabled</td></tr>
<tr><td>34</td><td>3.1</td><td>The table should be corrected as indicated by the shaded below.

(Error)
<table>
<tr><td rowspan="4">36</td><td>P021</td><td rowspan="4">L</td><td>General-purpose I/O ports</td></tr>
<tr><td>SIN1</td><td>Multi-function serial ch.1 serial data input pin</td></tr>
<tr><td>STOPWT</td><td>FlexRay Stopwatch input pin</td></tr>
<tr><td>INT0</td><td>INT0 external interrupt input pin</td></tr>
<tr><td rowspan="3">37</td><td>P022</td><td rowspan="3">K</td><td>General-purpose I/O ports</td></tr>
<tr><td>SOT1</td><td>Multi-function serial ch.1 serial data output pin</td></tr>
<tr><td>TXENB</td><td>FlexRay ch.B operation enable output pin</td></tr>
<tr><td rowspan="3">38</td><td>P023</td><td rowspan="3">K</td><td>General-purpose I/O ports</td></tr>
<tr><td>SCS1</td><td>Multi-function serial ch.1 serial chip select I/O pin</td></tr>
<tr><td>RXDB</td><td>FlexRay ch.B data input pin</td></tr>
<tr><td rowspan="3">39</td><td>P024</td><td rowspan="3">D</td><td>General-purpose I/O ports</td></tr>
<tr><td>PPG0</td><td>PPG ch.0 output pin</td></tr>
<tr><td>TXDB</td><td>FlexRay ch.B data output pin</td></tr>
</table>
(Correct)
<table>
<tr><td rowspan="4">36</td><td>P021</td><td rowspan="4">L</td><td>General-purpose I/O ports</td></tr>
<tr><td>SIN1</td><td>Multi-function serial ch.1 serial data input pin</td></tr>
<tr><td>TXENB</td><td>FlexRay ch.B operation enable output pin</td></tr>
<tr><td>INT0</td><td>INT0 external interrupt input pin</td></tr>
<tr><td rowspan="3">37</td><td>P022</td><td rowspan="3">K</td><td>General-purpose I/O ports</td></tr>
<tr><td>SOT1</td><td>Multi-function serial ch.1 serial data output pin</td></tr>
<tr><td>RXDB</td><td>FlexRay ch.B data input pin</td></tr>
<tr><td rowspan="3">38</td><td>P023</td><td rowspan="3">K</td><td>General-purpose I/O ports</td></tr>
<tr><td>SCS1</td><td>Multi-function serial ch.1 serial chip select I/O pin</td></tr>
<tr><td>TXDB</td><td>FlexRay ch.B data output pin</td></tr>
<tr><td rowspan="3">39</td><td>P024</td><td rowspan="3">D</td><td>General-purpose I/O ports</td></tr>
<tr><td>PPG0</td><td>PPG ch.0 output pin</td></tr>
<tr><td>STOPWT</td><td>FlexRay Stopwatch input pin</td></tr>
</table></td></tr>
</table>

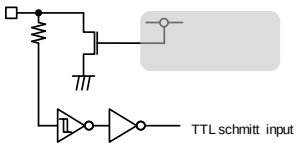
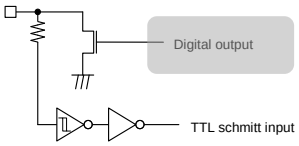
Page	Item	Change Results
39 40	3.1	The table should be corrected as indicated by the shaded below. (Error) (Correct)

(Error)

23	P021	L	General-purpose I/O ports
	SIN1		Multi-function serial ch.1 serial data input pin
	STOPWT		FlexRay Stopwatch input pin
	INT0		INT0 external interrupt input pin
24	P022	K	General-purpose I/O ports
	SOT1		Multi-function serial ch.1 serial data output pin
	TXENB		FlexRay ch.B operation enable output pin
25	P023	K	General-purpose I/O ports
	SCS1		Multi-function serial ch.1 serial chip select I/O pin
	RXDB		FlexRay ch.B data input pin
26	P024	D	General-purpose I/O ports
	PPG0		PPG ch.0 output pin
	TXDB		FlexRay ch.B data output pin

(Correct)

23	P021	L	General-purpose I/O ports
	SIN1		Multi-function serial ch.1 serial data input pin
	TXENB		FlexRay ch.B operation enable output pin
	INT0		INT0 external interrupt input pin
24	P022	K	General-purpose I/O ports
	SOT1		Multi-function serial ch.1 serial data output pin
	RXDB		FlexRay ch.B data input pin
25	P023	K	General-purpose I/O ports
	SCS1		Multi-function serial ch.1 serial chip select I/O pin
	TXDB		FlexRay ch.B data output pin
26	P024	D	General-purpose I/O ports
	PPG0		PPG ch.0 output pin
	STOPWT		FlexRay Stopwatch input pin

Page	Item	Change Results
57	3.4	The circuit structure of "Classification I" of "3.4 I/O circuit type" should be corrected as indicated by the shaded below. (Error)  (Correct) 
103	6.4.5	The shaded description should be added to " [bit7] MTIF (Main clock Timer Interrupt Flag) Main timer interrupt flag". (Correct) The MTIF bit is not set during return from the standby mode (power shut-down) because the internal reset is generated.
108	6.4.7	The shaded description should be added to "Note" of "6.4.7 CSTBR: (Clock STaBilization select Register) Clock Stabilization Select Register". (Correct) (Note) Note that if the clock supervisor function is enabled and the failure detection cycle is shorter than the cycle set in this register, a failure is detected during the oscillation stabilization wait time. The period of the failure detection cycle is as follow. 2^{12} x CR Oscillation time = approx. 40.96ms
115	6.4.14	The description of "[bit4] Reserved" should be corrected as indicated by the shaded below. (Error) Always write "1" to this bit. (Correct) Writing has no effect on operation.

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>117</td><td>6.4.15</td><td>
The description of " • CCSSCCR1: Address 052A$_H$ (Access: Byte, Half-word, Word)" should be deleted as indicated by the shaded below.

(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 colspan="3">RATESEL[2:0]</td><td colspan="5">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,W0</td><td>R,W0</td><td>R,W0</td><td>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R/W0</td><td>R/W0</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 colspan="3">RATESEL[2:0]</td><td colspan="5">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>R0,WX</td><td>R0,WX</td><td>R0,WX</td><td>R/W0</td><td>R/W0</td><td>Attribute</td></tr>
</table>
</td></tr>
<tr><td>121</td><td>6.4.18</td><td>
The description of "[bit7 to bit0] GRLP (clock GeaR LooP number select): Clock gear repeat time selection" should be deleted as indicated by the shaded below.

(Error)

GRLP	Number of steps
0000_0000	0
0000_0001	1
0000_0010	2
…	……
1111_1101	254
1111_1110	255
1111_1111	256

(Correct)

GRLP	Number of steps
0000_0000	1
0000_0001	2
0000_0010	3
…	……
1111_1101	254
1111_1110	255
1111_1111	256
</td></tr>
<tr><td>127</td><td>6.5.1</td><td>
The description of "Note" should be corrected as indicated by the shaded below.

(Error)

· During debug booting (E_DBCR:PLOCK =1), PLL cannot be stopped because MDI communication requires constant PLL clock supply.

(Correct)

· In debug operation, PLL cannot be stopped because MDI communication requires constant PLL clock supply.
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>128</td><td>6.5.1</td><td>The shaded description should be added to the "Note:" of "Clock control PLL clock frequency" table as shown below.

(Error)
(Note) PLLCR, or CCPSDIVR and CCPLLFBR, must be configured properly not to exceed the frequency (max).

(Correct)
(Note) ·PLLCR, or CCPSDIVR and CCPLLFBR, must be configured properly not to exceed the frequency (max).
·Set DIVR2, PICD for the peripheral clock so as not to exceed 40MHz.</td></tr>
<tr><td>128</td><td>6.5.1</td><td>The shaded description should be added to the "Note:" of " Microcontroller section clock control SSCG clock frequency" table as shown below.

(Error)
(Note) CCPSDIVR, or CCSSFBR0 and CCSSFBR1, must be configured properly not to exceed the frequency (max).

(Correct)
(Note) ·CCPSDIVR, or CCSSFBR0 and CCSSFBR1, must be configured properly not to exceed the frequency (max).
·Set DIVR2 for the peripheral clock so as not to exceed 40MHz.</td></tr>
<tr><td>128</td><td>6.5.1</td><td>The table should be corrected as indicated by the shaded below.

(Error)
<table>
<tr><th colspan="2">CCSSCCR1.RATESEL[2:0]
SSSCR1.RATESEL[2:0]</th><th colspan="3">CCSSFBR0.NDIV[5:0]
SDIVCR0.NDIV[5:0]</th></tr>
<tr><td>Modulation rate</td><td>Setting value</td><td>Division ratio range</td><td>Setting value lower limit</td><td>Setting value upper limit</td></tr>
</table>
(Correct)
<table>
<tr><th colspan="2">CCSSCCR1.RATESEL[2:0]</th><th colspan="3">CCSSFBR0.NDIV[5:0]</th></tr>
<tr><td>Modulation rate</td><td>Setting value</td><td>Division ratio range</td><td>Setting value lower limit</td><td>Setting value upper limit</td></tr>
</table></td></tr>
<tr><td>136</td><td>6.5.7</td><td>The description should be corrected as indicated by the shaded below.

(Error)
Also, during MDI high-speed communication (E_DBCR.PLOCK=1), and when CSELR.PCEN is cleared, control is in effect so that PLL reference clock continues to be supplied.

(Correct)
Also, during MDI high-speed communication, and when CSELR.PCEN is cleared, control is in effect so that PLL reference clock continues to be supplied.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>136</td><td>6.5.7</td><td>The description should be deleted as indicated by the shaded below.

(Error)
If values needs to be modified while debugging, for example, rewrite PLL-related registers while monitoring the E_DBCR.PLOCK value to make sure E_DBCR.PLOCK is always "0".</td></tr>
<tr><td>137</td><td>6.5.8</td><td>The description of “6.5.8 About PMU clock (PMUCLK)" should be deleted as indicated by the shaded below.

(Error)
(2) G divider must be set up to 4 times or more of the peripheral clock (PCLK1) frequency.

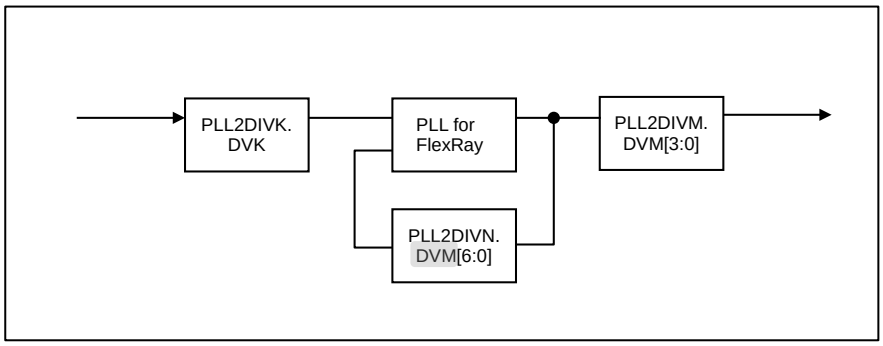
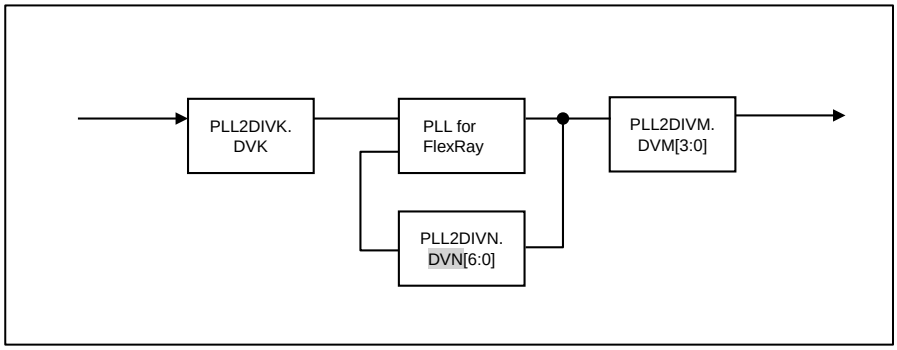
(Correct)
(2) G divider must be set up to 4 times or more of the peripheral clock (PCLK1) period.

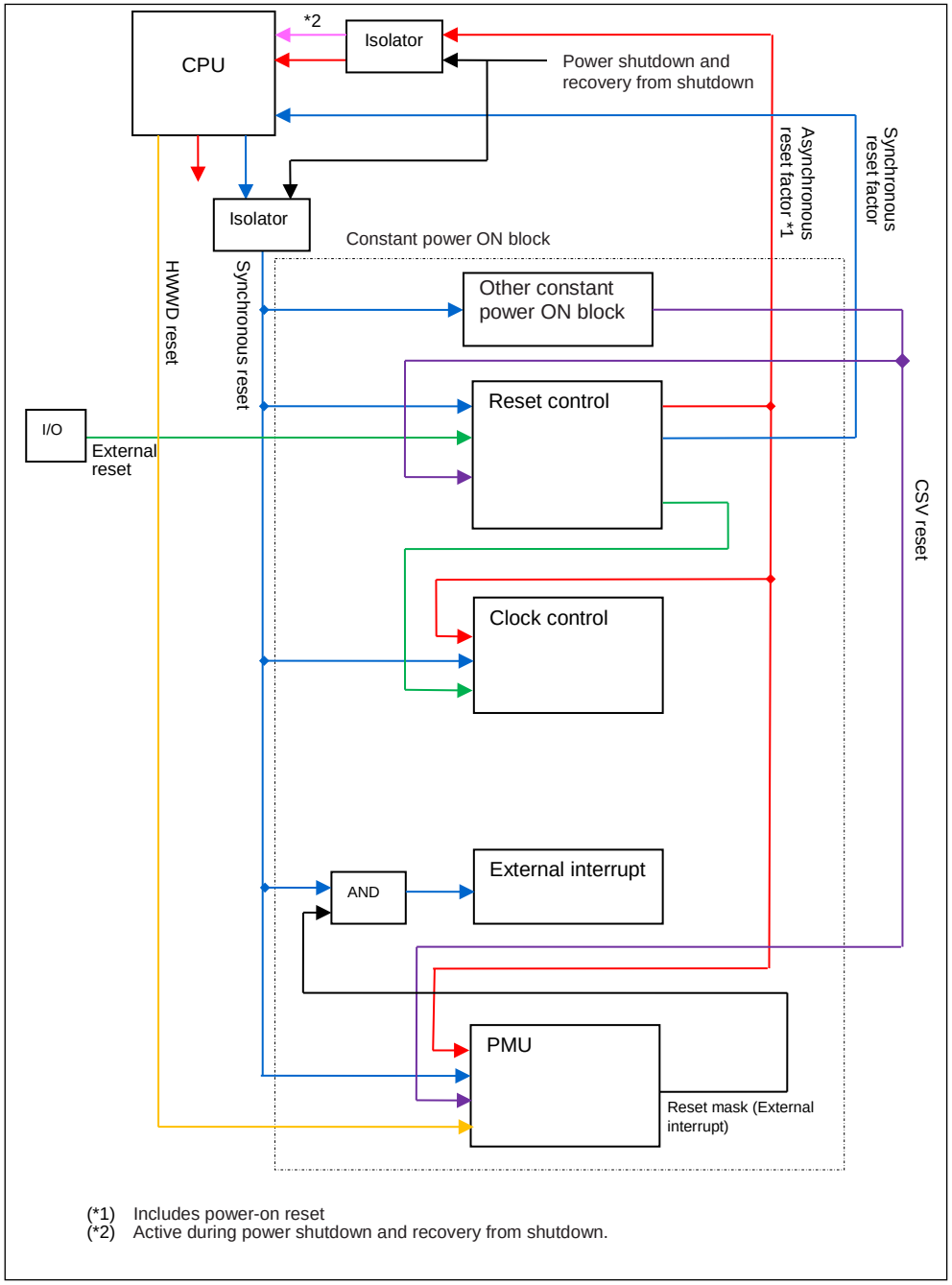
*The relevant part is two in this page.</td></tr>
<tr><td>147</td><td>7.4.7</td><td>The description of " 7.4.7 CLKR2:FlexRay PLL clock output control register" should be corrected as indicated by the shaded below.

(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>FPOVF</td><td>FPOVIR</td><td>FPOVIE</td><td colspan="2">Reserved</td><td>PLL2EN</td><td>CLKS1</td><td>CLKS0</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,WX</td><td>R(RM1),W</td><td>R/W</td><td>R/W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</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>FPOVF</td><td>FPOVIR</td><td>FPOVIE</td><td colspan="2">Reserved</td><td>PLL2EN</td><td>CLKS1</td><td>CLKS0</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,WX</td><td>R,W</td><td>R/W</td><td>R/W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R/W</td><td>Attribute</td></tr>
</table></td></tr>
<tr><td>147</td><td>7.4.7</td><td>The description of " [bit6] FPOVIR: FlexRay PLL alarm interrupt request flag" should be deleted as indicated by the shaded below.

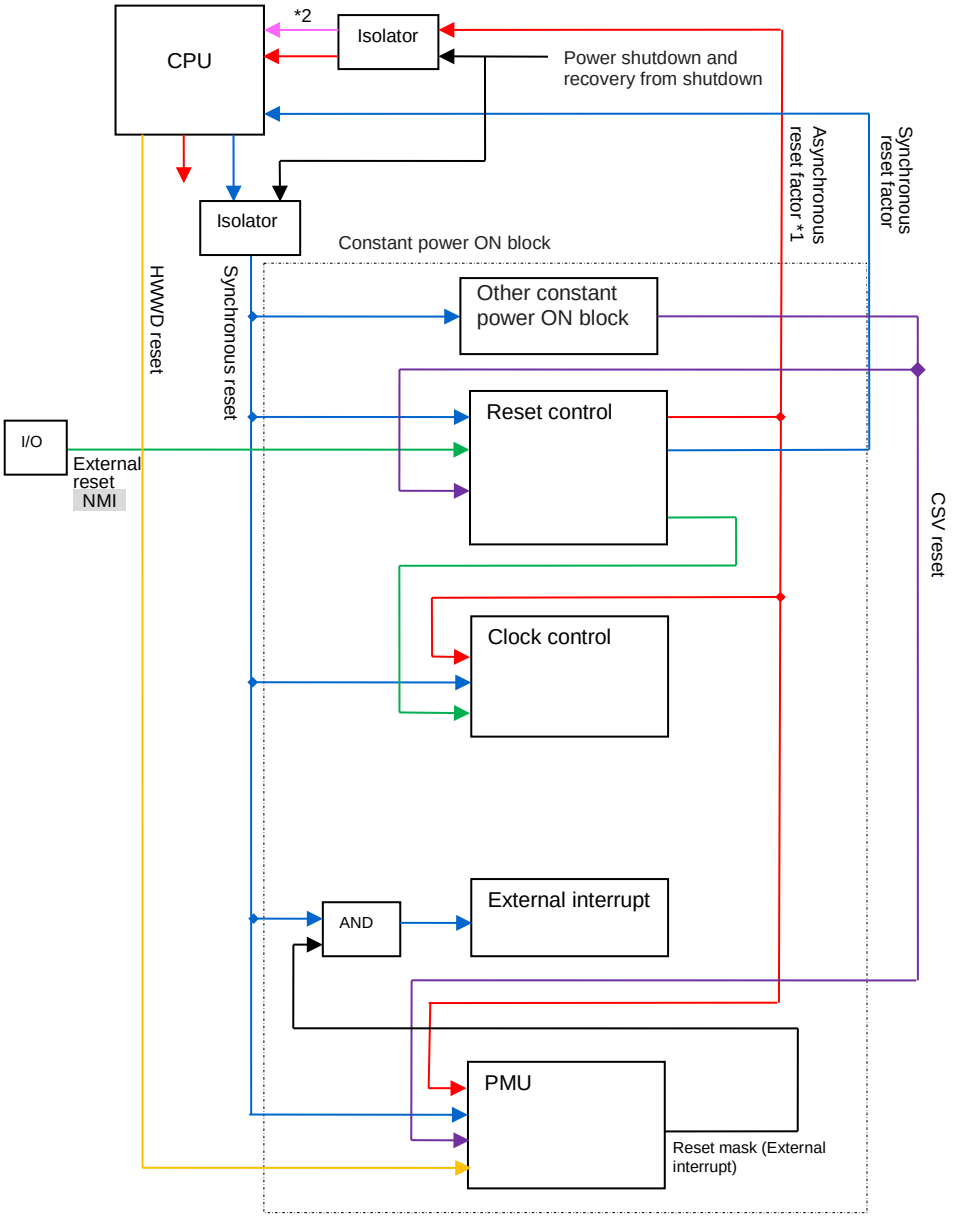
(Error)
While this bit is set to "1" and FlexRay PLL alarm interrupt request (FPOVIE) is set to "1", FlexRay PLL alarm interrupt is generated. "1" will be read during read/modify/write instructions.

(Correct)
While this bit is set to "1" and FlexRay PLL alarm interrupt request (FPOVIE) is set to "1", FlexRay PLL alarm interrupt is generated.</td></tr>
</table>

Page	Item	Change Results
147	7.4.7	The shaded description should be added to "[bit2] PLL2EN (FlexRay PLL selection enabled)". (Error) • It is prohibited to change the FlexRay PLL operation enable bit (PLL2EN) when CLKPLL (FlexRay) is selected as the clock source (CLKS[1:0]=10). (Correct) • It is prohibited to change the FlexRay PLL operation enable bit (PLL2EN) when CLKPLL (FlexRay) is selected as the clock source (CLKS[1:0]=10). • Set "0" to this bit before activating stand-by mode.
149	7.5	The figure of "7.5 Recommended Setting" should be corrected as indicated by the shaded below. (Error)  (Correct) 
153	7.8	The description of "■ Clock Automatic Gear" should be added as indicated by the shaded below. (Error) (4) Wait for the PLL stabilization wait time. (Correct) (4) Wait for the PLL stabilization wait time. The Stabilization wait time is 200µs.

Page	Item	Change Results
156	8.3	"Figure 8.3-1 Configuration Diagram of Reset" should be added as indicated by the shaded below. (Error) 

Page	Item	Change Results
156	8.3	(Correct)



(*1) Includes power-on reset
(*2) Active during power shutdown and recovery from shutdown.

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>163</td><td>8.4.4</td><td>The description of "Note" of "[bit7] PMUST (Power Management Unit STatus)" should be corrected as indicated by the shaded below.

(Error)
This register is initialized by a power-on reset, internal low-voltage reset, reset by simultaneous RSTX/NMIX asserts, and hardware watchdog timer reset.

(Correct)
This bit is just initialized by the power-on reset, the low-voltage detection, and the simultaneous assert of RSTX and NMIX only; is not initialized by the other reset factors. Therefore, confirm the reset factors before judging whether the status is the retuning from the power shut-down mode or not.</td></tr>
<tr><td>181</td><td>9.4.5</td><td>The description of " • PMUINTF2: Address 0596_H (Access: Byte, Half-word, Word)" should be corrected as indicated by the shaded below.

(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>Reserved</td><td>NIF</td><td>MTIF</td><td>Reserved</td><td colspan="4">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>R0(RM1),W0</td><td>R(RM1),W0</td><td>R(RM1),W0</td><td>R0(RM1),W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,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>Reserved</td><td>NIF</td><td>MTIF</td><td>Reserved</td><td colspan="4">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>R0(RM1),W0</td><td>R(RM1),W</td><td>R(RM1),W</td><td>R0(RM1),W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>R0,W0</td><td>Attribute</td></tr>
</table></td></tr>
<tr><td>182</td><td>9.4.5</td><td>The shaded description should be added to "[bit5] MTIF (Main Timer Interrupt Flag)".

(Correct)
The MTIF bit is not set during return from the standby mode (power shut-down) because the internal reset is generated.</td></tr>
<tr><td>204</td><td>11.3.1</td><td>The description of "Note" of "[bit6] MM (Main clock Missing)" should be corrected as indicated by the shaded below.

(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>

Page	Item	Change Results
206	11.4	The description of "Note" should be corrected as indicated by the shaded below. (Error) (Note) Please do not enable the PLL oscillation operation when the main clock is replaced the CR oscillator and it operates after detecting the main clock stop. (Correct) (Note) Please do not enable the PLL/SSCG oscillation operation when the main clock is replaced the CR oscillator and it operates after detecting the main clock stop.
219	13.4.1	The description should be corrected as indicated by the shaded below. (Error) If a power-on reset or a drop in the internal voltage is detected, the LVD_F bit is set to "1". (Correct) If a drop in the internal voltage is detected, the LVD_F bit is set to "1".
223	14.4.2	The description of "Note" of "[bit7] LVD5F_PD (Low Voltage Detect 5v Fall Power Down)" should be corrected as indicated by the shaded below. (Error) (Notes) When power down is changed from enabled to disabled (operation start) in this bit setting, enable the detection (OE=0) after 100 µs elapse. If the detection is enabled before 100 µs elapse, a detection flag may be set. (Correct) (Notes) When power down is changed from enabled to disabled (operation start) in this bit setting, enable the detection (LVD5F_OE=0) 100µs after setting LVD5F_OE=1. If the detection is enabled before 100 µs elapse, a detection flag may be set.
226	15.5	The shaded description should be added to "■ Recovering From Stop Mode". (Correct) The NMI request is not accepted during return from the standby mode (shut-down) because the internal reset is generated.
245	17.4.6	The description of "Note" of "[bit10 to bit0] D10 to D0: Single bit error generation address bits" should be corrected as indicated by the shaded below. (Error) (Absolute address) = (0001_4000_H) + (Offset address set with SEEARA + 2'b00) (Correct) (Absolute address) = (0000_4000_H) + (Offset address set with SEEARA + 2'b00)

Page	Item	Change Results
246	17.4.7	The description of "Note" of "[bit10 to bit0] D10 to D0: Single bit error generation address bits" should be corrected as indicated by the shaded below. (Error) (Absolute address) = (0001_4000$_H$) + (Offset address set with DEEARA + 2'b00) (Correct) (Absolute address) = (0000_4000$_H$) + (Offset address set with DEEARA + 2'b00)
248	17.4.9	The description of "Note" of "[bit10 to bit0] D10 to D0: Fake error generation address setting bits" should be corrected as indicated by the shaded below. (Error) (Absolute address) = (0001_4000$_H$) + (Offset address set with EFEARA + 2'b00) (Correct) (Absolute address) = (0000_4000$_H$) + (Offset address set with EFEARA + 2'b00)
276	18.4.16	The description of " [bit7] TEIE: Interrupt enable bit resulting from a diagnosis error" should be corrected as indicated by the shaded below. (Error) This bit is used to enable an interrupt resulting from a RAM diagnosis error for XBS RAM. (Correct) This bit is used to enable an interrupt resulting from a RAM diagnosis error for Backup RAM.
293	19.3	The description of "Note" of "19.3 Configuration" should be deleted as indicated by the shaded below. (Error) DMA controller 0x00000C00 - 0x00000DFF * (Correct) DMA controller 0x00000C00 - 0x00000DFF
296	19.4.1	The description of "Note" of "[bit1] CNER: Control parity error" should be corrected as indicated by the shaded below. (Error) (Note) This bit is not updated while this bit is "1", any one of the AER bits is "1", or the CNER bit is "1". (Correct) (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".

Page	Item	Change Results
299	19.4.2	The description of " [bit7 to bit4] DEN3 to DEN0: Data error" should be corrected as indicated by the shaded below. (Error) If DER[3] is "0", parity in bit7 to bit0 of the data bus is properly generated. (Correct) If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated.
299	19.4.2	The description of "[bit3 to bit0] AEN3 to AEN0: Address error" should be corrected as indicated by the shaded below. (Error) If AER[1] is "0", parity in bit23 to bit16 of the address bus is properly generated. (Correct) If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.
300	19.4.2	The description of “Note" of” [bit11] CEN: Control error" should be corrected as indicated by the shaded below. (Error) (Note) For RBEN=0, APBEN=0, and AHBEN=0, this bit is invalid and the same operation as when "0" is set in this bit occurs. (Correct) (Note) For RBEN=0 and APBEN=0, this bit is invalid and the same operation as when "0" is set in this bit occurs.
301	19.4.2	The description of " [bit7 to bit4] DEN3 to DEN0: Data error" should be corrected as indicated by the shaded below. (Error) If DER[3] is "0", parity in bit7 to bit0 of the data bus is properly generated. (Correct) If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated
301	19.4.2	The description of "Note" of " [bit7 to bit4] DEN3 to DEN0: Data error" should be corrected as indicated by the shaded below. (Error) (Notes) • For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid and the same operation as when "0000" is set in these bits occurs. (Correct) (Notes) • For RBEN= 0 and APBEN= 0, these bits are invalid and the same operation as when "0000" is set in these bits occurs.

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>301</td><td>19.4.2</td><td>The description of "[bit3 to bit0] AEN3 to AEN0: Address error" should be corrected as indicated by the shaded below.

(Error)
If AER[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.

(Correct)
If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated</td></tr>
<tr><td>301</td><td>19.4.2</td><td>The description of "Note" of " [bit3 to bit0] AEN3 to AEN0: Address error" should be corrected as indicated by the shaded below.

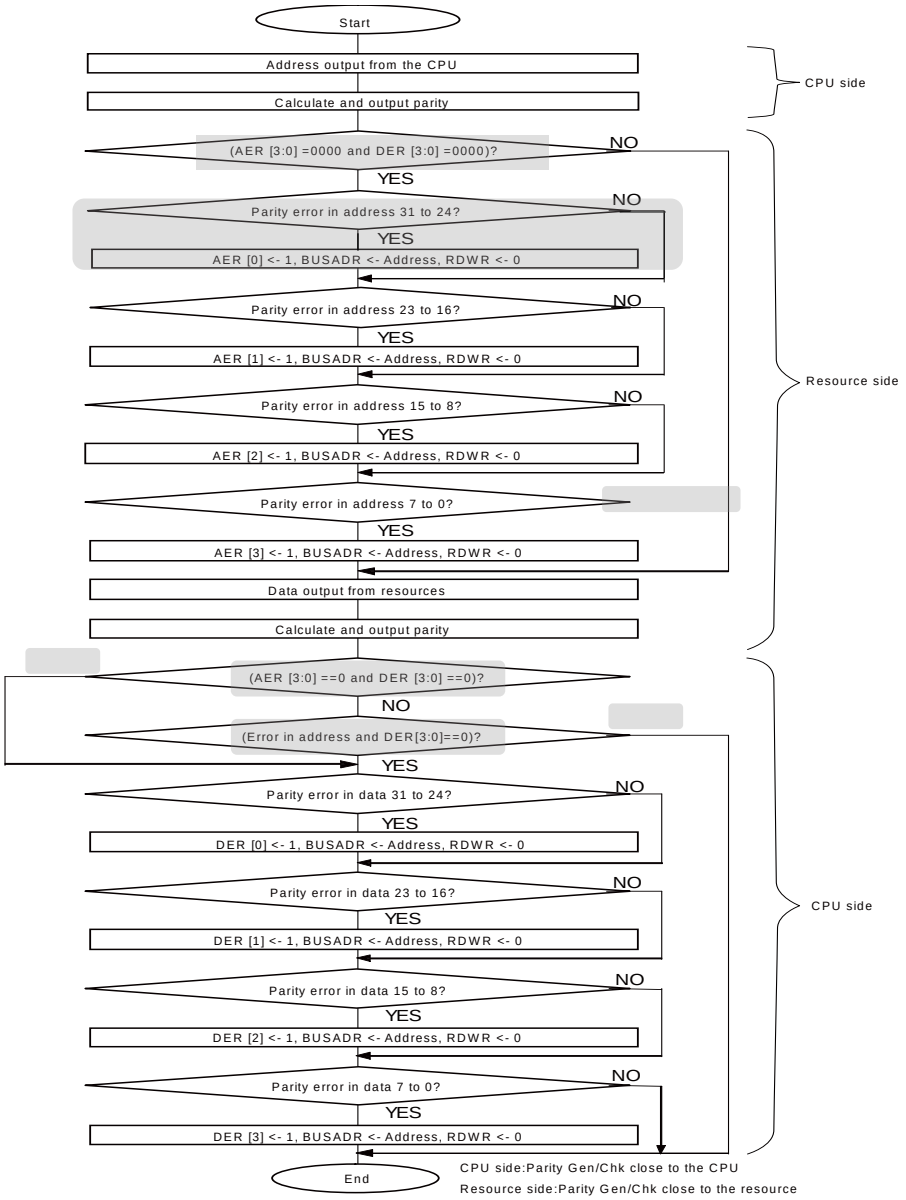
(Error)
(Notes) • For RBEN= 0, APBEN= 0, AHBEN= 0, these bits are invalid and the same operation as when "0000" is set in these bits occurs.

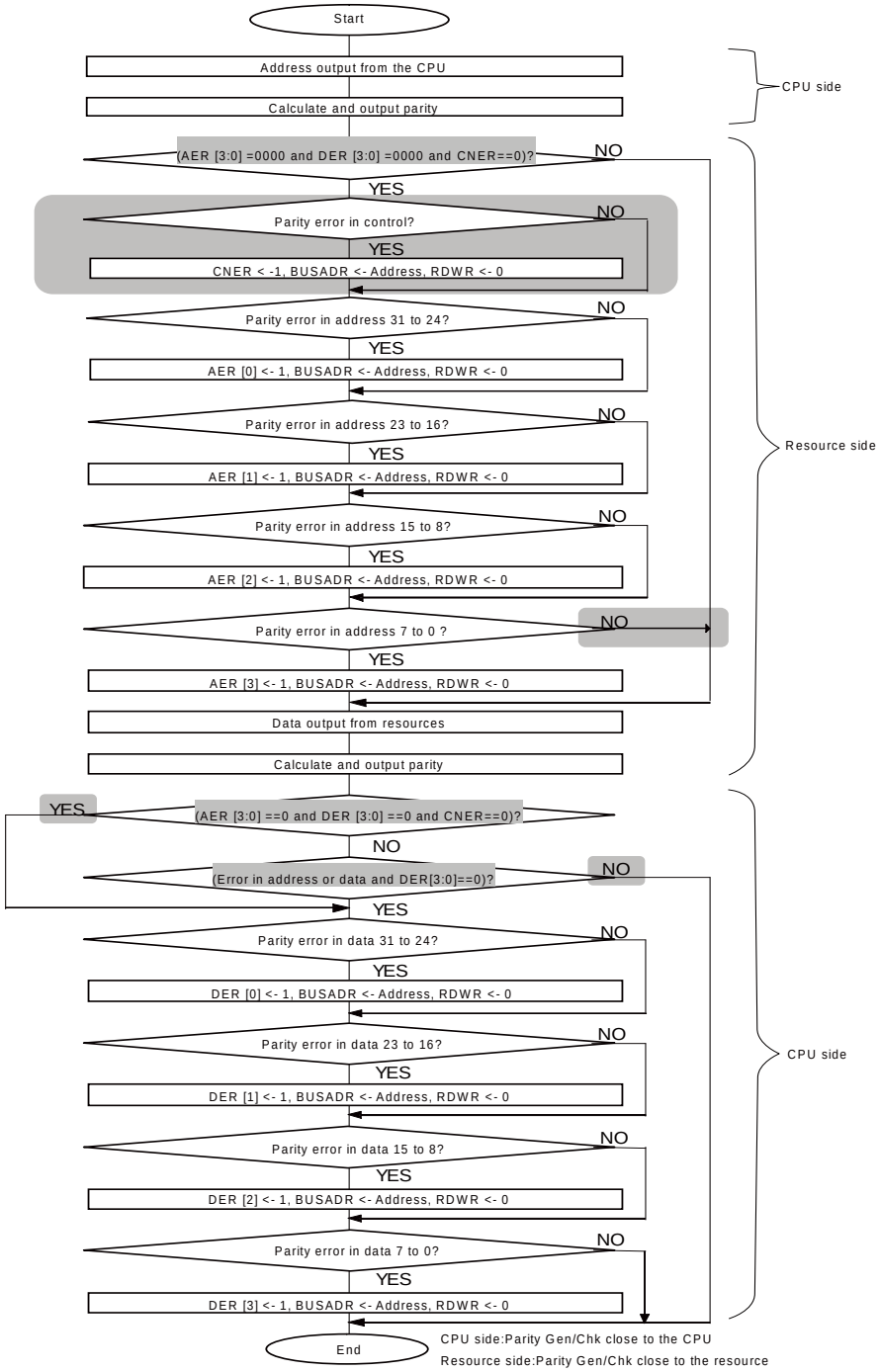
(Correct)
(Notes) • For RBEN= 0 and APBEN= 0, these bits are invalid and the same operation as when "0000" is set in these bits occurs.</td></tr>
<tr><td>303</td><td>19.5.1</td><td>The table should be corrected as indicated by the shaded below.

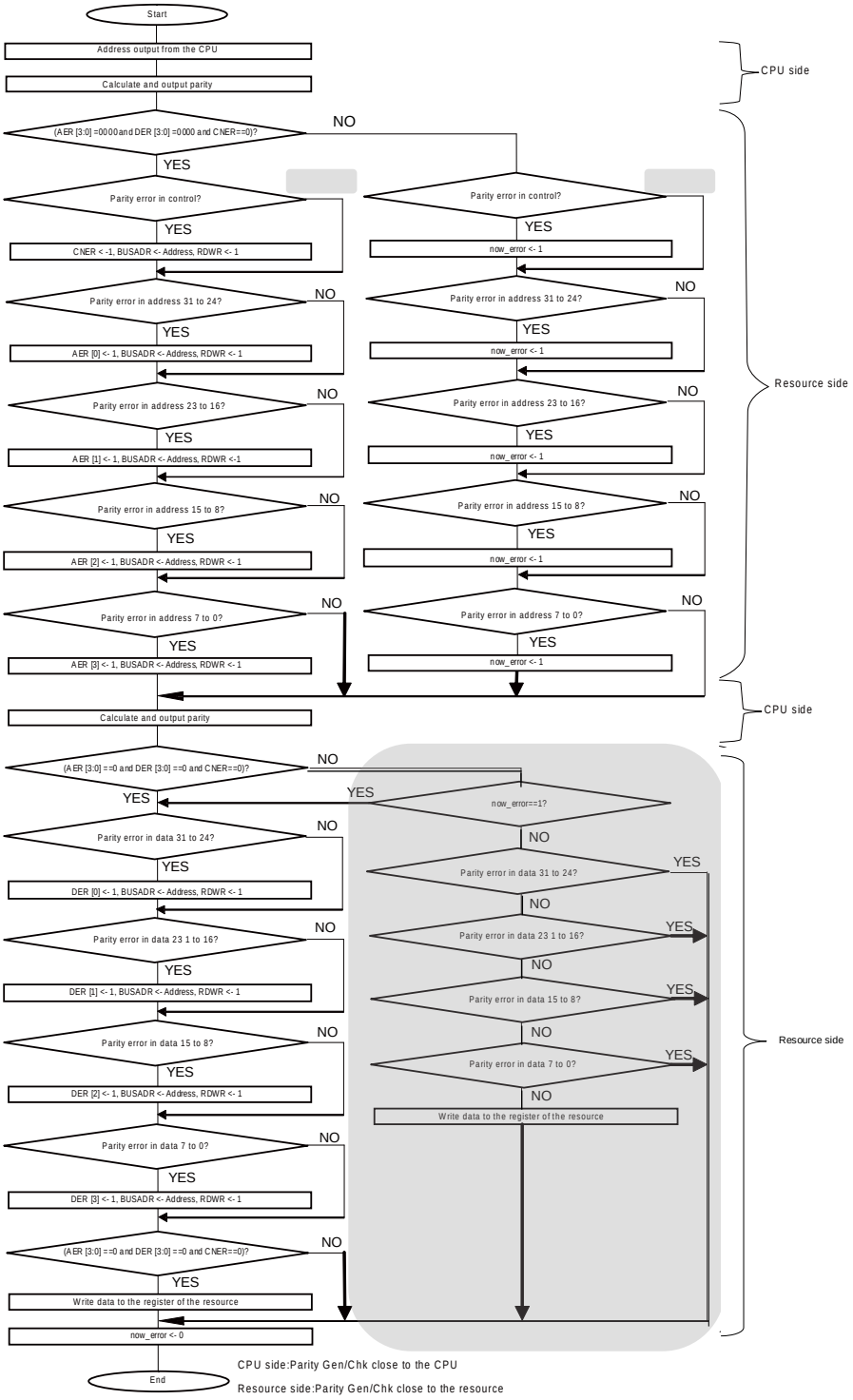
(Error)
<table>
<tr><td rowspan="3">Access size</td><td rowspan="3">Address</td><td colspan="4">BUSDIAGSR0(AHB:On-chip bus)
BUSDIAGSR1(APB:32bit peripheral bus)
BUSDIAGSR2(R-bus:16bit peripheral bus)</td></tr>
<tr><td>DER[0]</td><td>DER[1]</td><td>DER[2]</td><td>DER[3]</td></tr>
<tr><td>Data bit31-24</td><td>Data bit23-16</td><td>Data bit15-8</td><td>Data bit7-0</td></tr>
</table>

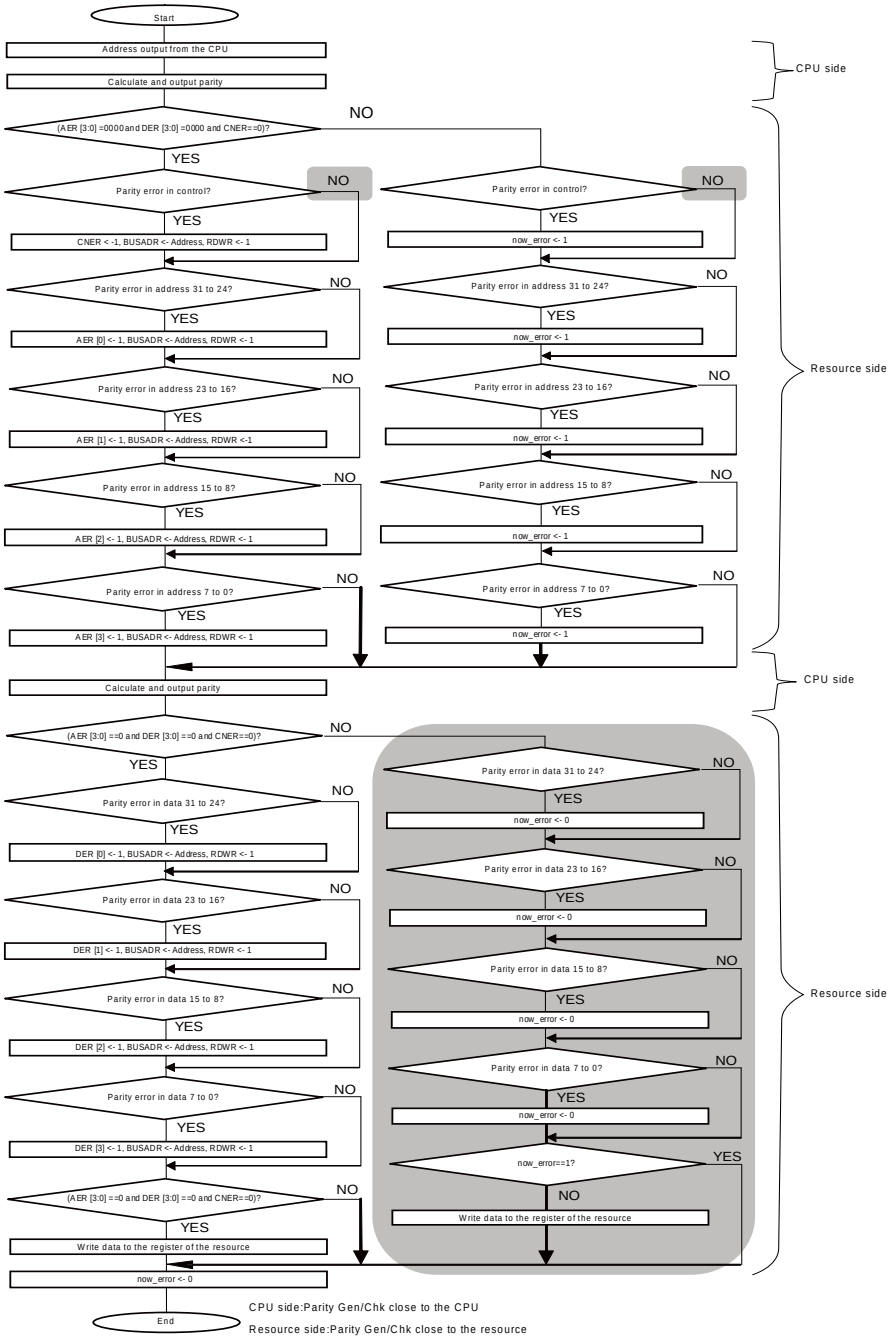
(Correct)
<table>
<tr><td rowspan="3">Access size</td><td rowspan="3">Address</td><td colspan="4">BUSDIGSR0(AHB:On-chip bus)
BUSDIGSR1/3(APB:32bit peripheral bus)
BUSDIGSR2/4(R-bus:16bit peripheral bus)</td></tr>
<tr><td>DER[0]</td><td>DER[1]</td><td>DER[2]</td><td>DER[3]</td></tr>
<tr><td>Data bit31-24</td><td>Data bit23-16</td><td>Data bit15-8</td><td>Data bit7-0</td></tr>
</table>
</td></tr>
</table>

Page	Item	Change Results
305	19.5.3	The description of "19.5.3 Notes" should be deleted as indicated by the shaded below. (Error) • When bus diagnosis test function of R-bus is used When it accesses the following resource area in the state (RBEN was set in "1" by the bus diagnosis test register, and either AEN[3:0] and CEN or DEN[3:0] was set in "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 (Correct) • When bus diagnosis test function of R-bus is used When it accesses the following resource area in the state (RBEN was set in "1" by the bus diagnosis test register, and either AEN[3:0] and CEN or DEN[3:0] was set in "1"), the error is detected. At this time, it might seem that the bus was diagnosed. At this time, the write access to the resource is executed. • Watchdog timer • Delay interrupt • Interrupt controller
306	19.5.4	The description of "• Pseudo data error setting operation" should be corrected as indicated by the shaded below. (Error) 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]. (Correct) 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 DEN[0] by the key code access, and do byte access [31:24].

Page	Item	Change Results
307	19.5.4	"■Bus Diagnosis Operation Flow" should be corrected as indicated by the shaded below. (Error)  	

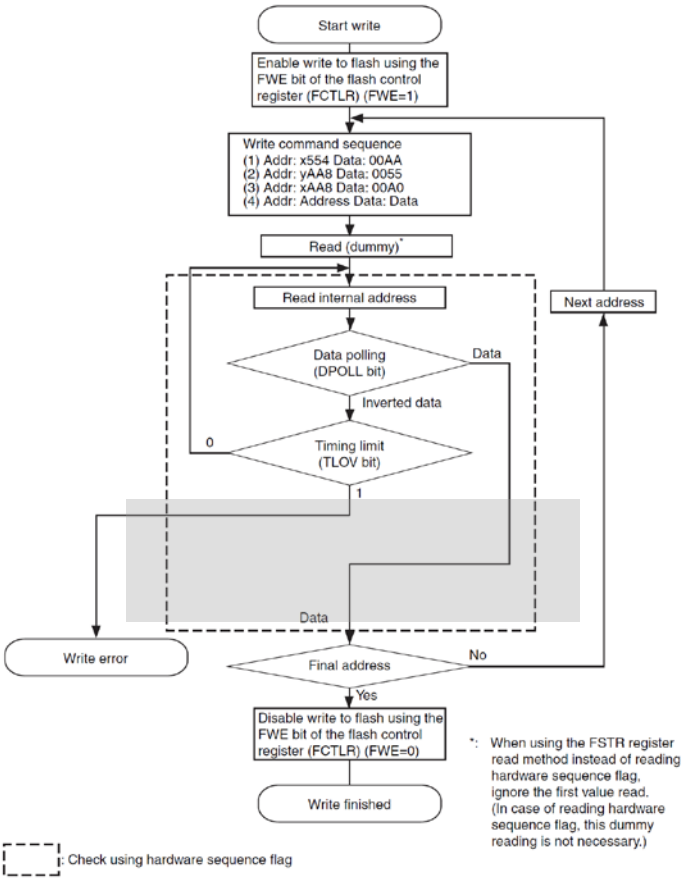
Page	Item	Change Results
307	19.5.4	(Correct) 

Page	Item	Change Results
308	19.5.4	"■Bus Diagnosis Operation Flow" should be corrected as indicated by the shaded below. (Error)  	

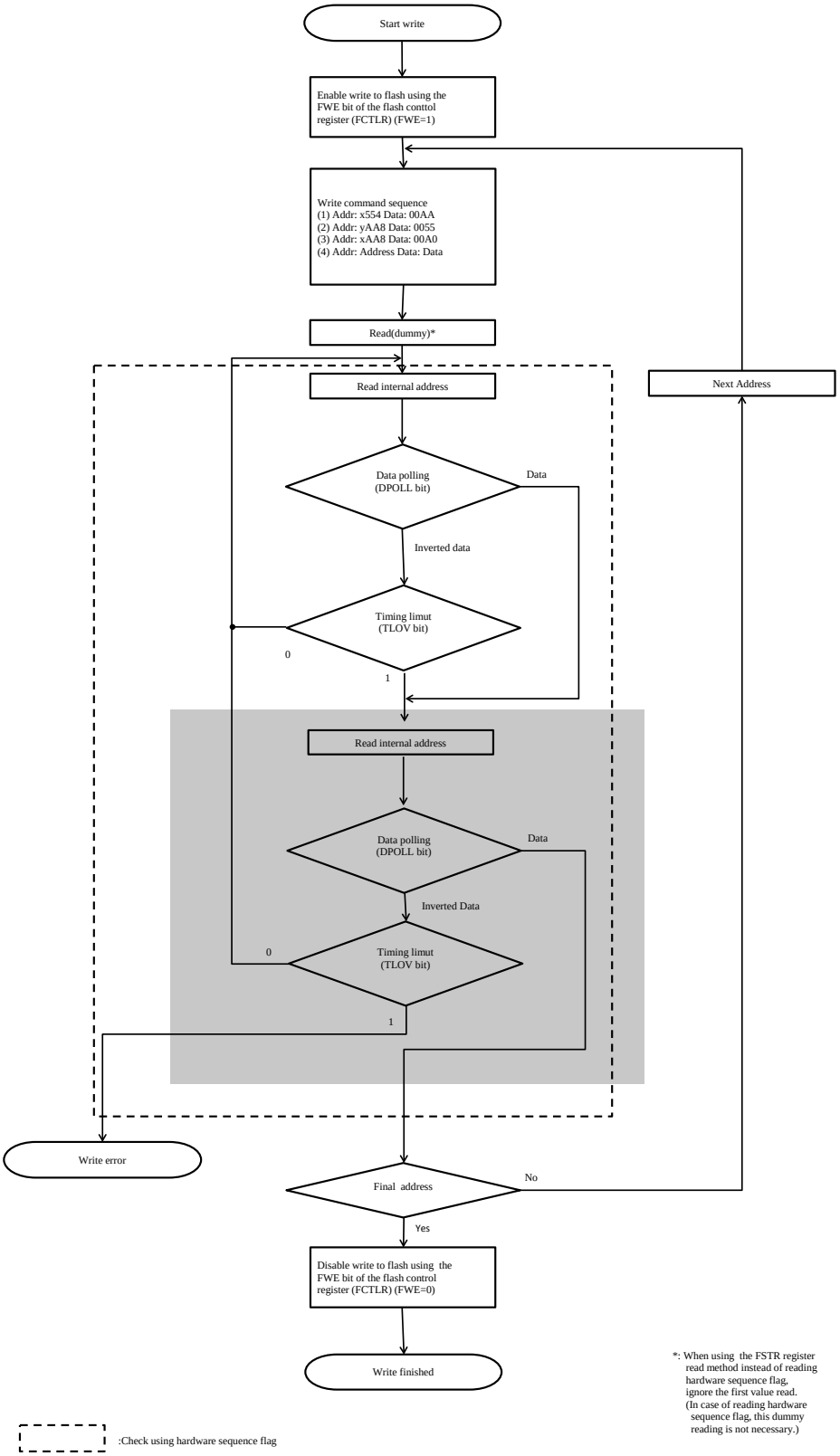
Page	Item	Change Results
308	19.5.4	(Correct) 

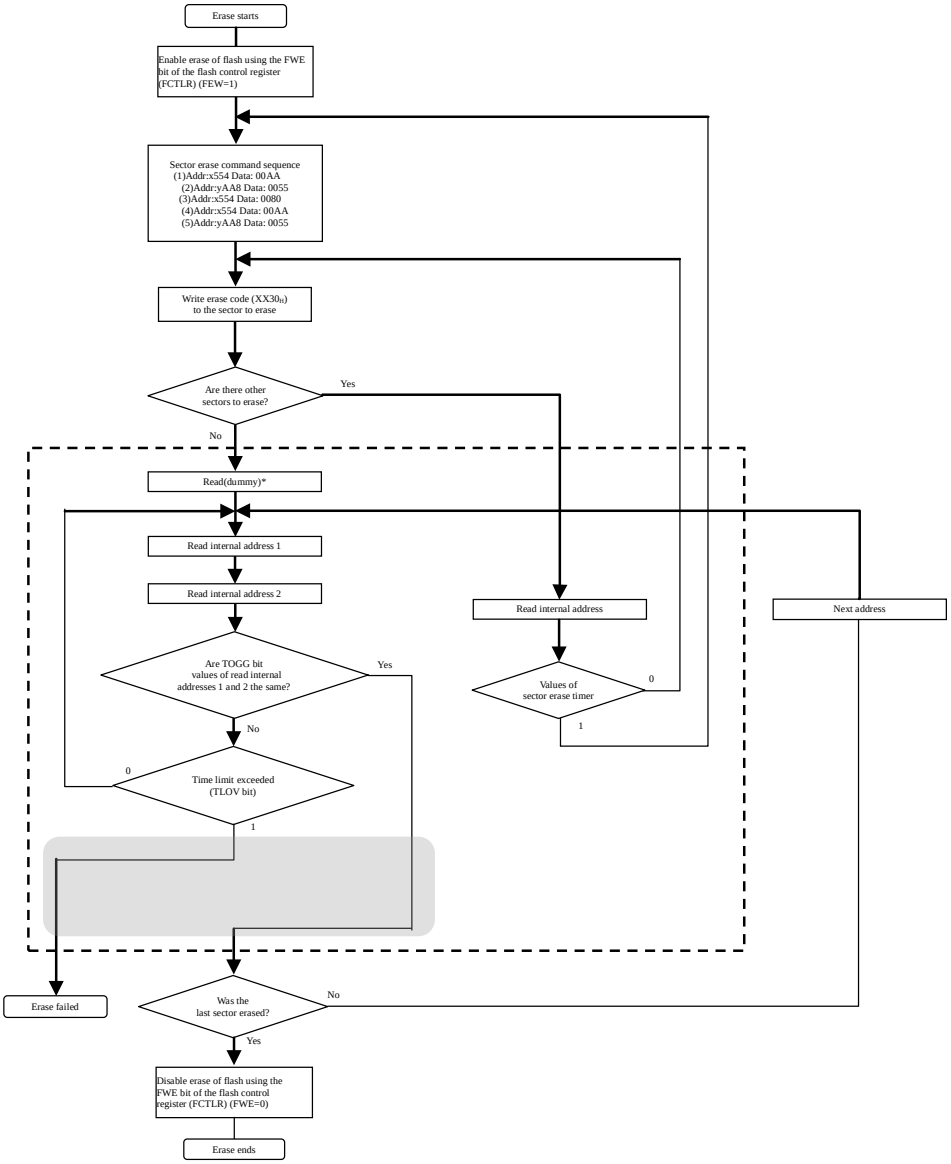
Page	Item	Change Results
309	19.5.4	The description of "Note" of "(3) Processing at error detection" should be corrected as indicated by the shaded below. (Error) (Note) Set the interrupt function address for the interrupt vector before the I flag of CCR is set to "enable". If the program is executed in following steps while the bus signal continuously malfunctions, CPU refers to the interrupt function address immediately after the CCR setting. Example: 1. Define the vector area. 2. Clear the Stack 3. Set the pin register 4. Set the interrupt permission (CCR I flag enabled) * 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. (Correct) (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.
342	23.4.32	The shaded description should be added to "[bit7] EXTNMI (External Non Maskable Interrupt Request) External NMI request". (Correct) NMI request is not retained during return from the standby mode (shut-down) because the internal reset is generated.
374	24.5.9	The description of "24.5.9 Setting when Using the Base Timer TIOA1 Pin" should be corrected as indicated by the shaded below. (Error) If the base timer TIOA1 pin is to be used, it must be set for output if base timer I/O mode 1 prevails or for input otherwise. If the base timer TIOA1 pin is to be used, it must be set for peripheral output if base timer I/O mode 1 prevails (see "■ Peripheral Output Assignment") or for peripheral input otherwise (see "■ Peripheral Input Assignment"). (Correct) If the base timer TIOA1 pin is to be used, it must be set for input if base timer I/O mode 1 prevails or for output otherwise. If the base timer TIOA1 pin is to be used, it must be set for peripheral input if base timer I/O mode 1 prevails (see "■ Peripheral Input Assignment") or for peripheral output otherwise (see "■ Peripheral Output Assignment").

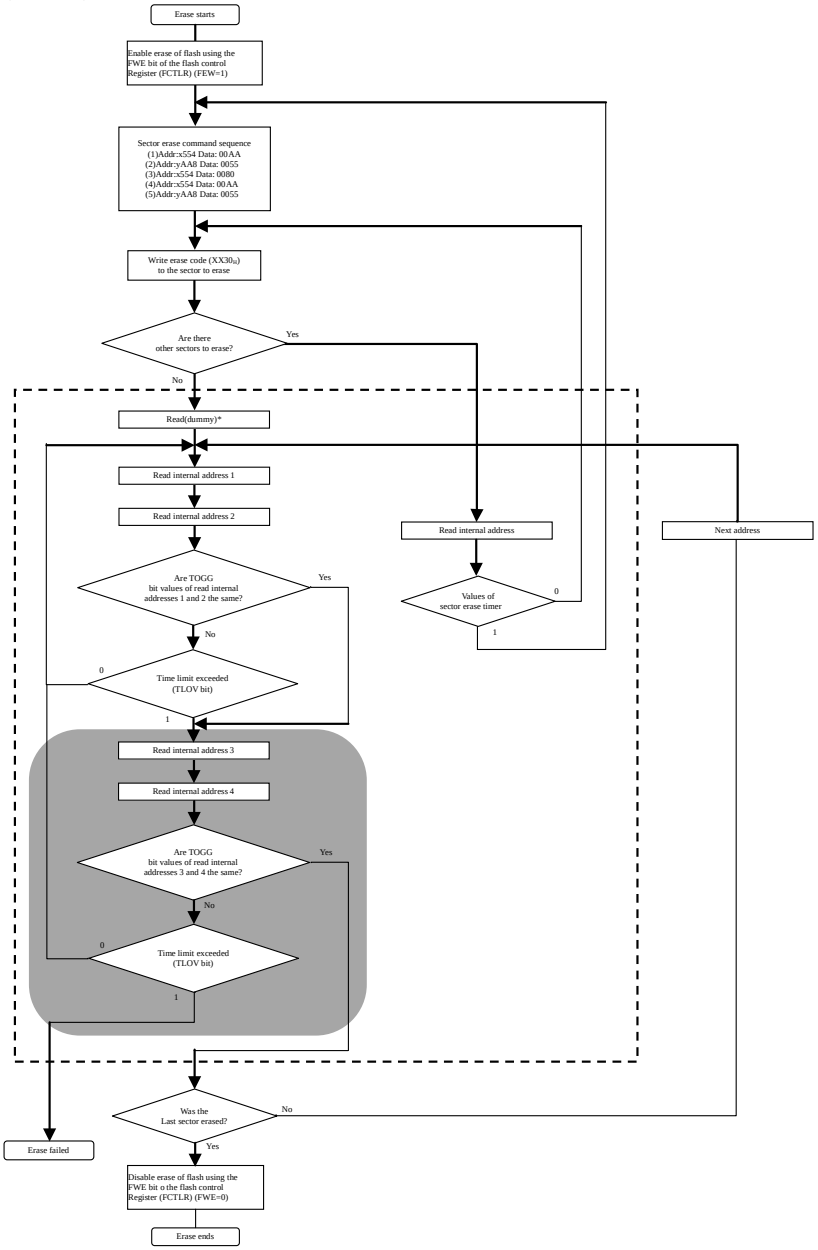
Page	Item	Change Results
386	26.4.1	The description of " • CUCR1:Address 04C4H (Access: Byte, Half-word, Word)" should be corrected as indicated by the shaded below. (Error) 7 6 5 4 3 2 1 0 bit - | - | STRT | - | INT | INTEN 0 0 0 0 0 0 0 0 Initial value R/W0 R0,WX R0,WX R,W R0,WX R0,WX R,W0 R/W Attribute (Correct) 7 6 5 4 3 2 1 0 bit - | - | STRT | - | INT | INTEN 0 0 0 0 0 0 0 0 Initial value R0,W0 R0,WX R0,WX R,W R0,WX R0,WX R,W0 R/W Attribute
453	30.2	The shaded description should be added to "• Error correction code (ECC) security function". (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 if reading data in the state of chip erase/sector erase. To read the data correctly in the state of erase ($FFFF_H$), always write "$FFFF_H$" to read the data.

Page	Item	Change Results
475	30.5.5	"Figure 30.5-2 Write procedure example" should be corrected as indicated by the shaded below. (Error) 

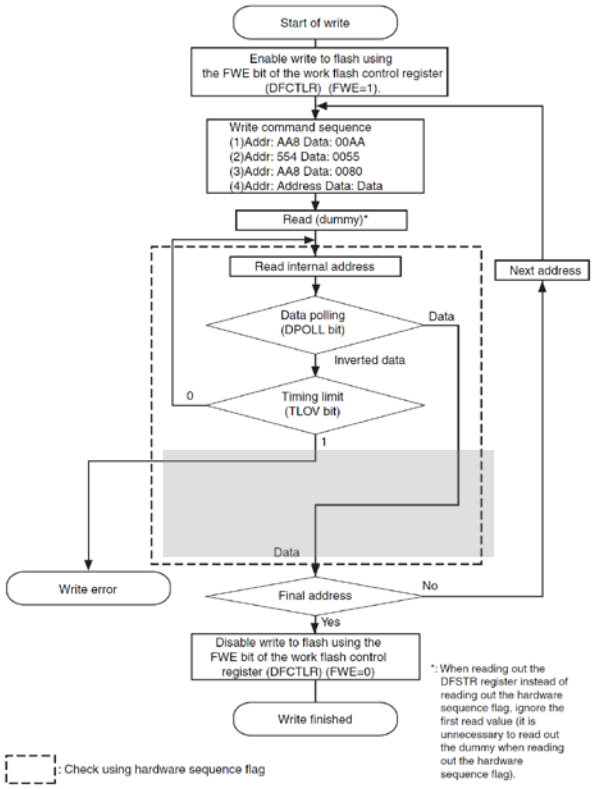
Page	Item	Change Results
475	30.5.5	(Correct)



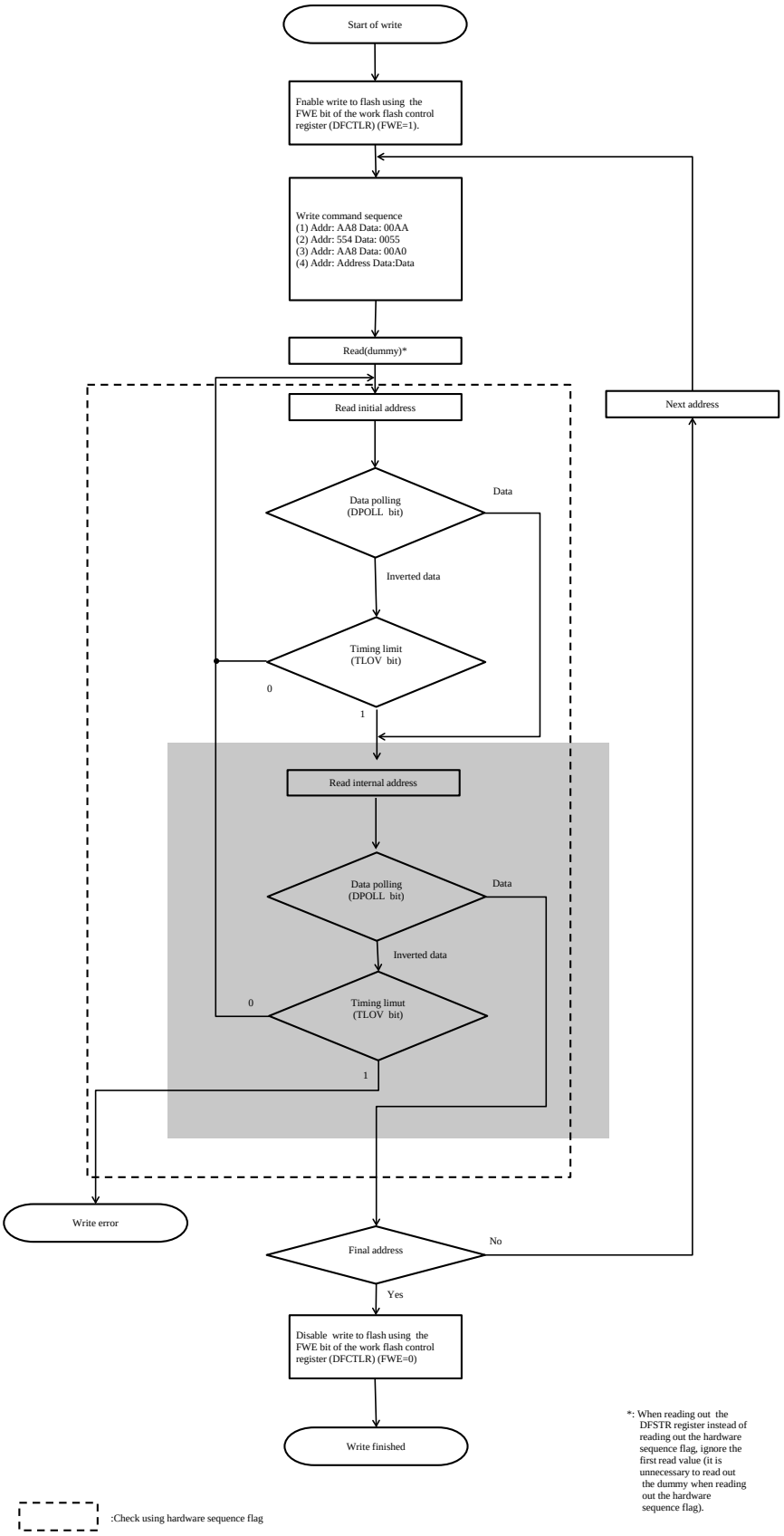
Page	Item	Change Results
477	30.5.7	"Figure 30.5-3 Sector erase procedure example" should be corrected as indicated by the shaded below. (Error) 

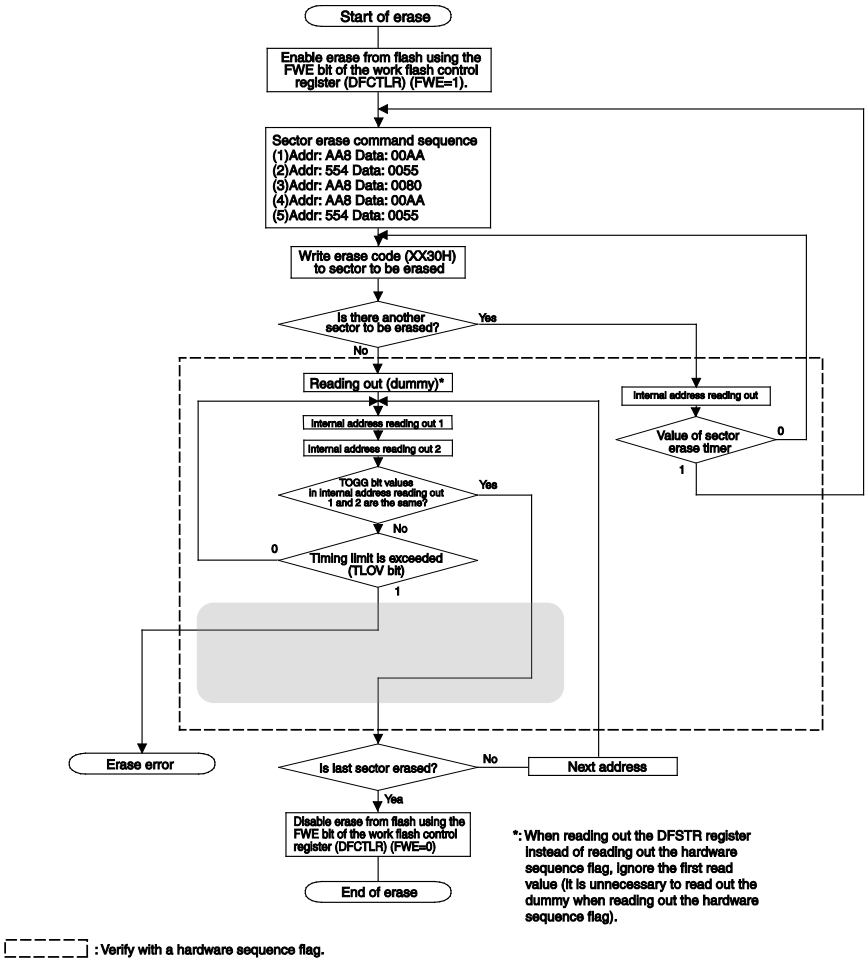
Page	Item	Change Results
477	30.5.7	(Correct) 

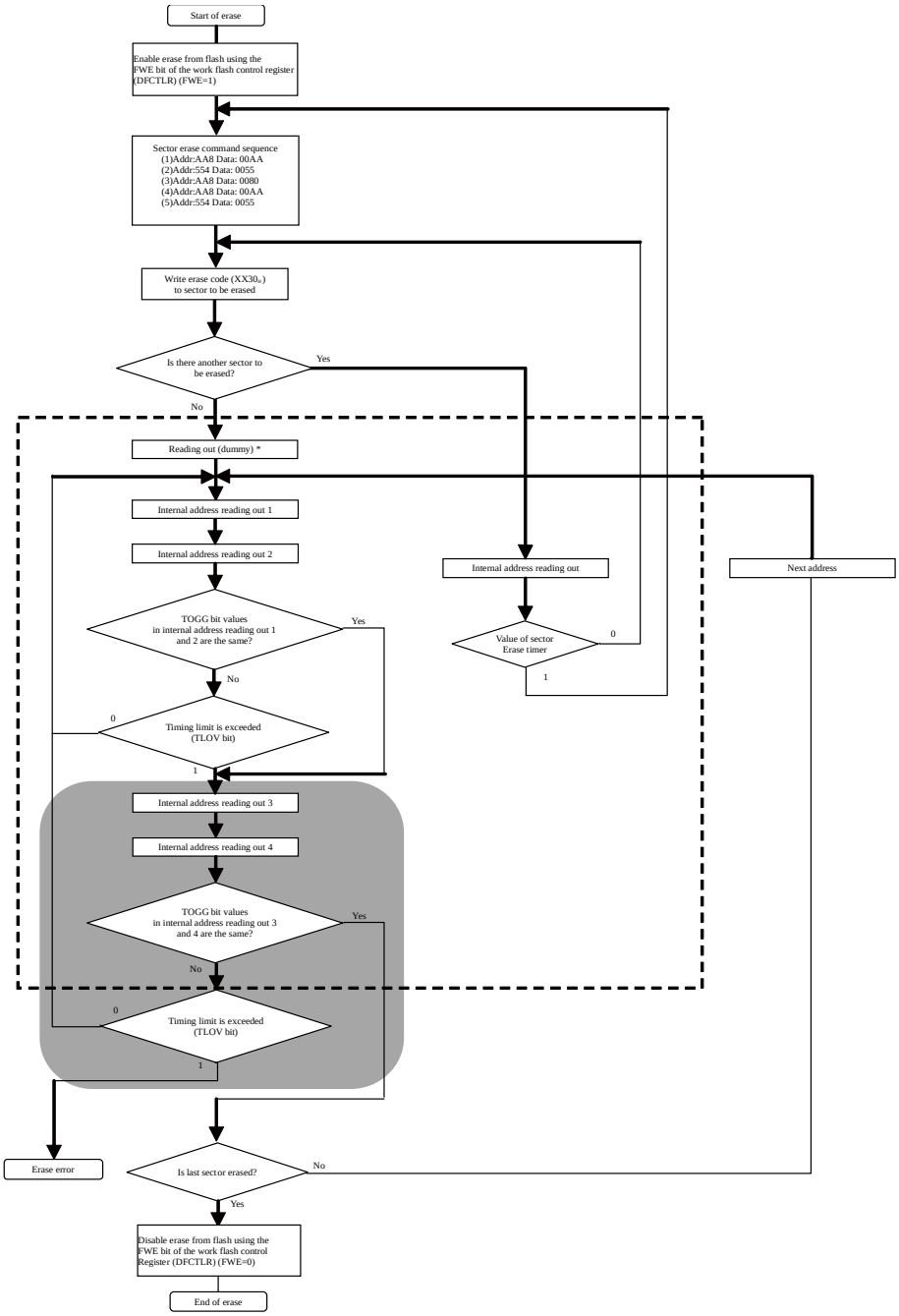
Page	Item	Change Results
478	30.5.7	The description of "Note" of "30.5.7 Sector Erase Command" should be corrected as indicated by the shaded below. (Error) • If a command other than the sector erase command is issued during sector erase including the timeout period, the flash memory returns to the read/reset 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) • The only write commands that can be accepted in a command time-out state are the add erased sector and erase suspend commands. The only command that can be accepted in a sector erase state is the erase suspend command.
483	31.2	The shaded description should be added to "31.2 Features". (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 if reading data in the state of chip erase/sector erase. To read the data correctly in the state of erase ($FFFF_H$), always write "$FFFF_H$" to read the data.

Page	Item	Change Results
497	31.5.5	"Figure 31.5-2 Write procedure example" should be corrected as indicated by the shaded below. (Error) 

Page	Item	Change Results
497	31.5.5	(Correct)



Page	Item	Change Results
500	31.5.7	"Figure 31.5-3 Sector erase procedure example" should be corrected as indicated by the shaded below. (Error) 

Page	Item	Change Results
500	31.5.7	(Correct) 

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>501</td><td>31.5.7</td><td>The description of "Note" of "31.5.7 Sector Erase Command" should be corrected as indicated by the shaded below.

(Error)
• If commands other than sector erase command are issued while erasing sector including the time out period, flash memory becomes read/reset state. In this case, because flash memory is reset, the sector erase command one or multiple prior to command that is issued is invalid. When sector erase is performed, reissue sector erase command from scratch.

(Correct)
• The only write commands that can be accepted in a command time-out state are the add erased sector and erase suspend commands. The only command that can be accepted in a sector erase state is the erase suspend command.</td></tr>
<tr><td>536</td><td>35.4.1.1</td><td>The description of "[bit0] SOE (Serial Output Enable) : Serial data output enable bit" should be corrected as indicated by the shaded below.

(Error)
When this bit is set to "0": The SOT pin functions as a general-purpose I/O port.

(Correct)
When this bit is set to "0": The SOT pin stops serial data output.</td></tr>
<tr><td>537</td><td>35.4.1.2</td><td>The description of bit attribution should be corrected as indicated by the shaded below.

(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">Reserved</td><td>FLSTE</td><td>FRIIE</td><td>FDRQ</td><td>FTIE</td><td>FSEL</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R/W0</td><td>R/W0</td><td>R,W0</td><td>R/W</td><td>R/W</td><td>R(RM1), W</td><td>R/W</td><td>R,W</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">Reserved</td><td>FLSTE</td><td>FRIIE</td><td>FDRQ</td><td>FTIE</td><td>FSEL</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>Initial value</td></tr>
<tr><td>R/W0</td><td>R/W0</td><td>R0,W0</td><td>R/W</td><td>R/W</td><td>R(RM1), W</td><td>R/W</td><td>R,W</td><td>Attribute</td></tr>
</table></td></tr>
</table>

Page	Item	Change Results
541	35.4.1.3	The description of "[bit0] FE1 (FIFO Enable 1) FIFO1 operation enable bit" should be corrected as indicated by the shaded below. (Error) • [UART] [CSIO] [LIN] When FIFO2 is set for the transmission FIFO (FCR1:FSEL="1"), FIFO1 contains data, and UART, CSIO, or LIN is enabled for transmission (SCR:TXE="1"), writing "1" to this bit starts transmission immediately. (Correct) • [UART] [CSIO] [LIN] When FIFO1 is set for the transmission FIFO (FCR1:FSEL="0"), FIFO1 contains data, and UART, CSIO, or LIN is enabled for transmission (SCR:TXE="1"), writing "1" to this bit starts transmission immediately.
543	35.4.1.4	The shaded description should be added to "Note" of "[bit15 to bit8] FBYTE2 (FIFO Byte 2): FIFO2 data count display bits" and "[bit7 to bit0] FBYTE1 (FIFO Byte 1): FIFO1 data count display bits". (Correct) • [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.
560	35.4.3.2	The shaded description should be added to "Note" of “[bit4] AWC: Access width control bit". (Correct) •To use SPI mode with slave mode, set "1" to this bit.
563	35.4.3.3	The shaded description should be added to "Note" of "[bit5] CSFE: Serial chip select format enable bit". (Correct) • When "1" is set to this bit, configure as follows. • Enable the reception FIFO. • Set the hold-delay value to "2" or more. (Set the CSHD7-0 bit of SCSTR1 register to "2" or more.) • Set the data length of a serial chip select to under 9 bits or over 10 bits, communicating to multi slave devices. • Setting the data length of each serial chip selects to under 9 bits or over 10 bits are prohibited. (Example of prohibition) Serial chip select terminal 0 = 9 bits Serial chip select terminal 1 = 10 bits (Example of permission) Serial chip select terminal 0 = 16 bits Serial chip select terminal 1 = 10 bits

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>564</td><td>35.4.3.3</td><td>The shaded description should be added to "[bit4, bit3] WT1, WT0: Data transmission/reception wait select bits".

(Correct)
(Note) • Set "00" to "WT1" and "WT0" bit, when all the conditions listed below are met.
• Using serial chip select function.
• Using SPI mode (SCR:SPI="1").
• Setting "01" to TBYTE register.
• Setting "1" to SCAM bit of SCSCR register.</td></tr>
<tr><td>565</td><td>35.4.3.4</td><td>The description of "• RDR/TDR" should be corrected as indicated by the shaded below.

(Error)
• RDR/TDR: Address 1504_H, 1528_H, $154C_H$, 1570_H (Access: Byte, Half-word, Word)

(Correct)
• RDR/TDR: Address 1504_H, 1528_H, $154C_H$, 1570_H (Access: Half-word, Word)</td></tr>
<tr><td>569</td><td>35.4.3.5</td><td>The description of " • SACSR: Address 1508_H, $152C_H$, 1550_H, 1574_H (Access: Byte, Half-word, Word)" should be corrected as indicated by the shaded below.

(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>TBEEN</td><td>CSEIE</td><td>CSE</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/W</td><td>R/W</td><td>R(RM1), 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>Reserved</td><td>TBEEN</td><td>CSEIE</td><td>CSE</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>R0,W0</td><td>R/W</td><td>R/W</td><td>R(RM1), W</td><td>RX,W0</td><td>RX,W0</td><td>R(RM1), W</td><td>Attribute</td></tr>
</table></td></tr>
<tr><td>569</td><td>35.4.3.5</td><td>The shaded description should be added to "[bit15] STST: Serial test bit".

(Correct)
(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).</td></tr>
<tr><td>575</td><td>35.4.3.8</td><td>The shaded description should be added to "Note" of "[bit9] SCAM: Serial chip select active retain bit".

(Correct)
• Set "0" to this bit, when all the conditions listed below are met.
• Using Master mode (SCR:MS="0").
• Using serial chip select function.
• Using SPI mode (SCR:SPI="1").
• Setting "01" to TBYTE register.
• Setting "00" to WT1 and WT0 bit of ESCR register.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>577</td><td>35.4.3.9</td><td>The shaded description should be added to " 35.4.3.9 Serial Chip Select Timing Register: SCSTR3-0".

(Correct)
(Note)•Setting value of the setup delay time (CSSU7-0) or hold delay time (CSHD7-0) should be satisfied either of the following condition.
Baud rate/2[ns] < Hold delay time[ns] + 3 × Bus clock[ns]
Hold delay time + Setup delay time < Baud rate - 2 × Bus clock[ns]</td></tr>
<tr><td>579</td><td>35.4.3.10</td><td>The shaded description should be added to "Note" of "[bit7 to bit0] TBYTE: Transfer data count display bits".

(Correct)
•Do NOT set "01" to this bit, when all the conditions listed below are met.
•Using Master mode (SCR:MS="0").
•Using serial chip select function.
•Using SPI mode (SCR:SPI="1").
•Setting "1" to SCAM bit of SCSCR register.
•Setting "00" to WT1 and WT0 bits of ESCR register.</td></tr>
<tr><td>581</td><td>35.4.4.1</td><td>The shaded description should be added to "[bit7] UPCL: Programmable clear bit" and "Note" of "[bit7] UPCL: Programmable clear bit".

(Correct)
This bit initializes the internal state of the LIN interface (v2.1).
When this bit is set to "1":
• The LIN interface (v2.1) is directly reset (software reset). In this case, the register settings are maintained. Note that any active transmission or reception is cut off immediately.
• The baud rate generator restarts by reloading the setting value of the BGR register.
• All transmission/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": There is no effect. A read always results in "0".

<table>
<tr><th rowspan="2">UPCL</th><th colspan="2">Programmable Clear Bit</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td>0</td><td>No effect</td><td rowspan="2">"0" always read</td></tr>
<tr><td>1</td><td>Programmable clear</td></tr>
</table>

(Notes) • Execute a programmable clear after disabling interrupts.
• When using a FIFO, disable the FIFO (FCR0:FE2, FE1="0") before you execute a programmable clear.
• Even after you execute a programmable clear (SSR:UPCL="1"), the value of the serial timer register is not cleared.
•Transmission/reception FIFO is not cleared by a programmable clear.</td></tr>
<tr><td>585</td><td>35.4.4.2</td><td>The shaded description should be added to "[bit4] FRE: Framing error flag bit".

(Correct)
(Note) •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.</td></tr>
</table>

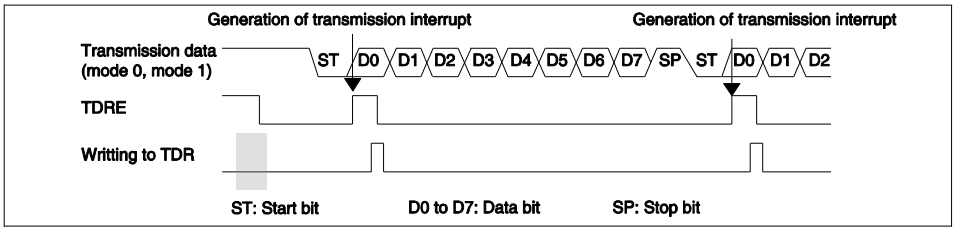
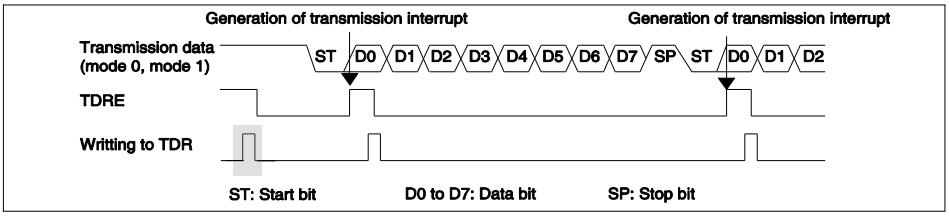
<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>591</td><td>35.4.4.5</td><td>The description of " • SACSR: Address 1508_H, $152C_H$, 1550_H, 1574_H (Access: Byte, Half-word, Word)" should be corrected as indicated by the shaded below.

(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>591</td><td>35.4.4.5</td><td>The description of " [bit14] BST: Reserved" should be corrected as indicated by the shaded below.

(Error)
bit[14] BST: Reserved
Always set this bit to "0".

(Correct)
bit[14] BST : Baud rate setting flag
This bit indicates that the auto baud rate adjustment was executed by "Sync Filed" reception.
This bit is cleared when the fifth falling edge of LIN bus with Sync Field is detected.
<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 effect</td><td>Disable auto baud rate adjustment function.</td></tr>
<tr><td>1</td><td>Enable auto baud rate adjustment function.</td></tr>
</table>
(Notes)
•This bit is fixed to "0" when the auto baud rate adjustment function is disabled (AUTE=0).
•This bit is cleared to "0" by executing programmable clear (SCR:UPCL="1").
•This bit is valid only when Sync Field detection flag is "1"(SACSR:SFD="1").
•Writing has no effect on operation.</td></tr>
<tr><td>594</td><td>35.4.4.5</td><td>The shaded description should be added to "[bit0] TMRE: Serial timer enable bit".

(Correct)
(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")</td></tr>
</table>

Page	Item	Change Results
609	35.4.5.2	The description of "Note" of "[bit2] RDRF: Reception data full flag bit" should be corrected as indicated by the shaded below. (Error) If all of the following conditions are met, SCL becomes "L" after 1-byte data reception, and the SCL state of "L" is canceled when the RDRF bit becomes "0": • Reception FIFO not used • DMA mode enabled (IBCR:DMA="1") • RDRF bit is "1" when reception is in progress at the 2nd or subsequent byte (IBSR:TRX="0") • IBCR:WSEL="0" (Correct) If all of the following conditions are met, SCL becomes "L" after 1-byte data reception, and the SCL state of "L" is canceled when the RDRF bit becomes "0": • Reception FIFO not used • DMA mode enabled (IBCR:DMA="1") • RDRF bit is "1" when reception is in progress at the 2nd or subsequent byte (IBSR:TRX="0") • IBCR:WSEL="1"
617	35.4.5.5	The description of "Note" of "[bit8] TINT: Timer interrupt flag" should be deleted as indicated by the shaded below. (Error) (Notes) • When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
627	35.5.1.4	"Figure 35.5-3 Timing of Transmission Interrupt Flag" should be corrected as indicated by the shaded below. (Error)  (Correct) 

Page	Item	Change Results
654 662 670 678	35.6.2.1 35.6.2.2 35.6.2.3 35.6.2.4	The description of "■ Register Settings" should be corrected as indicated by the shaded below. (Error) *: Bit settings depend on the condition. See Table for details. (Correct) *: Bit settings depend on the condition. See Table 35.6-3 for details.
692	35.6.2.6	The figure name should be corrected as indicated by the shaded below. (Error) Figure 35.6-24 Operation of Serial Chip Selection (Master Transmission (MS="0"), Normal Transfer (SPI="1"), SCINV="0") (Correct) Figure 35.6-24 Operation of Serial Chip Selection (Master Transmission (MS="0"), SPI Transfer (SPI="1"), SCINV="0")
694	35.6.2.6	The figure name should be corrected as indicated by the shaded below. (Error) Figure 35.6-27 Operation of Serial Chip Selection (Master Transmission (MS="0"),Normal Transfer (SPI="1"), SCINV="0") (Correct) Figure 35.6-27 Timing Adjustment (SPI Transfer (SPI="1"), SCINV="0")
Old revision 696 to 698	35.6.2.6	The section of "● Rounding Operation of Chip Selection Pin (Only Valid in Master Mode (SCR:MS=0))" should be deleted.

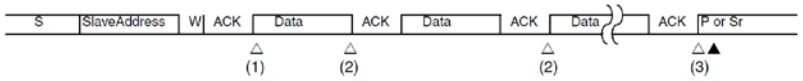
<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>704</td><td>35.7.1.1</td><td>
The table should be added as indicated by the shaded below.

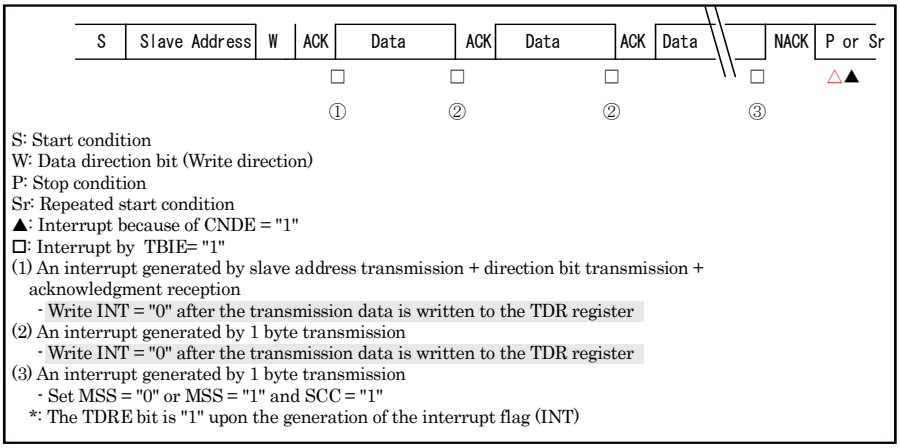
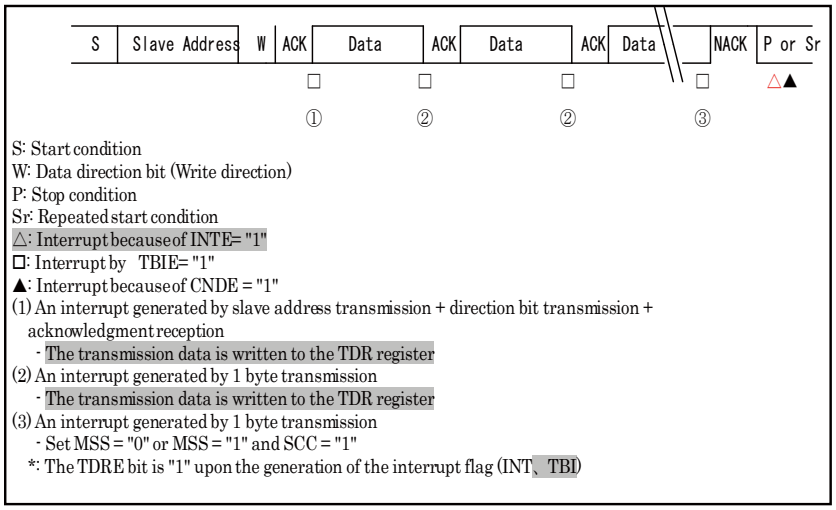
(Error)
<table>
<tr><th>Interrupt type</th><th>Interrupt request flag bit</th><th>Flag register</th><th>Interrupt factor</th><th>Interrupt factor enable bit</th><th>Clearing of interrupt request</th></tr>
<tr><td rowspan="3">Status</td><td>LBD</td><td>SSR</td><td>Lin break field detection</td><td>ESCR:LBIE</td><td>Writing "0" to the SSR:LBD</td></tr>
<tr><td>SFD</td><td>SACSR</td><td>SyncField is detected</td><td>SACSR: SFDE</td><td>Writing "0" to SyncField detection flag (SACSR:SFD)</td></tr>
<tr><td>TINT</td><td>SACSR</td><td>Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)</td><td>SACSR: TINTE</td><td>Writing "0" to timer interrupt flag bit (SACSR:TINT)</td></tr>
<tr><td rowspan="2">Input capture</td><td>ICP</td><td>ICS</td><td>1st falling edge of Lin Sync Field</td><td rowspan="2">ICS:ICE0</td><td rowspan="2">Disabling of ICP</td></tr>
<tr><td>ICP</td><td>ICS</td><td>5th falling edge of Lin Sync Field</td></tr>
</table>
*1: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

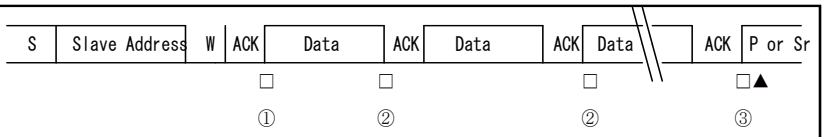
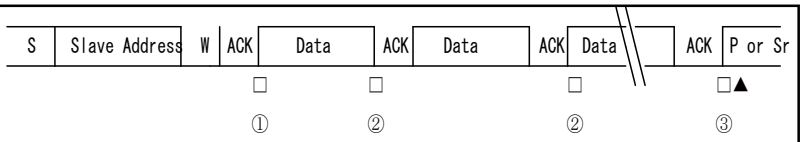
(Correct)
<table>
<tr><th>Interrupt type</th><th>Interrupt request flag bit</th><th>Flag register</th><th>Interrupt factor</th><th>Interrupt factor enable bit</th><th>Clearing of interrupt request</th></tr>
<tr><td rowspan="3">Status</td><td>LBD</td><td>SSR</td><td>Lin break field detection</td><td>ESCR:LBIE</td><td>Writing "0" to the SSR:LBD</td></tr>
<tr><td>SFD</td><td>SACSR</td><td>SyncField is detected</td><td>SACSR: SFDE</td><td>Writing "0" to SyncField detection flag (SACSR:SFD)</td></tr>
<tr><td>TINT</td><td>SACSR</td><td>Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)</td><td>SACSR: TINTE</td><td>Writing "0" to timer interrupt flag bit (SACSR:TINT)</td></tr>
<tr><td rowspan="2">Input Capture *2</td><td>ICP</td><td>ICS</td><td>1st falling edge of Lin Sync Field</td><td rowspan="2">ICS:ICE0</td><td rowspan="2">Disabling of ICP</td></tr>
<tr><td>ICP</td><td>ICS</td><td>5th falling edge of Lin Sync Field</td></tr>
</table>
*1: Set the TIE bit to "1" after the TDRE bit is cleared to "0".
*2: An interrupt does not generate by Input Capture when the auto baud rate adjustment function is enable (SACSR.AUTE="1").
</td></tr>
<tr><td>719</td><td>35.7.2.2</td><td>
The figure name should be corrected as indicated by the shaded below.

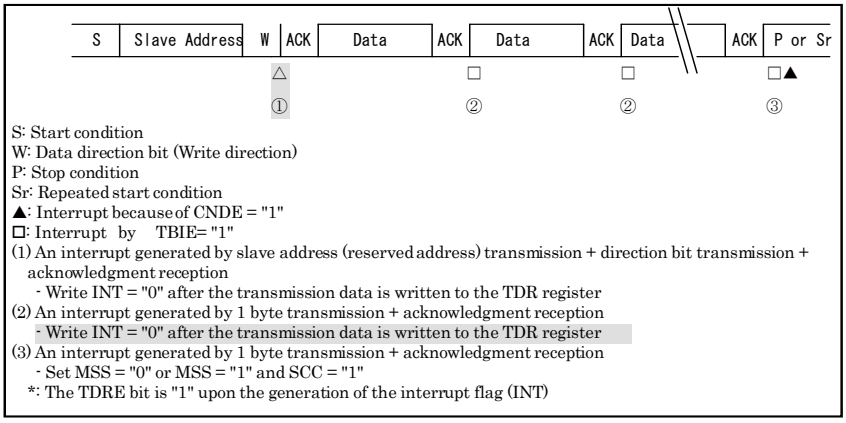
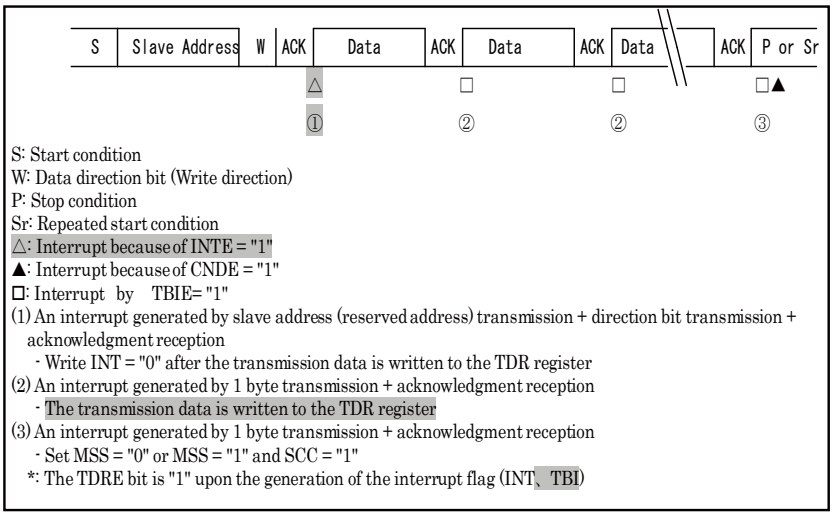
(Error)
Figure 35.7-16 LIN Break Field Reception to Sync Field Reception (in case where STMR is SFUR or smaller and SFLR or larger)

(Correct)
Figure 35.7-16 LIN Break Field Reception to Sync Field Reception (in case where STMR is larger than SFUR or smaller than SFLR)
</td></tr>
</table>

Page	Item	Change Results
758	35.8.3.4	"Figure 35.8-20 Master Transmission Interrupt (5)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1" → "0", IBSR:RSA="0", ACK Response)" should be corrected as indicated by the shaded below. (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 register (2) An interrupt generated by 1 byte transmission - Write WSEL="0", INT = "0" after the transmission data is written to the TDR register (3) An interrupt generated by 1 byte transmission - Set MSS = "0" or MSS = "1" and SCC = "1" *: The TDRE bit is "1" upon the generation of the interrupt flag (INT)
761	35.8.3.4	The figure name should be corrected as indicated by the shaded below. (Error) Figure 35.8-27 Master Transmission Interrupt (12)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response) (Correct) Figure 35.8-27 Master Transmission Interrupt (12)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)

Page	Item	Change Results
761	35.8.3.4	"Figure 35.8-27 Master Transmission Interrupt (12)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="1", IBSR:RSA="0", NACK Response)" should be corrected as indicated by the shaded below. (Error)  (Correct) 

Page	Item	Change Results
762	35.8.3.4	"Figure 35.8-29 Master Transmission Interrupt (14)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="1"→"0", IBSR:RSA="0", ACK Response)" should be corrected as indicated by the shaded below. (Error)  S: Start condition W: Data direction bit (Write direction) P: Stop condition Sr: Repeated start condition ▲: Interrupt because of CNDE = "1" □: Interrupt by 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 by TBIE= "1" (1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception - The transmission data is written to the TDR register (2) An interrupt generated by 1 byte transmission - Write WSEL="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 (TBIE)

Page	Item	Change Results
763	35.8.3.4	"Figure 35.8-30 Master Transmission Interrupt (15)-when FIFO is Disabled (SSR:DMA="1", IBCR:WSEL="0", IBSR:RSA="1")" should be corrected as indicated by the shaded below. (Error)  (Correct) 
777	35.8.4.3	The figure name should be corrected as indicated by the shaded below. (Error) Figure 35.8-48 Slave Reception Interrupt (6)-when FIFO is Enabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1") (Correct) Figure 35.8-48 Slave Reception Interrupt (6)-when FIFO is Disabled (SSR:DMA="0", IBCR:WSEL="0", IBSR:RSA="1")

Page	Item	Change Results
800	36.4.1	The description of "Note" of " [bit0] Init: Initialization bit" should be corrected as indicated by the shaded below. (Error) • 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. • To change the division ratio of the clock supplied to the CAN interface from the CAN prescaler register (BRTER), set "1" to the Init bit and change the CAN prescaler register with the CAN controller stopped (Correct) •When setting the INIT bit of the CAN control register (CTRLR) to "1", set the INIT bit of the CAN control register (CTRLR) to "1" after transmission is completed. When setting the INIT bit of the CAN control register to "1" during transmission, and then setting the INIT bit to "0" to transmit: • First set the INIT bit to "1". • Second make the message buffer with the transmission request bit (TxRqst) = "1" cancel transmission (set the TxRqst bit to "0"). • Third set the INIT bit to "0". After a 2-bit time passes, set the transmission request bit (TxRqst) of the message buffer to "1". • Before transiting to the low-power consumption mode (Stop mode or Clock mode) or before changing the source clock supplied to CAN controller, CAN controller must be initialized by setting Init bit to "1". • To change the clock divide ratio which supplies to the CAN interface by the following registers, set "1" to the Init bit and stop the CAN controller. • CAN bit timing register (BTR) • CAN prescaler extension register (BRPER) • CAN prescaler register (CANPRE)
806	36.4.1	The description of "■ CAN Interrupt Register (INTR)" should be corrected as indicated by the shaded below. (Error) • CAN Interrupt Register (upper byte): Address Base+08_H (Access: Byte, Half-word, Word) • CAN Interrupt Register (lower byte): Address Base+09_H (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+09_H (Access: Half-word, Word)

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>811</td><td>36.4.2</td><td>
The description of " [bit7 to bit0] Message Number: Message Number" should be deleted as indicated by the shaded below.

(Error)

Message Number	Function
00_H	Setting prohibited. If this value is set, it is interpreted as 20_H and 20_H is read out.
01_H to 20_H	Sets the message number for processing.
21_H to $3F_H$	Setting prohibited. If this value is set, it is interpreted as 01_H to $1F_H$ and the value interpreted is read out.

(Correct)

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.
</td></tr>
<tr><td>925</td><td>37.3.9.5</td><td>
The description of "37.3.9.5 Input Buffer Command Mask Register (IBCM)" should be corrected as indicated by the shaded below.

(Error)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STX RS	LDSS
0xD510	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(Correct)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STX RS	LDSS	LHSS
0xD510	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>1009</td><td>38.5.3</td><td>The title of "Single one-shot timer (GATE="0": When the trigger input and rising edge trigger are selected)" and "Single one-shot timer (GATE="0": gate input, TRGM: H input interval count)" in Figure 38.5-5 in "38.5.3. Single One-shot Operation" should be modified as follows:

(Error)
Single one-shot timer (GATE= "0": For trigger input/rising edge trigger selection)
(Correct)
Single one-shot timer (GATE= "1": Gate input and TRGM:H input interval count)

(Error)
Single one-shot timer (GATE= "0": Gate input and TRGM:H input interval count)
(Correct)
Single one-shot timer (GATE= "0": For trigger input/rising edge trigger selection)</td></tr>
<tr><td>1017</td><td>38.5.3</td><td>The description of "■ Capture Mode" should be corrected as indicated by the shaded below.

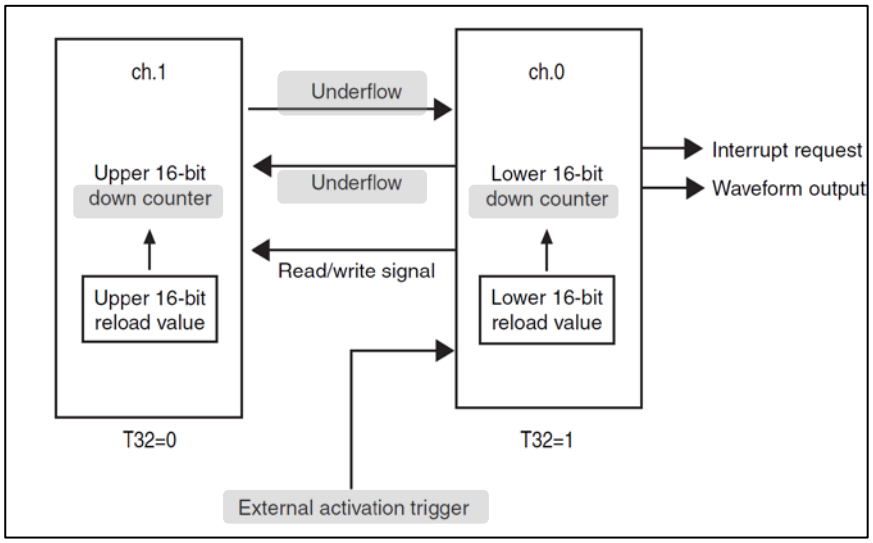
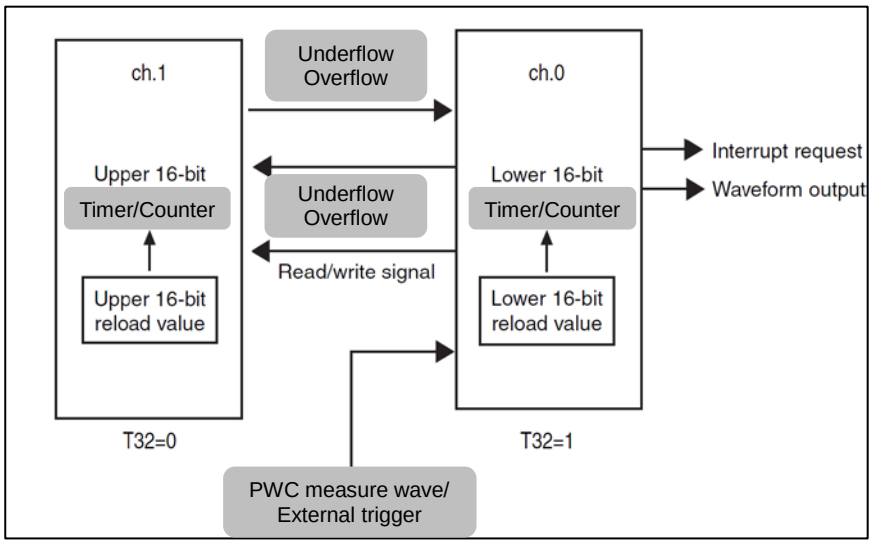
(Error)
In a mode other than trigger, a capture will not be performed at a retrigger. EF interrupt will also not be generated.

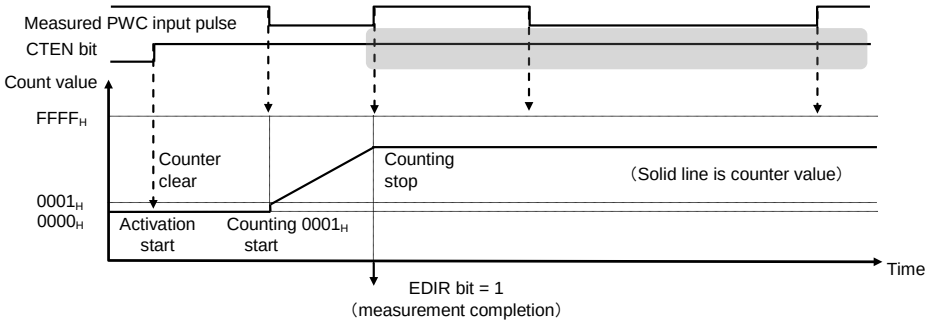
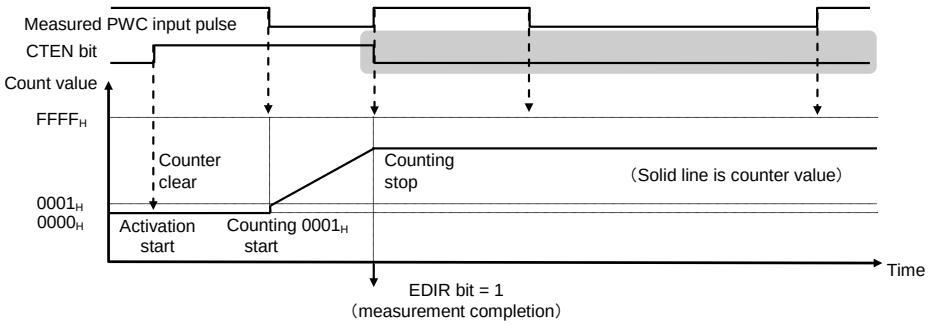
(Correct)
In a mode other than capture, a capture will not be performed at a retrigger. EF interrupt will also not be generated.</td></tr>
<tr><td>1070</td><td>40.4.1.2</td><td>The description of bit attribution should be corrected as indicated by the shaded below.

(Error)
<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>bit</th></tr>
<tr><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><td></td></tr>
<tr><td>0
R/W
R0,WX(*1)
R0,WX(*2)</td><td>0
R/W</td><td>0
R/W</td><td>0
R/W</td><td>0
R/W
R/W0(*1)</td><td>0
R/W</td><td>0
R/W</td><td>0
R0,W</td><td>Initial value
Attribute</td></tr>
</table>
(Correct)
<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>bit</th></tr>
<tr><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><td></td></tr>
<tr><td>0
R/W
R0,WX(*1)
R0,WX(*2)</td><td>0
R/W</td><td>0
R/W</td><td>0
R/W</td><td>0
R/W
R/W0(*1)</td><td>0
R/W</td><td>0
R,W</td><td>0
R0,W
R0,W(*1)</td><td>Initial value
Attribute</td></tr>
</table></td></tr>
<tr><td>1075</td><td>40.4.1.2</td><td>The shaded description should be added to "[BTxTMCR: bit1] CTEN (CounT ENable) Counter operation enable bit ".

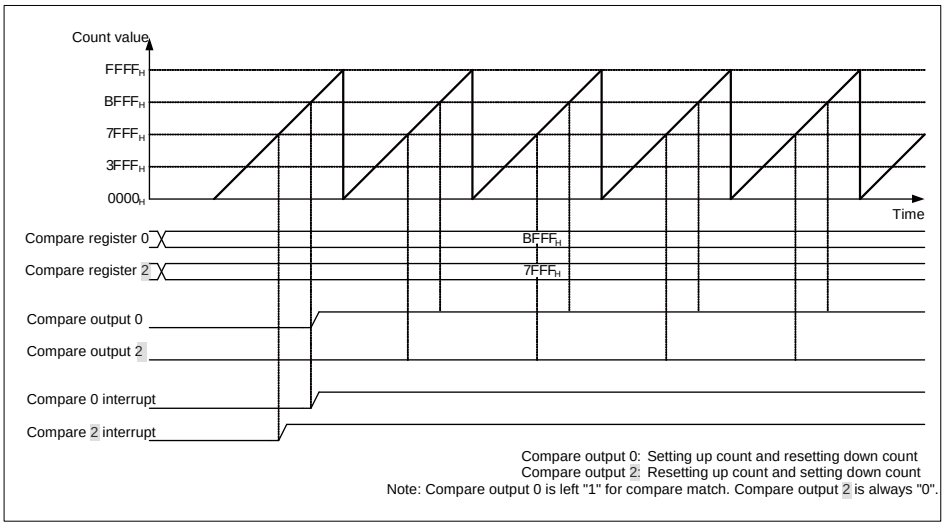
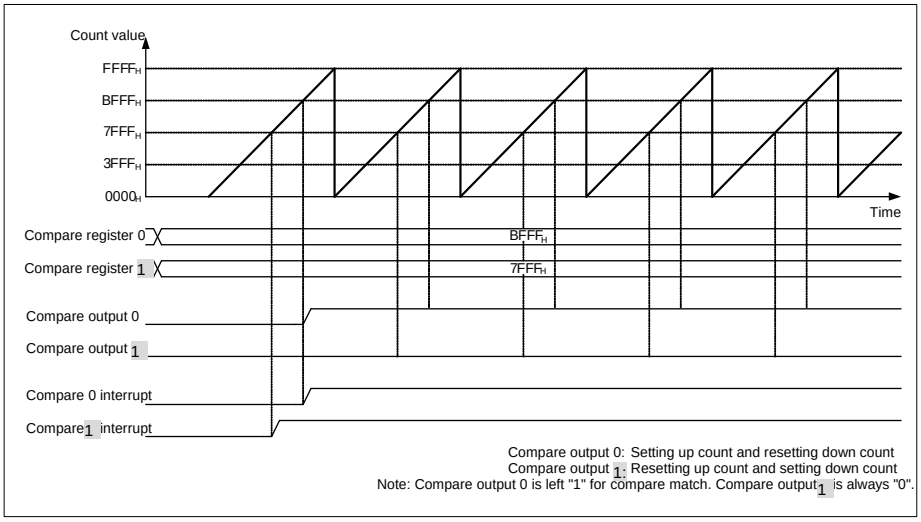
(Correct)
This bit is cleared to "0" when measurement is completed in single measurement mode of the PWC mode.</td></tr>
</table>

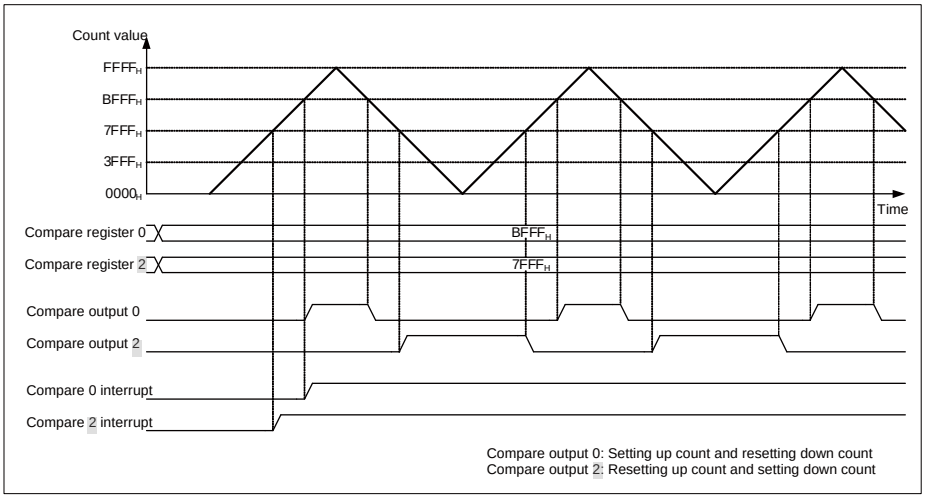
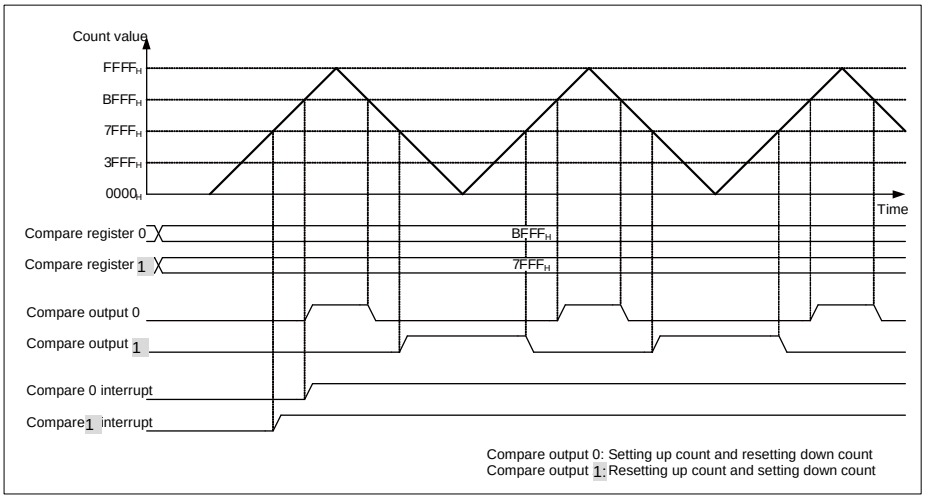
Page	Item	Change Results
1075	40.4.1.2	The shaded description should be added to " [BTxTMCR: bit0] STRG (Software TRiGger) Software trigger bit". (Correct) In the PWC mode the read value is "0"; this bit should be cleared to "0"
1076	40.4.1.3	The description of "Notes" of "40.4.1.3 I/O selection register: BTSEL01 (Base Timer SElect register ch.0 and ch.1)" should be corrected as indicated by the shaded below. (Error) • This register will not be initialized when reset mode is set (writing of (BT0TMCR.FMD=000/BT1TMCR.FMD=000)). (Correct) • Rewrite these registers after selecting base timer reset mode (FMD2 to FMD0 = 000) by setting the base timer x timer control registers (BTxTMCR).

Page	Item	Change Results
1099	40.5.4.4	"Figure 40.5-11 Configuration in 32-bit Timer Mode" should be corrected as indicated by the shaded below. (Error)  (Correct) 

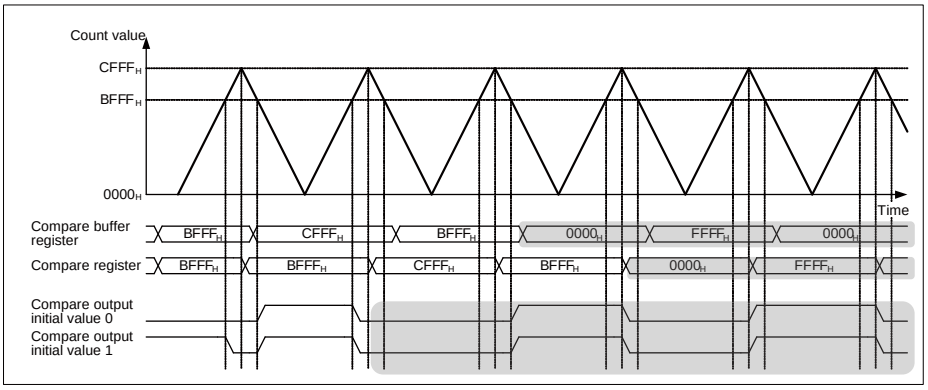
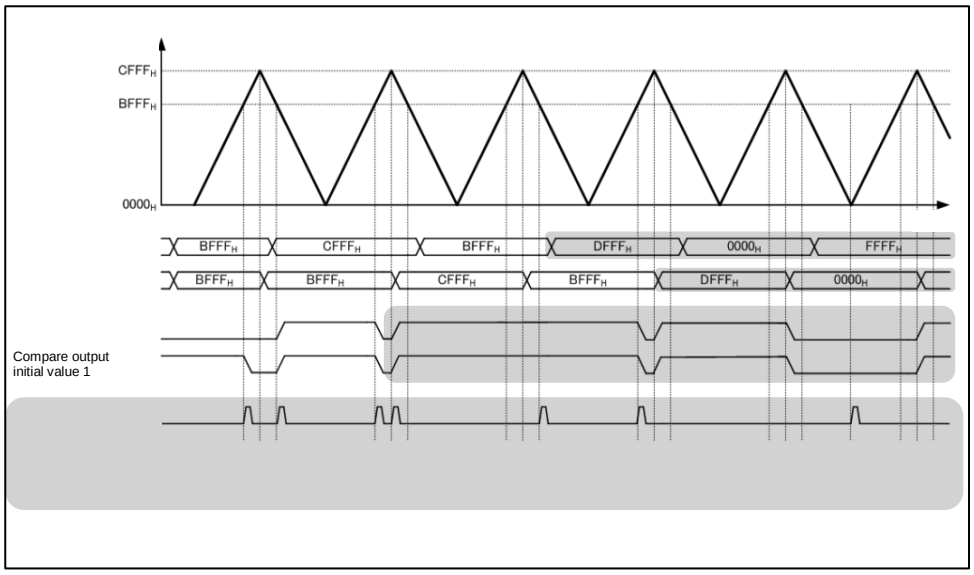
Page	Item	Change Results
1125	40.5.7.2	"Figure 40.5-26 Operation Example" should be corrected as indicated by the shaded below. (Error) Operation of L pulse width measurement in single measurement mode  (Correct) Operation of L pulse width measurement in single measurement mode 
1156	41.6	The description should be deleted as indicated by the shaded below. (Error) • If the activation trigger selection bits (GTRS0 to GTRS2:TSEL0[6:0], TSEL1[6:0]) are set to a value out of the specifiable ranges (000_0110 to 001_1111, 010_0010 to 011_1111, and 100_1000 to 111_1111), once disable PPG operation and write a correct value within specifiable ranges in order to restore the registers to normalcy.
1156	41.6	The shaded description should be added to "■ Application Notes". (Correct) • When PGMS is cleared from "1" to "0" after a matching, the pulse with specified duty is outputted in the next period (PPG cycle).

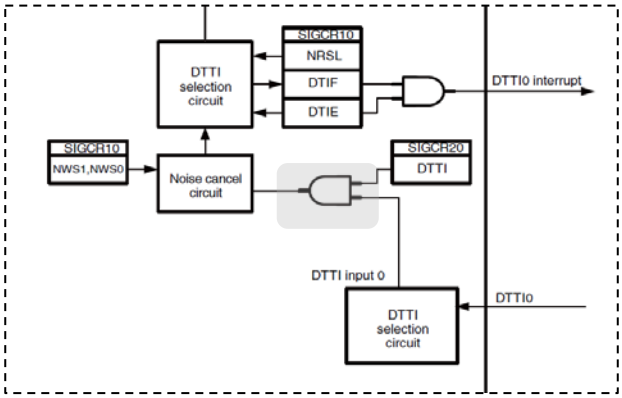
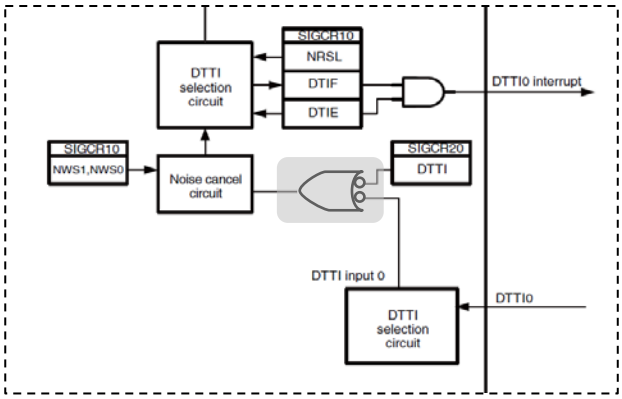
Page	Item	Change Results
1215	43.4.1.2	The description of "43.4.1.2 Input Capture State Control Register (ICS)" should be corrected as indicated by the shaded below. (Error) • IPCPxy: Address Base_addr+0x04$_H$ (Access: Half-word, Word) (Correct) • ICSxy: Address Base_addr+0x04$_H$ (Access: Half-word, Word)
1234	44.5.2	The description of "44.5.2 Operation of 16-bit Output Compare" should be deleted as indicated by the shaded below. (Error) 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. (Correct) The output compare is used to compare the value set in the specified compare register and the value of the 16-bit free-run timer.

Page	Item	Change Results
1238	44.5.2.2	"Figure 44.5-7 Operation of 16-bit Output Compare (Set/Reset Mode) #1" should be corrected as indicated by the shaded below. (Error)  (Correct) 

Page	Item	Change Results
1238	44.5.2.2	"Figure 44.5-8 Operation of 16-bit Output Compare (Set/Reset Mode) #2" should be corrected as indicated by the shaded below. (Error)  (Correct) 

Page	Item	Change Results
1239	44.5.2.3	The shaded description should be added to " 44.5.2.3 16-bit Output Compare Timing". (Correct) When the free-run timer is stopped or started during the output compare is enabled (OCSxy.CSTn=1), the output compare operation is as follows; ・When the initial counter value of free-run timer is same as the compare register value, the output compare generates a compare match signal concurrently with the start of the free-run timer. ・When the free-run timer is stopped at a free-run timer match with a compare register while output compare is running(OCSxy.CSTn=1), the output compare generates a compare match signal. However, the output compare doesn't generate a compare match signal under the following conditions. ・When any of the following applies while the free-run timer is running on up-count mode. ・An output compare buffer register was written concurrently with a compare matched timing when the output compare buffering function is disabled(OCSxy.BUFn=1). ・The free-run timer was stopped before 2 peripheral clock times from the output timing of the compare matched signal. ・When any of the following applies while the free-run timer is running on up/down-count mode. ・The output compare buffer function is disabled(OCSxy.BUFn=1). ・Transfer is activated when 0 of the free-run timer is detected(OCSxy.BTSn=0). ・The CMOD bit of the compare control register is set to "0"(OCSxy.CMOD=0). ・The MOD bit of the compare mode control register is set to "0"(OCMODxy.MODn=0). If the free-run timer is stopped at the counter value of the free-run timer matched a compare register value while the output compare is running(OCSxy.CSTn=1), after that the free-run timer is started while the free-run timer count value is matched the output compare register, the output compare may generate a compare matched signal respectively at the time of a stop and stop of the free-run timer. The output compare doesn't generate a compare match signal, when the free-run timer is stopped during the output compare is disabled (OCSxy.CSTn=0).

Page	Item	Change Results
1245	44.5.2.4	"Figure 44.5-17 Case #5 Where the Free-run Timer Is in Up/Down Count Mode" should be corrected as indicated by the shaded below. (Error)  (Correct) 
1247	44.5.3	The description of "44.5.3 Notes on Using 16-bit Output Compare" should be deleted as indicated by the shaded below. (Error) •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 "0000B".
1290	45.5.4.7	The shaded description should be added to "45.5.4.7 Power down". (Correct) (Note) Stop the A/D conversion before the transition to standby mode occurs.

Page	Item	Change Results
1294 1295	46.3	"Figure 46.3-1 Configuration of the Waveform Generator (0ch)" and "Figure 46.3-2 Configuration of the Waveform Generator (1ch)"should be corrected as indicated by the shaded below. (Error)  (Correct) 

Page	Item	Change Results
1297	46.4	The table should be corrected as indicated by the shaded below.

(Error)

Table 46.4-1 List of Waveform Generator Register

Address	+0	+1	+2	+3
$11A4_H$	DTTI selection register (DTSR)	Reserved	Reserved	Reserved
$11A8_H$	16-bit dead timer register 0 (TMRR0)		16-bit dead timer register 1 (TMRR1)	
$11AC_H$	16-bit dead timer register 2 (TMRR2)		Reserved	Reserved
$11B0_H$	16-bit dead timer state control register 0 (DTCR0)	16-bit dead timer state control register 1 (DTCR1)	16-bit dead timer state control register 2 (DTCR2)	Reserved
$11B4_H$	Reserved	16-bit dead timer reload interrupt register 0 (DTIR0)	Reserved	16-bit dead timer minus control register 0 (DTMNS0)
$11B8_H$	Reserved	Waveform control register 10 (SIGCR10)	Reserved	Waveform control register 20 (SIGCR20)
$11BC_H$	PPG output control register 0 (PICS0)			
$11C0_H$	16-bit dead timer register 3 (TMRR3)		16-bit dead timer register 4 (TMRR4)	
$11C4_H$	16-bit dead timer register 5 (TMRR5)		Reserved	Reserved
$11C8_H$	16-bit dead timer state control register 3 (DTCR3)	16-bit dead timer state control register 4 (DTCR4)	16-bit dead timer state control register 5 (DTCR5)	Reserved

(Correct)

Table 46.4-1 List of Waveform Generator Register

Address	+0	+1	+2	+3
$11A4_H$	DTTI selection register (DTSR)	Reserved	Reserved	Reserved
$11A8_H$	16-bit dead timer register 0 (TMRR0)		16-bit dead timer register 1 (TMRR1)	
$11AC_H$	16-bit dead timer register 2 (TMRR2)		Reserved	Reserved
$11B0_H$	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
$11B4_H$	Reserved	16-bit dead timer reload interrupt register 0 (DTIR0)	Reserved	16-bit dead timer minus control register 0 (DTMNS0)
$11B8_H$	Reserved	Waveform control register 10 (SIGCR10)	Reserved	Waveform control register 20 (SIGCR20)
$11BC_H$	PPG output control register 0 (PICS0)			
$11C0_H$	16-bit dead timer register 3 (TMRR3)		16-bit dead timer register 4 (TMRR4)	
$11C4_H$	16-bit dead timer register 5 (TMRR5)		Reserved	Reserved
$11C8_H$	16-bit dead timer state control register 3 (DTSCR3)	16-bit dead timer state control register 4 (DTSCR4)	16-bit dead timer state control register 5 (DTSCR5)	Reserved

Page	Item	Change Results
1300	46.4.1.3	The description of "46.4.1.3 16-bit Dead Timer State Control Register (DTCR)" should be corrected as indicated by the shaded below. (Error) 46.4.1.3 16-bit Dead Timer State Control Register (DTCR) The 16-bit dead timer state control register (DTCR) is used to control operation mode, permission of interrupt request, interrupt request flag, permission of GATE signal, and output level polarity of the waveform generator. • DTCR0: Address $11B0_H$ (Access: Byte, Half-word, Word) • DTCR3: Address $11C8_H$ (Access: Byte, Half-word, Word) (Correct) 46.4.1.3 16-bit Dead Timer State Control Register (DTSCR) The 16-bit dead timer state control register (DTSCR) is used to control operation mode, permission of interrupt request, interrupt request flag, permission of GATE signal, and output level polarity of the waveform generator. • DTSCR0: Address $11B0_H$ (Access: Byte, Half-word, Word) • DTSCR3: Address $11C8_H$ (Access: Byte, Half-word, Word)
1303	46.4.1.3	The description of "46.4.1.3 16-bit Dead Timer State Control Register (DTCR)" should be corrected as indicated by the shaded below. (Error) • DTCR1: Address $11B1_H$ (Access: Byte, Half-word, Word) • DTCR4: Address $11C9_H$ (Access: Byte, Half-word, Word) (Correct) • DTSCR1: Address $11B1_H$ (Access: Byte, Half-word, Word) • DTSCR4: Address $11C9_H$ (Access: Byte, Half-word, Word)
1306	46.4.1.3	The description of "46.4.1.3 16-bit Dead Timer State Control Register (DTCR)" should be corrected as indicated by the shaded below. (Error) • DTCR2: Address $11B2_H$ (Access: Byte, Half-word, Word) • DTCR5: Address $11CA_H$ (Access: Byte, Half-word, Word) (Correct) • DTSCR2: Address $11B2_H$ (Access: Byte, Half-word, Word) • DTSCR5: Address $11CA_H$ (Access: Byte, Half-word, Word)

<table>
<tr><th>Page</th><th>Item</th><th>Change Results</th></tr>
<tr><td>1322</td><td>46.5.1</td><td>
The table should be corrected as indicated by the shaded below.

(Error)
Table 46.5-1 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-1
<table>
<tr><th></th><th>16-bit dead timer 0/3</th><th>16-bit dead timer 1/4</th><th>16-bit dead timer 2/5</th></tr>
<tr><td>Interrupt request flag bit</td><td>16-bit dead timer state control register 0/3 (DTCR0/3) TMIF0: bit28</td><td>16-bit dead timer state control register 1/4 (DTCR1/4) TMIF1: bit20</td><td>16-bit dead timer state control register 2/5 (DTCR2/5) TMIF2: bit12</td></tr>
<tr><td>Interrupt request enable bit</td><td>16-bit dead timer state control register 0/3 (DTCR0/3) TMIE0: bit27</td><td>16-bit dead timer state control register 1/4 (DTCR1/4) TMIE1: bit19</td><td>16-bit dead timer state control register 2/5 (DTCR2/5) TMIE2: bit11</td></tr>
<tr><td>Interrupt factor</td><td>16-bit dead timer 0/3 underflow</td><td>16-bit dead timer 1/4 underflow</td><td>16-bit dead timer 2/5 underflow</td></tr>
</table>
(Correct)
Table 46.5-1 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-1
<table>
<tr><th></th><th>16-bit dead timer 0/3</th><th>16-bit dead timer 1/4</th><th>16-bit dead timer 2/5</th></tr>
<tr><td>Interrupt request flag bit</td><td>16-bit dead timer state control register 0/3 (DTSCR0/3) TMIF0: bit28</td><td>16-bit dead timer state control register 1/4 (DTSCR1/4) TMIF1: bit20</td><td>16-bit dead timer state control register 2/5 (DTSCR2/5) TMIF2: bit12</td></tr>
<tr><td>Interrupt request enable bit</td><td>16-bit dead timer state control register 0/3 (DTSCR0/3) TMIE0: bit27</td><td>16-bit dead timer state control register 1/4 (DTSCR1/4) TMIE1: bit19</td><td>16-bit dead timer state control register 2/5 (DTSCR2/5) TMIE2: bit11</td></tr>
<tr><td>Interrupt factor</td><td>16-bit dead timer 0/3 underflow</td><td>16-bit dead timer 1/4 underflow</td><td>16-bit dead timer 2/5 underflow</td></tr>
</table>
</td></tr>
<tr><td>1322</td><td>46.5.1</td><td>
The description of " Table 46.5-2 Interrupt Control Bits and Interrupt Factor of the Waveform Generator #1-2" should be corrected as indicated by the shaded below.

(Error)
For the waveform generator, TMIF0/TMIF1/TMIF2 (bit28/bit20/bit12) of the 16-bit dead timer state control register (DTCR) 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 DTCR register is either "000B" or "001B". If interrupt requests are enabled (DTCR register; TMIE0/TMIE1/TMIE2 (bit27/bit19/bit11)=1) in this state, an interrupt request is output to the interrupt controller.

(Correct)
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 "000B" or "001B". 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.
</td></tr>
</table>

Page	Item	Change Results
1325	46.5.2	The description of "Notes" of "■ Pins and Bit Setting" should be corrected as indicated by the shaded below. (Error) <Notes:> The RTO0 and RTO1 are controlled by the TMD2 to TMD0 of the 16-bit dead timer state control register (DTCR0). The RTO2 and RTO3 are controlled by the TMD5 to TMD3 of the DTCR1 register. The RTO4 and RTO5 are controlled by the TMD8 to TMD6 of the DTCR2 register. (Correct) <Notes:> 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.
1341	46.6	The description of "■ Notes on interrupts" should be corrected as indicated by the shaded below. (Error) • 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 (DTCR). • The interrupt flags (DTIF) must be cleared before setting "1" to the interrupt request enable bit (DTIE) of the waveform control register 1 (SIGCR1). (Correct) • 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).

Page	Item	Change Results
1373	49.5.4	The table should be corrected as indicated by the shaded below.

(Error)

Reset factor	Difference from when the OCD tool is not connected	
	Initialization range	Processing time
Power-on reset	Yes	Yes
RSTX pin input (irregular)		
RSTX pin input		
Watchdog reset 0 (irregular)		
Watchdog reset 0		
Watchdog reset 1 (irregular)		
Watchdog reset 1		
External low voltage detection reset (irregular)		
External low voltage detection reset		
Reset by detection of transition to wrong standby mode (irregular)		Yes
Reset by detection of transition to wrong standby mode		No
Internal low voltage detection reset		Yes
Flash security violation reset (irregular)		Yes
Flash security violation reset		No
Software reset (irregular)		Yes
Software reset		No

(Correct)

Reset factor	Difference from when the OCD tool is not connected	
	Initialization range	Processing time
Power-on reset	No	Yes
RSTX pin input (irregular)		
RSTX pin input		
Watchdog reset 0 (irregular)		
Watchdog reset 0		
Watchdog reset 1 (irregular)		
Watchdog reset 1		
External low voltage detection reset (irregular)		
External low voltage detection reset		
Reset by detection of transition to wrong standby mode (irregular)		Yes
Reset by detection of transition to wrong standby mode		No
Internal low voltage detection reset		Yes
Flash security violation reset (irregular)		Yes
Flash security violation reset		No
Software reset (irregular)		Yes
Software reset		No

Page	Item	Change Results
1399 1434	APPENDIX B:	The description of "Table B-1 I/O Map (MB91F583M/F584M/F585M) (22 / 35)" and "Table B-2 I/O Map (MB91F583S/F584S/F585S) (22 / 32)" should be corrected as indicated by the shaded below.

(Error)

001500H	SCR0/(IBCR0)[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000000-0	SSR0[R/W] B,H,W 0--00011	ESCR0/(IBSR0)[R/W] B,H,W 00000000	Multi Function Serial I/F 0 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
001504H	—/(RDR10/(TDR10))[R/W] H,W -------- -------- *3		RDR00/(TDR00)[R/W] B,H,W -------0 00000000*1		
001508H	SACSR0[R/W] B,H,W 0----000 00000000		STMR0[R] B,H,W 00000000 00000000		
00150CH	STMCR00[R/W] B,H,W 00000000 00000000		—/(SCSCR0/SFUR0) [R/W] B,H,W -------- -------- *3 *4		
001510H	—/(SCSTR30) [R/W] B,H,W -------- *3	—/(SCSTR20) [R/W] B,H,W -------- *3	—/(SCSTR10/SFLR10) [R/W] B,H,W -------- *3 *4	—/(SCSTR00/SFLR00) [R/W] B,H,W -------- *3 *4	
001514H	—	—/(SCSFR20) [R/W] B,H,W -------- *3	—/(SCSFR10) [R/W] B,H,W -------- *3	—/(SCSFR00) [R/W] B,H,W -------- *3	
001518H	—/(TBYTE30) [R/W] B,H,W -------- *3	—/(TBYTE20) [R/W] B,H,W -------- *3	—/(TBYTE10) [R/W] B,H,W -------- *3	TBYTE00 [R/W] B,H,W 00000000	
00151CH	BGR0[R/W] H,W 00000000 00000000		—/(ISMK0)[R/W] B,H,W -------- *2	—/(ISBA0)[R/W] B,H,W -------- *2	
001520H	FCR10[R/W] B,H,W 00-00100	FCR00[B/W] B,H,W -0000000	FBYTE20[R/W] B,H,W 00000000	FBYTE10[R/W] B,H,W 00000000	

(Correct)

001500H	SCR0/(IBCR0)[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000000-0	SSR0[R/W] B,H,W 0--00011	ESCR0/(IBSR0)[R/W] B,H,W 00000000	Multi Function Serial I/F 0 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
001504H	—/(RDR10/(TDR10))[R/W] H,W -------- -------- *3		RDR00/(TDR00)[R/W] B,H,W -------0 00000000*1		
001508H	SACSR0[R/W] B,H,W 0----000 00000000		STMR0[R] B,H,W 00000000 00000000		
00150CH	STMCR0[R/W] B,H,W 00000000 00000000		—/(SFUR0) [R/W] B,H,W -------- -------- *3 *4		
001510H	—	—	—/(SFLR10) [R/W] B,H,W -------- *3 *4	—/(SFLR00) [R/W] B,H,W -------- *3 *4	
001514H	—	—	—	—	
001518H	—	—	—	—	
00151CH	BGR0[R/W] H,W 00000000 00000000		—/(ISMK0)[R/W] B,H,W -------- *2	—/(ISBA0)[R/W] B,H,W -------- *2	
001520H	FCR10[R/W] B,H,W 00-00100	FCR00[R/W] B,H,W -0000000	FBYTE20[R/W] B,H,W 00000000	FBYTE10[R/W] B,H,W 00000000	

Page	Item	Change Results
1399 1434	APPENDIX B:	The description of "Table B-1 I/O Map (MB91F583M/F584M/F585M) (22 / 35)" and "Table B-2 I/O Map (MB91F583S/F584S/F585S) (22 / 32)" should be corrected as indicated by the shaded below.

(Error)

001524H	SCR1[R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000000-0	SSR1[R/W] B,H,W 0—00011	ESCR1[R/W] B,H,W 00000000	Multi Function Serial I/F 1 *1 Byte access is possible only for access to lower 8 bits. *3 Reserved because CSIO mode is not set immediately after reset *4 Reserved because LIN2.1 mode is not set immediately after reset
001528H	—/(RDR11/(TDR11))[R/W] B,H,W -------- -------- *3		RDR01/(TDR01)[R/W] B,H,W -------0 00000000 *1		
00152CH	SACSR1[R/W] B,H,W 0----000 00000000		STMR1[R] B,H,W 00000000 00000000		
001530H	STMCR1[R/W] B,H,W 00000000 00000000		—/(SCSCR1/SFUR1) [R/W] B,H,W -------- -------- *3 *4		
001534H	—/(SCSTR31) [R/W] B,H,W -------- *3	—/(SCSTR21) [R/W] B,H,W -------- *3	—/(SCSTR11/SFLR11) [R/W] B,H,W -------- *3 *4	—/(SCSTR01/SFLR01) [R/W] B,H,W -------- *3 *4	
001538H	—	—/(SCSFR21) [R/W] B,H,W -------- *3	—/(SCSFR11) [R/W] B,H,W -------- *3	—/(SCSFR01) [R/W] B,H,W -------- *3	
00153CH	—/(TBYTE31) [R/W] B,H,W -------- *3	—/(TBYTE21) [R/W] B,H,W -------- *3	—/(TBYTE11) [R/W] B,H,W -------- *3	TBYTE01[R/W] B,H,W 00000000	
001540H	BGR1[R/W] H,W 00000000 00000000		—	—	
001544H	FCR11[R/W] B,H,W 00-00100	FCR01[B/W] B,H,W -0000000	FBYTE21[R/W] B,H,W 00000000	FBYTE11[R/W] B,H,W 00000000	

(Correct)

001524H	SCR1[R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000000-0	SSR1[R/W] B,H,W 0—00011	ESCR1[R/W] B,H,W 00000000	Multi Function Serial I/F 1 *1 Byte access is possible only for access to lower 8 bits. *3 Reserved because CSIO mode is not set immediately after reset *4 Reserved because LIN2.1 mode is not set immediately after reset
001528H	—/(RDR11/(TDR11))[R/W] H,W -------- -------- *3		RDR01/(TDR01)[R/W] B,H,W -------0 00000000 *1		
00152CH	SACSR1[R/W] B,H,W 0----000 00000000		STMR1[R] B,H,W 00000000 00000000		
001530H	STMCR1[R/W] B,H,W 00000000 00000000		—/(SCSCR1/SFUR1) [R/W] B,H,W -------- -------- *3 *4		
001534H	—/(SCSTR31) [R/W] B,H,W -------- *3	—/(SCSTR21) [R/W] B,H,W -------- *3	—/(SCSTR11/SFLR11) [R/W] B,H,W -------- *3 *4	—/(SCSTR01/SFLR01) [R/W] B,H,W -------- *3 *4	
001538H	—	—	—	—	
00153CH	—	—	—	TBYTE01[R/W] B,H,W 00000000	
001540H	BGR1[R/W] H,W 00000000 00000000		—	—	
001544H	FCR11[R/W] B,H,W 00-00100	FCR01[R/W] B,H,W -0000000	FBYTE21[R/W] B,H,W 00000000	FBYTE11[R/W] B,H,W 00000000	

Page	Item	Change Results
1400	APPENDIX B:	The description of "Table B-1 I/O Map (MB91F583M/F584M/F585M) (23 / 35)" should be corrected as indicated by the shaded below.

(Error)

001548H	SCR2/(IBCR2)[R/W] B,H,W 0--00000	SMR2[R/W] B,H,W 000000-0	SSR2[R/W] B,H,W 0--00011	ESCR2/(IBSR2)[R/W] B,H,W 00000000	Multi Function Serial I/F 2 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
00154CH	—/(RDR12/(TDR12))[R/W] H,W -------- -------- *3		RDR02/(TDR02)[R/W] B,H,W -------0 00000000 *1		
001550H	SACSR2[R/W] B,H,W 0----000 00000000		STMR2[R] B,H,W 00000000 00000000		
001554H	STMCR2[R/W] B,H,W 00000000 00000000		—/(SCSCR2/SFUR2) [R/W] B,H,W -------- -------- *3 *4		
001558H	—/(SCSTR32) [R/W] B,H,W -------- *3	—/(SCSTR22) [R/W] B,H,W -------- *3	—/(SCSTR12/SFLR12) [R/W] B,H,W -------- *3 *4	—/(SCSTR02/SFLR02) [R/W] B,H,W -------- *3 *4	
00155CH	—	—/(SCSFR22) [R/W] B,H,W -------- *3	—/(SCSFR12) [R/W] B,H,W -------- *3	—/(SCSFR02) [R/W] B,H,W -------- *3	
001560H	—/(TBYTE32) [R/W] B,H,W -------- *3	—/(TBYTE22) [R/W] B,H,W -------- *3	—/(TBYTE12) [R/W] B,H,W -------- *3	TBYTE02[R/W] B,H,W 00000000	
001564H	BGR2[R/W] H,W 00000000 00000000		—/(ISMK2)[R/W] B,H,W -------- *2	—/(ISBA2)[R/W] B,H,W -------- *2	
001568H	FCR12[R/W] B,H,W 00-00100	FCR02[B/W] B,H,W -0000000	FBYTE22[R/W] B,H,W 00000000	FBYTE12[R/W] B,H,W 00000000	

(Correct)

001548H	SCR2/(IBCR2)[R/W] B,H,W 0--00000	SMR2[R/W] B,H,W 000000-0	SSR2[R/W] B,H,W 0--00011	ESCR2/(IBSR2)[R/W] B,H,W 00000000	Multi Function Serial I/F 2 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
00154CH	—/(RDR12/(TDR12))[R/W] H,W -------- -------- *3		RDR02/(TDR02)[R/W] B,H,W -------0 00000000 *1		
001550H	SACSR2[R/W] B,H,W 0----000 00000000		STMR2[R] B,H,W 00000000 00000000		
001554H	STMCR2[R/W] B,H,W 00000000 00000000		—/(SCSCR2/SFUR2) [R/W] B,H,W -------- -------- *3 *4		
001558H	—/(SCSTR32) [R/W] B,H,W -------- *3	—/(SCSTR22) [R/W] B,H,W -------- *3	—/(SCSTR12/SFLR12) [R/W] B,H,W -------- *3 *4	—/(SCSTR02/SFLR02) [R/W] B,H,W -------- *3 *4	
00155CH	—	—	—	—	
001560H	—	—	—	TBYTE02[R/W] B,H,W 00000000	
001564H	BGR2[R/W] H,W 00000000 00000000		—/(ISMK2)[R/W] B,H,W -------- *2	—/(ISBA2)[R/W] B,H,W -------- *2	
001568H	FCR12[R/W] B,H,W 00-00100	FCR02[R/W] B,H,W -0000000	FBYTE22[R/W] B,H,W 00000000	FBYTE12[R/W] B,H,W 00000000	

Page	Item	Change Results
1400	APPENDIX B:	The description of "Table B-1 I/O Map (MB91F583M/F584M/F585M) (23 / 35)" should be corrected as indicated by the shaded below.

(Error)

00156CH	SCR3/(IBCR3)[R/W] B,H,W 0--00000	SMR3[R/W] B,H,W 000000-0	SSR3[R/W] B,H,W 0--00011	ESCR3/(IBSR3)[R/W] B,H,W 00000000	Multi Function Serial I/F 3 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
001570H	—/(RDR13/(TDR13))[R/W] H,W -------- -------- *3		RDR03/(TDR03)[R/W] B,H,W -------0 00000000 *1		
001574H	SACSR3[R/W] B,H,W 0----000 00000000		STMR3[R] B,H,W 00000000 00000000		
001578H	STMCR3[R/W] B,H,W 00000000 00000000		—/(SCSCR3/SFUR3) [R/W] B,H,W -------- -------- *3 *4		
00157CH	—/(SCSTR33) [R/W] B,H,W -------- *3	—/(SCSTR23) [R/W] B,H,W -------- *3	—/(SCSTR13/SFLR13) [R/W] B,H,W -------- *3 *4	—/(SCSTR03/SFLR03) [R/W] B,H,W -------- *3 *4	
001580H	—	—/(SCSFR23) [R/W] B,H,W -------- *3	—/(SCSFR13) [R/W] B,H,W -------- *3	—/(SCSFR03) [R/W] B,H,W -------- *3	
001584H	—/(TBYTE33)[R/W] B,H,W -------- *3	—/(TBYTE23)[R/W] B,H,W -------- *3	—/(TBYTE13)[R/W] B,H,W -------- *3	TBYTE03[R/W] B,H,W 00000000	
001588H	BGR3[R/W] H,W 00000000 00000000		—/(ISMK3)[R/W] B,H,W -------- *2	—/(ISBA3)[R/W] B,H,W -------- *2	
00158CH	FCR13[R/W] B,H,W 00-00100	FCR03[B/W] B,H,W -0000000	FBYTE23[R/W] B,H,W 00000000	FBYTE13[R/W] B,H,W 00000000	

(Correct)

00156CH	SCR3/(IBCR3)[R/W] B,H,W 0--0□000	SMR3[R/W] B,H,W 000000-0	SSR3[R/W] B,H,W 0--00011	ESCR3/(IBSR3)[R/W] B,H,W 00000000	Multi Function Serial I/F 3 *1 Byte access is possible only for access to lower 8 bits. *2 Reserved because I2C 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
001570H	—/(RDR13/(TDR13))[R/W] H,W -------- -------- *3		RDR03/(TDR03)[R/W] B,H,W -------0 00000000 *1		
001574H	SACSR3[R/W] B,H,W 0----000 00000000		STMR3[R] B,H,W 00000000 00000000		
001578H	STMCR3[R/W] B,H,W 00000000 00000000		—/(SCSCR3/SFUR3) [R/W] B,H,W -------- -------- *3 *4		
00157CH	—/(SCSTR33) [R/W] B,H,W -------- *3	—/(SCSTR23) [R/W] B,H,W -------- *3	—/(SCSTR13/SFLR13) [R/W] B,H,W -------- *3 *4	—/(SCSTR03/SFLR03) [R/W] B,H,W -------- *3 *4	
001580H	—	—	—	—	
001584H	—	—	—	TBYTE03[R/W] B,H,W 00000000	
001588H	BGR3[R/W] H,W 00000000 00000000		—/(ISMK3)[R/W] B,H,W -------- *2	—/(ISBA3)[R/W] B,H,W -------- *2	
00158CH	FCR13[R/W] B,H,W 00-00100	FCR03[R/W] B,H,W -0000000	FBYTE23[R/W] B,H,W 00000000	FBYTE13[R/W] B,H,W 00000000	

1410 1442	APPENDIX B:	The description of "Table B-1 I/O Map (MB91F583M/F584M/F585M) (33 / 35)" and "Table B-2 I/O Map (MB91F583S/F584S/F585S) (30 / 32)" should be corrected as indicated by the shaded below.

(Error)

00D310H	MHDS[R/W] W -0000000 -0000000 -0000000 10000000	FlexRay MHD *5

(Correct)

00D310H	MHDS[R/W] W -0000000 -0000000 -0000000 00000000	FlexRay MHD *5

MN705-00015-2v0-E

Spansion • Controller Manual

32-BIT MICROCONTROLLER

FR family FR81S

MB91580M/S Series

HARDWARE MANUAL

June 2014 Revision 2.0

Published Spansion Inc.

Edited Marketing Communication Dept.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2012-2014 Spansion All rights reserved. Spansion®, the Spansion logo, MirrorBit®, MirrorBit® Eclipse™, ORNAND™ and combinations thereof, are trademarks and registered trademarks of Spansion LLC in the United States and other countries. Other names used are for informational purposes only and may be trademarks of their respective owners.