



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

Continuity of Specifications

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

Continuity of Ordering Part Numbers

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

For More Information

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

About Cypress

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

32-BIT MICROCONTROLLER

FR60

MB91470/480 Series

HARDWARE MANUAL

32-BIT MICROCONTROLLER
FR60
MB91470/480 Series
HARDWARE MANUAL

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

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

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

FUJITSU SEMICONDUCTOR LIMITED

MB91470/480 Series

PREFACE

■ Objectives and intended reader

Thank you for using Fujitsu semiconductor products.

The MB91470/480 series is a standard microcontroller that has a 32-bit high-performance RISC CPU as well as built-in I/O resources and bus control mechanisms for embedded controller that requires high-performance and high-speed CPU processing. The MB91470/480 series has built-in RAM (for data) to increase the speed at which the CPU executes instructions.

This manual is intended for engineers who will develop products using the MB91470/480 series and describes the functions and operations of the MB91470/480 series. Read this manual thoroughly.

For more information on instructions, see the "FR Family 32-bit microcontroller Instruction Manual".

Note: FR, which is an abbreviation of FUJITSU RISC controller, is a product of FUJITSU SEMICONDUCTOR LIMITED.

■ Trademark

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Structure of This Manual

This manual consists of the following 23 chapters and an appendix.

CHAPTER 1 OVERVIEW

This chapter provides basic information required to understand the MB91470/480 series, and covers features, a block diagram, and package dimension.

CHAPTER 2 HANDLING DEVICES

This chapter provides precautions on handling the device.

CHAPTER 3 CPU AND CONTROL UNIT

This chapter provides basic information required to understand the CPU core functions of the MB91470/480 series. It covers architecture, specifications, and instructions.

CHAPTER 4 EXTERNAL BUS INTERFACE

The external bus interface controller controls the interfaces with the internal bus for LSI and with external memory and I/O devices. This chapter explains each function of the external bus interface.

CHAPTER 5 I/O PORTS

This chapter outlines the I/O ports and describes the configuration and functions of their registers.

CHAPTER 6 INTERRUPT CONTROLLER

This chapter explains the overview of the interrupt controller, the configuration and functions of registers, and interrupt controller operation.

CHAPTER 7 EXTERNAL INTERRUPT AND NMI CONTROLLER

This chapter describes the overview of the external interrupt and NMI controller, the configuration and functions of registers, and operation of the external interrupt and NMI controller.

CHAPTER 8 REALOS-RELATED HARDWARE

The REALOS-related hardware is used by the real-time OS. Accordingly, these functions cannot be used by user programs if using REALOS. This chapter explains the overview of the delayed interrupt module and bit search module, the configuration and functions of the registers, and the operation of the delayed interrupt module and bit search module.

CHAPTER 9 16-BIT RELOAD TIMER

This chapter describes the overview of the reload timer, the configuration and functions of registers, and the reload timer operation.

CHAPTER 10 TIMING GENERATOR

This chapter explains the overview of the timing generator, the configuration and functions of registers, and operation of the timing generator.

CHAPTER 11 PPG

This chapter explains the overview of the PPG, the configuration and functions of registers, and the PPG operation.

CHAPTER 12 MULTI-FUNCTION TIMER

This chapter explains the overview of the multi-function timer, the configuration and functions of registers, and operation of the multi-function timer.

CHAPTER 13 BASE TIMER

This chapter provides an overview of the base timer, summarizes its register configuration and functions, and describes its operations.

CHAPTER 14 UP/DOWN COUNTER

This chapter describes the function and operation of 8/16-bit up/down counter.

CHAPTER 15 MULTI-FUNCTION SERIAL INTERFACE

This chapter describes the functions and operations of the multi-function serial interface.

CHAPTER 16 8/10-BIT A/D CONVERTER

This chapter describes the overview of the 8/10-bit A/D converter, the configuration and functions of registers, and the operation of the 8/10-bit A/D converter.

CHAPTER 18 12-BIT A/D CONVERTER

This chapter describes the overview of the 12-bit A/D converter, the configuration and functions of registers, and the operation of the 12-bit A/D converter.

CHAPTER 17 CLOCK MONITOR

This chapter explains the function and the operation of the clock monitor.

CHAPTER 19 MULTIPLICATION AND ADDITION CALCULATOR

This chapter outlines the multiplication and addition calculator and describes its register configuration/function and operations.

CHAPTER 20 DMA CONTROLLER (DMAC)

This chapter describes the DMAC, the configuration and functions of registers, and DMAC operation.

CHAPTER 21 FLASH MEMORY

This chapter explains the overview of the flash memory, the configuration and functions of registers, and the flash memory operation.

CHAPTER 22 SERIAL PROGRAMMING CONNECTION

This chapter explains the basic configuration for serial programming, pins used for serial on-board programming, the connection example for serial programming, and system configuration for flash microcontroller programmer.

MB91470/480 Series

CHAPTER 23 WILD REGISTER CONTROL BLOCK

This chapter describes the register configuration, functions and timer operations of the wild register control block.

APPENDIX

The appendix describes pin states in each CPU state, notes on using the little-endian areas, a list of FR family instructions, and notes on using MB91470/480 series.

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

Copyright ©2006-2011 FUJITSU SEMICONDUCTOR LIMITED All rights reserved.

MB91470/480 Series

CONTENTS

CHAPTER 1	OVERVIEW	1
1.1	Overview	2
1.2	Block Diagram	7
1.3	Pin Assignment	9
1.4	Package Dimension	13
1.5	List of Pin Functions	18
1.6	I/O Circuit Type	29
CHAPTER 2	HANDLING DEVICES	33
2.1	Precautions on Handling the Device	34
CHAPTER 3	CPU AND CONTROL UNIT	39
3.1	Memory Space	40
3.2	Memory Map	41
3.3	Internal Architecture	43
3.4	Programming Model	48
3.4.1	Registers	49
3.5	Data Structure	55
3.6	Memory Map	57
3.7	Divergence Instructions	58
3.8	EIT (Exception, Interruption, and Trap)	61
3.9	Operating Mode	72
3.9.1	Bus Modes	73
3.9.2	Mode Setting	74
3.9.3	Note	76
3.10	Reset (Device Initialization)	77
3.10.1	Reset Level	78
3.10.2	Reset Factor	79
3.10.3	Reset Sequence	81
3.10.4	Oscillation Stabilization Wait Time	83
3.10.5	Reset Operation Mode	85
3.11	Clock Generation Control	87
3.11.1	Selection of the Source Clock	88
3.11.2	PLL Control	89
3.11.3	Oscillation Stabilization Wait and PLL Lock Wait Time	92
3.11.4	Clock Distribution	94
3.11.5	Clock Divider	95
3.11.6	Block Diagram of Clock Generation Control Unit	96
3.11.7	Explanation of Register Details for Clock Generation Control Unit	97
3.11.8	Peripheral Circuit Functions in the Clock Controller	112
3.12	Device State Control	116

CHAPTER 4	EXTERNAL BUS INTERFACE	127
4.1	Features of the External Bus Interface	128
4.2	External Bus Interface Registers	131
4.2.1	Area Select Registers (ASR0 to ASR2)	132
4.2.2	Area Configuration Registers (ACR0 to ACR2)	133
4.2.3	Area Wait Registers (AWR0 to AWR2)	139
4.2.4	Chip Select Enable Register (CSER)	143
4.3	Chip Select Area	144
4.4	Endian and Bus Access	145
4.4.1	Relationship between Data Bus Widths and Control Signals	146
4.4.2	Big Endian Bus Access	147
4.4.3	Little Endian Bus Access	153
4.4.4	External Access	157
4.5	Ordinary Bus Interface	161
4.6	Address/Data Multiplex Interface	168
4.7	Procedure for Setting a Register	171
CHAPTER 5	I/O PORTS	173
5.1	Overview of I/O Port	174
5.2	Block Diagrams of I/O Port	175
5.2.1	Normal I/O Port	176
5.2.2	External Interrupt Input I/O Port	178
5.2.3	Analog Input I/O Port	180
5.2.4	External Bus Interface I/O Port	182
5.2.5	Multi-Function Timer I/O Port	184
5.3	I/O Port Registers	186
CHAPTER 6	INTERRUPT CONTROLLER	203
6.1	Overview of Interrupt Controller	204
6.2	Interrupt Controller Registers	205
6.3	Block Diagram of Interrupt Controller	209
6.4	Register Details Explanation of Interrupt Controller	210
6.5	Operation of Interrupt Controller	212
CHAPTER 7	EXTERNAL INTERRUPT AND NMI CONTROLLER	217
7.1	Overview of External Interrupt/NMI Controller	218
7.2	Registers of External Interrupt/NMI Controller	220
7.3	Operation of External Interrupt/NMI Controller	222
CHAPTER 8	REALOS-RELATED HARDWARE	227
8.1	Delayed Interrupt Module	228
8.2	Bit Search Module	230
CHAPTER 9	16-BIT RELOAD TIMER	235
9.1	Overview of 16-bit Reload Timer	236
9.2	16-bit Reload Timer Register	237
9.2.1	Control Status Register (TMCSR)	238

MB91470/480 Series

9.2.2	16-bit Timer Register (TMR)	240
9.2.3	16-bit Reload Register (TMRLR)	241
9.3	Operation of 16-bit Reload Timer	242
CHAPTER 10 TIMING GENERATOR		245
10.1	Overview of Timing Generator	246
10.2	Block Diagram of Timing Generator	247
10.3	Registers of Timing Generator	249
10.3.1	Timing Generator Control Register (TTCR0/TTCR1)	251
10.3.2	Compare Register (COMP0/COMP2/COMP4/COMP6, COMP1/COMP3/COMP5/COMP7)	253
10.4	Operation of Timing Generator	255
CHAPTER 11 PPG		257
11.1	Overview of PPG	258
11.2	Block Diagram of PPG	260
11.3	Registers of PPG	265
11.3.1	PPG Operation Mode Control Registers (PPGC0 to PPGC15)	267
11.3.2	Reload Registers (PRLH0 to PRLH15, PRLL0 to PRLL15)	270
11.3.3	PPG Trigger Register (TRG)	271
11.3.4	Output Inversion Register (REVC)	272
11.3.5	GATE Function Control Registers (GATEC0/GATEC4/GATEC8/GATEC12)	273
11.4	Operation Explanation of PPG	274
CHAPTER 12 MULTI-FUNCTION TIMER		281
12.1	Overview of the Multi-function Timer	282
12.2	Block Diagram of the Multi-function Timer	286
12.3	Pins of the Multi-function Timer	301
12.4	Multi-function Timer Register	303
12.4.1	Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5) /Compare Clear Register (CPCLRH0 to CPCLRH5, CPCLRL0 to CPCLRL5)	310
12.4.2	Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)	312
12.4.3	Timer State Control Register (TCCSH0 to TCCSH5, TCCSL0 to TCCSL5, TCCSM0 to TCCSM5)	313
12.4.4	A/D Trigger Control Register (ADTRGC0 to ADTRGC5)	320
12.4.5	Free-run Timer Selection Register (FRS0 to FRS9)	322
12.4.6	Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11) /Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11)	327
12.4.7	Compare Control Register (OCSH0 to OCSH11, OCSL0 to OCSL11)	329
12.4.8	Compare Mode Control Register (OCMOD0/OCMOD1)	334
12.4.9	Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)	336
12.4.10	Input Capture State Control/PPG Output Control Register (ICSH23/ICSH67, ICSL23/ICSL67, PICSH01/PICSH45, PICSL01/PICSL45)	337
12.4.11	16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)	345
12.4.12	16-bit Dead Timer Control Register (DTCR0 to DTCR5)	346
12.4.13	Waveform Control Register (SIGCR1/SIGCR2)	352
12.4.14	A/D Activation Compare Register (ADCOMPB0 to ADCOMP5, ADCOMP0 to ADCOMP5, ADTGCE0/ADTGCE1, ADTGSEL0/ADTGSEL1, ADTGBUF0/ADTGBUF1)	356

12.5	Multi-function Timer Interrupt	364
12.6	Operation of the Multi-function Timer	367
12.6.1	Operation of 16-bit Free-run Timer	368
12.6.2	Operation of Free-run Timer Selector	376
12.6.3	Operation of 16-bit Output Compare	379
12.6.4	16-bit Input Capture Operation	392
12.6.5	Waveform Generator Operation	394
12.6.5.1	Operation of Timer Mode	399
12.6.5.2	Operation during Dead Time Timer Mode	401
12.6.5.3	DTTI Pin Control Operation	404
12.6.6	A/D Activation Compare Operation	406
12.7	Notes on Using the Multi-function Timer	412
12.8	Example Program for Multi-function Timer	417

CHAPTER 13 BASE TIMER 421

13.1	Overview of the Base Timer	422
13.2	Block Diagrams of the Base Timer	424
13.3	Base Timer's Registers	427
13.4	Operations of the Base Timer	431
13.5	32-bit Mode Operations	433
13.6	Notes of Using the Base Timer	435
13.7	Base Timer Interrupts	437
13.8	Base Timer Description by Function Mode	438
13.8.1	PWM Function	439
13.8.1.1	Timer Control Register (BTnTMCR) for PWM Timer	440
13.8.1.2	PWM Period Setting Register (BTnPCSR)	446
13.8.1.3	PWM Duty Setting Register (BTnPDUT)	447
13.8.1.4	Timer Register (BTnTMR)	448
13.8.1.5	16-bit PWM Timer Operation	449
13.8.1.6	One-shot Operation	450
13.8.1.7	Interrupt Factors and Timing Chart	451
13.8.1.8	Output Waveforms	452
13.8.2	PPG Function	453
13.8.2.1	Timer Control Register (BTnTMCR) for PPG Timer	454
13.8.2.2	"L"-width Setting Reload Register (BTnPRLL)	460
13.8.2.3	"H"-width Setting Reload Register (BTnPRLH)	461
13.8.2.4	Timer Register (BTnTMR)	462
13.8.2.5	16-bit PPG Timer Operation	463
13.8.2.6	Continuous Operation	464
13.8.2.7	One-shot Operation	465
13.8.2.8	Interrupt Factors and Timing Chart	467
13.8.3	Reload Timer Function	468
13.8.3.1	Timer Control Register (BTnTMCR) for Reload Timer	469
13.8.3.2	Period Setting Register (BTnPCSR)	475
13.8.3.3	Timer Register (BTnTMR)	476
13.8.3.4	16-bit Reload Timer Operation	477
13.8.4	PWC Function	480

MB91470/480 Series

13.8.4.1	Timer Control Register (BTnTMCR) for PWC Timer	481
13.8.4.2	Data Buffer Register (BTnDTBF)	487
13.8.4.3	PWC Operation	488
CHAPTER 14	UP/DOWN COUNTER	497
14.1	Overview of Up/Down Counter	498
14.2	Block Diagram of Up/Down Counter	500
14.3	Register of Up/Down Counter	501
14.3.1	Up/Down Count Register (UDCR)	502
14.3.2	Reload Compare Register (RCR)	503
14.3.3	Counter Status Register (CSR)	504
14.3.4	Counter Control Register (CCR)	506
14.4	Operation of Up/Down Counters	509
CHAPTER 15	MULTI-FUNCTION SERIAL INTERFACE	517
15.1	Characteristics of Multi-function Serial Interface	518
15.2	UART (Asynchronous Serial Interface)	520
15.3	Overview of UART (Asynchronous Serial Interface)	521
15.4	Registers of UART (Asynchronous Serial Interface)	522
15.4.1	Serial Control Register (SCR)	524
15.4.2	Serial Mode Register (SMR)	527
15.4.3	Serial Status Register (SSR)	529
15.4.4	Extended Communication Control Register (ESCR)	532
15.4.5	Reception Data Register / Transmission Data Register (RDR0/RDR1/TDR0/TDR1)	534
15.4.6	Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	538
15.4.7	FIFO Control Register 1 (FCR1)	540
15.4.8	FIFO Control Register 0 (FCR0)	543
15.4.9	FIFO Byte Register (FBYTE1/FBYTE2)	546
15.5	Interrupts of UART	548
15.5.1	Occurrence of Reception Interrupts and Flag Set Timing	550
15.5.2	Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing	551
15.5.3	Occurrence of Transmission Interrupts and Flag Set Timing	553
15.5.4	Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing	554
15.6	Operation of UART	555
15.7	Dedicated Baud Rate Generator	560
15.7.1	Setting Baud Rate	561
15.8	Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)	565
15.9	Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)	567
15.10	Notes on UART Mode	571
15.11	CSIO (Clock Synchronous Serial Interface)	572
15.12	Overview of CSIO (Clock Synchronous Serial Interface)	573
15.13	Registers of CSIO (Clock Synchronous Serial Interface)	574
15.13.1	Serial Control Register (SCR)	576
15.13.2	Serial Mode Register (SMR)	579
15.13.3	Serial Status Register (SSR)	582
15.13.4	Extended Communication Control Register (ESCR)	585

MB91470/480 Series

15.13.5	Reception Data Register / Transmission Data Register (RDR/TDR)	587
15.13.6	Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	590
15.13.7	FIFO Control Register 1 (FCR1)	592
15.13.8	FIFO Control Register 0 (FCR0)	594
15.13.9	FIFO Byte Register (FBYTE1/FBYTE2)	597
15.14	Interrupts of CSIO (Clock Synchronous Serial Interface)	599
15.14.1	Occurrence of Reception Interrupts and Flag Set Timing	600
15.14.2	Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing	601
15.14.3	Occurrence of Transmission Interrupts and Flag Set Timing	603
15.14.4	Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing	604
15.15	Operation of CSIO (Clock Synchronous Serial Interface)	605
15.16	Dedicated Baud Rate Generator	617
15.16.1	Setting Baud Rate	618
15.17	Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)	621
15.18	Notes on CSIO Mode	623
15.19	I ² C Interface	624
15.20	Overview of I ² C Interface	625
15.21	Registers of I ² C Interface	626
15.21.1	I ² C Bus Control Register (IBCR)	628
15.21.2	Serial Mode Register (SMR)	633
15.21.3	I ² C Bus Status Register (IBSR)	635
15.21.4	Serial Status Register (SSR)	639
15.21.5	Reception Data Register / Transmission Data Register (RDR/TDR)	642
15.21.6	7-bit Slave Address Mask Register (ISMK)	644
15.21.7	7-bit Slave Address Register (ISBA)	645
15.21.8	Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	646
15.21.9	FIFO Control Register 1 (FCR1)	647
15.21.10	FIFO Control Register 0 (FCR0)	649
15.21.11	FIFO Byte Register (FBYTE1/FBYTE2)	653
15.22	Interrupts of I ² C Interface	655
15.22.1	Operation of I ² C Interface Communication	657
15.22.2	Master Mode	658
15.22.3	Slave Mode	676
15.22.4	Bus Error	680
15.23	Dedicated Baud Rate Generator	681
15.23.1	Example of I ² C Flowcharts	683
15.24	Notes on I ² C Mode	695

CHAPTER 16 8/10-BIT A/D CONVERTER 697

16.1	Overview of the 8/10-bit A/D Converter	698
16.2	Configuration of the 8/10-bit A/D Converter	700
16.3	Pin of the 8/10-bit A/D Converter	704
16.4	Registers of the 8/10-bit A/D Converter	705
16.4.1	A/D Channel Control Register (ADCH)	707
16.4.2	A/D Mode Setting Register (ADMD)	709
16.4.3	A/D Control Status Register (ADCS)	712
16.4.4	A/D Data Register (ADCD)	715

MB91470/480 Series

16.4.5	Analog Input Control Register (AICR)	717
16.5	Interrupt of the 8/10-bit A/D Converter	718
16.6	Operation Explanation of the 8/10-bit A/D Converter	719
16.7	A/D conversion Data Protection Function of the 8/10-bit A/D Converter	723
16.8	Using Memorandum of the 8/10-bit A/D Converter	724
16.9	Notes on Using the 8/10-bit A/D Converter	725
CHAPTER 17 CLOCK MONITOR		727
17.1	Overview of the Clock Monitor	728
17.2	Clock output enable register of the Clock Monitor	730
CHAPTER 18 12-BIT A/D CONVERTER		731
18.1	Overview of the 12-bit A/D Converter	732
18.2	Configuration the 12-bit A/D Converter	734
18.3	Pin the 12-bit A/D Converter	737
18.4	Registers the 12-bit A/D Converter	738
18.4.1	A/D Channel Control Register (ADCH)	739
18.4.2	A/D Mode Setting Register (ADMD)	741
18.4.3	A/D Control Status Register (ADCS)	744
18.4.4	A/D Data Register (ADCD)	747
18.4.5	Analog Input Control Register (AICR)	749
18.5	Interrupt the 12-bit A/D Converter	750
18.6	Operation Explanation the 12-bit A/D Converter	751
18.7	A/D conversion Data Protection Function the 12-bit A/D Converter	754
18.8	Differential input mode the 12-bit A/D Converter	755
18.9	Using Memorandum of the 12-bit A/D Converter	757
18.10	Notes on Using the 12-bit A/D Converter	758
CHAPTER 19 MULTIPLICATION AND ADDITION CALCULATOR		761
19.1	Overview of the Multiplication and Addition Calculator	762
19.2	Block Diagram of the Multiplication and Addition Calculator	763
19.3	Instruction Definitions of the Multiplication and Addition Calculator	765
19.4	Register List of the Multiplication and Addition Calculator	766
19.5	Register Description of the Multiplication and Addition Calculator	767
19.6	Principles of Operation of the Multiplication and Addition Calculator	774
19.7	Details on Instructions of the Multiplication and Addition Calculator	778
19.8	Notes on Using the Sum-of-Products Circuit	784
CHAPTER 20 DMA CONTROLLER (DMAC)		785
20.1	Overview of the DMAC	786
20.2	Detailed Explanation of the DMAC Registers	789
20.2.1	DMAC ch.0,ch.1,ch.2,ch.3,ch.4 Control/Status Registers A	790
20.2.2	DMAC ch.1,ch.2,ch.3,ch.4 Control/Status Registers B	794
20.2.3	DMAC ch.1,ch.2,ch.3,ch.4 Transfer Source/Transfer Destination Address Setting Registers	800
20.2.4	DMAC ch.1,ch.2,ch.3,ch.4 DMAC All-Channel Control Register	802
20.3	Explanation of the DMAC Operation	804

20.3.1	Overview of the DMAC Operation	805
20.3.2	Setting a Transfer Request	807
20.3.3	Transfer Sequence	808
20.3.4	General Aspects of DMA Transfer	810
20.3.5	Addressing Mode	812
20.3.6	Data Types	813
20.3.7	Transfer Count Control	814
20.3.8	CPU Control	815
20.3.9	Operation Start	816
20.3.10	Transfer Request Acceptance and Transfer	817
20.3.11	Clearing Peripheral Interrupts by DMA	818
20.3.12	Temporary Stopping	819
20.3.13	Operation End/Stopping	820
20.3.14	Stopping due to an Error	821
20.3.15	DMAC Interrupt Control	822
20.3.16	DMA Transfer during Sleep	823
20.3.17	Channel Selection and Control	824
20.4	Operation Flowcharts of the DMAC	826
20.5	Data Bus of the DMAC	828
CHAPTER 21 FLASH MEMORY		831
21.1	Overview of Flash Memory	832
21.2	Flash Memory Registers	835
21.2.1	Flash Control/Status Register (FLCR)	836
21.2.2	Flash Wait Register (FLWC)	838
21.3	Explanation of Flash Memory Operation	839
21.4	Flash Memory Automatic Algorithms	841
21.4.1	Command Sequence	842
21.4.2	Confirming Automatic Algorithm Execution States	846
21.5	Details of Programming and Erasing Flash Memory	851
21.5.1	Read/Reset State	852
21.5.2	Programming Data	853
21.5.3	Erasing Data (Chip Erase)	855
21.5.4	Erasing Data (Sector Erase)	856
21.5.5	Suspending Sector Erasure	858
21.5.6	Resuming Sector Erasure	859
21.5.7	Continuous Mode Operation	860
21.6	Restrictions on Data Polling Flag (DQ7) and How to Avoid Problems	862
21.7	Notes on Flash Memory Programming	865
CHAPTER 22 SERIAL PROGRAMMING CONNECTION		867
22.1	Overview of the Serial Programming Connection	868
CHAPTER 23 WILD REGISTER CONTROL BLOCK		873
23.1	Overview of Wild Register Control Block	874
23.2	Registers of Wild Register Control Block	875
23.2.1	Wild Register Enable Register (WREN)	876

MB91470/480 Series

23.2.2	Wild Register Address Register (WA)	877
23.2.3	Wild Register Data Register (WD)	878
23.3	Operations of Wild Register Control Block	879
23.4	Restrictions and Notes	880
APPENDIX		881
APPENDIX A	I/O Map	882
APPENDIX B	Interrupt Vector	902
APPENDIX C	Pin States in Each CPU State	906
APPENDIX D	Notes when Little Endian Region is used	910
APPENDIX E	Instruction List	915
APPENDIX F	Precautions when Using	930
INDEX		933

MB91470/480 Series

MB91470/480 Series

Major changes in this edition

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

Page	Changes (For details, refer to main body.)	
836	CHAPTER 21 FLASH MEMORY	Corrected Summary.
837	21.2 Flash Memory Registers 21.2.1 Flash Control/Status Register (FLCR)	Corrected [bit1] WE: Write enabled.
840	21.3 Explanation of Flash Memory Operation	Corrected ■ Automatic Algorithm Execution States.
842	21.4 Flash Memory Automatic Algorithms 21.4.1 Command Sequence	Corrected "Table 21.4-1 Command Sequence Table".
843		Corrected ■ Data Write Command.
844		Corrected ■ Sector Erase Command.
846	21.4.2 Confirming Automatic Algorithm Execution States ■ RDY Bit	Corrected item name. Ready/Busy Signal (RDY/BUSYX) → RDY Bit
847	21.4.2 Confirming Automatic Algorithm Execution States ■ Hardware Sequence Flag	Corrected "Table 21.4-2 Hardware Sequence Flag Status List".
848		Corrected [bit7] DPOLL: Data polling flag.
856	21.5 Details of Programming and Erasing Flash Memory 21.5.4 Erasing Data (Sector Erase)	Added ■ Restrictions on Data Polling Flag (DQ7).
857		Corrected "Figure 21.5-2 Example of Sector Erasing Procedure".
860 , 861	21.5.7 Continuous Mode Operation	Added Item.
862 to 864	21.6 Restrictions on Data Polling Flag (DQ7) and How to Avoid Problems	Added Item.

CHAPTER 1

OVERVIEW

This chapter provides basic information required to understand the MB91470/480 series, and covers features, a block diagram, and package dimension.

- 1.1 Overview
- 1.2 Block Diagram
- 1.3 Pin Assignment
- 1.4 Package Dimension
- 1.5 List of Pin Functions
- 1.6 I/O Circuit Type

1.1 Overview

The MB91470/480 series is a line of Fujitsu's general-purpose 32-bit RISC microcontrollers designed for embedded control applications which require high-speed processing.

■ Features of FR CPU

- 32-bit RISC, load/store architecture, 5-stage pipeline
- Maximum operating frequency of 80 MHz (PLL clock multiplier)
- 16-bit fixed-length instructions (basic instruction)
- Instruction execution speed: One instruction per cycle
- Memory-to-memory transfer instructions, bit processing instructions, and barrel shift instructions: instructions appropriate for embedded applications
- Function entry and exit instructions, multi-load/store instructions of register content: instructions compatible with C language
- Register interlock function: facilitate assembly-language coding
- Built-in multiplier/instruction-level support
Signed 32-bit multiplication: 5 cycles
Signed 16-bit multiplication: 3 cycles
- Interrupts (saving of PC and PS): 6 cycles (16 priority levels)
- Harvard architecture enabling simultaneous execution of both program access and data access
- Instruction compatible with the FR family

■ External Bus Interface

- Maximum operating frequency: 40MHz
- 16-bit address full output (64 Kbytes space) capability
- 8/16-bit data output
- Use of unused data/address pins as general-purpose I/O ports
- Totally independent 3-area chip select outputs that can be set at minimum of 64 Kbytes
- Support of interface for various memory (SRAM, ROM/Flash)
- Basic bus cycle: 2 cycles
- Automatic wait cycle generator that can be programmed for each area and can insert waits
- External wait cycle using RDY input

■ Internal Memory

	MB91470 series		MB91480 series	
	144 pins		100 pins	
	Flash	Mask	Flash	Mask
256 Kbytes/16 Kbytes	MB91F475	-	MB91F482	MB91482
384 Kbytes/24 Kbytes	MB91F478	-	MB91F486	MB91486
512 Kbytes/32 Kbytes	MB91F479	-	MB91F487	MB91487

MB91470/480 Series

■ I/O Port

- Capable of pull-up control per pin
- Capable of reading pin level directly

■ External Interrupt Input

- Include one non-maskable interrupt (NMI) pin
- Use for wake up at stop

■ Bit Search Module (for REALOS)

Function for searching for the first 1-to-0 change bit position from MSB (upper bit) in each word

■ 16-bit Reload Timer

- Includes 1 channel for REALOS
- Internal clock can be selected using divide by 2/8/32.

■ Timing Generator

The delay start of the PPG timers can be executed synchronously between the timers.

■ 8/16-bit PPG Timer

■ Multi-function Timer

- 16-bit free-run timer

- Input capture

Interface with free-run timer

- Output compare

Interface with free-run timer

- A/D activating compare

Interface with free-run timer

- Waveform generator

Various waveforms are generated by using output compare output, 16-bit PPG timer and 16-bit dead timer.

■ Base Timer

Only one timer function can be selected from the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer.

■ 8/16-bit Up-Down Counter

■ Multi-function Serial Interface

- Full-duplex double buffer
- With 16-byte FIFO
- Asynchronous (start-stop synchronization) communication, clock synchronous communication, I²C standard mode (Max 100kbps), I²C high-speed mode (selectable various modes at maximum of 400kbps)
- Selectable parity On/Off
- Each channel has built-in baud rate generator.
- Error detection function for parity, frame and overrun errors

- External clock can be used as transfer clock.
- With I²C function

■ 8/10-bit A/D Converter (Successive Comparison Type)

- 8/10-bit resolution selectable
- Conversion Time: 1.2μs (minimum conversion time for 33 MHz peripheral clock (CLKP))
1.2μs (minimum conversion time for 40 MHz peripheral clock (CLKP))

■ 12-bit A/D Converter (successive approximation type)

- 12-bit resolution selectable
- Conversion Time: 2.0μs (minimum conversion time for 33 MHz peripheral clock (CLKP))
2.2μs (minimum conversion time for 40 MHz peripheral clock (CLKP))
- Capable of differential input mode

■ Clock Monitor Function

Output the peripheral clock (CLKP) divided by 2/4/8/16/32/64/128/256

■ Multiplication and Addition Calculator

- RAM:
 - Instruction RAM (I-RAM): 256 × 16 bits
 - Factor RAM (X-RAM): 64 × 32 bits
 - Variable RAM (Y-RAM): 64 × 32 bits
- High-speed multiplication and addition (7-stage pipelining)
- Product addition (32 bits × 32 bits + 72 bits)
- Operation result is extracted rounded from 72 bits to 32 bits or 72-bit result data reading.

■ DMAC (DMA Controller)

- Five channels or less can operate at the same time.
- The transfer can be started by two transfer factors (built-in peripheral interrupt and software).
- Addressing mode: 32-bit full addressing (increase/decrease/fix)
- Transfer mode (burst transfer/step transfer/block transfer)
- The transferring data size can be selected from 8/16/32 bits.
- Multi-byte can be transferred (by software).

■ Wild Register Function

Replace instruction/data at the target address (within the built-in Flash/ROM area only)

■ Other Features

- Has a built-in oscillation circuit as a clock source for which PLL multiplication can be selected.
- INITX is provided as a reset pin.
- Additionally, a watchdog timer reset and software resets are provided.
- Stop mode and sleep mode supported as low-power consumption modes
- Clock division ratio setting function
- Built-in time-base timer
- CMOS 0.18μm technology
- Power supply: 1-power supply [V_{cc}=4.0V to 5.5V]
- 1.9V is supplied for the internal circuit by the internal step-down circuit.

■ Package Lineup

Series name Package	MB91470 series	MB91480 series	
	MB91F475 MB91F478 MB91F479	MB91F482 MB91F486 MB91F487	MB91482 MB91486 MB91487
FPT-100P-M20 (LQFP-0.50 mm)	-	○	○
FPT-100P-M06 (QFP-0.65 mm)	-	○	○
FPT-144P-M12 (LQFP-0.40 mm)	○	-	-
FPT-144P-M27 (LQFP-0.40 mm)	○	-	-
BGA-144P-M06 (FBGA-0.80 mm)	○	-	-

■ Components of Each Model

Characteristics	MB91470/480 series common EVA	MB91470 series			MB91480 series		
	MB91FV470	MB91F475	MB91F478	MB91F479	MB91F487 MB91487	MB91F486 MB91486	MB91F482 MB91482
Pin number	224 pins	144 pins			100 pins		
Built-in Flash/ ROM capacity	512 Kbytes (Flash)	256 Kbytes (Flash)	384 Kbytes (Flash)	512 Kbytes (Flash)	512 Kbytes (Flash/ROM)	384 Kbytes (Flash/ROM)	256 Kbytes (Flash/ROM)
Built-in RAM capacity	40 Kbytes	16 Kbytes	24 Kbytes	32 Kbytes	32 Kbytes	24 Kbytes	16 Kbytes
External bus	Yes	Yes			-		
I/O ports	160	113			77		
External interrupts	NMI + 16 channels	NMI + 10 channels			NMI + 10 channels		
Reload timer	2 channels	2 channels			2 channels		
Timing generator	2 units	1 unit			2 units		
PPG	8-bit × 16 channels 16-bit × 8 channels	8-bit × 8 channels 16-bit × 4 channels (PPG output: 8 channels)			8-bit × 16 channels 16-bit × 8 channels (PPG output: 10 channels)		

(Continued)

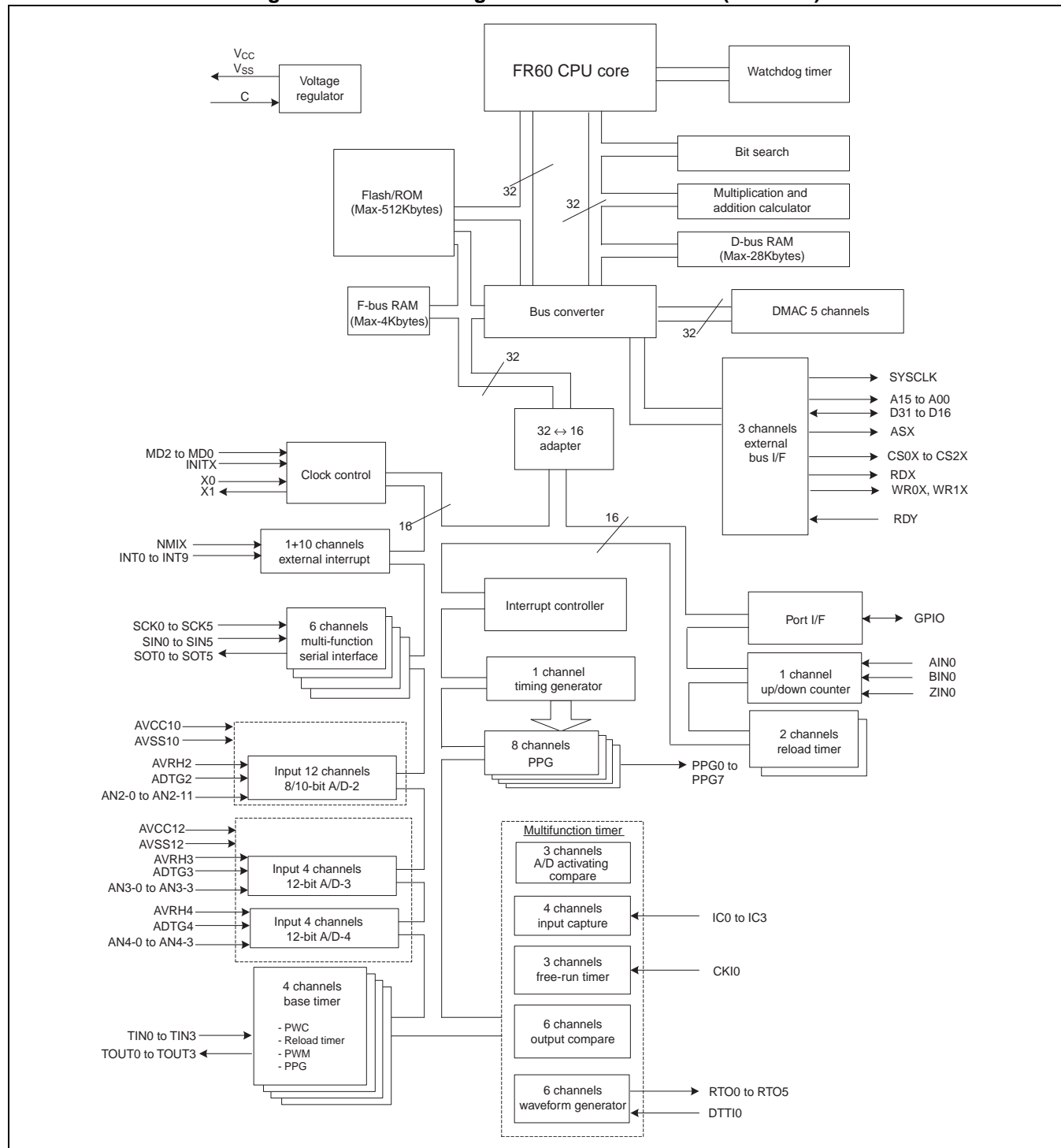
(Continued)

Characteristics	MB91470/480 series common EVA	MB91470 series			MB91480 series		
	MB91FV470	MB91F475	MB91F478	MB91F479	MB91F487 MB91487	MB91F486 MB91486	MB91F482 MB91482
Multi-function timer	2 units	1 unit			2 units		
Free-run timer	6 channels	3 channels			6 channels		
OCU	12 channels	6 channels			2 channels		
ICU	8 channels	4 channels			8 channels		
A/D activation compare	6 channels	3 channels			6 channels		
Waveform generator	12 channels	6 channels			12 channels		
Base timer	6 channels	4 channels			4 channels		
Up/down counter	2 channels	1 channel			-		
Multi-function serial interface	6 units	6 units			3 units		
8/10-bit A/D converter	4 channels × 2 units 16 channels × 1 unit	12 channels × 1 unit			4 channels × 2 units 10 channels × 1 unit		
12-bit A/D converter	4 channels × 2 units	4 channels × 2 units			-		
Clock monitor	1 unit	-			1 unit		
Multiplication and addition calculator	1 unit	1 unit			1 unit		
DMAC	5 channels	5 channels			5 channels		
Wild register	16 channels	16 channels			16 channels		
Debug function	DSU4	-			-		

1.2 Block Diagram

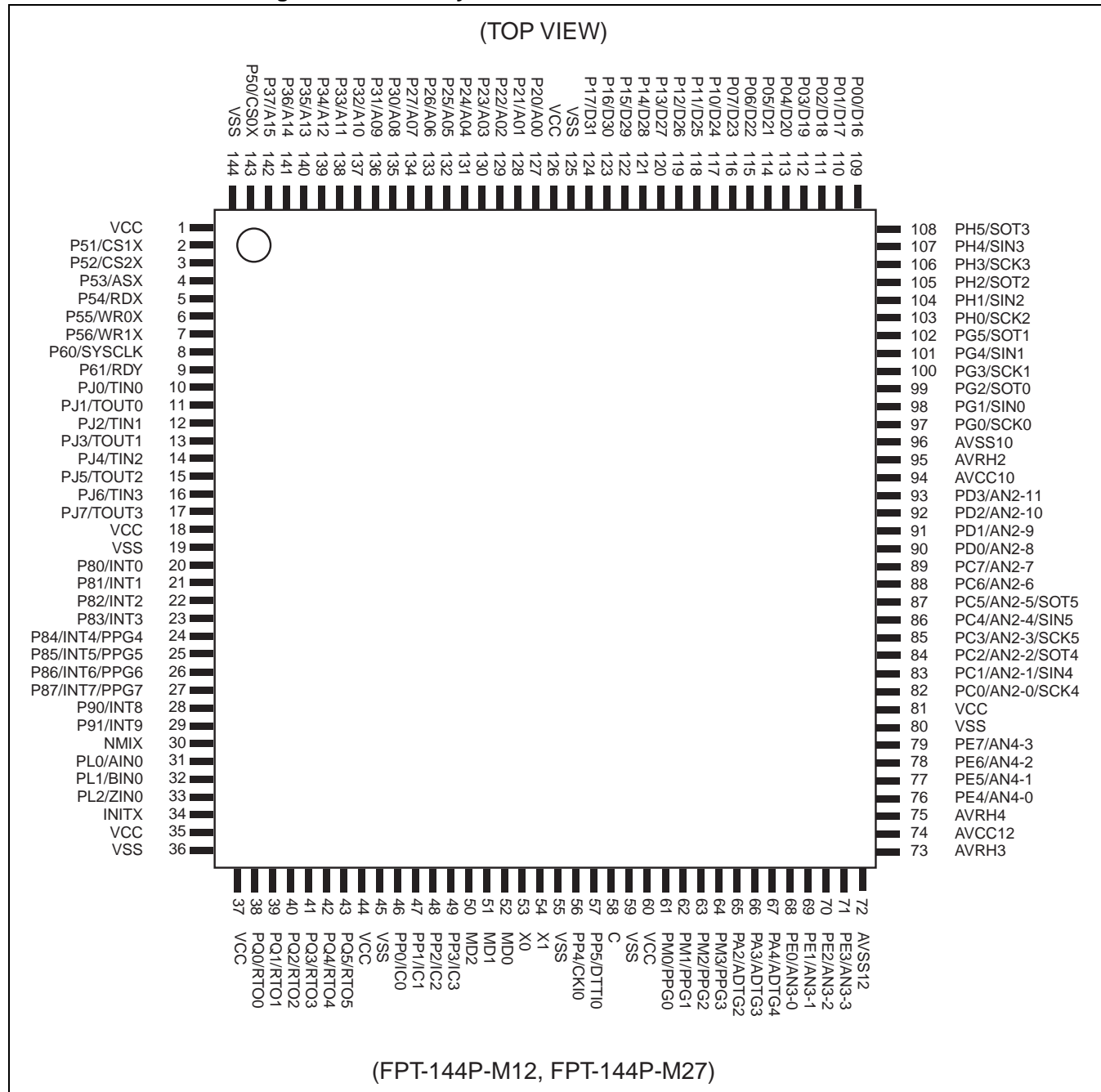
■ Block Diagram of MB91470 Series (144 Pins)

Figure 1.2-1 Block Diagram of MB91470 Series (144 Pins)



MB91470/480 Series**1.3 Pin Assignment**

This section shows the pin assignment and package dimension of MB91470/480 series.

■ LQFP-144 (MB91470 Series)**Figure 1.3-1 Pin Layout of MB91470 Series - LQFP144 Pin**

■ **FBGA-144 (MB91470 Series)**

Figure 1.3-2 Pin Layout of MB91480 Series - FBGA - 144 Pin

▼ Index	1	2	3	4	5	6	7	8	9	10	11	12	13	
A	1	48	47	46	45	44	43	42	41	40	39	38	37	A
B	2	49	88	87	86	85	84	83	82	81	80	79	36	B
C	3	50	89	120	119	118	117	116	115	114	113	78	35	C
D	4	51	90	121	144	143	142	141	140	139	112	77	34	D
E	5	52	91	122						138	111	76	33	E
F	6	53	92	123						137	110	75	32	F
G	7	54	93	124						136	109	74	31	G
H	8	55	94	125						135	108	73	30	H
J	9	56	95	126						134	107	72	29	J
K	10	57	96	127	128	129	130	131	132	133	106	71	28	K
L	11	58	97	98	99	100	101	102	103	104	105	70	27	L
M	12	59	60	61	62	63	64	65	66	67	68	69	26	M
N	13	14	15	16	17	18	19	20	21	22	23	24	25	N
	1	2	3	4	5	6	7	8	9	10	11	12	13	

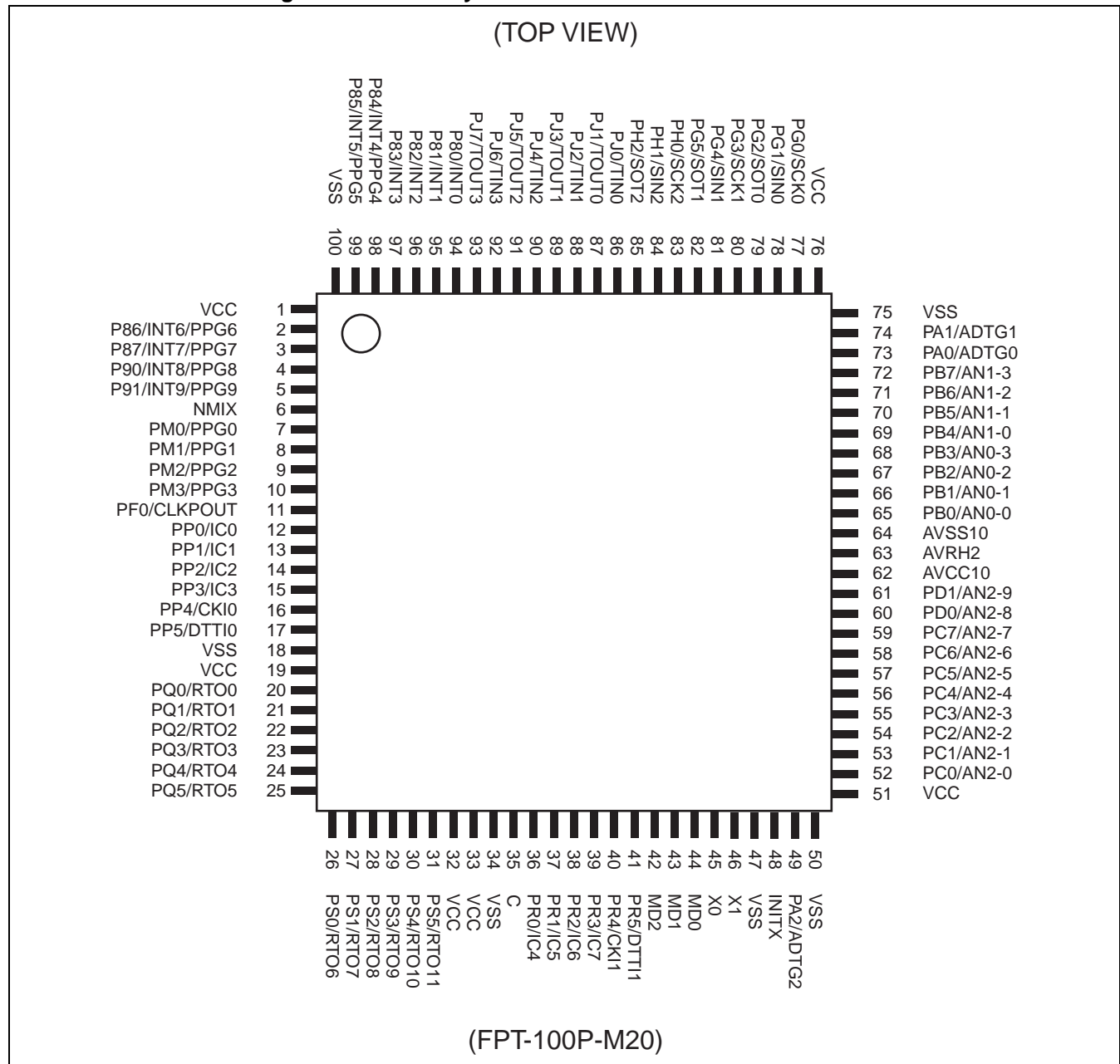
TOP VIEW

(BGA-144P-M06)

MB91470/480 Series

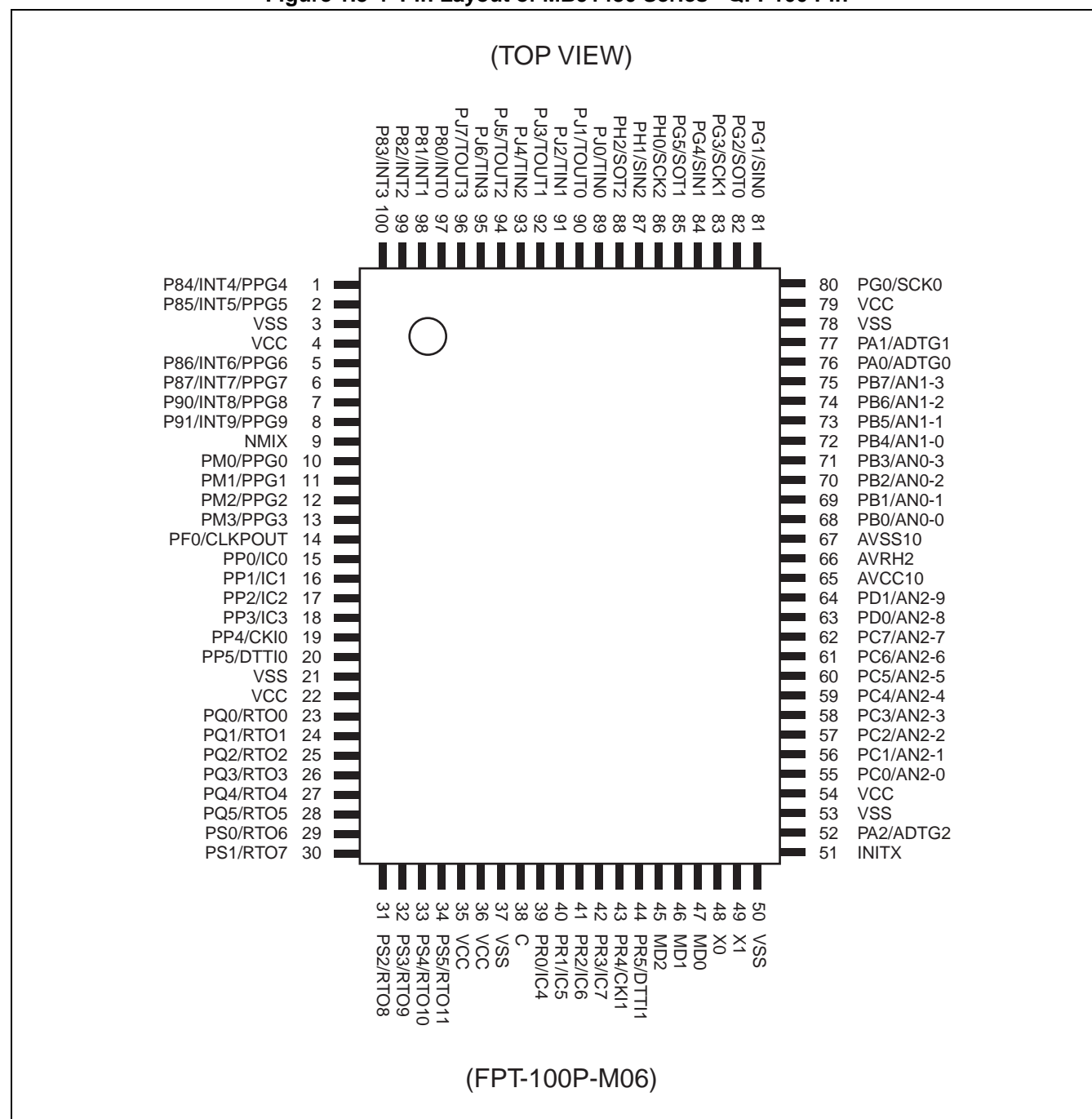
■ LQFP-100 (MB91480 Series)

Figure 1.3-3 Pin Layout of MB91480 Series - LQFP100 Pin



■ QFP-100 (MB91480 Series)

Figure 1.3-4 Pin Layout of MB91480 Series - QFP100 Pin



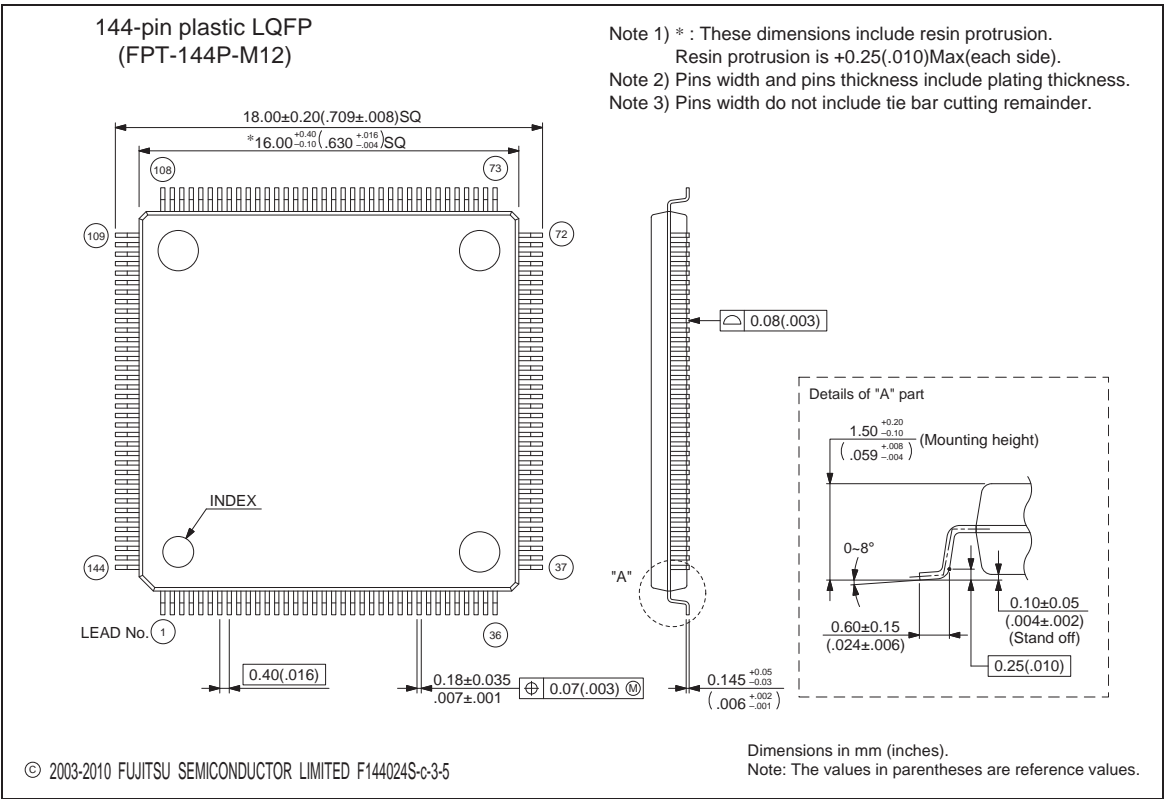
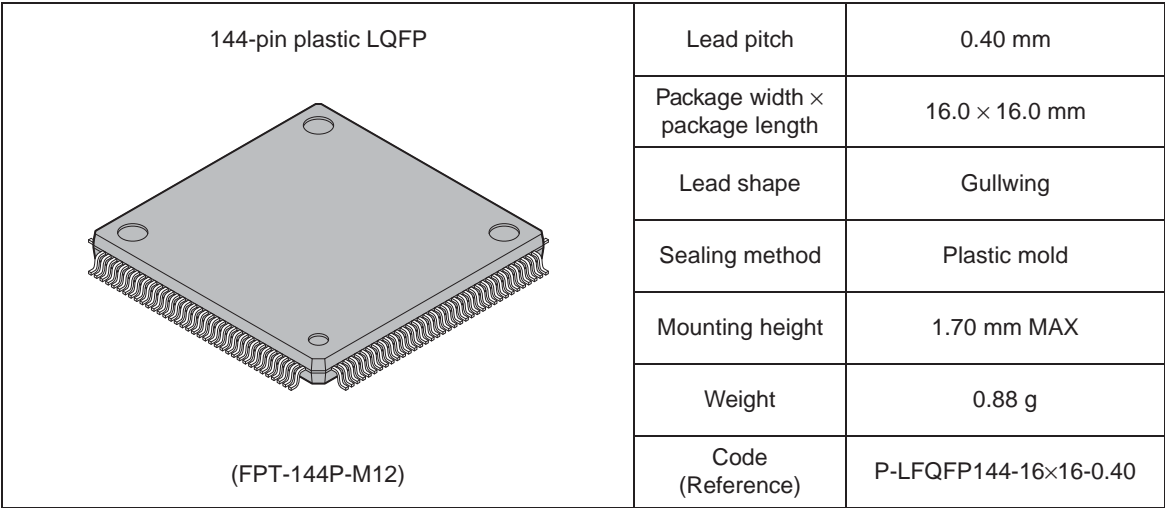
MB91470/480 Series

1.4 Package Dimension

This section shows the package dimension used in MB91470/480 series.

■ Package Dimension (LQFP-144P-M12)

Figure 1.4-1 Package Dimension of FPT-144P-M12

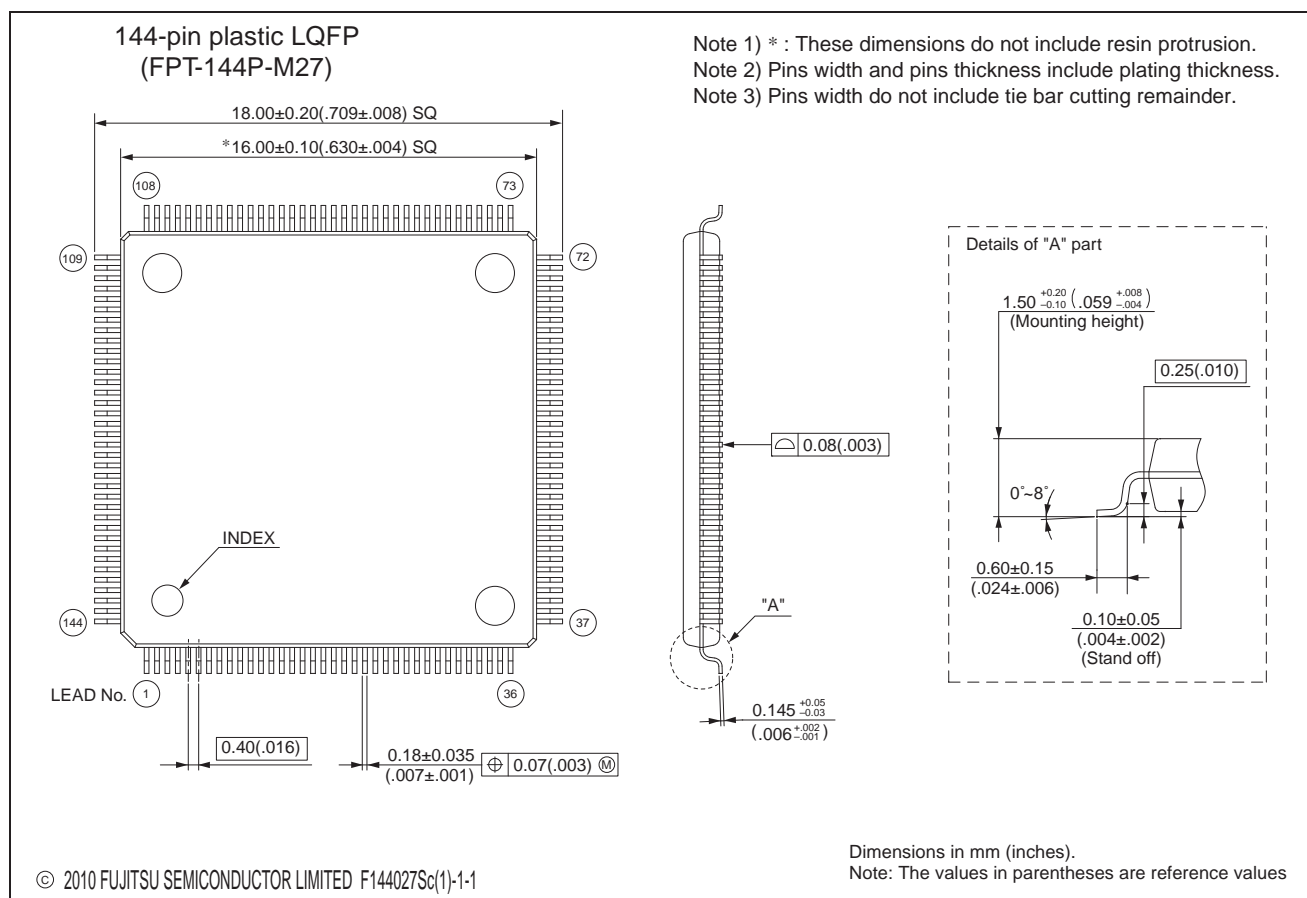


Please check the latest package dimension at the following URL.
<http://edevice.fujitsu.com/package/en-search/>

■ Package Dimension (FPT-144P-M27)

Figure 1.4-2 Package Dimension of FPT-144P-M27

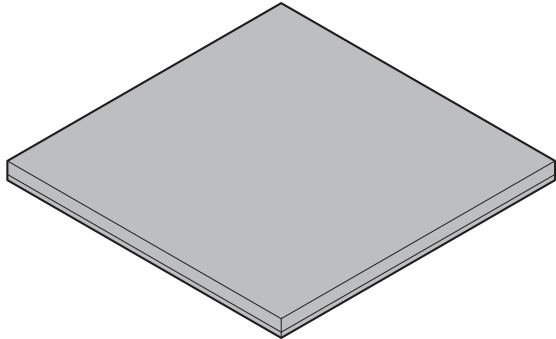
<p>144-pin plastic LQFP</p> <p>(FPT-144P-M27)</p>	Lead pitch	0.40 mm
	Package width × package length	16.0 mm × 16.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm Max
	Weight	0.88 g
	Code (Reference)	P-LFQFP144-16×16-0.40

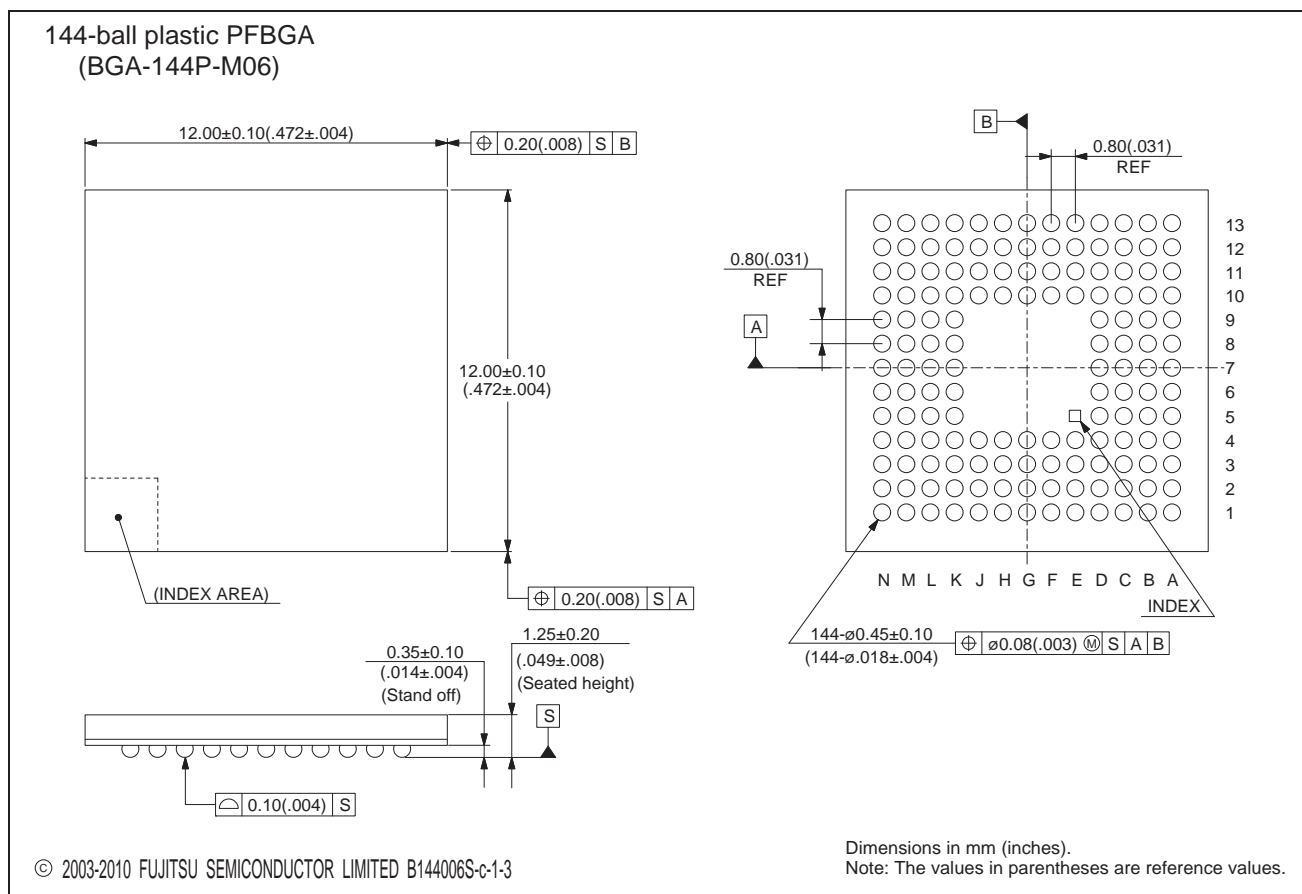


Please check the latest package dimension at the following URL.
<http://edevice.fujitsu.com/package/en-search/>

■ Package Dimension (BGA-144P-M06)

Figure 1.4-3 Package Dimension of BGA-144P-M06

<p>144-ball plastic PFBGA</p>  <p>(BGA-144P-M06)</p>	Ball pitch	0.80 mm
	Package width × package length	12.00 × 12.00 mm
	Lead shape	Soldering ball
	Sealing method	Plastic mold
	Ball size	Ø0.45 mm
	Mounting height	1.45 mm Max.
	Weight	0.32 g

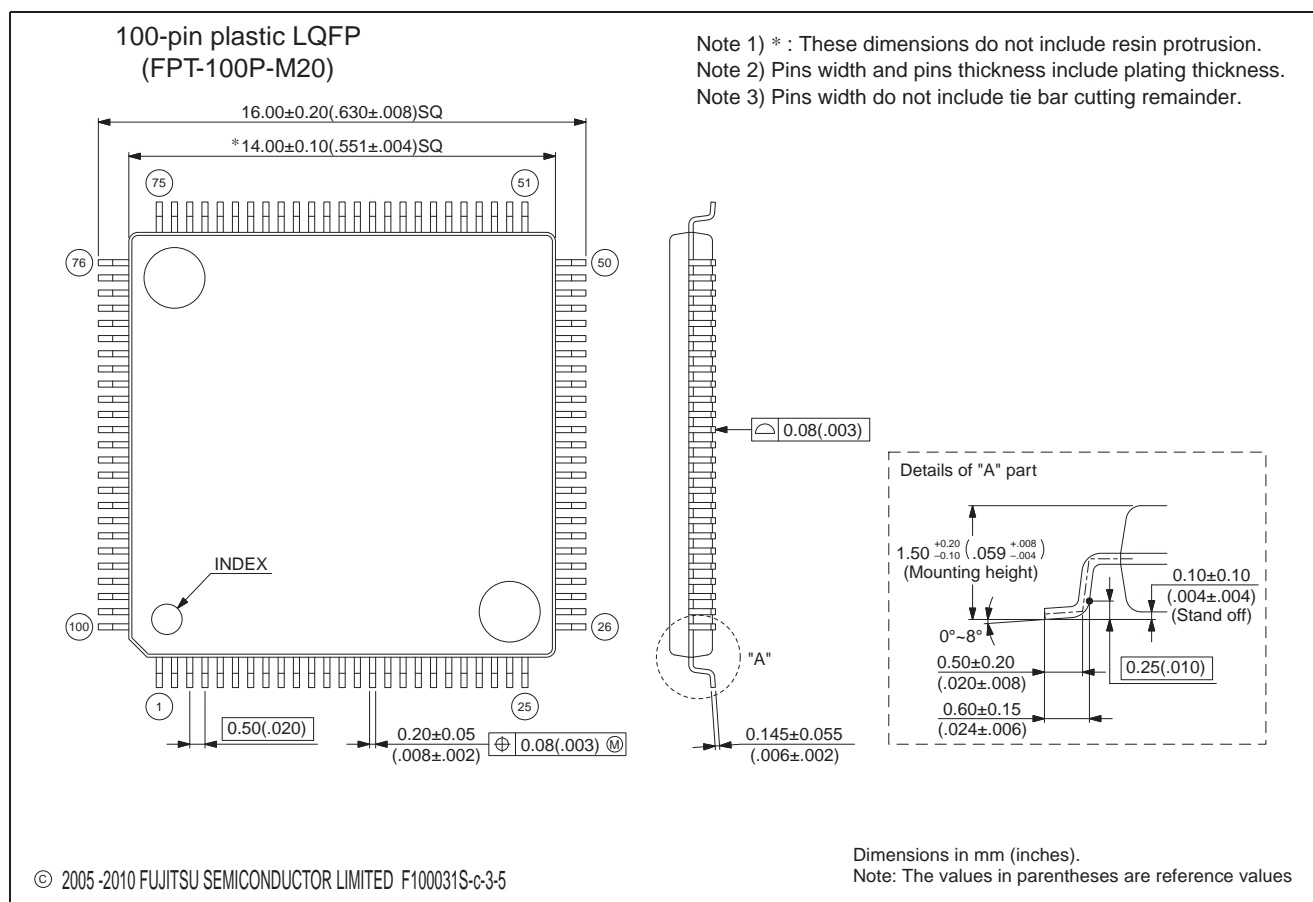


Please check the latest package dimension at the following URL.
<http://edevic.fujitsu.com/package/en-search/>

■ Package Dimension (LQFP-100P-M20)

Figure 1.4-4 Package Dimension of FPT-100P-M20

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

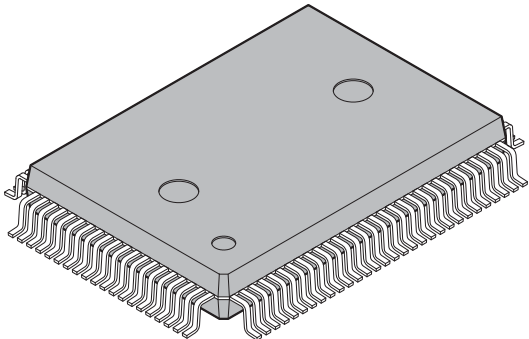


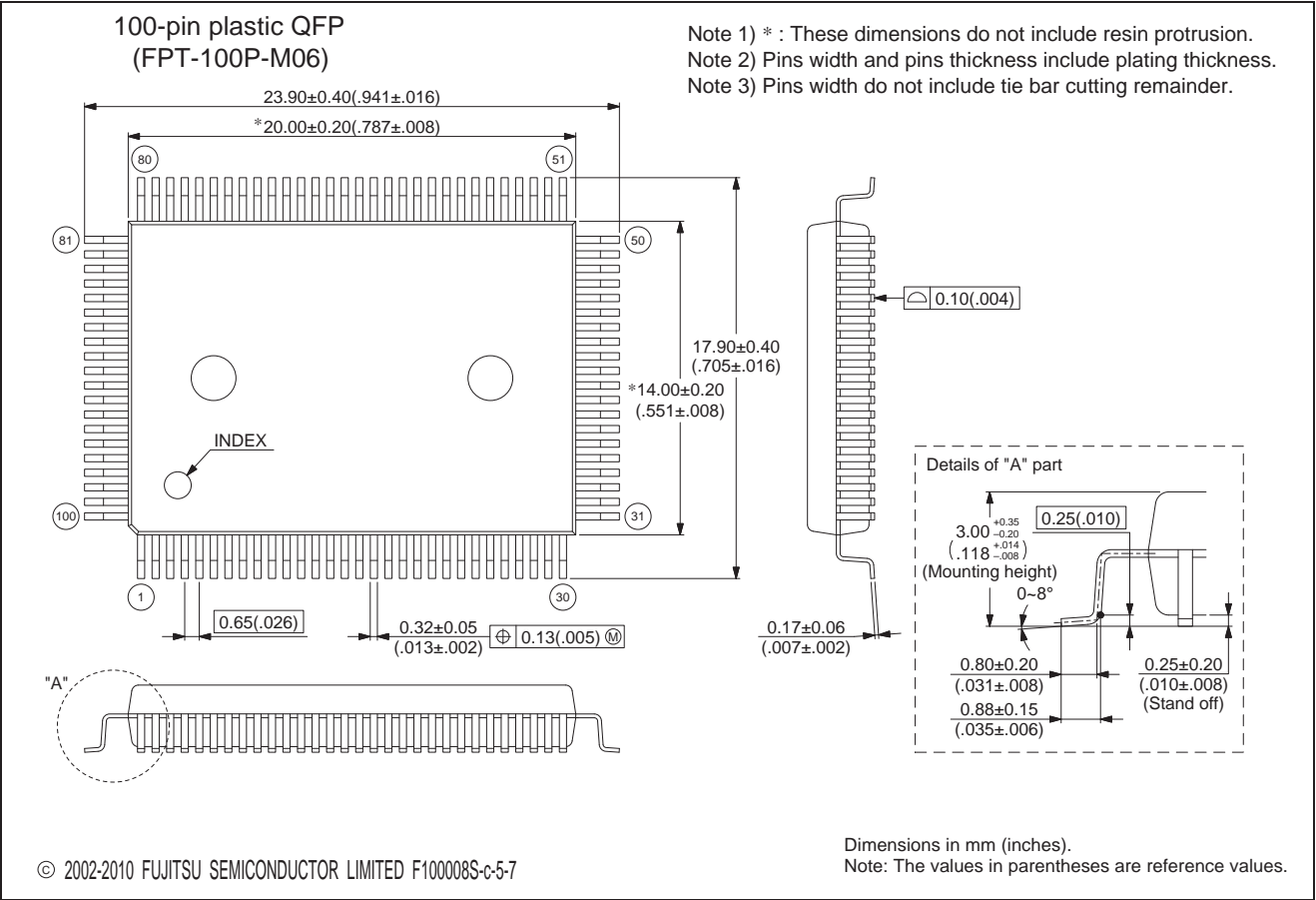
Please check the latest package dimension at the following URL.
<http://edevic.fujitsu.com/package/en-search/>

MB91470/480 Series

■ Package Dimension (QFP-100P-M06)

Figure 1.4-5 Package Dimension of FPT-100P-M06

 <p>100-pin plastic QFP</p> <p>(FPT-100P-M06)</p>	Lead pitch	0.65 mm
	Package width × package length	14.00 × 20.00 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	3.35 mm MAX
	Code (Reference)	P-QFP100-14×20-0.65



Please check the latest package dimension at the following URL.
<http://edevice.fujitsu.com/package/en-search/>

1.5 List of Pin Functions

Table 1.5-1 lists the pin functions.

■ List of Pin Functions

Table 1.5-1 List of Pin Functions (1 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
50	M6	42	45	MD2	H,K	Mode terminal 2. Setting terminal determines the basic operation mode. Connect to Vcc or Vss. The circuit type of Flash memory models is K.
51	N6	43	46	MD1	H,K	Mode terminal 1. Setting terminal determines the basic operation mode. Connect to Vcc or Vss. The circuit type of Flash memory models is K.
52	K5	44	47	MD0	H,K	Mode terminal 0. Setting terminal determines the basic operation mode. Connect to Vcc or Vss. The circuit type of Flash memory models is K.
53	L6	45	48	X0	A	Clock (oscillator) input
54	K6	46	49	X1	A	Clock (oscillator) output
34	L1	48	51	INITX	I	External reset input
30	J4	6	9	NMIX	H	NMI (Non Maskable Interrupt) input
109	A12	-	-	D16	C	External data bus input/output pin bit 16
				P00		General purpose input/output port
110	B12	-	-	D17	C	External data bus input/output pin bit17
				P01		General purpose input/output port
111	A11	-	-	D18	C	External data bus input/output pin bit18
				P02		General purpose input/output port
112	B11	-	-	D19	C	External data bus input/output pin bit19
				P03		General purpose input/output port
113	C12	-	-	D20	C	External data bus input/output pin bit20
				P04		General purpose input/output port
114	B10	-	-	D21	C	External data bus input/output pin bit21
				P05		General purpose input/output port
115	A10	-	-	D22	C	External data bus input/output pin bit22
				P06		General purpose input/output port
116	C11	-	-	D23	C	External data bus input/output pin bit23
				P07		General purpose input/output port

Table 1.5-1 List of Pin Functions (2 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
117	C10	-	-	D24	C	External data bus input/output pin bit24
				P10		General purpose input/output port
118	B9	-	-	D25	C	External data bus input/output pin bit25
				P11		General purpose input/output port
119	A9	-	-	D26	C	External data bus input/output pin bit26
				P12		General purpose input/output port
120	D10	-	-	D27	C	External data bus input/output pin bit27
				P13		General purpose input/output port
121	C9	-	-	D28	C	External data bus input/output pin bit28
				P14		General purpose input/output port
122	B8	-	-	D29	C	External data bus input/output pin bit29
				P15		General purpose input/output port
123	A8	-	-	D30	C	External data bus input/output pin bit30
				P16		General purpose input/output port
124	D9	-	-	D31	C	External data bus input/output pin bit31
				P17		General purpose input/output port
127	A7	-	-	A00	C	External address bus output pin bit0
				P20		General purpose input/output port
128	B7	-	-	A01	C	External address bus output pin bit1
				P21		General purpose input/output port
129	C7	-	-	A02	C	External address bus output pin bit2
				P22		General purpose input/output port
130	D7	-	-	A03	C	External address bus output pin bit3
				P23		General purpose input/output port
131	A6	-	-	A04	C	External address bus output pin bit4
				P24		General purpose input/output port
132	B6	-	-	A05	C	External address bus output pin bit5
				P25		General purpose input/output port
133	C6	-	-	A06	C	External address bus output pin bit6
				P26		General purpose input/output port
134	D6	-	-	A07	C	External address bus output pin bit7
				P27		General purpose input/output port
135	A5	-	-	A08	C	External address bus output pin bit8
				P30		General purpose input/output port
136	B5	-	-	A09	C	External address bus output pin bit9
				P31		General purpose input/output port

Table 1.5-1 List of Pin Functions (3 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
137	C5	-	-	A10	C	External address bus output pin bit10
				P32		General purpose input/output port
138	D5	-	-	A11	C	External address bus output pin bit11
				P33		General purpose input/output port
139	A4	-	-	A12	C	External address bus output pin bit12
				P34		General purpose input/output port
140	B4	-	-	A13	C	External address bus output pin bit13
				P35		General purpose input/output port
141	C4	-	-	A14	C	External address bus output pin bit14
				P36		General purpose input/output port
142	A3	-	-	A15	C	External address bus output pin bit15
				P37		General purpose input/output port
143	A2	-	-	CS0X	C	External chip select 0 output
				P50		General purpose input/output port
2	B2	-	-	CS1X	C	External chip select 1 output
				P51		General purpose input/output port
3	C1	-	-	CS2X	C	External chip select 2 output
				P52		General purpose input/output port
4	C2	-	-	ASX	C	External address strobe output
				P53		General purpose input/output port
5	B3	-	-	RDX	C	External read strobe output
				P54		General purpose input/output port
6	D2	-	-	WR0X	C	External write strobe output. This terminal corresponded to external data bus input/output bit31 to bit24.
				P55		General purpose input/output port
7	D1	-	-	WR1X	C	External write strobe output. This terminal corresponded to external data bus input/output bit23 to bit16.
				P56		General purpose input/output port
8	C3	-	-	SYSCLK	C	External clock output
				P60		General purpose input/output port
9	D3	-	-	RDY	C	External ready input
				P61		General purpose input/output port
20	G2	94	97	INT0	D	External interrupt 0 input
				P80		General purpose input/output port
21	G3	95	98	INT1	D	External interrupt 1 input
				P81		General purpose input/output port

Table 1.5-1 List of Pin Functions (4 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
22	G4	96	99	INT2	D	External interrupt 2 input
				P82		General purpose input/output port
23	H1	97	100	INT3	D	External interrupt 3 input
				P83		General purpose input/output port
24	H2	98	1	INT4	D	External interrupt 4 input
				PPG4		PPG timer 4 output
				P84		General purpose input/output port
25	H3	99	2	INT5	D	External interrupt 5 input
				PPG5		PPG timer 5 output
				P85		General purpose input/output port
26	H4	2	5	INT6	D	External interrupt 6 input
				PPG6		PPG timer 6 output
				P86		General purpose input/output port
27	J1	3	6	INT7	D	External interrupt 7 input
				PPG7		PPG timer 7 output
				P87		General purpose input/output port
28	J2	4	7	INT8	D	External interrupt 8 input
				PPG8		PPG timer 8 output
				P90		General purpose input/output port
29	J3	5	8	INT9	D	External interrupt 9 input
				PPG9		PPG timer 9 output
				P91		General purpose input/output port
-	-	-	-	INT10	D	External interrupt 10 input
				PPG10		PPG timer 10 output
				P92		General purpose input/output port
-	-	-	-	INT11	D	External interrupt 11 input
				PPG11		PPG timer 11 output
				P93		General purpose input/output port
-	-	-	-	INT12	D	External interrupt 12 input
				PPG12		PPG timer 12 output
				P94		General purpose input/output port
-	-	-	-	INT13	D	External interrupt 13 input
				PPG13		PPG timer 13 output
				P95		General purpose input/output port
-	-	-	-	INT14	D	External interrupt 14 input
				PPG14		PPG timer 14 output
				P96		General purpose input/output port

Table 1.5-1 List of Pin Functions (5 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
-	-	-	-	INT15	D	External interrupt 15 input
				PPG15		PPG timer 15 output
				P97		General purpose input/output port
-	-	73	76	ADTG0	D	8/10-bit A/D converter 0 external trigger input
				PA0		General purpose input/output port
-	-	74	77	ADTG1	D	8/10-bit A/D converter 1 external trigger input
				PA1		General purpose input/output port
65	L9	49	52	ADTG2	D	8/10-bit A/D converter 2 external trigger input
				PA2		General purpose input/output port
66	K9	-	-	ADTG3	D	12-bit A/D converter 3 external trigger input
				PA3		General purpose input/output port
67	N10	-	-	ADTG4	D	12-bit A/D converter 4 external trigger input
				PA4		General purpose input/output port
-	-	65	68	AN0-0	G	8/10-bit A/D converter 0 analog 0 input
				PB0		General purpose input/output port
-	-	66	69	AN0-1	G	8/10-bit A/D converter 0 analog 1 input
				PB1		General purpose input/output port
-	-	67	70	AN0-2	G	8/10-bit A/D converter 0 analog 2 input
				PB2		General purpose input/output port
-	-	68	71	AN0-3	G	8/10-bit A/D converter 0 analog 3 input
				PB3		General purpose input/output port
-	-	69	72	AN1-0	G	8/10-bit A/D converter 1 analog 0 input
				PB4		General purpose input/output port
-	-	70	73	AN1-1	G	8/10-bit A/D converter 1 analog 1 input
				PB5		General purpose input/output port
-	-	71	74	AN1-2	G	8/10-bit A/D converter 1 analog 2 input
				PB6		General purpose input/output port
-	-	72	75	AN1-3	G	8/10-bit A/D converter 1 analog 3 input
				PB7		General purpose input/output port
82	J12	52	55	AN2-0	G	8/10-bit A/D converter 2 analog 0 input
				SCK4 (SCL4)		Multi-function serial interface 4 clock input/output (SCL4 at I ² C mode)
				PC0		General purpose input/output port
83	J13	53	56	AN2-1	G	8/10-bit A/D converter 2 analog 1 input
				SIN4		Multi-function serial interface 4 data input (Not used at I ² C mode)
				PC1		General purpose input/output port

Table 1.5-1 List of Pin Functions (6 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
84	K10	54	57	AN2-2	G	8/10-bit A/D converter 2 analog 2 input
				SOT4 (SDA4)		Multi-function serial interface 4 data output (SDA4 at I ² C mode)
				PC2		General purpose input/output port
85	J11	55	58	AN2-3	G	8/10-bit A/D converter 2 analog 3 input
				SCK5 (SCL5)		Multi-function serial interface 5 clock input/output (SCL5 at I ² C mode)
				PC3		General purpose input/output port
86	H12	56	59	AN2-4	G	8/10-bit A/D converter 2 analog 4 input
				SIN5		Multi-function serial interface 5 data input (Not used at I ² C mode)
				PC4		General purpose input/output port
87	H13	57	60	AN2-5	G	8/10-bit A/D converter 2 analog 5 input
				SOT5 (SDA5)		Multi-function serial interface 5 data output (SDA5 at I ² C mode)
				PC5		General purpose input/output port
88	J10	58	61	AN2-6	G	8/10-bit A/D converter 2 analog 6 input
				PC6		General purpose input/output port
89	H11	59	62	AN2-7	G	8/10-bit A/D converter 2 analog 7 input
				PC7		General purpose input/output port
90	H10	60	63	AN2-8	G	8/10-bit A/D converter 2 analog 8 input
				PD0		General purpose input/output port
91	G13	61	64	AN2-9	G	8/10-bit A/D converter 2 analog 9 input
				PD1		General purpose input/output port
92	G12	-	-	AN2-10	G	8/10-bit A/D converter 2 analog 10 input
				PD2		General purpose input/output port
93	G11	-	-	AN2-11	G	8/10-bit A/D converter 2 analog 11 input
				PD3		General purpose input/output port
68	M10	-	-	AN3-0/ AN3-0P	G	12-bit A/D converter 3 analog 0 input (in normal input mode). 12-bit A/D converter 3 analog 0 (+) side input (in differential input mode)
				PE0		General purpose input/output port
69	L10	-	-	AN3-1/ AN3-0N	G	12-bit A/D converter 3 analog 1 input (in normal input mode). 12-bit A/D converter 3 analog 0(-) side input (in differential input mode)
				PE1		General purpose input/output port

Table 1.5-1 List of Pin Functions (7 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
70	N11	-	-	AN3-2/ AN3-1P	G	12-bit A/D converter 3 analog 2 input (in normal input mode). 12-bit A/D converter 3 analog 1(+) side input (in differential input mode)
				PE2		General purpose input/output port
71	N12	-	-	AN3-3/ AN3-1N	G	12-bit A/D converter 3 analog 3 input (in normal input mode). 12-bit A/D converter 3 analog 1(-) side input (in differential input mode)
				PE3		General purpose input/output port
76	L12	-	-	AN4-0/ AN4-0P	G	12-bit A/D converter 4 analog 0 input (in normal input mode). 12-bit A/D converter 4 analog 0(+) side input (in differential input mode)
				PE4		General purpose input/output port
77	M11	-	-	AN4-1/ AN4-0N	G	12-bit A/D converter 4 analog 1 input (in normal input mode). 12-bit A/D converter 4 analog 0(-) side input (in differential input mode)
				PE5		General purpose input/output port
78	K12	-	-	AN4-2/ AN4-1P	G	12-bit A/D converter 4 analog 2 input (in normal input mode). 12-bit A/D converter 4 analog 1(+) side input (in differential input mode)
				PE6		General purpose input/output port
79	K13	-	-	AN4-3/ AN4-1N	G	12-bit A/D converter 4 analog 3 input (in normal input mode). 12-bit A/D converter 4 analog 1(-) side input (in differential input mode)
				PE7		General purpose input/output port
-	-	11	14	CLKPOUT	D	Clock monitor output
				PF0		General purpose input/output port
-	-	-	-	PF1	D	General purpose input/output port
-	-	-	-	PF2	D	General purpose input/output port
-	-	-	-	PF3	D	General purpose input/output port
-	-	-	-	PF4	D	General purpose input/output port
-	-	-	-	PF5	D	General purpose input/output port
-	-	-	-	PF6	D	General purpose input/output port
-	-	-	-	PF7	D	General purpose input/output port

Table 1.5-1 List of Pin Functions (8 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
97	F11	77	80	SCK0 (SCL0)	D	Multi-function serial interface 0 clock input/output (SCL0 at I ² C mode)
				PG0		General purpose input/output port
98	F10	78	81	SIN0	D	Multi-function serial interface 0 data input (Not used at I ² C mode)
				PG1		General purpose input/output port
99	E13	79	82	SOT0 (SDA0)	D	Multi-function serial interface 0 data output (SDA0 at I ² C mode)
				PG2		General purpose input/output port
100	E12	80	83	SCK1 (SCL1)	D	Multi-function serial interface 1 clock input/output (SCL1 at I ² C mode)
				PG3		General purpose input/output port
101	E11	81	84	SIN1	D	Multi-function serial interface 1 data input (Not used at I ² C mode)
				PG4		General purpose input/output port
102	E10	82	85	SOT1 (SDA1)	D	Multi-function serial interface 1 data output (SDA1 at I ² C mode)
				PG5		General purpose input/output port
103	D13	83	86	SCK2 (SCL2)	D	Multi-function serial interface 2 clock input/output (SCL2 at I ² C mode)
				PH0		General purpose input/output port
104	D12	84	87	SIN2	D	Multi-function serial interface 2 data input (Not used at I ² C mode)
				PH1		General purpose input/output port
105	D11	85	88	SOT2 (SDA2)	D	Multi-function serial interface 2 data output (SDA2 at I ² C mode)
				PH2		General purpose input/output port
106	C13	-	-	SCK3 (SCL3)	D	Multi-function serial interface 3 clock input/output (SCL3 at I ² C mode)
				PH3		General purpose input/output port
107	B13	-	-	SIN3	D	Multi-function serial interface 3 data input (Not used at I ² C mode)
				PH4		General purpose input/output port
108	A13	-	-	SOT3 (SDA3)	D	Multi-function serial interface 3 data output (SDA3 at I ² C mode)
				PH5		General purpose input/output port
10	E2	86	89	TIN0	D	Base timer 0 input
				PJ0		General purpose input/output port
11	E1	87	90	TOUT0	D	Base timer 0 output
				PJ1		General purpose input/output port

Table 1.5-1 List of Pin Functions (9 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
12	D4	88	91	TIN1	D	Base timer 1 input
				PJ2		General purpose input/output port
13	E3	89	92	TOUT1	D	Base timer 1 output
				PJ3		General purpose input/output port
14	F2	90	93	TIN2	D	Base timer 2 input
				PJ4		General purpose input/output port
15	F1	91	94	TOUT2	D	Base timer 2 output
				PJ5		General purpose input/output port
16	E4	92	95	TIN3	D	Base timer 3 input
				PJ6		General purpose input/output port
17	F3	93	96	TOUT3	D	Base timer 3 output
				PJ7		General purpose input/output port
31	K1	-	-	AIN0	D	8/16-bit up count input terminal for up/down counter 0
				PL0		General purpose input/output port
32	K2	-	-	BIN0	D	8/16-bit down count input terminal for up/down counter 0
				PL1		General purpose input/output port
33	K3	-	-	ZIN0	D	8/16-bit reset input terminal for up/down counter 0
				PL2		General purpose input/output port
61	L8	7	10	PPG0	D	PPG timer 0 output
				PM0		General purpose input/output port
62	K8	8	11	PPG1	D	PPG timer 1 output
				PM1		General purpose input/output port
63	N9	9	12	PPG2	D	PPG timer 2 output
				PM2		General purpose input/output port
64	M9	10	13	PPG3	D	PPG timer 3 output
				PM3		General purpose input/output port
46	M5	12	15	IC0	D	Trigger input of input capture 0
				PP0		General purpose input/output port
47	N5	13	16	IC1	D	Trigger input of input capture 1
				PP1		General purpose input/output port
48	K4	14	17	IC2	D	Trigger input of input capture 2
				PP2		General purpose input/output port
49	L5	15	18	IC3	D	Trigger input of input capture 3
				PP3		General purpose input/output port
56	M7	16	19	CKI0	D	External clock input terminal of free-run timer ch.0 to ch.2
				PP4		General purpose input/output port

Table 1.5-1 List of Pin Functions (10 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
57	L7	17	20	DTTI0	D	Input signal controlled multi-function timer 0 waveform generator output RTO0 to RTO5
				PP5		General purpose input/output port
38	M2	20	23	RTO0	J	Waveform generator output of multi-function timer 0
				PQ0		General purpose input/output port
39	N3	21	24	RTO1	J	Waveform generator output of multi-function timer 0
				PQ1		General purpose input/output port
40	M3	22	25	RTO2	J	Waveform generator output of multi-function timer 0
				PQ2		General purpose input/output port
41	L2	23	26	RTO3	J	Waveform generator output of multi-function timer 0
				PQ3		General purpose input/output port
42	M4	24	27	RTO4	J	Waveform generator output of multi-function timer 0
				PQ4		General purpose input/output port
43	N4	25	28	RTO5	J	Waveform generator output of multi-function timer 0
				PQ5		General purpose input/output port
-	-	36	39	IC4	D	Trigger input of input capture 4
				PR0		General purpose input/output port
-	-	37	40	IC5	D	Trigger input of input capture 5
				PR1		General purpose input/output port
-	-	38	41	IC6	D	Trigger input of input capture 6
				PR2		General purpose input/output port
-	-	39	42	IC7	D	Trigger input of input capture 7
				PR3		General purpose input/output port
-	-	40	43	CKI1	D	External clock input terminal of free-run timer ch.3 to ch.5
				PR4		General purpose input/output port
-	-	41	44	DTTI1	D	Input signal controlled multi-function timer 1 waveform generator output RTO6 to RTO11
				PR5		General purpose input/output port
-	-	26	29	RTO6	J	Waveform generator output of multi-function timer 1
				PS0		General purpose input/output port
-	-	27	30	RTO7	J	Waveform generator output of multi-function timer 1
				PS1		General purpose input/output port
-	-	28	31	RTO8	J	Waveform generator output of multi-function timer 1
				PS2		General purpose input/output port
-	-	29	32	RTO9	J	Waveform generator output of multi-function timer 1
				PS3		General purpose input/output port

Table 1.5-1 List of Pin Functions (11 / 11)

Pin No.				Pin Name	I/O Circuit Type *	Function
MB91470 Series		MB91480 Series				
LQFP-144	FBGA-144	LQFP-100	QFP-100			
-	-	30	33	RTO10	J	Waveform generator output of multi-function timer 1
				PS4		General purpose input/output port
-	-	31	34	RTO11	J	Waveform generator output of multi-function timer 1
				PS5		General purpose input/output port

*: For the I/O circuit type, refer to "1.6 I/O Circuit Type".

[Power supply and GND pins]

Pin No.				Pin Name	Function
MB91470 Series		MB91480 Series			
LQFP-144	FBGA-144	LQFP-100	QFP-100		
1	B1			VCC	Power-supply pins. Use all of these pins at equal potential.
18	F4	1	4		
35	M1	19	22		
37	N2	32	35		
44	L3	33	36		
60	M8	51	54		
81	K11	76	79		
126	D8				
19	A1			VSS	GND pins. Use all of these pins at equal potential.
36	G1	18	21		
45	N1	34	37		
55	L4	47	50		
59	N7	50	53		
80	N8	75	78		
125	L11	100	3		
144	C8				
58	K7	35	38	C	Capacitor coupling pin for internal regulator
94	G10	62	65	AVCC10	Analog power-supply pin for 8/10-bit A/D converter 0/1/2
96	F12	64	67	AVSS10	Analog GND pin for 8/10-bit A/D converter 0/1/2
74	M12	-	-	AVCC12	Analog power-supply pin for 12-bit A/D converter 3/4
72	N13	-	-	AVSS12	Analog GND pin for 12-bit A/D converter 3/4
-	-	-	-	AVRH0	Analog reference power-supply pin for 8/10-bit A/D converter 0
-	-	-	-	AVRH1	Analog reference power-supply pin for 8/10-bit A/D converter 1
95	F13	63	66	AVRH2	Analog reference power-supply pin for 8/10-bit A/D converter 2
73	M13	-	-	AVRH3	Analog reference power-supply pin for 12-bit A/D converter 3
75	L13	-	-	AVRH4	Analog reference power-supply pin for 12-bit A/D converter 4

MB91470/480 Series

1.6 I/O Circuit Type

Table 1.6-1 shows the I/O circuit type.

I/O Circuit Type

Table 1.6-1 I/O Circuit Type (1 / 3)

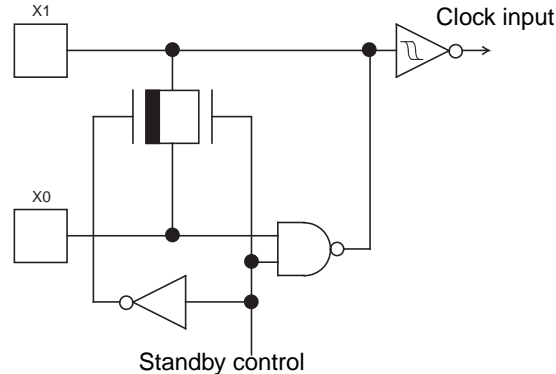
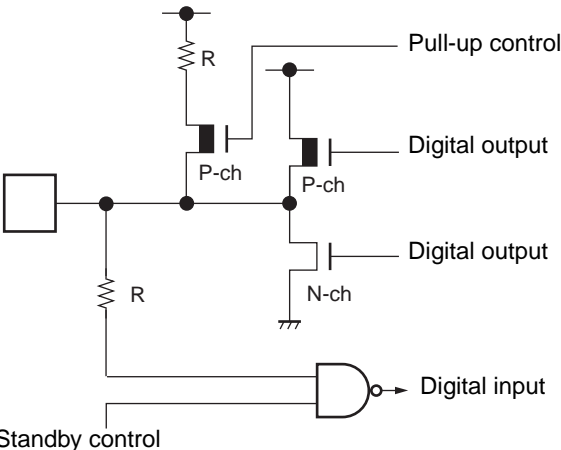
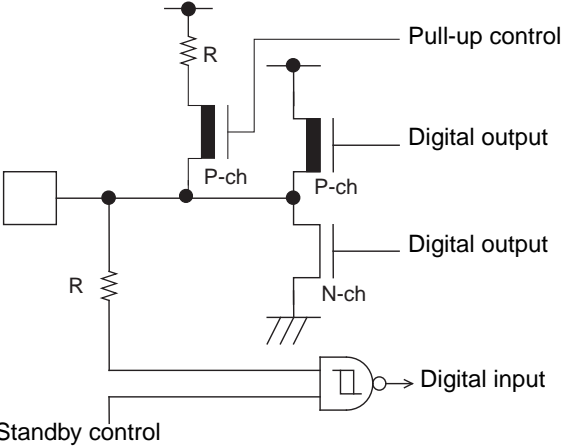
Type	Circuit	Remarks
A		Approx. 1MΩ oscillation feedback resistor for high-speed (source oscillation for main clock)
C		<ul style="list-style-type: none">• CMOS-level output• CMOS level input• Standby control provided• With pull-up control
D		<ul style="list-style-type: none">• CMOS-level output• CMOS level hysteresis input• Standby control provided• With pull-up control

Table 1.6-1 I/O Circuit Type (2 / 3)

Type	Circuit	Remarks
G	<p>Labels in diagram: Pull-up control, Digital output, P-ch, Digital output, N-ch, Digital input, Standby control, Analog input.</p>	<ul style="list-style-type: none"> • Analog/CMOS level hysteresis I/O pin • CMOS-level output • CMOS level hysteresis input (with standby control) • Analog input (Operates as an analog input when the corresponding AICR register bit is "1".) • With pull-up control
H	<p>Labels in diagram: P-ch, N-ch, Digital input.</p>	<ul style="list-style-type: none"> • CMOS level hysteresis input • No standby control
I	<p>Labels in diagram: P-ch, N-ch, Digital input.</p>	<ul style="list-style-type: none"> • CMOS level hysteresis input • With pull-up resistance • No standby control

Table 1.6-1 I/O Circuit Type (3 / 3)

Type	Circuit	Remarks
J	<p>Diagram J: This circuit features a pull-up resistor R connected to a P-channel MOSFET (P-ch) gate. The P-ch MOSFET's source is connected to VDD, and its drain is connected to a digital output. Another P-ch MOSFET gate is controlled by a pull-up control signal, with its source connected to VDD and its drain connected to the digital output. An N-channel MOSFET (N-ch) gate is connected to a digital output, with its source connected to ground and its drain connected to the digital output. A digital input is connected to a pull-up resistor R and a standby control signal through an AND gate. The standby control signal is also connected to the AND gate.</p>	<ul style="list-style-type: none"> • CMOS-level output • CMOS level hysteresis input • Standby control provided • With pull-up control
K	<p>Diagram K: This circuit consists of a stack of four N-channel MOSFETs (N-ch). The top N-ch MOSFET gate is connected to a control signal, with its source connected to VDD and its drain connected to the gate of the second N-ch MOSFET. The second N-ch MOSFET source is connected to VDD, and its drain is connected to the gate of the third N-ch MOSFET. The third N-ch MOSFET source is connected to VDD, and its drain is connected to the gate of the fourth N-ch MOSFET. The fourth N-ch MOSFET source is connected to VDD, and its drain is connected to a mode input through a resistor R. The mode input is also connected to ground through a resistor R.</p>	<p>Flash memory models only</p> <ul style="list-style-type: none"> • CMOS-level input • With high voltage control for Flash test

CHAPTER 2

HANDLING DEVICES

This chapter provides precautions on handling the device.

2.1 Precautions on Handling the Device

2.1 Precautions on Handling the Device

This section explains precautions on handling the device.

■ Precautions on Handling Device

● Preventing latch-up

Latch up phenomenon may occur with CMOS IC, when a voltage higher than VCC or lower than VSS is applied to either the input or output terminals, or when a voltage is applied between VCC and VSS that exceeds the rated voltage. When latch up occurs, a significant power-supply current surge results, which may damage some elements due to the excess heat, so great care must be taken to ensure that the maximum rating is never exceeded during use.

● Treatment of Unused Input Pins

Do not leave an unused input pin open, since it may cause a malfunction. Handle by, for example, using a pull-up or pull-down resistor.

● Power pins

In products with multiple VCC and VSS pins, the pins of the same potential are internally connected in the device to avoid abnormal operations including latch-up. However, you must connect the pins to the same potential power supply and a ground line externally to lower the electro-magnetic emission level, to prevent abnormal operation of strobe signals caused by the rise in the ground level, and to conform to the total output current rating.

Moreover, connect the current supply source with the VCC and VSS pins of this device at the low impedance.

It is also advisable to connect a ceramic capacitor of approximately 0.1 μF as a bypass capacitor between VCC and VSS near this device.

● Crystal oscillator circuit

Noise near the X0 and X1 pins may cause the device to malfunction. Design the printed circuit board so that X0, X1, the crystal oscillator (or ceramic oscillator), and the bypass capacitor to ground are located as close to the device as possible.

It is strongly recommended to design the PC board artwork with the X0 and X1 pins surrounded by ground plane because stable operation can be expected with such a layout. Please ask the crystal maker to evaluate the oscillational characteristics of the crystal and this device.

● About mode pins (MD0 to MD2)

Connect mode pins (MD0 to MD2) direct to VCC or VSS pins.

If pull-up or pull-down is necessary to change the mode pin level in rewriting the built-in FLASH or other processes, to prevent a device from accidentally entering the test mode due to noise, suppress the value of the resistor to use for pull-up or pull-down as low as possible, design the printed circuit board so that the layouts of the mode pins and VCC or VSS pins can be as near as possible to minimize the connection impedance.

● Operation at start-up

Be sure to execute setting initialized reset (INIT) with INITX pin immediately after start-up.

Immediately after that, also, hold the "L"-level input to the INITX pin for the stabilization wait time required for the oscillator circuit to take the oscillation stabilization wait time for the oscillator circuit and the stabilization wait time for the regulator (For INIT via the INITX pin, the oscillation stabilization wait time setting is initialized to the minimum value).

● Order of power turning ON/OFF

Use the following procedure for turning the power on or off. If not using the A/D converter, connect AVCC = VCC and AVSS = VSS. Turn on the power supply in the sequence VCC → AVCC → AVRH, and turn off the power in the reverse sequence.

● Source oscillation input when turning on the power

When turning the power on, maintain clock input until the device is released from the oscillation stabilization wait state.

● Caution for operation during PLL clock mode

Even if the oscillator comes off or the clock input stops with the PLL clock selected for MB91470/480 series, MB91470/480 series may continue to operate at the free-run frequency of the PLL's internal self-oscillating oscillator circuit.

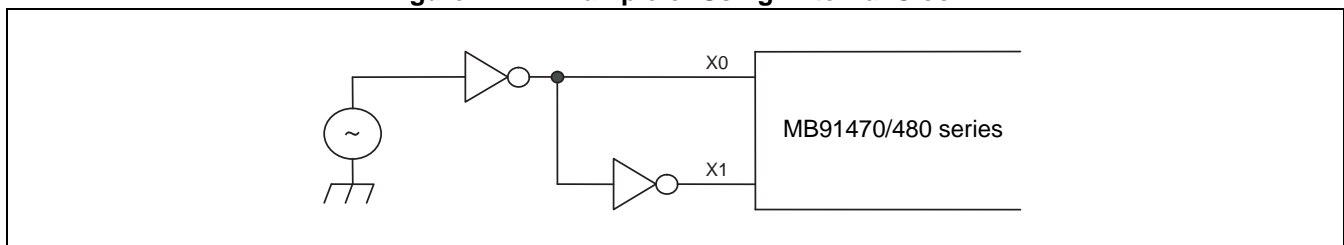
Performance of this operation, however, cannot be guaranteed.

● Using an external clock

When using an external clock, you must always input clock signals with opposite phase from X0 pin to X1 pin simultaneously. However, as the X1 pin halts with an output at the "H" level during STOP mode, insert a resistor of approximately 1kΩ externally to prevent a conflict between the two outputs if using STOP mode (oscillation stop mode).

The figure below shows an example of how to use an external clock.

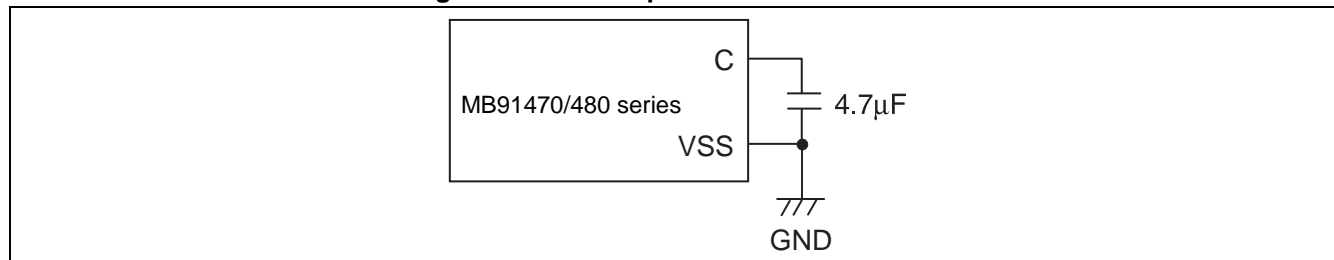
Figure 2.1-1 Example of Using External Clock



● C pin

As MB91470/480 series includes an internal regulator, always connect a bypass capacitor of approximately 4.7 μ F to the C pin for use by the regulator.

Figure 2.1-2 Example of C Pin Connection



● Software reset on the synchronous mode

Be sure to meet the following two conditions before setting 0 to the SRST bit of STCR (standby control register) when the software reset is used on the synchronous mode.

- Set the interrupt enable flag (I-Flag) to interrupt disable (I-Flag=0).
- Not used NMI

● Precautions at power-on

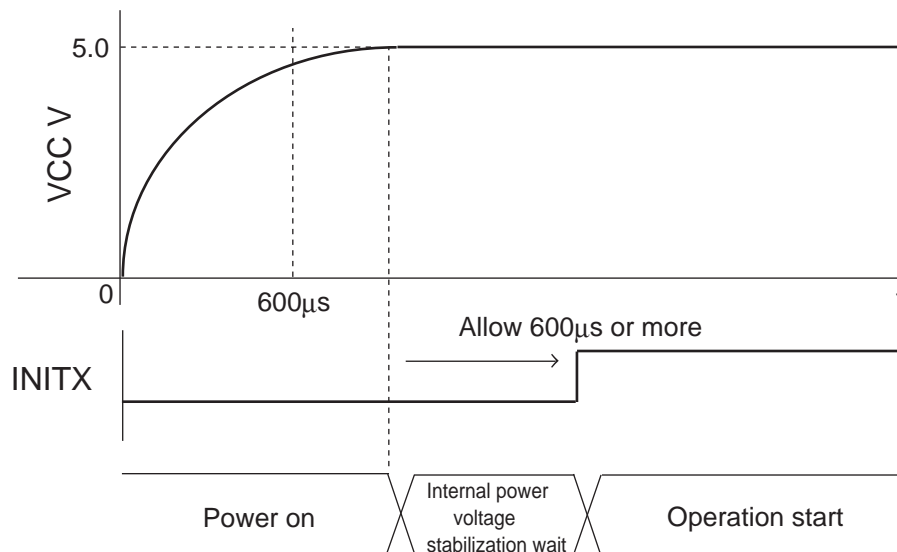
To prevent the device from malfunctioning due to overshoot of the embedded voltage reduction circuit, the voltage rising time at power-on is required to be 600 μ s (between 0.0V and 5.0V) or more. In addition, as it takes 600 μ s since the power voltage stabilization (after rising) until the internal voltage becomes stabilized, during which time INITX is required to be input continuously.

When the voltage rising time is less than 600 μ s (between 0.0V and 5.0V), as it takes 2ms* since the power voltage stabilization (after rising) until the internal voltage becomes stabilized, during which time INITX is required to be input continuously.

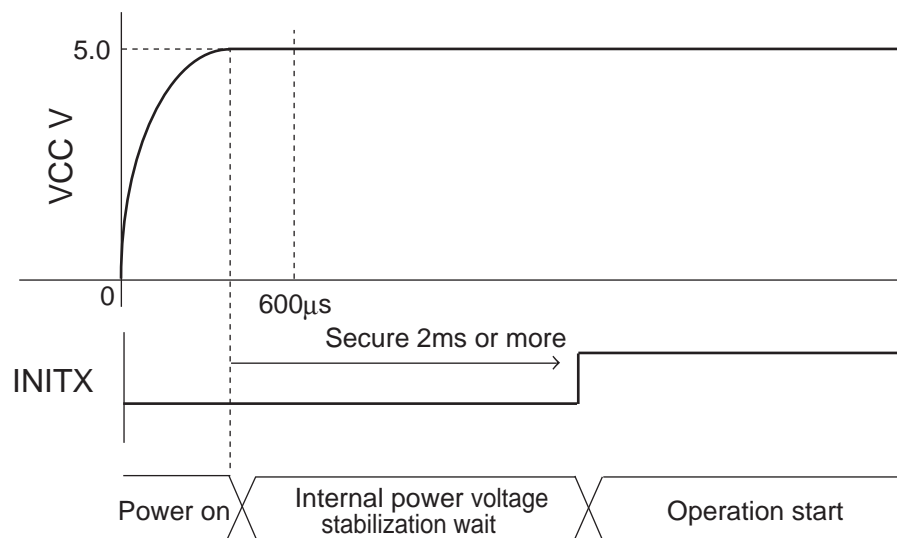
*: The internal power stabilization wait time when the voltage rising time is less than 600 μ s (between 0.0V and 5.0V) is in portion to the capacitance value of the bypass capacitor attached to pin C of this device. This 2ms is the value when pin C=4.7 μ F, and when pin C=9.4 μ F, the internal power stabilization wait time is 4ms.

Figure 2.1-3 Power Supply Rising Standard

- When the voltage rising time is $600\mu\text{s}$ (between 0.0 and 5.0V) or more:



- When the voltage rising time is $600\mu\text{s}$ (between 0.0 and 5.0V) or less:



CHAPTER 3

CPU AND CONTROL UNIT

This chapter provides basic information required to understand the CPU core functions of the MB91470/480 series. It covers architecture, specifications, and instructions.

- 3.1 Memory Space
- 3.2 Memory Map
- 3.3 Internal Architecture
- 3.4 Programming Model
- 3.5 Data Structure
- 3.6 Memory Map
- 3.7 Divergence Instructions
- 3.8 EIT (Exception, Interruption, and Trap)
- 3.9 Operating Mode
- 3.10 Reset (Device Initialization)
- 3.11 Clock Generation Control
- 3.12 Device State Control

3.1 Memory Space

The logical address space of the MB91470/480 series is 4GB (2^{32} addresses) and the CPU performs linear access.

■ Direct Addressing Area

The undermentioned area of the address space is used for I/O.

This area is called the direct addressing area and the operand address can be specified directly in the instruction.

A direct area is different depending on the size of the accessed data as follows.

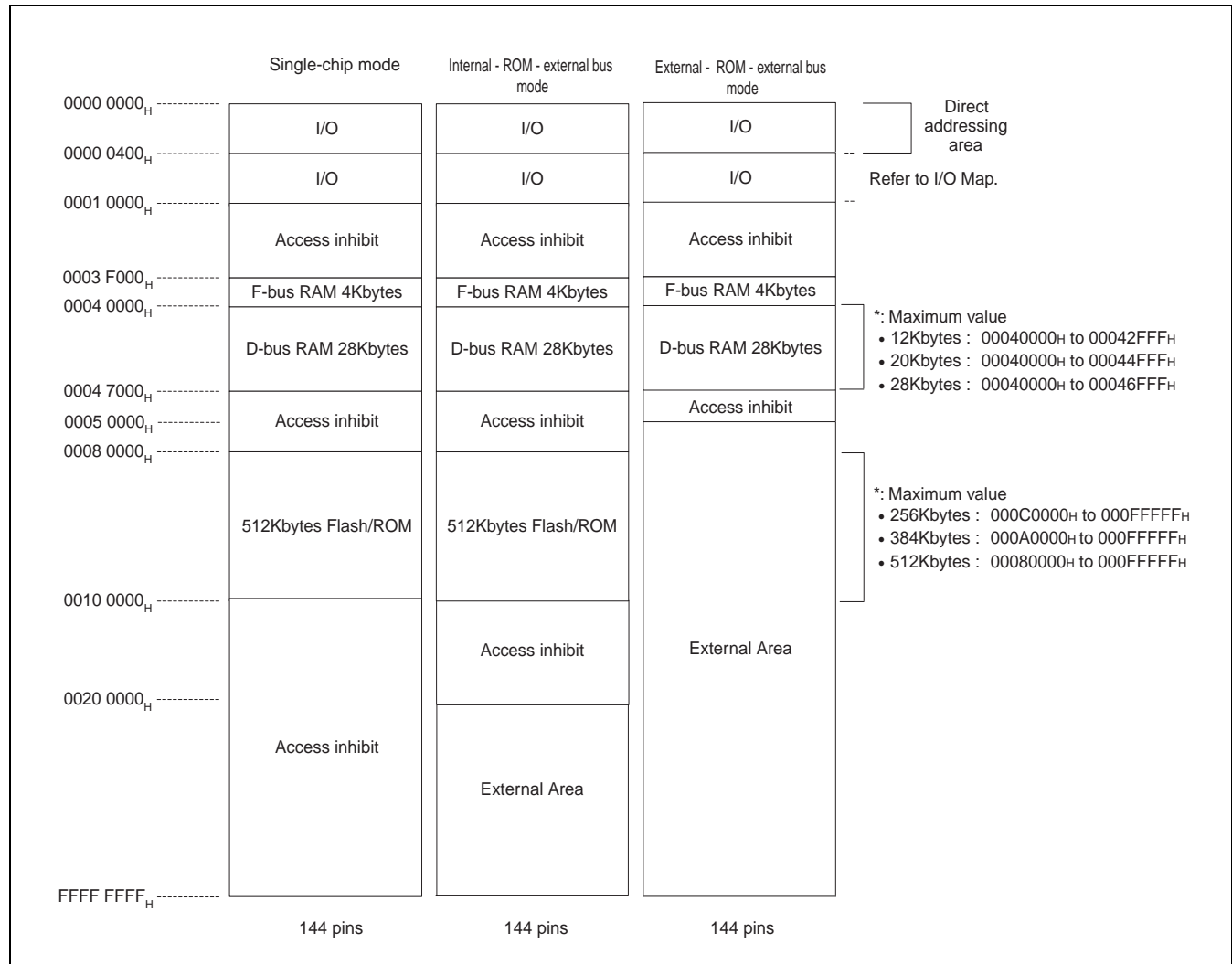
- Byte data access: 000_H to 0FF_H
- Half word data access: 000_H to 1FF_H
- Word data access: 000_H to 3FF_H

MB91470/480 Series

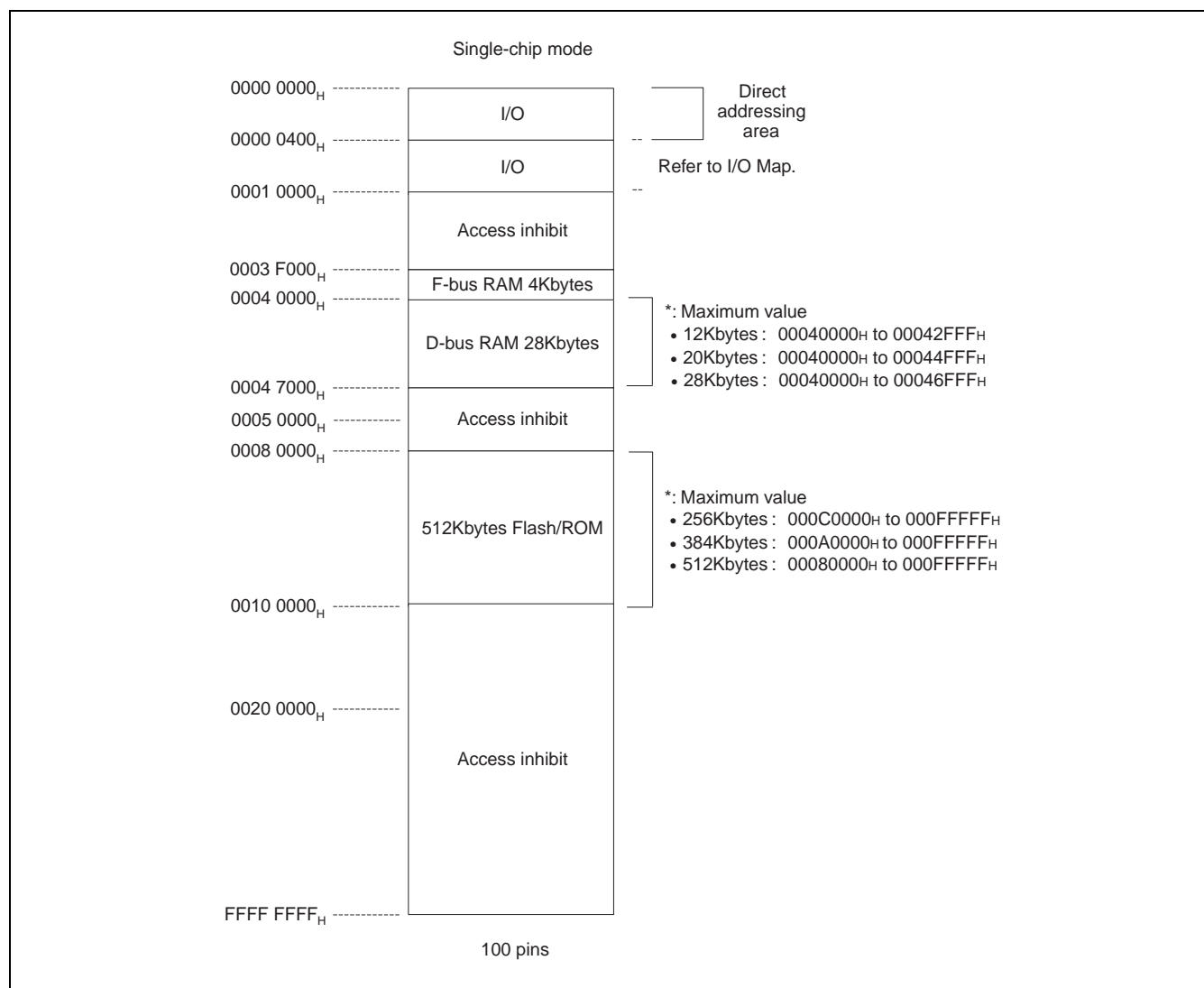
3.2 Memory Map

This section shows the memory map of the MB91470/480 series.

■ Memory Map (MB91470 Series)



■ **Memory Map (MB91480 Series)**



The mode is determined by the mode vector fetch performed after INITX is negated.
(See "■ Operating Mode" in Section "3.9 Operating Mode" for details on setting the mode.)

MB91470/480 Series

3.3 Internal Architecture

As well as adopting RISC architecture, the MB91470/480 series CPU is a high performance core featuring high function commands for embedded applications.

■ Features of Internal Architecture

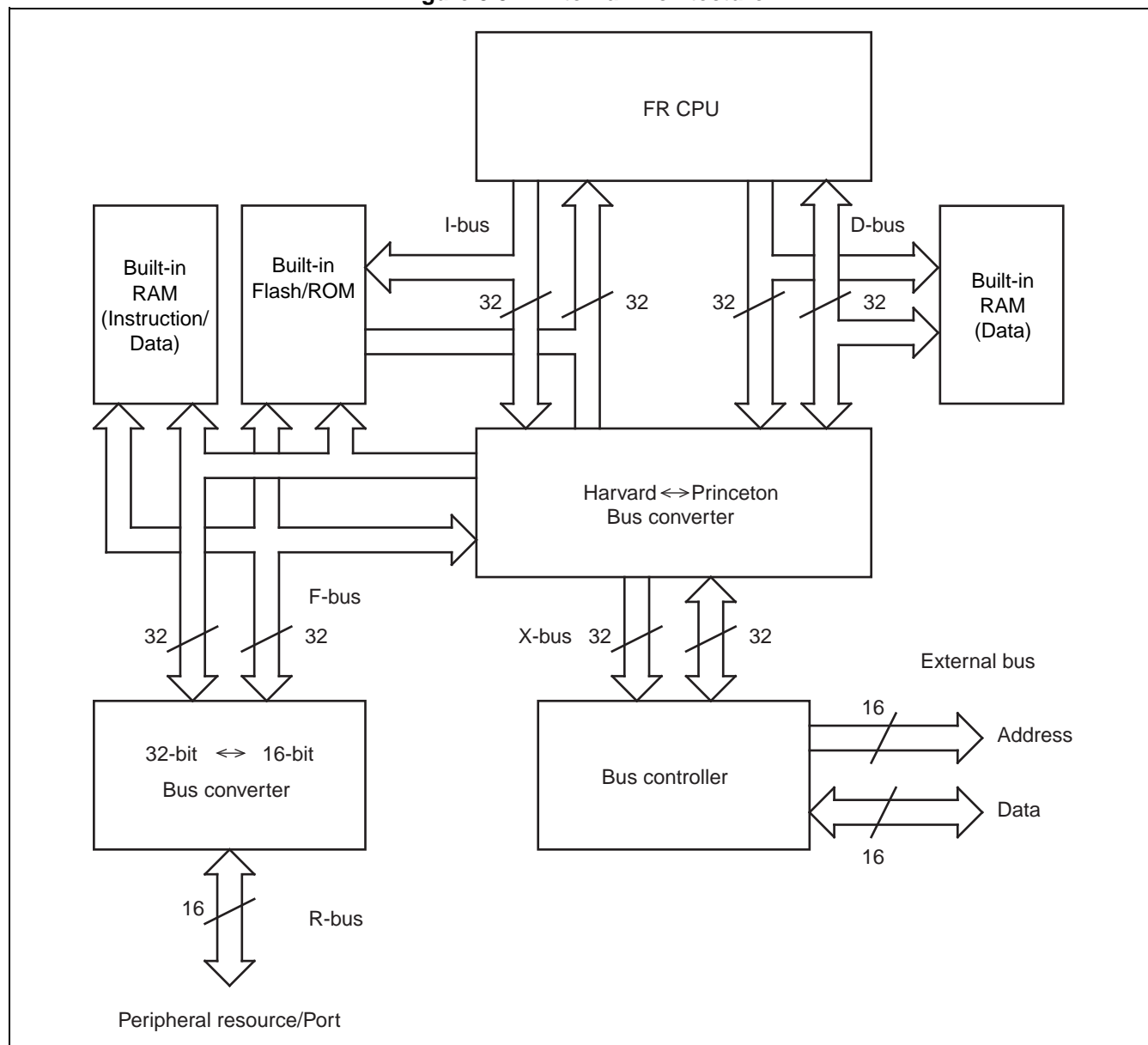
- Adoption of RISC architecture
 - Basic instruction: one instruction one cycle
- 32-bit architecture
 - General-purpose registers: 32 bits × 16 registers
- Linear memory space of 4GB
- Installing of multipliers
 - 32-bit by 32-bit multiplication: 5 cycles
 - 16-bit by 16-bit multiplication: 3 cycles
- Reinforcement of interruption processing function
 - High-speed response speed (6 cycles)
 - Support of multiple interruption
 - Level mask function (16 levels)
- Reinforcement of instruction for I/O operation
 - Memory-to-memory transfer instruction
 - Bit manipulation instruction
- High code efficiency
 - Basic instruction word length: 16 bits
- Low-power consumption
 - Sleep mode, stop mode
- Clock division ratio setting function

■ Internal Architecture

The CPU of the FR family uses the Harvard architecture with separate instruction bus and data bus.

A 32-bit ↔ 16-bit bus converter is connected to the 32-bit bus (F-bus) to provide an interface between the CPU and peripheral resources. A Harvard ↔ Princeton bus converter is connected to the I-bus and D-bus to provide an interface between the CPU and the bus controller.

Figure 3.3-1 Internal Architecture



MB91470/480 Series

■ CPU

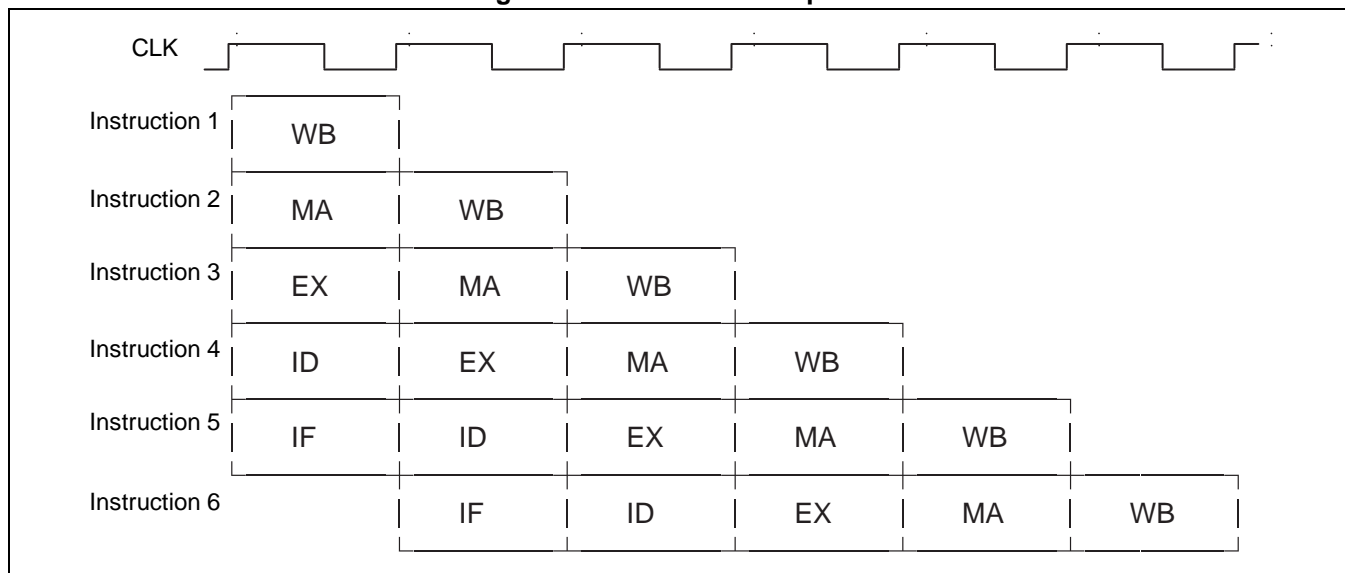
CPU is compactly implementation for the FR family architecture for 32-bit RISC.

A five-stage instruction pipeline is used to enable execution of one instruction per cycle.

The pipeline is composed of the following stages.

- Instruction fetch (IF): The instruction address is output, and the instruction is fetched.
- Instruction decipherment (ID): The decipherment does the fetched instruction. The register is read.
- Execution (EX): The operation is executed.
- Memory access (MA): Loading into or storing the memory is accessed.
- Write-back (WB): Writes the result of operation (or loaded memory data) to a register.

Figure 3.3-2 Instruction Pipeline



The instruction is never in any order executed. Accordingly, if instruction A enters the pipeline before instruction B, instruction A always reaches write-back stage before instruction B.

As a rule, the instruction is executed at the speed of one instruction per cycle. A number of cycles are required to execute commands for the load store commands accompanying memory wait, branch commands that do not have delay slots and multiple cycle commands. Also, instruction execution speed drops if supply of instructions is delayed.

■ 32-bit ↔ 16-bit Bus Converter

Acts as an interface between the F-bus which uses high-speed 32-bit access and the R-bus which uses 16-bit access to enable the CPU to access data to the internal peripheral circuits.

The bus converter converts 32-bit access from the CPU into two 16-bit accesses and performs R-bus access. Restrictions on the access width apply for some internal peripheral circuits.

■ Harvard ↔ Princeton Bus Converter

Arbitrates instruction access and data access from the CPU to provide a smooth interface to the external buses.

In CPU, the instruction bus and the data bus are the independent Harvard architecture structures. On the other hand, the bus controller that controls the external bus has a single-bus Princeton architecture structure. This bus converter ranks the priority order for command access and data access of the CPU and controls access to the bus controller. This mechanism causes the prioritizing of external bus access to be optimized.

■ Overview of Instructions

FR family supports logic operation, bit operation, and direct addressing commands that are optimized for embedded applications, as well as the command system of the normal RISC. The list of the instruction set is shown in "APPENDIX E Instruction List". Excellent memory efficiency is achieved because each instruction is 16-bit long (some instructions are 32 or 48 bits in length).

The instruction set can be divided into the following function groups.

- Arithmetic operation
- Loading and store
- Divergence
- Logical operation and bit operation
- Direct addressing
- The others

● Arithmetic operation

It has standard arithmetic operation commands (addition, subtraction, and comparison) and shift commands (logic shift and arithmetic operation shift). Operations with carry that are used for multi-word length operations and operations that do not change the flag which are convenient for address calculations are enabled for addition and subtraction.

32-bit \times 32-bit and 16-bit \times 16-bit multiplication instructions and a 32-bit/32-bit step division instruction are also provided.

Immediate value transfer instructions for setting immediate values to registers and register-to-register transfer instructions are also provided.

All arithmetic operation instructions use the multiplication and division register and the general-purpose registers in the CPU to perform the operation.

● Loading and store

Load and store instructions read or write data in external memory. They are also used to read or write to the peripheral resources (I/O) on the chip.

Loading and store have three kinds of access lengths of the byte, the half-word, and the word. In addition to general register indirect memory addressing, the register indirect memory addressing with displacement and register indirect memory addressing with register increment/decrement are possible for certain commands.

● Divergence

It is an instruction of the divergence, the call, the interruption, and the return. There are two types of branch commands; one type features a delay slot while the other does not. They can be optimized in accordance with the purpose. See Section "3.7 Divergence Instructions" for details of the divergence instruction.

● Logical operation and bit operation

Logical operation commands can perform AND, OR, and EOR logical operations between general-purpose registers or between a general-purpose register and the memory (and I/O). Moreover, the bit operation instruction can operate the content of the memory (and I/O) directly. The register of the memory addressing is generally indirect.

- Direct addressing

Direct addressing commands are used to access between I/O and general-purpose registers or between I/O and the memory. The I/O address can be accessed quickly and efficiently by direct specification within the command instead of indirectly to the register. Indirect memory addressing to the register with register increment/decrement is also enabled for some commands.

- The others

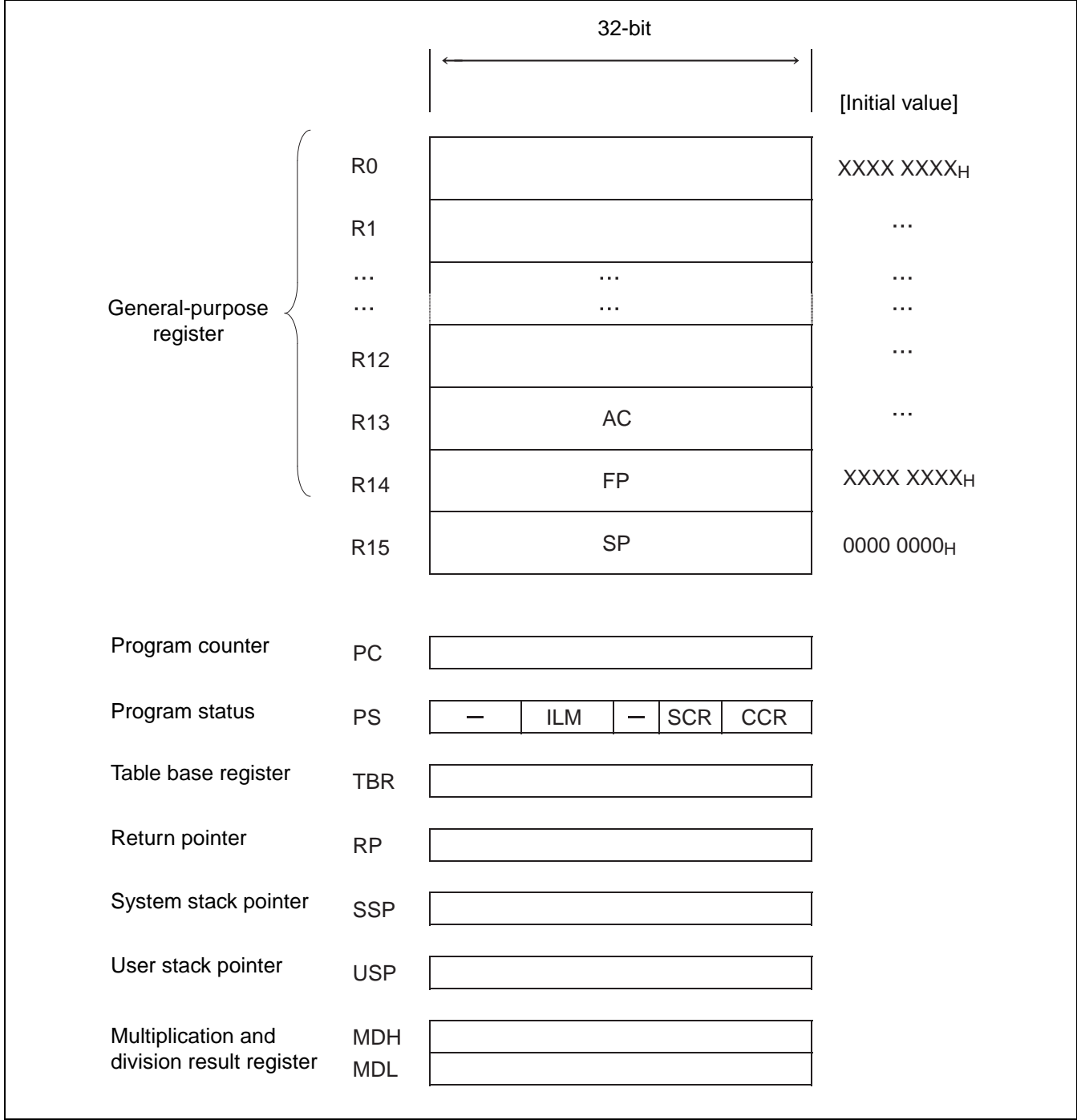
Other instructions include instructions for setting the flags in the PS register, performing stack operations, and performing sign and zero-extended operations. Function entry and exit instructions for use with high-level languages and register multi-load/store instructions are also provided.

3.4 Programming Model

This section explains the basic programming model and each register.

■ Basic Programming Model

Figure 3.4-1 Basic Programming Model



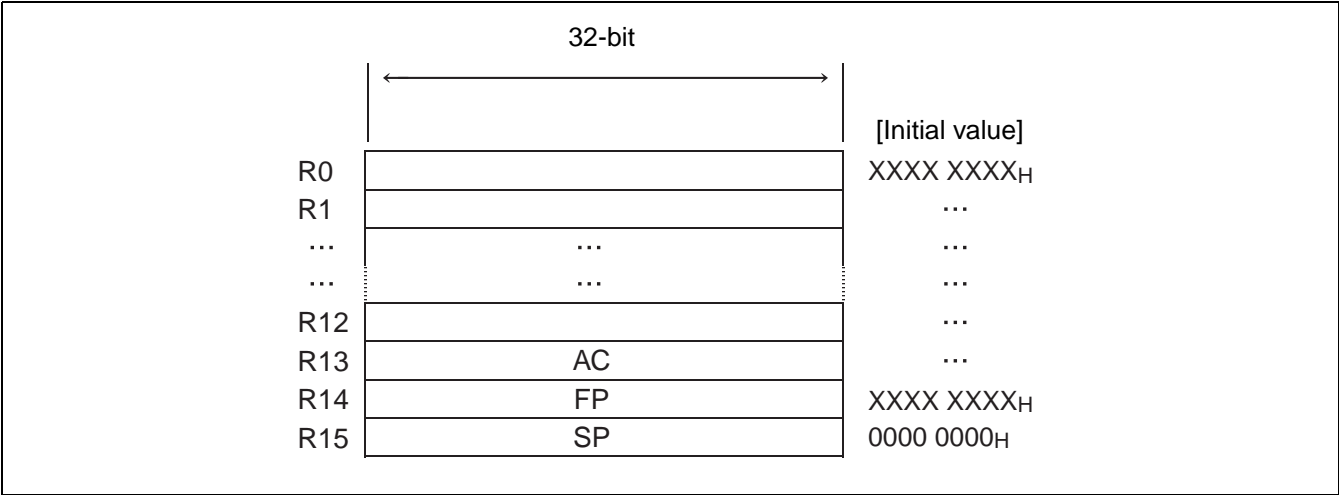
MB91470/480 Series

3.4.1 Registers

This section explains each register.

■ General-purpose Register

Figure 3.4-2 General-purpose Register



Registers R0 to R15 are a general-purpose register. They are used as accumulator for various types of operation and as pointer for memory access.

The following of the 16 registers are expected to have special uses, so some commands are emphasized.

R13: Virtual accumulator

R14: Frame pointer

R15: Stack pointer

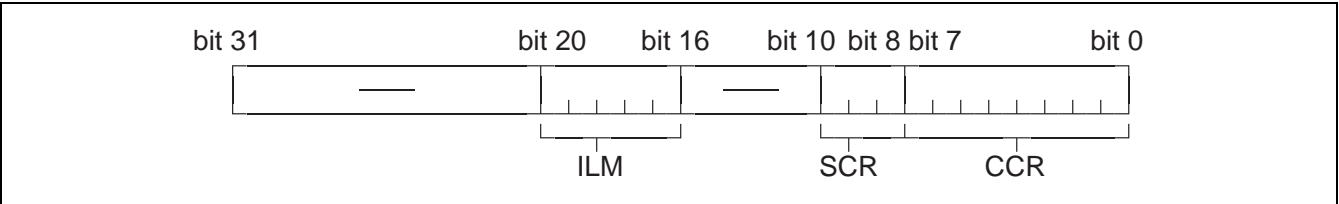
R0 to R14 of the initial value by reset are undefined. R15 is 00000000_H (SSP value).

■ Program Status (PS)

This register retains the program status and is separated into three parts, namely, ILM, SCR, and CCR.

All bits undefined in figure are reserved bit. Reading always returns "0".

Writing has no effect.



■ Condition Code Register (CCR)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
-	-	S	I	N	Z	V	C	--00XXXX _B

[bit5] S: Stack flag

The stack pointer used as R15 is specified.

Value	Description
0	SSP is used as R15. Automatically goes to "0" when an EIT occurs. (However, the value saved on the stack is the value before the bit is cleared.)
1	USP is used as R15.

Cleared to "0" by a reset.

Set to "0" when executing the RETI instruction.

[bit4] I: Interrupt enable flag

Permission and interdiction of the user interruption demand are controlled.

Value	Description
0	User interruption interdiction. Cleared to "0" when the INT instruction is executed. (However, the value saved on the stack is the value before the bit is cleared.)
1	User interruption permission. Masking of user interrupt requests is controlled by the value stored in the ILM.

Cleared to "0" by a reset.

[bit3] N: Negative flag

Indicates the sign when an operation result is represented as a two's-complement integer.

Value	Description
0	It is indicated that operation result was a positive value.
1	It is indicated that operation result was a negative value.

Initial state by reset is irregular.

[bit2] Z: Zero flag

It is shown operation result was 0.

Value	Description
0	It is indicated that operation result was the values other than 0.
1	It is indicated that operation result was 0.

Initial state by reset is undefined.

[bit1] V: Overflow flag

Operands used for calculations are defined as integers expressed in complements of 2, and whether or not an overflow occurs as a result of the calculation is indicated.

Value	Description
0	It is indicated not to have caused the overflow as a result of the operation.
1	It is indicated to have caused the overflow as a result of the operation.

Initial state by reset is undefined.

[bit0] C: Carrying flag

Indicates whether an operation resulted in a borrow or a carry from the most significant bit.

Value	Description
0	It is indicated that neither a carry nor a borrow occurred.
1	It is indicated that a carry or a borrow occurred.

Initial state by reset is undefined.

■ System Condition Code Register (SCR)

	bit10	bit9	bit8	Initial value
	D1	D0	T	XX0 _B

[bit10, bit9] D1, D0: Flag for step division

The middle data of step division execution time is maintained.

Do not change while executing the division processing.

During step division when other processing is performed, re-start of the step division is guaranteed by saving and returning to the PS register value.

Initial state by reset is irregular.

Set based on the value of the dividend and divisor during the execution of the DIV0S instruction.

Forcibly cleared by execution of the DIV0U instruction.

[bit8] T: Step trace trap flag

It is a flag which specifies whether to make the step trace trap effective.

Value	Description
0	Step trace trap invalidity.
1	Step trace trap effective. In this case, all user NMIs and user interrupts are disabled.

Initialized to "0" by a reset.

The emulator uses the function of the step trace trap. When the emulator is used, the use cannot be done in user program.

■ **ILM**

bit20	bit19	bit18	bit17	bit16	Initial value
ILM4	ILM3	ILM2	ILM1	ILM0	01111 _B

This register stores the interrupt level mask value and the value set in the ILM is used as the level mask.

The interrupt request to be input to the CPU is accepted only when its interrupt level is stronger than the level indicated by this ILM.

As for the level value, 0 ("00000_B") is the strongest, and 31 ("11111_B") is weakest.

There is a limitation in the value which can be set from the program.

When former value is 16 to 31

New values can only be set in the range 16 to 31. Executing an instruction that sets a value between 0 and 15 results in (specified value + 16) being transferred.

When former value is 0 to 15

Any value of 0 to 31 can be set.

Initialized to 15 ("01111_B") by reset.

[Notes of PS register]

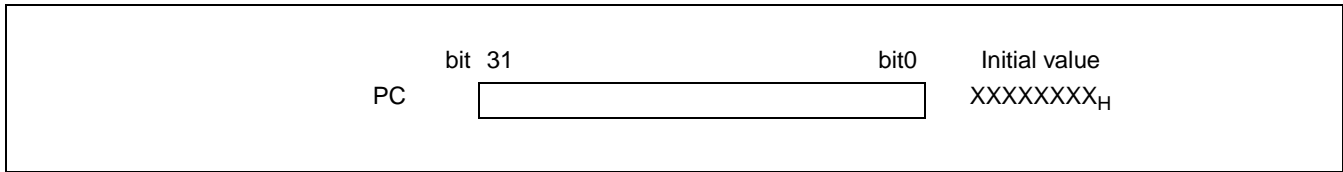
As some instructions pre-process the PS register, the following exception operations may cause a break to occur in an interrupt processing routine when using the debugger or the updating of the PS flag.

In either case, the system is designed to re-execute correctly after the EIT returns and therefore processing before and after the EIT is executed correctly.

1. The following operations may occur when (a) user interrupt/NMI is received, (b) step execution is performed, (c) break occurs in a data event or emulator menu in an instruction immediately preceding DIV0U/DIV0S instruction.
 - (1) D0 and D1 flags precede and are renewed.
 - (2) EIT processing routine (user interruption, NMI, or emulator) is executed.
 - (3) After returning from EIT, DIV0U/DIV0S instructions are executed and the D0 and D1 flags are updated to the same value as (1).
2. When each ORCCR/STILM/MOV Ri and PS instructions are executed to permit interrupting with the user interruption and the NMI factor generated, the following operations are done.
 - (1) The PS register precedes and is updated.
 - (2) Execute an EIT processing routine (user interrupt or NMI).
 - (3) After returning from EIT, the above instructions are executed and the PS register is updated to the same value as (1).

MB91470/480 Series

■ Program Counter (PC)



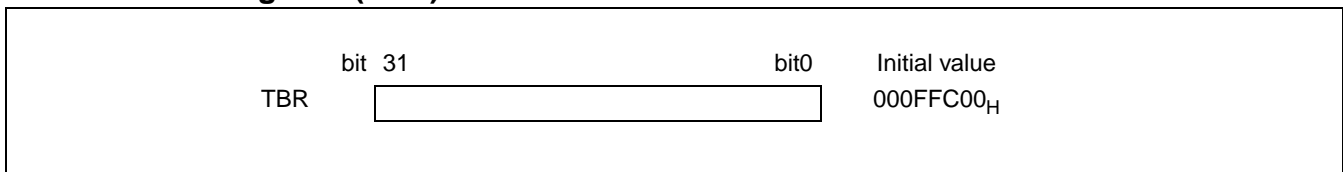
The address of the executed instruction is shown with the program counter.

Bit0 is set to "0" when the PC is updated during instruction execution. Bit0 can only go to "1" in the case when an odd-numbered address is specified as a branch destination address.

However, bit0 is ignored in this case also and instructions must be located at addresses that are a multiple of two.

The initial value by reset is irregular.

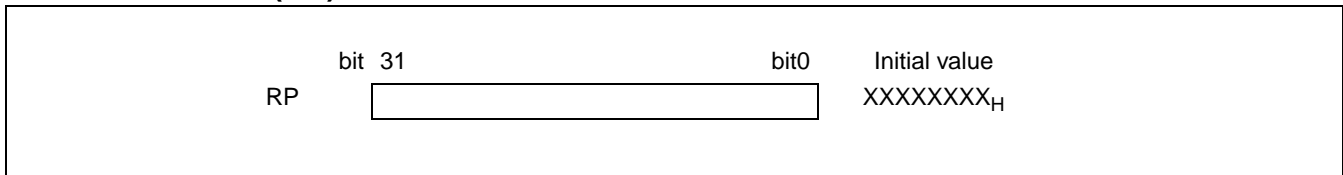
■ Table Base Register (TBR)



The table base register stores the start address of the vector table used in EIT processing.

The initial value by reset is "000FFC00_H".

■ Return Pointer (RP)



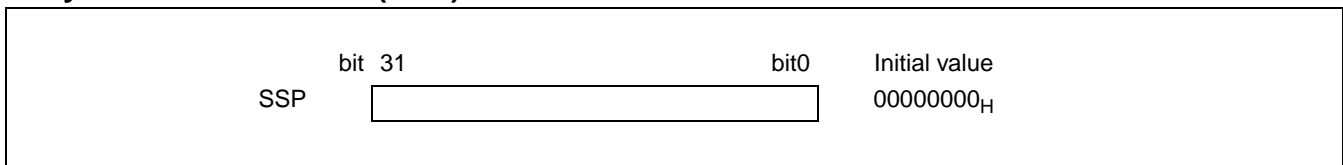
The address which returns from the sub routine is maintained with the return pointer.

The value of PC is forwarded to this RP at CALL instruction execution time.

The content of RP is forwarded to PC at RET instruction execution time.

The initial value by reset is irregular.

■ System Stack Pointer (SSP)



The SSP is the system stack pointer.

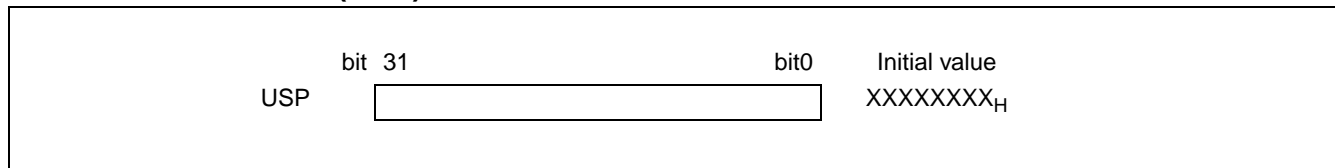
Functions as R15 when the S flag is "0".

The SSP can also be specified explicitly.

Also used as the stack pointer specifying the stack that saves the PS and PC when EIT occurs.

The initial value by reset is "00000000_H".

■ User Stack Pointer (USP)



The USP is the user stack pointer.

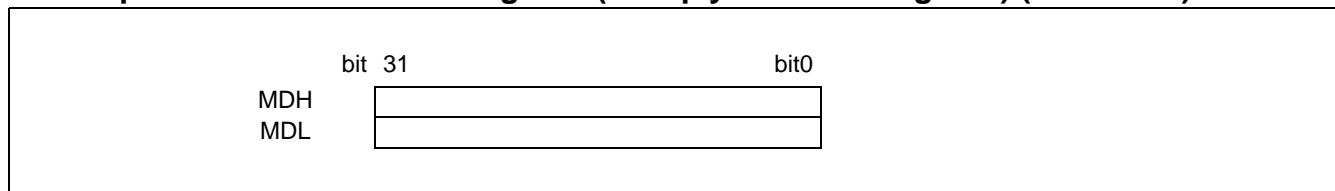
Functions as R15 when the S flag is "1".

The USP can also be specified explicitly.

The initial value by reset is irregular.

The use cannot be done in the RETI instruction.

■ Multiplication and Division Register (Multiply & Divide Register) (MDH/MDL)



This register is the register for multiplication and division, and 32-bit lengths respectively.

The initial value by reset is irregular.

Multiplication execution time

When performing 32-bit × 32-bit multiplication, 64-bit length calculation results are stored in the multiplication/division results storage register in the following format.

MDH: Higher 32 bits

MDL: Lower 32 bits

When 16 bits × 16 bits are multiplied, the result is stored as follows.

MDH: Undefined

MDL: Result of 32 bits

Division execution time

When beginning to calculate, the dividend is stored in MDL.

When division is performed using the DIV0S/DIV0U, DIV1, DIV2, DIV3, and DIV4S commands, the results are stored in MDL and MDH.

MDH: Surplus

MDL: Quotient

MB91470/480 Series

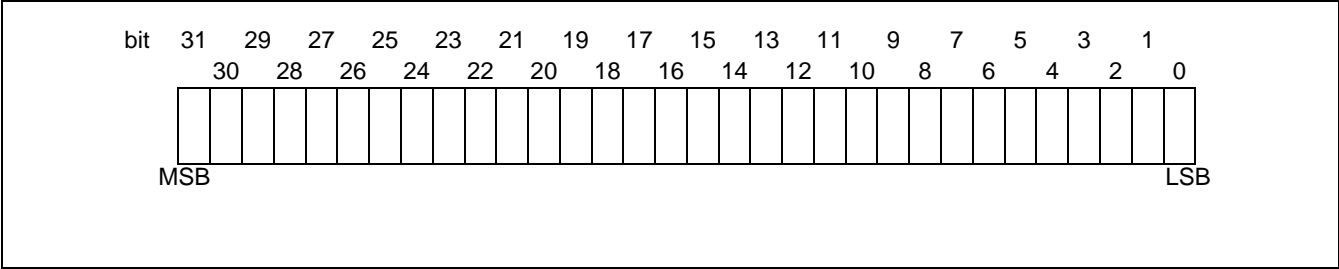
3.5 Data Structure

This section explains the bit ordering, byte ordering, and word alignment.

■ Bit Ordering

In the FR family, the little endian has been adopted as a bit ordering.

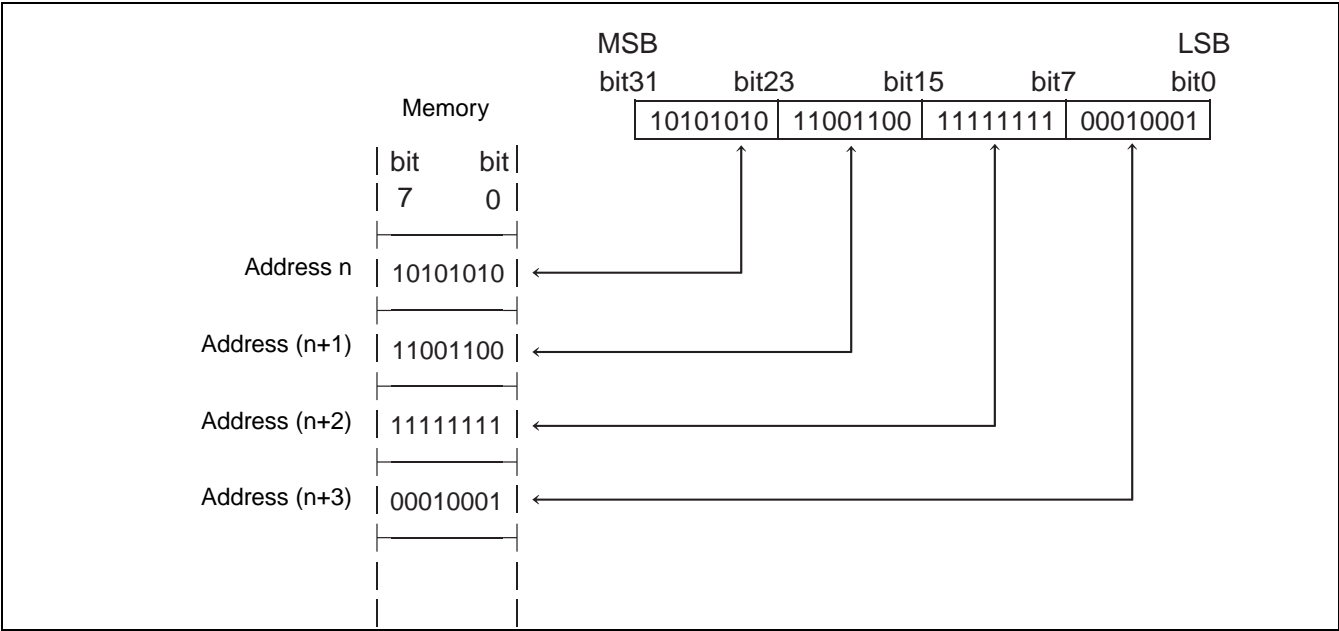
Figure 3.5-1 Bit Ordering



■ Byte Ordering

In the FR family, the big endian has been adopted as byte ordering.

Figure 3.5-2 Byte Ordering



■ Word Alignment

● Program access

It is necessary to arrange the program of the FR family in the address of the multiple of two.

Bit0 of PC is set to "0" when the PC is updated during instruction execution. The bit can only go to "1" in the case when an odd-numbered address is specified as a branch destination address.

However, bit0 is ignored in this case also and instructions must be located at addresses that are a multiple of two.

There is no odd-number address exception.

● Data access

In the FR family, when data access is performed, forced alignment is provided to addresses as follows in accordance with their width.

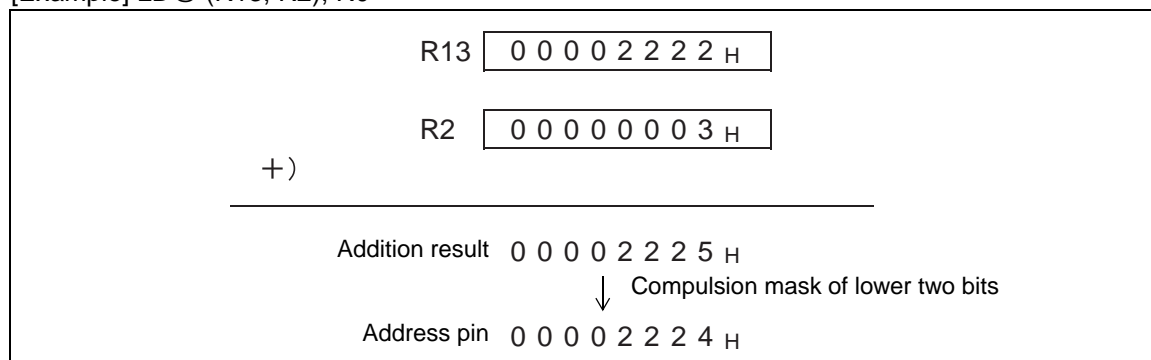
Word access : The address is a multiple of four (The lowest 2 bits are forcibly set to "00_B").

Half-word access : The address is a multiple of two (The lowest bit is forcibly set to "0").

Byte access : ----

When word or half-word data is accessed, 0 is forcibly set to some bits, which are the calculation results of the effective address. For example, in the @(R13, Ri) addressing mode, the pre-addition register is used for calculations as it is (even though the lowest bit is 1), and the lower bits of the addition results will be masked. The register prior to the calculation is not masked.

[Example] LD@ (R13, R2), R0



MB91470/480 Series

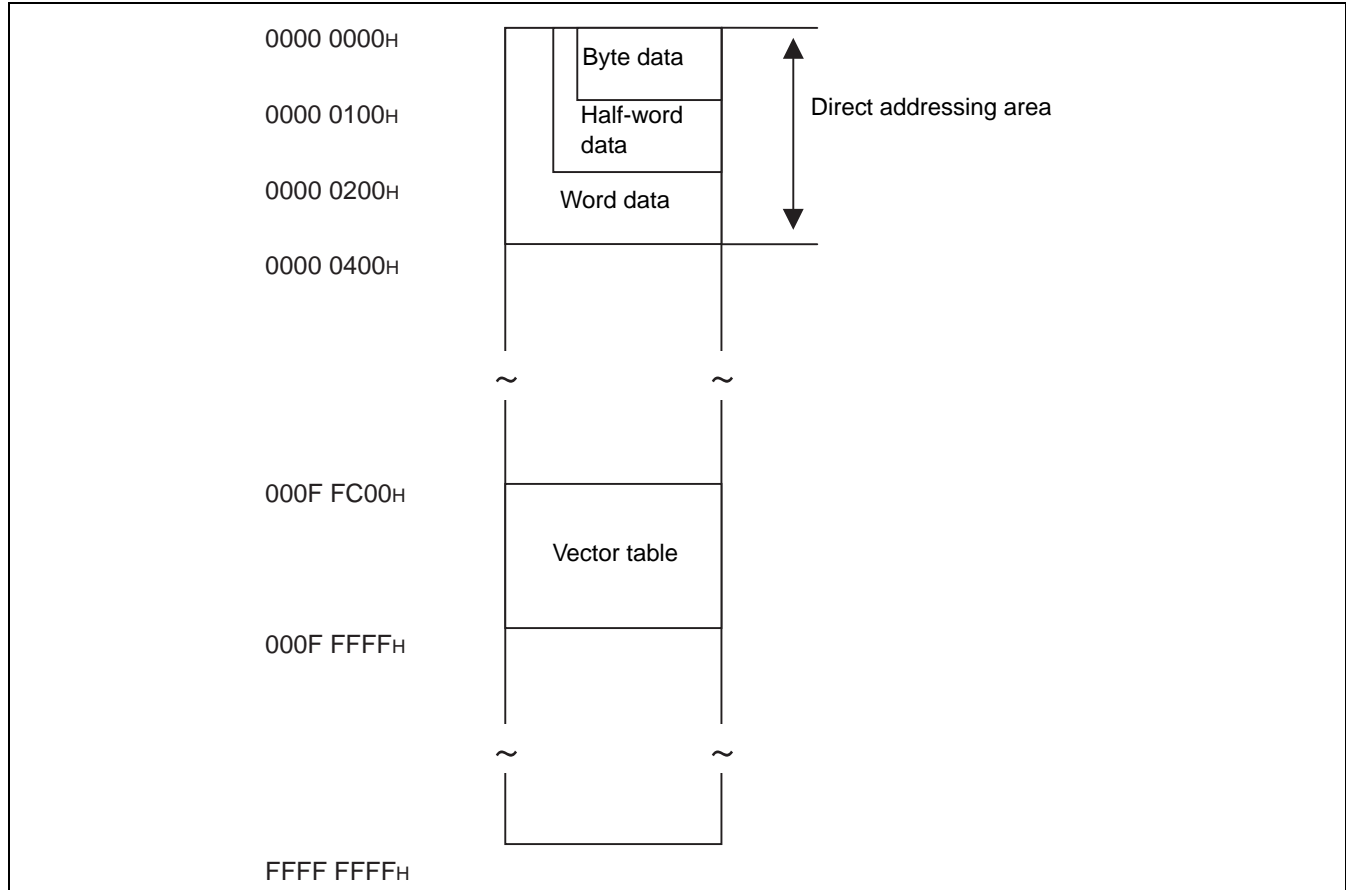
3.6 Memory Map

This section describes a memory map of the MB91470/480 series.

■ Memory Map

The address space is 32-bit linear.

Figure 3.6-1 Memory Map



■ Direct Addressing Area

The undermentioned area of the address space is an area for I/O. This area can directly specify the operand address in the instruction using the direct addressing.

The size of the address area of direct possible addressing is different in each data length.

- Byte data (8-bit) : 000_H to 0FF_H
- Half-word data (16-bit) : 000_H to 1FF_H
- Word data (32-bit) : 000_H to 3FF_H

■ Vector Table Initial Area

The area of 000FFC00_H to 000FFFFF_H is EIT vector table initial area.

The vector table used for EIT processing can be allocated to an arbitrary address by rewriting the TBR, but it is allocated to this address on initialization through reset.

3.7 Divergence Instructions

In the FR family, whether the operations are with or without delay slots can be specified for the branch command.

■ Operation with Delay Slot

● Instruction

The instructions with the notation shown below perform a branch operation with a delay slot.

JMP:D	@Ri	CALL:D	label12	CALL:D	@Ri	RET:D
BRA:D	label9	BNO:D	label9	BEQ:D	label9	BNE:D label9
BC:D	label9	BNC:D	label9	BN:D	label9	BP:D label9
BV:D	label9	BNV:D	label9	BLT:D	label9	BGE:D label9
BLE:D	label9	BGT:D	label9	BLS:D	label9	BHI:D label9

● Operation explanation

Operations with delay slots branch out after executing the command placed just after the branch command (called a "delay slot") before executing the branch destination command. As the instruction in the delay slot is executed prior to the branch, the apparent execution speed is one cycle. The NOP command must be placed as an alternative if an effective command cannot be inserted in the delay slot.

[Example]

```

;      Row of instruction
      ADD    R1, R2    ;
      BRA:D  LABEL    ; Divergence instructions
      MOV    R2, R3    ; Delay slot ... Executed before branch.
      ...
LABEL:ST      R3,@R4    ; Branch destination

```

The command placed in the delay slot is executed regardless of whether the branch condition for the condition branch command will be met or not.

For delay branch commands, the execution order of the partial command seems to be reversed, but this applies only to PC update operations, and other operations (i.e. update and refer to register) are absolutely executed in the described order.

A concrete explanation is done as follows.

- The Ri to be referred to for the JMP:D@Ri/CALL:D@Ri command will not be affected even if the command within the delay slot updates the Ri.

[Example]

```

LDI:32  #Label,  R0
JMP:D    @R0      ; Branches out to Label
LDI:8    #0,      R0    ; Has no effect on branch destination address
      ...

```

- The RP to be referred by the RET:D command will not be affected even if the command within the delay slot updates the RP.

[Example]

```
RET:D                ; Branches to the address in RP set previously.
MOV    R8,    RP    ; Has no effect on return operation.
...
```

- Flags to be referred by the Bcc:D rel instruction are also not affected by the delay slot instruction.

[Example]

```
ADD    #1,    R0    ; Flag change
BC:D    Overflow    ; Branches based on execution result of above instruction.
AND CCR #0          ; Do not refer to this flag update in the above-mentioned branch
                        instruction.
...
```

- If the instruction in the delay slot for the CALL:D instruction refers RP, it reads the value after updating by CALL:D instruction.

[Example]

```
CALL:D Label        ; Updating RP and branching
MOV    RP,    R0    ; RP of an execution result in the above-mentioned
                        CALL:D is forwarded.
...
```

● Restrictions

- Instruction that can be placed in the delay slot

Only instructions that satisfy the following conditions can be executed in the delay slot.

- 1-cycle command
- No branch instruction
- Instruction whose operation is not affected even though the order is changed

The "1-cycle command" is a command with "1", "a", "b", "c", or "d" described in the cycle number column within the command list.

- Step trace trap

Step trace trap is not generated between executing the branch command with the delay slot and the delay slot.

- Interrupts and NMI

Interrupts and NMI cannot be received between execution of a branch instruction with a delay slot and the delay slot.

- Undefined instruction exception

No undefined instruction exception occurs if the delay slot contains an undefined instruction. At this time, undefined instruction operates as NOP instruction.

■ Operation without Delay Slot

● Instruction

Instructions written as follows perform a branch operation without a delay slot:

JMP @Ri	CALL label12	CALL @Ri	RET
BRA label9	BNO label9	BEQ label9	BNE label9
BC label9	BNC label9	BN label9	BP label9
BV label9	BNV label9	BLT label9	BGE label9
BLE label9	BGT label9	BLS label9	BHI label9

● Operation explanation

When the delay slot is not used, instructions are executed in the sequence they are coded. The next instruction will not be executed prior to the branch.

[Example]

```

;      Row of instruction
      ADD    R1, R2    ;
      BRA    LABEL    ; Branch instruction (delay slot none)
      MOV    R2, R3    ; Not executed
      ...
LABEL: ST     R3, @R4 ; Divergence destination

```

Execution cycle number for branch commands without delay slots will be 2 cycles branched, or 1 cycle non-branched.

As the appropriate command cannot be inserted into the delay slot, the command code efficiency can be improved more than the branch command with delay slot described the NOP.

A balance between execution speed and code efficiency can be struck by selecting either the operation with the delay slot when effective commands can be set in the delay slot or the operation without the delay slot when effective commands cannot be set.

3.8 EIT (Exception, Interruption, and Trap)

EIT, which is the generic term for "Exception", "Interrupt", and "Trap" indicates that the program is suspended due to events generated while running the current program and another program is being executed.

The exception is an event which occurs in relation to the context under execution.

Execution continues from the instruction that caused the exception.

The interruption is an event which occurs without any relation to the context under execution. The event factor is hardware.

The trap is an event which occurs in relation to the context under execution. Some traps, such as system calls, are specified in program. Execution continues from the instruction after the instruction that caused the trap.

■ Feature of EIT

- Multiple interrupt is supported to the interruption.
- It is a level mask function (15 levels are available to the user) to the interruption.
- Trap instruction (INT)
- EIT (hardware/software) for emulator startup

■ EIT Factor

The following is used as an EIT factor.

- Reset
- User interruption (internal resource and external interruption)
- NMI
- Delayed interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap
- Coprocessor absent trap
- Coprocessor error trap

■ Return from EIT

To return from EIT, RETI instruction is executed.

■ Interrupt Level

Interrupt levels are 0 to 31 and are managed by five bits.

The allocation of each level is as follows.

Table 3.8-1 Interrupt Level

Level		Interrupt Factor	Note
Binary	Decimal		
00000	0	(System reservation)	If the original value of ILM is between 16 and 31, the value of this range cannot be set in the ILM with program.
...	
...	
00011	3	(System reservation)	
00100	4	INTE instruction	
		Step trace trap	
00101	5	(System reservation)	
...	
...	
01110	14	(System reservation)	
01111	15	NMI (for user)	
10000	16	Interrupt	When ILM is set, it is a user interruption interdiction.
10001	17	Interrupt	
...	
...	
11110	30	Interrupt	
11111	31	-	When ICR is set, it is an interruption interdiction.

It is a level of 16 to 31 that the operation is possible.

Undefined command exceptions, coprocessor absent traps, coprocessor error traps, and INT commands are not affected for interruption levels. Moreover, ILM may not be changed.

■ I Flag

It is a flag which specifies the permission and interdiction of the interruption. Contained in bit4 of CCR in the PS register.

Value	Description
0	Interruption interdiction. Cleared to "0" when the INT instruction is executed. (However, the value saved on the stack is the value before the bit is cleared.)
1	Interruption permission. The mask processing of the interruption demand is controlled by the value which ILM maintains.

■ ILM

It is PS register (bit20 to bit16) which maintains the interrupt level mask value.

The interrupt request to be input to the CPU is accepted only when its interrupt level is stronger than the level indicated by this ILM.

The highest level is 0 ("00000_B") and the lowest level is 31 ("11111_B").

There is a limitation in the value which can be set from the program. When the original value is between 16 and 31, new values can only be set in the range 16 to 31. Executing an instruction that sets a value between 0 and 15 results in (specified value + 16) being transferred.

When former value is 0 to 15, any value of 0 to 31 can be set. Use the STILM instruction to set the ILM.

■ Level Mask to Interruption and NMI

When an NMI or interrupt request occurs, the interrupt level (see Table 3.8-1) corresponding to the interrupt factor is compared with the level mask set in the ILM. And, when the following condition consists, the mask is done, and the demand is not accepted.

Interrupt levels of factor \geq level mask value

■ Interrupt Control Register (ICR)

This register is located in the interrupt controller and specifies the level for each interrupt request. ICR is prepared for each of the interruption demand input. ICR is mapped on the I/O space and is accessed by CPU through the bus.

● The ICR bit make-up

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
-	-	-	R	R/W	R/W	R/W	R/W	

[bit4] ICR4

This bit is always "1".

[bit3 to bit0] ICR3 to ICR0

Lower four bits of the interrupt level for the corresponding interrupt factor. The read and write are possible.

ICR can set the value within the range of 16 to 31 together with bit4.

● ICR mapping

Table 3.8-2 Interruption Factor, Interruption Control Register, and Interruption Vector

Interrupt Factor	Interruption Control Register		Corresponding Interruption Vector		
			Number		Address
			Hexadecimal	Decimal	
IRQ00	ICR00	00000440 _H	10 _H	16	TBR + 3BC _H
IRQ01	ICR01	00000441 _H	11 _H	17	TBR + 3B8 _H
IRQ02	ICR02	00000442 _H	12 _H	18	TBR + 3B4 _H
...
...
IRQ45	ICR45	0000046D _H	3D _H	61	TBR + 308 _H
IRQ46	ICR46	0000046E _H	3E _H	62	TBR + 304 _H
IRQ47	ICR47	0000046F _H	3F _H	63	TBR + 300 _H

TBR initial value: 000F FC00_H

Reference: Refer to "CHAPTER 6 INTERRUPT CONTROLLER".

■ **System Stack Pointer (SSP)**

<div> <div> <div>bit 31</div> <div>bit0</div> </div> <div>Initial value</div> </div> <div> <div>SSP</div> <div>00000000_H</div> </div>		
--	--	--

The SSP is used as the pointer to the stack used to save and restore data when an EIT is accepted or a return operation occurs.

8 is deducted from the content during EIT processing, and 8 is added when returning from EIT in line with execution of the RETI command.

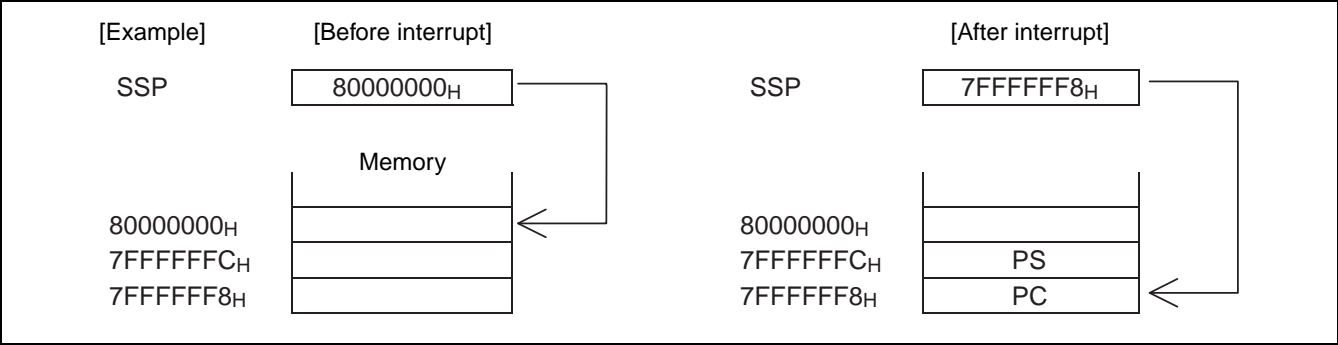
The initial value by reset is "00000000_H".

The SSP can also be used as general-purpose register R15 when the S flag in the CCR is "0".

■ Interrupt Stack

The value in the PC or PS is saved to or restored from an area pointed to by the system stack pointer (SSP). After an interrupt, the PC is stored at the address contained in the SSP and PS is stored at the (SSP + 4) address.

Figure 3.8-1 Interrupt Stack



■ Table Base Register (TBR)



It is a register which shows the first address of the vector table for EIT.
The vector address is generated by adding the TBR and the offset value determined for each EIT factor.
The initial value by reset is "000FFC00_H".

■ EIT Vector Table

The vector region for EIT is 1 KB region starting at address indicated by the TBR.
Each vector consists of four bytes and the relationship between the vector number and vector address is as follows.

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

vctadr: Vector address
vctofs: Vector offset
vct: Vector number

The lower two bits of the addition result are always treated as "00_B".
The region of 000FFC00_H to 000FFFFFF_H is an initial region of the vector table by reset.
A special function is partially allocated to the vector.

■ Multiple EIT Processing

When a number of EIT factors are simultaneously generated, the CPU selects and accepts one EIT factor, and after executing the EIT sequence, the detection of EIT factors is repeated.

When EIT factors are detected if there are no more EIT factors that can be accepted, the handler command for the last EIT factor accepted will be executed.

Accordingly, if more than one EIT factor occurs at the same time, the sequence for executing the handler for each EIT is determined by the following two elements:

- (1) Priority level of EIT factor acceptance
- (2) How other factors can be masked when one factor is accepted

■ Priority of EIT Factor

The priority for accepting EIT factors is the order for selecting factors executing EIT sequence that saves the PS and PC, updates the PC (on demand), and executes mask processing for other factors.

The handler of the factor previously accepted is not previously executed necessarily.

The priority of the EIT factor acceptance is shown in Table 3.8-3.

Table 3.8-3 Priority of EIT Factor Acceptance and Mask to Other Factors

Priority of acceptance	Factor	Mask to other factors
1	Reset	Other factors are annulled.
2	Undefined instruction exception	Cancellation
3	INTE instruction	ILM = 4 Other factors are annulled.
4	INT instruction	I flag = 0
5	Coprocessor absent trap Coprocessor error trap	-
6	User interrupt	ILM = level of accepted factor
7	NMI (for user)	ILM = 15
8	NMI (for emulator)	ILM = 4
9	Step trace trap	ILM = 4

Considering the mask processing for other EIT factors after an EIT factor is accepted, the sequence for executing the handlers for EIT factors that occur simultaneously is as shown in Table 3.8-4.

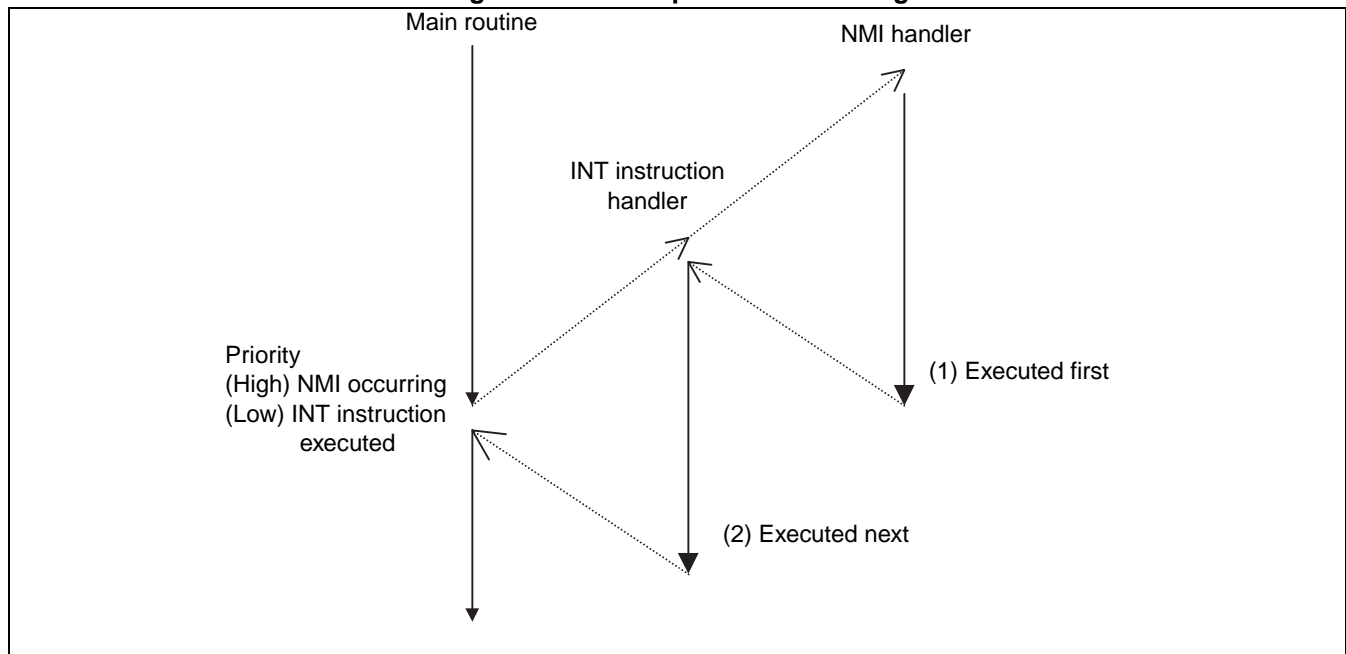
Table 3.8-4 Execution Sequence of EIT Handler

Execution sequence of handler	Factor
1	Reset *
2	Undefined instruction exception
3	INTE instruction *
4	Step trace trap
5	NMI (for user)
6	INT instruction
7	User interrupt
8	Coprocessor absent trap and coprocessor error trap

*: Other factors are annulled.

[Example]

Figure 3.8-2 Multiple EIT Processing



■ EIT Operation

In the following explanation, the transfer source "PC" is address of the instruction at which each EIT factor was detected.

Similarly, the "address of the following instruction" has the following meaning depending on the instruction at which each EIT was detected.

- At LDI: $32 \rightarrow PC + 6$
- At LDI: 20, COPOP, COPLD, COPST, COPSV $\rightarrow PC + 4$
- At the other instructions $\rightarrow PC + 2$

● Operation of user interruption and NMI

When a user interrupt or user NMI interrupt request occurs, the following sequence is used to determine whether or not to accept the request.

[Right or wrong judgment of interruption demand acceptance]

- (1) The interruption levels of requests that are generated simultaneously are compared, and the one with the highest level (the smallest numeric value) will be selected.
For the level used for the comparison, the value held in the corresponding ICR is used for a maskable interrupt and the predefined constant is used for the NMI.
- (2) If a number of interruption requests with the same level are generated, the interruption request with the smallest interruption number will be selected.
- (3) When the interrupt level is greater than or equal to the level mask value, the interrupt request is masked and is not accepted.
To (4) at interrupt levels $<$ level mask value.
- (4) When the selected interruption request is an interruption that can be masked, the interruption request will be masked and will not be accepted if the I flag is 0. To (5) if I flag is one.
To (5) regardless of the I flag value when the selected interruption demand is NMI.
- (5) If the above conditions are satisfied, the interrupt request is accepted at the instruction processing boundary.

If a user interrupt or NMI request is accepted when an EIT request is detected, the CPU operation is as follows based on the interrupt number of the accepted interrupt request.

Note: () in the [operation] shows the address which the register indicates.

[Operation]

- | | |
|---|----------------------|
| (1) SSP - 4 | \rightarrow SSP |
| (2) PS | \rightarrow (SSP) |
| (3) SSP - 4 | \rightarrow SSP |
| (4) Address of the following instruction | \rightarrow (SSP) |
| (5) Interrupt levels of accepted demand | \rightarrow ILM |
| (6) "0" | \rightarrow S flag |
| (7) (TBR + vector offset of accepted interruption demand) | \rightarrow PC |

Detection of any new EITs is performed after the interrupt sequence completes and before the initial instruction of the interrupt handler is executed. If an EIT that is able to be accepted is found at this time, the CPU changes to the EIT processing sequence.

● Operation of INT instruction

INT #u8

Branches to the interrupt handler at the vector indicated by u8.

[Operation]

- (1) SSP - 4 → SSP
- (2) PS → (SSP)
- (3) SSP - 4 → SSP
- (4) PC + 2 → (SSP)
- (5) "0" → I flag
- (6) "0" → S flag
- (7) (TBR + 3FC_H - 4 × u8) → PC

● Operation of INTE instruction

INTE

Branches to the interrupt handler for the vector with vector number #9.

[Operation]

- (1) SSP - 4 → SSP
- (2) PS → (SSP)
- (3) SSP - 4 → SSP
- (4) PC + 2 → (SSP)
- (5) "00100_B" → ILM
- (6) "0" → S flag
- (7) (TBR + 3D8_H) → PC

Do not use the INTE command during the INTE command and step trace trap processing routine.

Moreover, EIT is not generated while executing the step by INTE.

● Operation of step trace trap

If the T flag in the SCR in the PS is set to enable the step trace function, a trap occurs after each instruction and execution breaks.

[Condition of step trace trap detection]

- (1) T flag = 1
- (2) Not a delayed branch instruction.
- (3) Executing code other than an INTE instruction or step trace trap processing routine.
- (4) If the above conditions are satisfied, execution breaks at each instruction boundary.

[Operation]

- | | |
|--|----------|
| (1) SSP - 4 | → SSP |
| (2) PS | → (SSP) |
| (3) SSP - 4 | → SSP |
| (4) Address of the following instruction | → (SSP) |
| (5) "00100 _B " | → ILM |
| (6) "0" | → S flag |
| (7) (TBR + 3CC _H) | → PC |

When step trace traps are enabled by setting the T flag, NMI for users and user interruption are disabled.

Moreover, EIT by the INTE instruction is not generated.

In the FR family, the trap is generated from the following instruction by which T flag is set.

● **Operation of undefined instruction exception**

An undefined instruction exception occurs if an undefined instruction is detected during instruction decoding.

[Detection condition of undefined instruction exception]

- (1) It is detected that it is undefined instruction at the decipherment of the instruction.
- (2) Located at other than a delay slot. (Not located immediately after a delayed branch instruction.)
- (3) If the above conditions are satisfied, an undefined instruction exception is triggered and execution breaks.

[Operation]

- | | |
|-------------------------------|----------|
| (1) SSP - 4 | → SSP |
| (2) PS | → (SSP) |
| (3) SSP - 4 | → SSP |
| (4) PC | → (SSP) |
| (5) "0" | → S flag |
| (6) (TBR + 3C4 _H) | → PC |

The address saved as the PC is the address of the instruction at which the undefined instruction exception was detected.

● **Coprocessor absent trap**

When a coprocessor command using an unmounted coprocessor is executed, a coprocessor absent trap will be generated.

[Operation]

- | | |
|--|----------|
| (1) SSP - 4 | → SSP |
| (2) PS | → (SSP) |
| (3) SSP - 4 | → SSP |
| (4) Address of the following instruction | → (SSP) |
| (5) "0" | → S flag |
| (6) (TBR + 3E0 _H) | → PC |

● Coprocessor error trap

If an error occurs when the coprocessor is used and then the coprocessor instruction that operates the coprocessor is executed, a coprocessor error trap will be generated.

[Operation]

- | | |
|--|----------|
| (1) SSP - 4 | → SSP |
| (2) PS | → (SSP) |
| (3) SSP - 4 | → SSP |
| (4) Address of the following instruction | → (SSP) |
| (5) "0" | → S flag |
| (6) (TBR + 3DC _H) | → PC |

● Operation of RETI instruction

The RETI instruction is an instruction which returns from EIT processing routine.

[Operation]

- | | |
|-------------|-------|
| (1) (R15) | → PC |
| (2) R15 + 4 | → R15 |
| (3) (R15) | → PS |
| (4) R15 + 4 | → R15 |

The RETI instruction must be executed with the S flag set to "0".

■ Note

● Delay slot

In the delay slot of the branch instruction, there is a restriction concerning EIT.

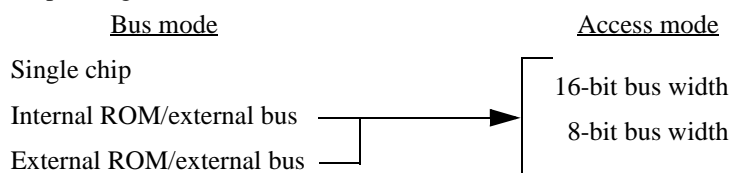
Please refer to Section "3.7 Divergence Instructions" for details of the divergence instruction.

3.9 Operating Mode

This section explains the operating mode of the MB91470/480 series.

■ Operating Mode

The operating mode includes the bus mode and the access mode.



● Bus modes

Bus mode indicates the mode that controls the internal ROM operations and external access function operations and is specified using the mode set up terminals (MD2, MD1, and MD0) and ROMA bit contents within the mode data.

● Access mode

The access mode controls the width of the external data bus and is specified by the WTH1 and WTH0 bits in the mode data and the DBW1 and DBW0 bits in ACR0 to ACR2 (Area Configuration Registers).

MB91470/480 Series

3.9.1 Bus Modes

In the MB91470/480 series, there are three bus modes shown next.
Please refer to Section "3.2 Memory Map" for details.

■ Bus Mode 0 (Single-Chip Mode)

In this mode, internal I/O, internal RAM, and internal Flash/ROM are enabled, and access to all other areas is disabled. The external pins can be used by either the peripheral resources or general-purpose ports. The pin does not work as a external bus pin.

■ Bus Mode 1 (Internal ROM/External Bus Mode)

In this mode, internal I/O, internal RAM, and internal Flash/ROM are enabled, and access to an area that enables the external access is handled as access to the external space. A part of an external terminal functions as a external bus terminal.

■ Bus Mode 2 (External ROM/External Bus Mode)

In this mode, internal I/O and internal RAM are enabled, access to internal Flash/ROM is disabled, and access to an area that enables the external access and to the internal Flash/ROM space is handled as access to the external space.

A part of an external terminal functions as a external bus terminal.

3.9.2 Mode Setting

On the FR family, the operation mode is set by the mode pins (MD2, MD1, and MD0) and mode data.

■ Mode Pin

The MD2, MD1, and MD0 pins specify operation in relation to the mode vector and reset vector fetch. Settings other than those listed in the table are prohibited.

Mode pins			Mode name	Reset vector access area	Remarks
MD2	MD1	MD0			
0	0	0	Internal ROM mode vector	Internal	
0	0	1	External ROM mode vector	External	Bus width is set by mode register.

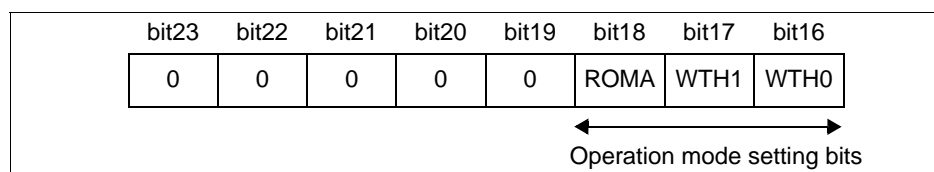
■ Mode Data

The data written to the internal mode register (MODR) by the mode vector fetch (see Section "3.10.3 Reset Sequence") is called the mode data.

After the mode register is set, the device operates in accordance with the operation mode set in the register.

The mode data is set by all types of reset. The mode data cannot be set by the user program.

<Detailed explanation of mode data>



[bit23 to bit19] Reserved bits

Always set to "00000_B". Operation is not guaranteed if a value other than "00000_B" is set.

[bit18] ROMA (Internal Flash/ROM enable bit)

This bit sets whether to enable internal Flash/ROM area.

ROMA	Function	Remarks
0	External ROM mode	Internal Flash/ROM area (80000 _H to FFFFF _H) becomes an external area.
1	Internal ROM mode	Internal Flash/ROM area (80000 _H to FFFFF _H) is enabled.

[bit17, bit16] WTH1, WTH0 (Bus width specification bits)

Set the bus width specification in external bus mode.

This value is set by DBW1 and DBW0 bits of ACR0 (CS0 area) in the external bus mode.

WTH1	WTH0	Function	Remarks
0	0	8-bit bus width	External bus mode
0	1	16-bit bus width	External bus mode
1	0	—	(Setting disabled)
1	1	Single-chip mode	Single-chip mode

3.9.3 Note

This section explains notes on setting the operation mode.

■ Note

The mode data set in the mode vector must be located as byte data at "000FFFF8_H".

As the FR family uses big endian as byte endian, place in the most significant byte (bit31 to bit24) as shown below.

		bit31	bit24	bit23	bit16	bit15	bit8	bit7	bit0
Incorrect	000FFFF8 _H	XXXXXXXX _B		XXXXXXXX _B		XXXXXXXX _B		Mode Data	
Correct	000FFFF8 _H	Mode Data		XXXXXXXX _B		XXXXXXXX _B		XXXXXXXX _B	
	000FFFC _H	Reset Vector							

MB91470/480 Series

3.10 Reset (Device Initialization)

This section describes the reset operation.

■ Overview

When reset factors are generated, the device suspends all programs and hardware operations and initializes the status. This state is called the reset state.

On removal of the reset factor, the device starts the program and hardware operation from its initialized state. The series of operations from the reset state to the start of operations is called the reset sequence.

The following table shows the reset factor, reset level, reset operation mode, and oscillation stabilization wait time after releasing set initialization reset (INIT).

Reset factor	Reset level			Reset operation mode	Oscillation stabilization wait time after releasing set initialization reset (INIT)
	System initialization reset (SINIT) - High -	Set initialization reset (INIT) - Medium -	Operation initialization reset (RST) - Low -		
External INITX pin	Issue	Issue	Issue	Normal (asynchronous) reset operation only	Minimum wait time (OS1, OS0=00 _B)
Watchdog reset	Do not issue	Issue	Issue	Normal (asynchronous) reset operation only	No oscillation stabilization wait time
Software reset	Do not issue	Do not issue	Issue	Normal (asynchronous) reset operation or synchronous reset operation	No oscillation stabilization wait time

3.10.1 Reset Level

The reset operation for MB91470/480 series is divided into three levels, each of which is triggered by different causes and performs different initialization. The following describes each reset level.

■ System Initialization Reset (SINIT)

Reset to initialize all systems is called a system initialization reset (SINIT). The main content initialized by system initialization reset (SINIT) is as follows.

[Initialization part by system initialization reset (SINIT)]

- Oscillation stabilization wait time (OS1 and OS0 bits in standby control register (STCR))
- All parts initialized by set initialization reset (INIT)

Please refer to the explanation of each function for details.

Always use the INITX pin to trigger a system initialization reset (SINIT) after the power is turned on.

■ Set Initialization Reset (INIT)

Reset to initialize all settings except for the oscillation stabilization wait time is called set initialization reset (INIT).

The main content initialized by set initialization reset (INIT) is as follows.

[Initialization part by set initialization reset (INIT)]

- All settings related to internal clock (clock source selection, PLL control, divide ratio setting)
- All settings concerning CS0 region of external bus
- All settings concerning state of other terminal
- All parts initialized by operation initialization reset (RST)

Please refer to the explanation of each function for details.

■ Operation Initialization Reset (RST)

Reset to initialize program operation is called an operation initialization reset (RST).

When a set initialization reset (INIT) is performed, the operation initialization reset (RST) is performed also.

The main content initialized by operation initialization reset (RST) is as follows.

[Initialization part by operation initialization reset (RST)]

- Program operation
- CPU and internal bus
- Register settings in peripheral circuits
- I/O port setting
- Operation mode of device (setting of bus mode and width of external bus)

Please refer to the explanation of each function for details.

3.10.2 Reset Factor

This section describes each reset factor and the associated reset level of this device.

■ Reset Factor

Reset factors that were generated in the past can be identified by reading the reset source register (RSRR).

(Refer to Section "3.11.6 Block Diagram of Clock Generation Control Unit" and Section "3.11.7 Explanation of Register Details for Clock Generation Control Unit" in the Section "3.11 Clock Generation Control" for details of the registers and flags referred to below.)

■ INITX Terminal Input (System Initialization Reset Terminal)

The INITX external pin acts as the system initialization reset terminal.

A system initialization reset (SINIT) request is generated while a low level input is applied to this pin.

System initialization reset (SINIT) demand is released by inputting the High level to this terminal.

When system initialization reset (SINIT) is generated at the request of this terminal, the bit15: INIT bit within the reset source register (RSRR) will be set.

The system initialization reset (SINIT) at the request of this terminal is the strongest of all reset factors and will be handled in priority to all other inputs, operations, and statuses.

Always use the INITX pin to trigger a system initialization reset (SINIT) after turning on the power.

Immediately after the power is turned on, maintain low level input to the INITX terminal for the oscillation stabilization wait time requested by the oscillation circuit to acquire the oscillation stabilization wait time for the oscillation circuit. (When an SINIT is triggered by the INITX pin, the oscillation stabilization wait time is initialized to its minimum value.)

- Generation factor: Low level input to external INITX terminal
- Release factor: High level input to external INITX terminal
- Generation level: System initialization reset (SINIT)
- Correspondence flag: bit15: INIT

■ Watchdog Reset

The watchdog timer will be activated by writing to the watchdog timer control register (RSRR). Then a watchdog reset request occurs per cycle specified in bit9 and bit8: WT1 and WT0 bits in the RSRR.

Watchdog reset request is set initialization reset (INIT) demand. After the request is accepted, and when a set initialization reset (INIT) or operation initialization reset (RST) is generated, the watchdog reset request will be cancelled.

When a set initialization reset (INIT) is generated by a watchdog reset request, the bit13: WDOG bit within the reset source register (RSRR) will be set.

When a set initialization reset (INIT) is generated by a watchdog reset request, the setup for the oscillation stabilization wait time will not be initialized.

- Generation factor: Specified cycle elapsed on the watchdog timer.
- Release factor: Generation of a set initialization reset (INIT) or operation initialization reset (RST).
- Generation level: Set initialization reset (INIT)
- Correspondence flag: bit13: WDOG

■ STCR: SRST Bit Writing (Software Reset)

When "0" is written to the bit4: SRST bit within the standby control register (STCR), a software reset request will be generated.

Software reset request is operation initialization reset (RST) demand.

When the request is accepted and operation initialization reset (RST) is generated, the software reset request will be cancelled.

When operation initialization reset (RST) is generated by a software reset request, the bit11: SRST bit within the reset source register (RSRR) will be set.

Operation initialization reset (RST) through the software reset request is generated only after all bus accesses are stopped when the bit9: SYNCR bit within the time-base counter control register (TBCR) is set (synchronous reset mode).

- Generation factor: Writing "0" to bit4: SRST bit of the standby control register (STCR).
- Release factor: Generation of operation initialization reset (RST)
- Generation level: Operation initialization reset (RST)
- Correspondence flag: bit11: SRST

Note:

For using software reset on the synchronous mode, see the limitations of the bit9:SYNCR bit of TBCR (time-base counter control register).

3.10.3 Reset Sequence

The device begins the execution of the reset sequence by the disappearance of the reset factor. The operation of the reset sequence is different depending on the reset level. The content of the operation for the reset sequence at each reset level is explained.

■ System Initialization Reset (SINIT) Release Sequence

This reset is triggered by an external INITX pin input.

On release of a system initialization reset (SINIT) request, the device performs the following operations in the order.

- (1) Releasing the system initialization reset (SINIT)
- (2) Set initialization reset (INIT) state and beginning of internal clock operation
- (3) Releases the set initialization reset (INIT) and changes to the oscillation stabilization wait state
- (4) The device remains in the operation initialization reset (RST) state during the minimum oscillation stabilization wait time (set by bit3, bit2: OS1 and OS0 in STCR=00_B). Internal clock stopping.
- (5) Operation initialization reset (RST) state and beginning of internal clock operation
- (6) Releases the operation initialization reset (RST) and changes to the normal operating state
- (7) Reading of mode vector from address 000FFFF8_H
- (8) Writes the mode vector to the MODR (mode register)
- (9) Reading of reset vector from address 000FFFC_H
- (10) Writing of the reset vector in PC (program counter)
- (11) Starting a program from the address contained in the PC (program counter)

■ Set Initialization Reset (INIT) Release Sequence

This reset is triggered by a watchdog reset.

On release of a set initialization reset (INIT) request, the device performs the following operations in the order.

- (1) Releasing the set initialization reset (INIT)
- (2) Operation initialization reset (RST) state and beginning of internal clock operation
- (3) Releases the operation initialization reset (RST) and changes to the normal operating state
- (4) Reading of mode vector from address 000FFFF8_H
- (5) Writes the mode vector to the MODR (mode register)
- (6) Reading of reset vector from address 000FFFC_H
- (7) Writing of the reset vector in PC (program counter)
- (8) Starting a program from the address contained in the PC (program counter)

■ Operation Initialization Reset (RST) Release Sequence

This reset is triggered by a software reset.

On release of an operation initialization reset (RST) request, the device performs the following operations in the order.

- (1) Releases the operation initialization reset (RST) and changes to the normal operating state
- (2) Reading of mode vector from address 000FFFF8_H
- (3) Writes the mode vector to the MODR (mode register)
- (4) Reading of reset vector from address 000FFFFC_H
- (5) Writing of the reset vector in PC (program counter)
- (6) Starting a program from the address contained in the PC (program counter)

MB91470/480 Series**3.10.4 Oscillation Stabilization Wait Time**

Automatically transits to oscillation stabilization waiting status when the source oscillation of the device has been suspended or when returning from a status with such possibility. This function prevents the unstable oscillator output that occurs when the oscillation first starts from being used.

During the oscillation stabilization wait time, the internal and external clock provision is suspended, only built-in time-base counter operates, and pauses until the stabilization waiting time set by the standby control register (STCR) has expired.

Hereafter, details of oscillate stabilization wait operation are explained.

■ Triggers for the Oscillation Stabilization Wait

The factor is shown below.

- When set initialization reset (INIT) is released by INITX pin factor

The device goes to the oscillation stabilization wait state immediately after a set initialization reset (INIT) is released by INITX pin factor.

The device goes to the operation initialization reset (RST) state after the oscillation stabilization wait time elapses.

The oscillation stabilization wait time is set to its minimum value by an INITX pin initialization, therefore the oscillation stabilization wait time is required by an input width of the INITX pin.

Furthermore, the device transits to the operation initialization reset (RST) state without changing to the oscillation stabilization wait state immediately after the set initialization reset (INIT) due to the watchdog reset factor is canceled.

- Recovery from stop mode

The device goes to the oscillation stabilization wait state immediately after the stop mode is released by an input of a valid external interrupt request (including NMIs). If the mode is released by the INITX pin factor, it goes to the set initialization reset (INIT) state, and then to the oscillation stabilization wait state after INIT is canceled.

The device goes to the state corresponding to the factor that stop mode is released after the oscillation stabilization wait time elapses.

- When recovering due to input of a valid external interrupt request (including NMIs) →

Transits to normal operation state

- When recovering due to a system initialization reset (SINIT) request by INITX pin factor→

Transits to system initialization reset (SINIT) state

■ Select Oscillation Stabilization Wait Time

The oscillation stabilization wait time is timed with built-in time-base counter.

When generation factors for oscillation stabilization waiting arise and it transits to the oscillation stabilization waiting status, built-in time-base counter begins measurement of the oscillation stabilization wait time after being initialized once.

4 types of oscillation stabilization wait time can be selected and set using the bit3 and bit2: OS1 and OS0 bits of the standby control register (STCR).

The setting that has been once selected will not be initialized by other than system initialization reset (SINIT) using the external INITX terminal. Set initialization resets (INIT) and operation initialization resets (RST) maintain the oscillation stabilization wait time setting before reset.

The four available oscillation stabilization wait time settings are intended for use in the following situations.

- OS1, OS0 = 00_B : No oscillation stabilization wait time
(used when the PLL and oscillator do not halt in stop mode)
- OS1, OS0 = 01_B : Oscillation stabilization wait time (short)
(used with an external clock input or when the oscillator does not halt in stop mode)
- OS1, OS0 = 10_B : Oscillation stabilization wait time (medium)
(used with an oscillator that is quick to stabilize such as a ceramic oscillator)
- OS1, OS0 = 11_B : Oscillation stabilization wait time (long)
(used with a standard crystal oscillator or similar)

Always use the INITX pin to trigger a system initialization reset (SINIT) after turning on the power.

In the following cases, maintain the low level input to the INITX pin for the stabilization wait time required by the oscillation circuit to acquire the oscillation stabilization wait time of the oscillation circuit. (When an SINIT is triggered by the INITX pin, the oscillation stabilization wait time is initialized to its minimum value.)

- INITX pin input immediately after power on
- INITX pin input during STOP mode with the oscillation halted

Accordingly, input a "L" level to the INITX pin for a period that satisfies the oscillation stabilization wait time for the main clock to allow the oscillation to stabilize.

MB91470/480 Series

3.10.5 Reset Operation Mode

There are two modes for operation initialization resets (RST), namely, normal (asynchronous) reset mode and synchronous reset mode, and which operation mode is to be used is set by the bit9: SYNCR bit of the time-base counter control register (TBCR).

This mode setting is initialized only by set initialization reset (INIT).

Set initialization reset (INIT) always does the reset action asynchronously.

Hereafter, each mode operation is explained.

■ Normal Reset Operation

The operation whereby the device goes to the operation initialization reset (RST) state immediately after an operation initialization reset (RST) request occurs is called normal reset operation.

When a reset (RST) request is received in this mode, the device goes to the reset (RST) state immediately regardless of the current status of internal bus access.

Results of the bus access performed at the time of transition to each status cannot be guaranteed under this mode. However, the operation initialization reset (RST) request can be accepted reliably.

It will be normal reset mode when the bit9: SYNCR bit within the time-base counter control register (TBCR) is "0".

The initial value after a set initialization reset (INIT) is normal reset mode.

■ Synchronous Reset Operation

The operation whereby the device goes to the operation initialization reset (RST) state only once all bus access halts after an operation initialization reset (RST) request occurs is called synchronous reset operation.

In this mode, the device does not change to the reset (RST) state while internal bus access is in progress even though a reset (RST) request may be present.

When the above request is accepted, a sleep request is issued to the internal bus. The device goes to the operation initialization reset (RST) state once each bus halts operation and goes to the sleep state.

As all bus accesses are stopped when transiting to each status under this mode, the results of all bus accesses can be guaranteed.

However, if bus access does not stop for some reason, no requests can be accepted during that time. However, even in this case, a set initialization reset (INIT) still occurs immediately.

When ready requests (RDY) are continually input into the external bus interface, and bus wait is valid, the bus access have not been halted.

This will be the synchronous reset mode when the bit9: SYNCR bit within the time-base counter control register (TBCR) is "1".

The initial value after a set initialization reset (INIT) is to return to normal reset mode.

Notes:

- Transfer of the DMA controller will be stopped on receiving each request, so transition to each status does not need to be delayed.
 - Refer to the limitations of the bit9:SYNCR bit of TBCR (time-base counter control register) for the using software reset of the synchronous mode.
-

MB91470/480 Series

3.11 Clock Generation Control

This section explains the clock generation control.

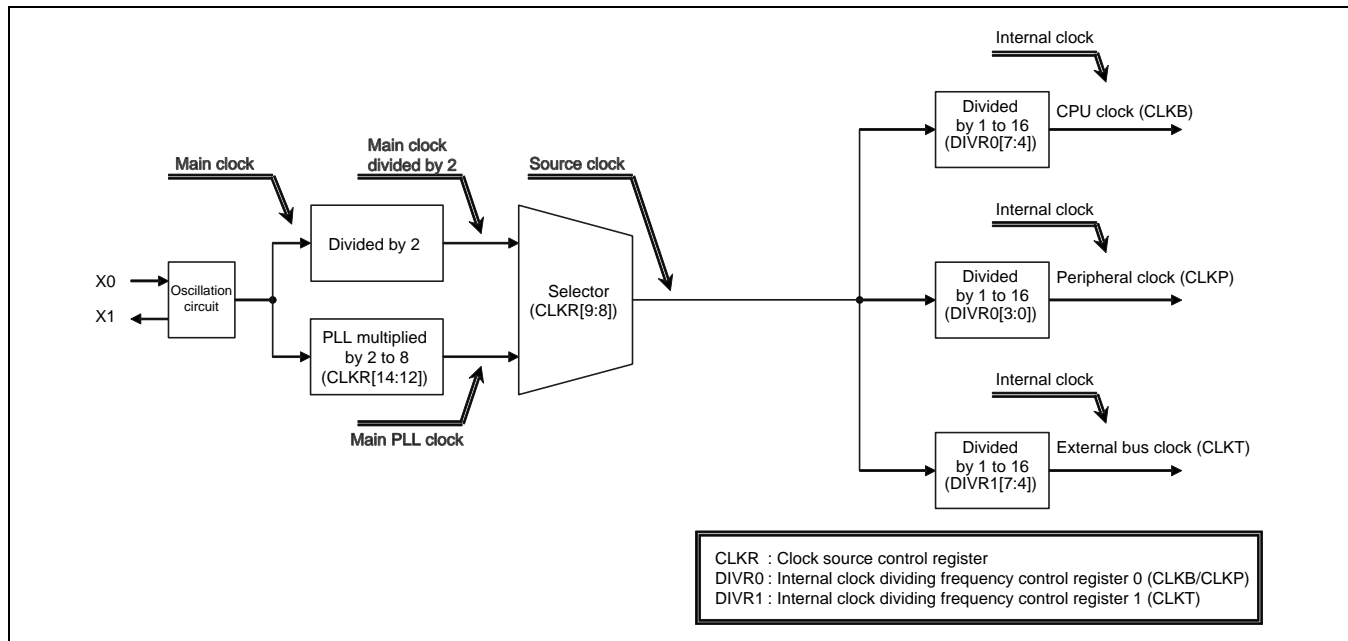
■ Overview of Clock Generation Control

The internal operation clocks on this model are generated as follows.

- Source clock generation: The base clock is generated from the main clock divided by two or by using the PLL oscillation.
- Generation of each internal clock: The source clock is divided to generate the operation clocks supplied to each block.

Hereafter, each clock generation and the control are explained.

Refer to Section "3.11.6 Block Diagram of Clock Generation Control Unit" and "3.11.7 Explanation of Register Details for Clock Generation Control Unit" for details of the registers and flags.



3.11.1 Selection of the Source Clock

This section describes how the source clock is selected.

■ Selection of the Source Clock

All clock sources including the external bus clock are supplied from within MB91470/480 series.

The external oscillator pins and internal oscillation circuit can be switched at any time while the main clock is in operation.

- Main clock: Generated from the X0 and X1 pin inputs and intended for use as the high-speed clock.

The source clock can be selectively generated from the following clocks.

- Main clock divided by two
- Main clock multiplied using the PLL

Selection of the source clock is controlled by the clock source control register (CLKR) setting.

3.11.2 PLL Control

Operation (oscillation) enable and disable and the multiplier ratio setting can be set for the PLL oscillation circuit for the main clock.

Each control is done by setting clock source control register (CLKR).

Hereafter, the content of each control is explained.

■ PLL Enabling Operation

The value of bit10: PLL1EN bit of the clock source control register (CLKR) enables or halts the main PLL oscillation.

The PLL1EN bit is initialized to "0" after a set initialization reset (INIT) to halt the main PLL oscillation. The output of the main PLL cannot be selected as the source clock while it is halted.

Once program operation has started, set the multiplier ratio for the main PLL to use it as the source clock and enable operation, and then wait for the PLL lock wait time to elapse before switching the source clock. Using the time-base timer interrupt to time the PLL lock wait is recommended.

The PLL cannot be halted while the main PLL output is selected as the source clock.

When you wish to stop the PLL such as when changing to stop mode, select the main clock divided by two as the source clock first before halting the PLL.

■ PLL Multiplication Rate

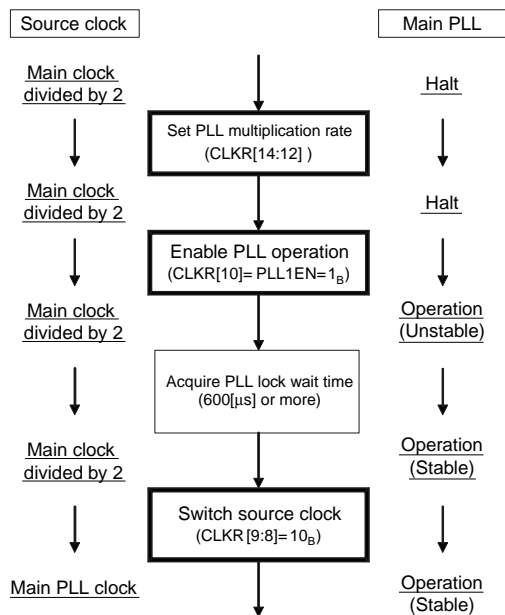
The multiplication rate for the main PLL is set up by the bit14 to bit12: PLL1S2, PLL1S1, and PLL1S0 bits of the clock source control register (CLKR).

All bits are initialized to "0" after a set initialization reset (INIT).

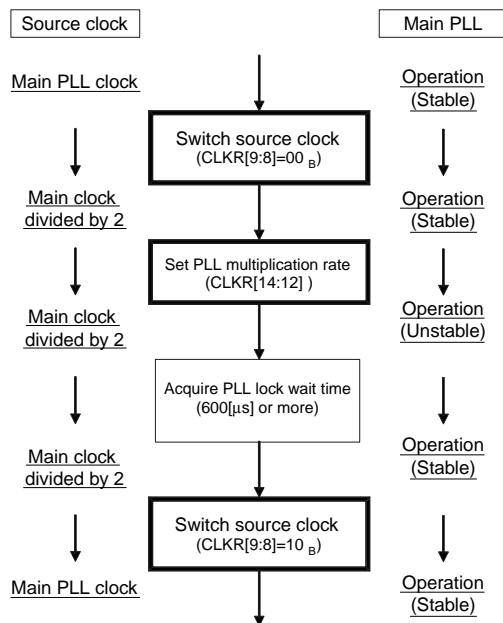
After program operation starts when changing the PLL multiplier ratio to a value different to its initial setting, always make the change before or at the same time as enabling PLL operation. After changing the multiplier ratio, wait for the lock wait time before switching the source clock. Using the time-base timer interrupt to time the PLL lock wait is recommended.

If you want to change the PLL multiplier ratio during operation, first change the source clock to something other than the PLL. After changing the multiplier ratio, wait for the lock wait time before switching the source clock, as in the case above.

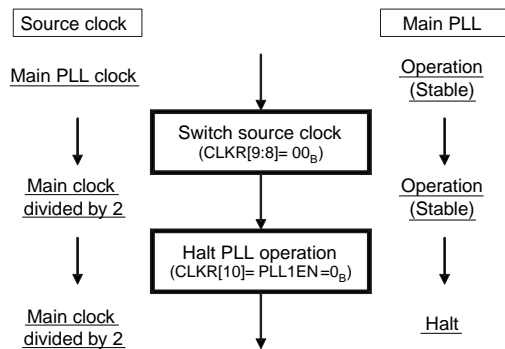
[Procedure for enabling main PLL operation => Switching source clock]

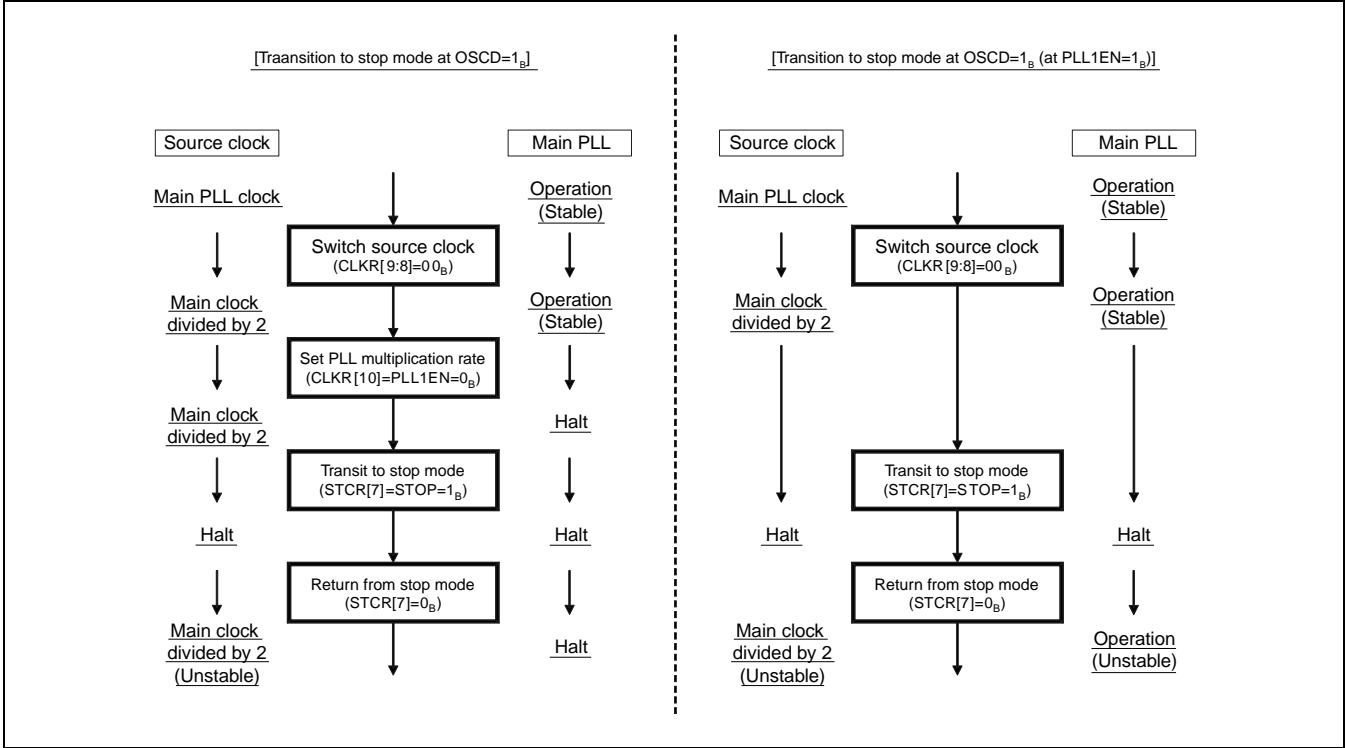


[Procedure for changing main PLL multiplication rate]



[Procedure for halting main PLL operation]





3.11.3 Oscillation Stabilization Wait and PLL Lock Wait Time

If the operation of the clock selected as the source clock is not stable, an oscillation stabilization wait time is required. (See Section "3.10.4 Oscillation Stabilization Wait Time".)

A wait time while the PLL locks is required after the PLL starts operating to allow the output to stabilize at the specified frequency.

This section describes the wait time used in various situations.

■ Wait Time after Power Supply is Turned on

An oscillation stabilization wait time for the main clock oscillation circuit is required first after the power is turned on.

Setting for oscillation stabilization wait time is initialized to the minimum value through input from the INITX terminal (system initialization reset terminal), so this oscillation stabilization wait time will be acquired from the time for inputting the low level to the INITX terminal input.

As the PLL is still not enabled in this state, the lock wait time does not need to be considered in this case.

■ Wait Time after System/Setting is Initialized

When a set initialization reset (INIT) is released after the system initialization reset (SINIT) is canceled, the device goes to the oscillation stabilization wait state. Here, the set oscillation stabilization wait time is internally generated.

As the setting time is initialized to its minimum value for the initial oscillation stabilization wait state after the INITX pin input ends, this state ends quickly and the device changes to the operation initialization reset (RST) state.

Under such statuses, no operation of any PLL is enabled, so the lock wait time does not need to be considered at this stage.

■ Wait Time after Enabling PLL Operation

If you intend to enable the PLL from the halted state after program operation starts, the output of the PLL cannot be used until the lock wait time has elapsed.

If main PLL is not selected as the source clock, program execution can continue while waiting for the PLL to lock. Using the time-base timer interrupt to time the PLL lock wait is recommended.

■ Wait Time after Changing the PLL Multiplier Ratio

After program operation starts if you want to change the multiplier ratio for the PLL while it is running, the output of the PLL cannot be used until the lock wait time has elapsed.

If main PLL is not selected as the source clock, program execution can continue while waiting for the PLL to lock. Using the time-base timer interrupt to time the PLL lock wait is recommended.

■ Wait Time after Recovering from Stop Mode

After program operation starts, the oscillation stabilization wait time set by the program is generated internally after recovering from stop mode.

If the device is set to halt the oscillation circuit for the clock selected as the source clock during stop mode, the time which adds the oscillation stabilization wait time for the oscillation circuit and the lock wait time for the PLL must be used as the wait time. Always set the oscillation stabilization wait times before changing to stop mode.

If the device is set not to halt the oscillation circuit for the clock selected as the source clock during stop mode, the PLL is not halted automatically. Accordingly, no oscillation stabilization wait time is required unless you halt the PLL.

It is recommended that you set the minimum value of the oscillation stabilization wait time before changing to stop mode.

3.11.4 Clock Distribution

The internal clocks for each function are generated from the source clock which is generated from the main clock respectively.

There are a total of three different internal clocks and the divide ratio can be set independently for each clock.

The each internal clock is explained as follows.

■ CPU Clock (CLKB)

It is a clock used for CPU, an internal memory, and an internal bus.

The circuit which uses this clock is as follows.

- CPU
- Internal RAM, internal Flash/ROM
- Bit search module
- I-bus, D-bus, F-bus, X-bus
- DMA controller
- Multiplication and addition calculator

Do not set a combination of multiplier ratio and divide ratio that results in the upper-limit frequency being exceeded.

■ Peripheral Clock (CLKP)

This is the clock used by the peripheral resources and the peripheral bus.

The circuit which uses this clock is as follows.

- Peripheral (surrounding) bus
- Clock controller (bus interface part only)
- Interrupt controller
- I/O port
- Peripheral resources such as the external interrupt inputs and 16-bit timer.

Do not set a combination of multiplier ratio and divide ratio that results in the upper-limit frequency being exceeded.

■ External Bus Clock (CLKT)

It is a clock used for the external bus interface.

The circuit which uses this clock is as follows.

- External bus interface
- External SYSCLK output

Do not set a combination of multiplier ratio and divide ratio that results in the upper-limit frequency being exceeded.

3.11.5 Clock Divider

The source clock divide ratio can be set independently for each internal clock. The best operation frequency for each circuit can be set by this function.

■ Clock Divider

The division rate is set up using internal clock dividing frequency set registers 0 (DIVR0) and 1 (DIVR1). There are 4 setting bits that support each clock in each register, and (register set up value + 1) will be the division rate for the base clock of that clock. The duty ratio is always 50% even if an odd-numbered divide ratio is set.

If the setting is modified, the new divide ratio applies from the next rising edge on the clock signal.

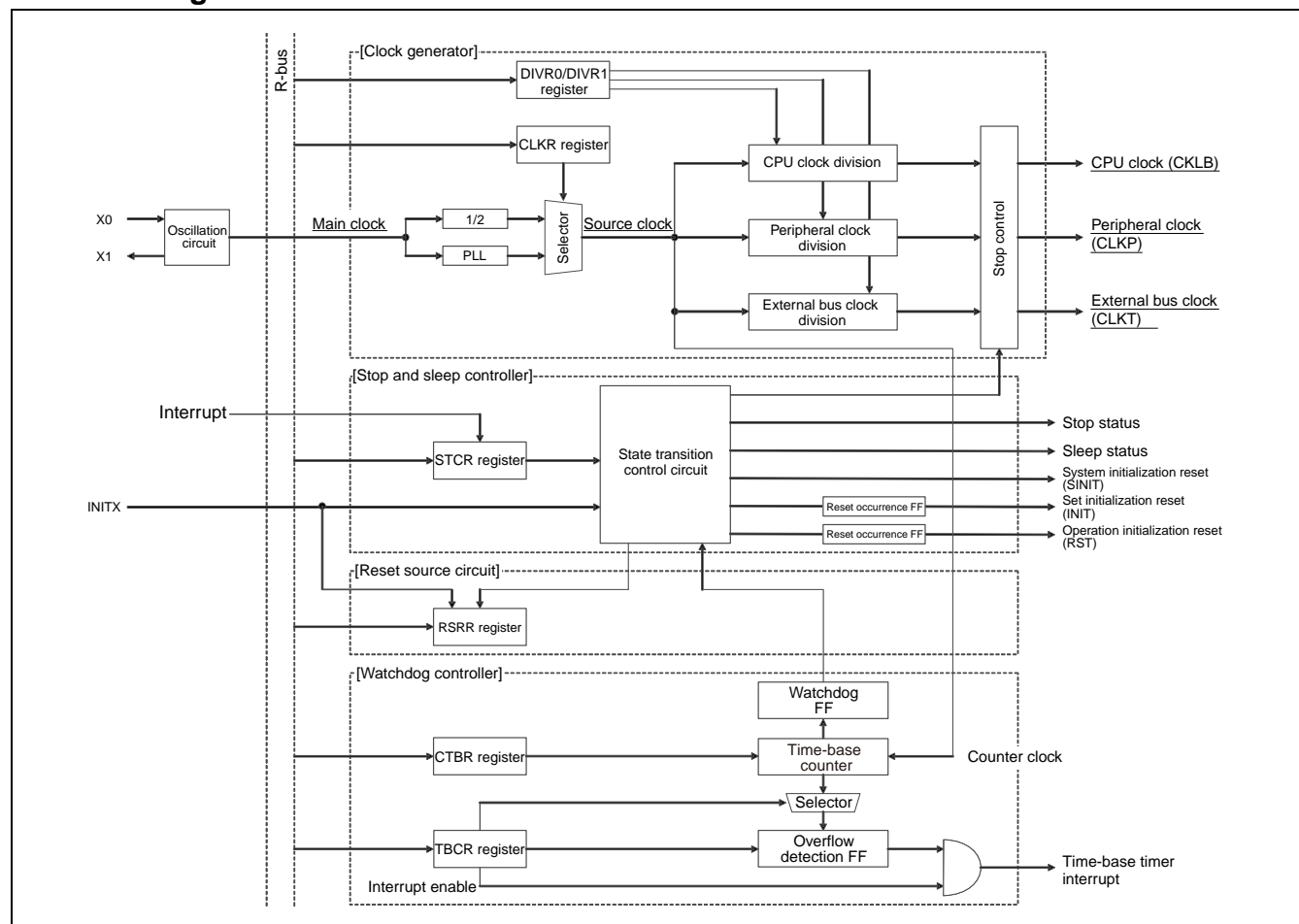
The divide ratio setting is not initialized by an operation initialization reset (RST) and the setting prior to the reset remains. The setting is only initialized by a set initialization reset (INIT). If changing the source clock from its initial setting to a higher speed, always set the divide ratio beforehand.

Operation is not guaranteed if the combination of the source clock selection, main PLL multiplier ratio setting, and divide ratio setting results in the upper-limit frequency being exceeded. Great care must be taken (in particular not to adopt the wrong order with modification settings for the source clock selection).

3.11.6 Block Diagram of Clock Generation Control Unit

The block diagram of clock generation control unit is shown as follows. Refer to Section "3.11.7 Explanation of Register Details for Clock Generation Control Unit" for detailed explanations of the register within the figure.

■ Block Diagram of Clock Generation Control Unit



MB91470/480 Series

3.11.7 Explanation of Register Details for Clock Generation Control Unit

This section describes the register of the clock generation control unit.

■ Reset Source Register/Watchdog Timer Control Register (RSRR)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
00000480 _H	INIT	-	WDOG	-	SRST	-	WT1	WT0
	R	R	R	R	R	R	R/W	R/W
Initial value (INITX pin)	1	-	0	-	0	-	0	0
Initial value (INIT)	*	*	*	x	x	*	0	0
Initial value (RST)	x	x	x	*	*	x	0	0
R/W : Readable/writable								
R : Read only								
* : Vary depending on the source.								
× : Not initialized								

This register retains reset factors that were generated just beforehand and performs cycle setting and initiation control of the watchdog timer.

After reading, the maintained reset factor is cleared when this register is read. If a number of resets are generated before reading, the reset factor flags accumulate, and a number of the flags will be set.

Writing a synchronous setting value to the WT1 and WT0 bits in this register, starts the watchdog timer. The watchdog timer keeps working until reset (RST) is generated after that.

[bit15] INIT (INITialize reset occurred)

Indicates whether a reset triggered by the INITX pin input (SINIT) has occurred.

0	No SINIT has occurred due to an INITX pin input.
1	SINIT has occurred due to an INITX pin input.

- Initialized to "0" after a read.
- read only. Writing has no effect on the bit values.

[bit14] (reserved bit)

[bit13] WDOG (WatchDOG reset occurred)

Indicates whether a reset triggered by the watchdog timer (INIT) has occurred.

0	No INIT has occurred due to the watchdog timer.
1	INIT has occurred due to the watchdog timer.

- Initialized to "0" after a read and after a reset triggered by the INITX pin input (SINIT).
- read only. Writing has no effect on the bit values.

[bit12] (reserved bit)

[bit11] SRST (Software ReSeT occurred)

Indicates whether reset (RST) by the SRST bit (software reset) of the STCR register is generated or not.

0	No RST has occurred due to a software reset.
1	RST has occurred due to a software reset.

- Initialized to "0" after a read and after a reset triggered by the INITX pin input (SINIT).
- read only. Writing has no effect on the bit values.
- Refer to the limitations of the bit9:SYNCR bit of TBCR (time-base counter control register) for the using software reset of the synchronous mode.

[bit10] (reserved bit)

[bit9, bit8] WT1, WT0 (Watchdog interval Time select)

Sets the period of the watchdog timer.

The period of the watchdog timer is selected from the following four settings based on the value written to these bits.

WT1	WT0	Occurring watchdog reset
0	0	$\phi \times 2^{17}$ (Initial value)
0	1	$\phi \times 2^{19}$
1	0	$\phi \times 2^{21}$
1	1	$\phi \times 2^{23}$

(ϕ is the period of the source clock.)

- Initialized to "00_B" by reset (RST).
- Read is enabled while write is valid only once after reset (RST), thereafter write will be invalid.

MB91470/480 Series

■ Standby Control Register (STCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000481 _H	STOP	SLEEP	HIZ	SRST	OS1	OS0	-	OSCD1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INITX pin)	0	0	1	1	0	0	-	1
Initial value (INIT)	0	0	1	1	0	0	1	1
Initial value (RST)	0	0	x	1	x	x	x	x
R/W : Readable/writable								
× : Not initialized								

It is a register which controls the operation mode of the device.

Transits to two standby modes, namely stop and sleep, controls the terminals under the stop mode and carries out oscillation stop control, sets up oscillation stabilization wait time, and issues software resets.

Note:

To place the device in standby mode, use the synchronous standby mode (set with bit8: SYNCs bit of the time-base counter control register (TBCR)) and be sure to use the following sequence:

```
// -- STCR Write
LDI    #_STCR, R0          // STCR register (0481H)
LDI    #value_of_standby, R1 // value_of_standby is the data to write to STCR.
STB    R1, @R0             // Writing in standby control register (STCR)

// -- CTBR Write
LDI    #_CTBR, R2          ; CTBR register (0483H)
LDI    #0xA5, R1           ; Clear command (1)
STB    R1, @R2             ; Write A5H to CTBR
LDI    #0x5A, R1           ; Clear command (2)
STB    R1, @R2             ; Write 5AH to CTBR (time-base counter clear)

LDUB    @R0, R1            ; STCR read (start the synchronous standby shifting)
LDUB    @R0, R1            ; Dummy re-read of STCR
NOP
NOP
NOP
NOP
NOP
```

The following describes the functions of each bit in the standby control register (STCR).

[bit7] STOP (STOP mode)

Changes the device to stop mode. When 1 is written to both bit6: SLEEP bit and this bit, this bit is given priority and the device transits to the stop mode.

0	Does not change to stop mode. (Initial value)
1	Changes to stop mode.

- Initialized to "0" by a reset (RST) and by stop recovery factor.
- Read and write are possible.

[bit6] SLEEP (SLEEP mode)

The transition to sleep mode is directed. When 1 is written to both bit7: STOP bit and this bit, bit7: STOP bit has precedence, and the device transits to the stop mode.

0	Does not change to sleep mode. (Initial value)
1	Changes to sleep mode.

- Initialized to "0" by a reset (RST) and by sleep recovery factor.
- Read and write are possible.

[bit5] HIZ (HIZ mode)

The state of the terminal at the stop mode is controlled.

0	The state of the terminal before shifting to the stop mode is maintained.
1	Terminal output is put into the state of high impedance in the stop mode. (Initial value)

- Initialized to "0" by reset (INIT).
- Read and write are possible.

[bit4] SRST (Software ReSeT)

Invokes a software reset (RST).

0	Generates a software reset.
1	Does not generate a software reset. (Initial value)

- Initialized to 1 by reset (RST).
- Read and write are possible. The value read is always "1".

[bit3, bit2] OS1, OS0 (Oscillation Stabilization time select)

Sets the oscillation stabilization wait time to use after a reset (INIT) or recovery from stop mode.

The length of the oscillation stabilization wait time is selected from the following four settings based on the value written to these bits.

OS1	OS0	Oscillation stabilization wait time	When main oscillation 10MHz	When main oscillation 20MHz
0	0	$\phi \times 2^1$ (Initial value)	400 ns	200 ns
0	1	$\phi \times 2^{11}$	408 μ s	204 μ s
1	0	$\phi \times 2^{16}$	13.1 ms	6.55 ms
1	1	$\phi \times 2^{22}$	838 ms	419 ms

(ϕ is the period of the source clock and is twice the period of the main clock.)

- Initialized to "00_B" by a reset triggered by the INITX pin input (SINIT).
- Read and write are possible.

[bit1] (reserved bit)

Reserved bit. Always write "1" to this bit on MB91470/480 series.

[bit0] OSCD1 (Oscillation Disable mode for XIN1)

Controls whether the oscillation halts for the main oscillation input (X0 and X1) during stop mode.

0	The main oscillation does not halt during stop mode.
1	The main oscillation halts during stop mode. (Initial value)

- Initialized to "1" by reset (INIT).
- Read and write are possible.

■ **Time-base Counter Control Register (TBCR)**

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
00000482 _H	TBIF	TBIE	TBC2	TBC1	TBC0	-	SYNCR	SYNCS
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	0	0	x	x	x	-	0	0
Initial value (RST)	0	0	x	x	x	x	x	x
R/W : Readable/writable								
x : Not initialized								

It is a register which controls the time-base timer interruption etc.

Enables time-base timer interruption, selects interruption interval time, and also sets up option functions for reset operations.

The following describes the functions of each bit in the time-base counter control register (TBCR).

[bit15] TBIF (time-base timer Interrupt Flag)

The interrupt flag for the time-base timer.

Indicates that the time-base counter has expired the set interval time (bit13 to bit11: TBC2 to TBC0 bits).

While interruption generation is enabled (TBIE = 1) by the bit14: TBIE bit, when this bit is "1", a time-base timer interruption request is generated.

Clear factor	Writing of 0 by instruction
Set factor	After specified interval time has elapsed (detection of a falling edge on the output of the time-base counter)

- Initialized to 0 by reset (RST).
- Read and write are possible. However, only 0 can be written to this bit, and writing 1 does not change the bit value.
Moreover, the read value in the read modification write (RMW) system instruction always becomes 1.

[bit14] TBIE (time-base timer Interrupt Enable)

It is a time-base timer interruption demand output permission bit.

Controls output of interrupt requests when the interval time set for the time-base counter elapses. While this bit is "1", bit15: TBIF bit will be "1", and a time-base timer interruption request is generated.

0	Disable output of time-base timer interrupt requests. (Initial value)
1	Enable output of time-base timer interrupt requests.

- Initialized to "0" by reset (RST).
- Read and write are possible.

[bit13 to bit11] TBC2, TBC1, TBC0 (time-base timer Counting time select)

The interval time of the time-base counter used with the time-base timer is set.

The interval time is selected from the following eight settings based on the value written to these bits.

TBC2	TBC1	TBC0	Timer interval time	For a 20MHz source oscillation and a x4 PLL multiplier
0	0	0	$\phi \times 2^{11}$	25 μ s
0	0	1	$\phi \times 2^{12}$	51.2 μ s
0	1	0	$\phi \times 2^{13}$	102.4 μ s
0	1	1	$\phi \times 2^{22}$	52.4 ms
1	0	0	$\phi \times 2^{23}$	104.9 ms
1	0	1	$\phi \times 2^{24}$	209.7 ms
1	1	0	$\phi \times 2^{25}$	419.4 ms
1	1	1	$\phi \times 2^{26}$	838.9 ms

(ϕ is the period of the source clock.)

- The initial value is undefined. Be sure to set a value before enabling the interrupt.
- Read and write are possible.

[bit10] (reserved bit)

Reserved bit. The value when read is undefined. Writing has no effect on the operation.

[bit9] SYNCR (SYNChronous Reset enable)

It is a synchronous reset operation permission bit.

This bit specifies whether normal reset operation or synchronous reset operation is executed when an operation initialization reset (RST) request. Normal reset operation performs a reset (RST) immediately. Synchronous reset operation performs an operation initialization reset (RST) after all bus accesses have stopped.

0	Normal reset operation (Initial value)
1	Synchronous reset operation

- Initialized to "0" by reset (INIT).
- Read and write are possible.

Limitation:

Meet two the following requirement before setting 0 to the SRST bit of STCR (standby control register) at using software reset of the synchronous mode.

- Set interrupt enable flag (I-Flag) to interrupt disable (I-Flag=0).
- Do not use NMI.

[bit8] SYNCS (SYNChronous Standby enable)

It is a synchronous standby operation permission bit.

Always set to "1" when using standby modes (sleep or stop mode).

0	Normal standby operation (Initial value)
1	Synchronous standby operation

- Initialized to "0" by reset (INIT).
- Read and write are possible.

Note:

Set to the synchronous standby operation by setting "1" for transiting to the standby mode.

MB91470/480 Series

■ Time-base Counter Clear Register (CTBR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000483 _H	D7	D6	D5	D4	D3	D2	D1	D0
	W	W	W	W	W	W	W	W
Initial value (INIT)	x	x	x	x	x	x	x	x
Initial value (RST)	x	x	x	x	x	x	x	x
W : Write only								
× : Not initialized								

It is a register to initialize the time-base counter.

Successively writing "A5_H" and then "5A_H" to this register clears all bits of the time-base counter to "0" immediately after the "5A_H" value is written. Although there is no limit on the length of time between writing "A5_H" and "5A_H", if you write a value other than "5A_H" after writing "A5_H", the counter is not cleared the next time you write "5A_H" unless you first write "A5_H" again.

The reading value of this register is undefined.

Note:

When this register is used to clear the time-base counter, the oscillation stabilization wait time, watchdog timer period, and time-base timer period change temporarily.

■ Clock Source Control Register (CLKR)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
00000484 _H	-	PLL1S2	PLL1S1	PLL1S0	-	PLL1EN	CLKS1	CLKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	-	0	0	0	-	0	0	0
Initial value (RST)	x	x	x	x	x	x	x	x
R/W : Readable/writable								
× : Not initialized								

This register is used to select the source clock and to control the main PLL.

This register selects the source clock. The register is also used to enable the main PLL and set the multiplier ratio.

[bit15] (reserved bit)

Reserved bit. Always write "0" to this bit on MB91470/480 series.

[bit14 to bit12] PLL1S2,PLL1S1,PLL1S0 (PLL1 ratio Select 2 to 0)

The main PLL multiplier ratio selection bits.

The main PLL multiplier ratio can be selected from the following combinations.

Modifying these bits while the main PLL is selected as the source clock is prohibited.

Do not specify a setting that will result in the upper-limit frequency for the device being exceeded.

PLL1S2	PLL1S1	PLL1S0	Main PLL multiply-by rate	When main oscillation 10MHz	When main oscillation 20MHz
0	0	0	$\times 1$ (equal)	Setting disabled	Setting disabled
0	0	1	$\times 2$ (2 multiplication)	Setting disabled	$\phi = 25$ ns (at 40 MHz)
0	1	0	$\times 3$ (3 multiplication)	Setting disabled	$\phi = 16.6$ ns (at 60 MHz)
0	1	1	$\times 4$ (4 multiplication)	$\phi = 25.0$ ns (at 40 MHz)	$\phi = 12.5$ ns (at 80 MHz)
1	0	0	$\times 5$ (5 multiplication)	$\phi = 20.0$ ns (at 50 MHz)	Setting disabled
1	0	1	$\times 6$ (6 multiplication)	$\phi = 16.6$ ns (at 60 MHz)	Setting disabled
1	1	0	$\times 7$ (7 multiplication)	$\phi = 14.3$ ns (at 70 MHz)	Setting disabled
1	1	1	$\times 8$ (8 multiplication)	$\phi = 12.5$ ns (at 80 MHz)	Setting disabled

(ϕ is the period of the main PLL clock.)

- Initialized to "000_B" by reset (INIT).
- Read and write are possible.

[bit11] (reserved bit)

Reserved bit. Always write "0" to this bit on MB91470/480 series.

[bit10] PLL1EN (PLL1 ENable)

The operation enable bit for the main PLL.

Modifying this bit while the main PLL is selected as the source clock is prohibited.

Selecting the main PLL as the source clock while this bit is "0" is prohibited.

See the bit9 and bit8: CLKS1 and CLKS0 bits settings.

If bit0: OSCD1 of STCR is "1", the main PLL stops during stop mode even if this bit is "1".

After the device returns from the stop mode, the main PLL is enabled again.

0	Main PLL stopped (Initial value)
1	Main PLL enabled

- Initialized to "0" by reset (INIT).
- Read and write are possible.

[bit9, bit8] CLKS1,CLKS0 (CLock source Select)

Sets the source clock to use.

The source clock is selected from the following three settings based on the value written to these bits.

CLKS1	CLKS0	Source clock setting
0	0	Main divided by two (Initial value)
0	1	Setting disabled
1	0	Main PLL
1	1	Setting disabled

- Initialized to "00_B" by reset (INIT).
- Read and write are possible.

Note:

Changing the value of bit8: CLKS0 when bit9: CLKS1 is "1" is prohibited.

[Combinations able to be modified]
00 _B →10 _B
10 _B →00 _B

Setting other than one of the above combinations is prohibited.

■ Internal Clock Dividing Frequency Set Register 0 (DIVR0)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
00000486 _H	B3	B2	B1	B0	P3	P2	P1	P0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	0	0	0	0	0	0	1	1
Initial value (RST)	x	x	x	x	x	x	x	x
R/W : Readable/writable								
× : Not initialized								

This register controls the ratios for dividing the source clock to generate each internal clock.

Under this register, the division rate between the CPU clock (CLKB) and peripheral clock (CLKP) will be set.

Operation is not guaranteed if the combination of the source clock selection, main PLL multiplier ratio setting, and divide ratio setting results in the upper-limit frequency being exceeded. Please take great care with this point. Also take care not to make a mistake in the sequence when changing the source clock selection.

When settings for this register are modified, after the set up, the division rate after modification from the next clock will be valid.

[bit15 to bit12] B3,B2,B1,B0 (clkB divide select 3 to 0)

These are clock dividing frequency ratio set bits of the CPU clock (CLKB).

Sets the clock divide ratio for the CPU clock (CLKB).

The value written to these bits selects the division rate to the source clock (clock frequency) for the CPU clock (CLKB) from the 16 types shown in the following table.

Do not set a divide ratio that will result in the upper-limit frequency for the device being exceeded.

B3	B2	B1	B0	Clock division ratio
0	0	0	0	ϕ
0	0	0	1	$\phi \times 2$ (divided by 2)
0	0	1	0	$\phi \times 3$ (divided by 3)
0	0	1	1	$\phi \times 4$ (divided by 4)
0	1	0	0	$\phi \times 5$ (divided by 5)
0	1	0	1	$\phi \times 6$ (divided by 6)
0	1	1	0	$\phi \times 7$ (divided by 7)
0	1	1	1	$\phi \times 8$ (divided by 8)
...
1	1	1	1	$\phi \times 16$ (divided by 16)

(ϕ is the period of the source clock.)

- Initialized to "0000_B" by reset (INIT).
- Read and write are possible.

[bit11 to bit8] P3,P2,P1,P0 (clkP divide select 3 to 0)

These are clock dividing frequency ratio set bits of the peripheral clock (CLKP).

Sets the clock divide ratio for the peripheral clock (CLKP).

The value written to these bits selects the division ratio to the source clock (clock frequency) for the peripheral clock (CLKP) from the 16 types shown in the following table.

Do not set a divide ratio that will result in the upper-limit frequency for the device being exceeded.

P3	P2	P1	P0	Clock division ratio
0	0	0	0	ϕ
0	0	0	1	$\phi \times 2$ (divided by 2)
0	0	1	0	$\phi \times 3$ (divided by 3)
0	0	1	1	$\phi \times 4$ (divided by 4)
0	1	0	0	$\phi \times 5$ (divided by 5)
0	1	0	1	$\phi \times 6$ (divided by 6)
0	1	1	0	$\phi \times 7$ (divided by 7)
0	1	1	1	$\phi \times 8$ (divided by 8)
...
1	1	1	1	$\phi \times 16$ (divided by 16)

(ϕ is the period of the source clock.)

- Initialized to "0011_B" by reset (INIT).
- Read and write are possible.

■ **Internal Clock Dividing Frequency Set Register 1 (DIVR1)**

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000487 _H	T3	T2	T1	T0	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	0	0	0	0	0	0	0	0
Initial value (RST)	x	x	x	x	x	x	x	x
R/W : Readable/writable								
x : Not initialized								

This register controls the ratios for dividing the source clock to generate each internal clock.

This register sets the divide ratio for the clock used by the external bus clock (CLKT).

Operation is not guaranteed if the combination of the source clock selection, main PLL multiplier ratio setting, and divide ratio setting results in the upper-limit frequency being exceeded. Please take great care with this point. Also take care not to make a mistake in the sequence when changing the source clock selection.

When settings for this register are modified, after the set up, the division rate after modification from the next clock will be valid.

[bit7 to bit4] T3,T2,T1,T0 (clkT divide select 3 to 0)

These are clock dividing frequency ratio set bits of the external bus clock (CLKT).

Sets the clock divide ratio for the clock used by the external bus clock (CLKT).

The value written to these bits selects the division rate to the source clock (clock frequency) for external bus clock (CLKT) from the 16 types shown in the following table.

Do not set a divide ratio that will result in the upper-limit frequency for the device being exceeded.

T3	T2	T1	T0	Clock division ratio
0	0	0	0	ϕ
0	0	0	1	$\phi \times 2$ (divided by 2)
0	0	1	0	$\phi \times 3$ (divided by 3)
0	0	1	1	$\phi \times 4$ (divided by 4)
0	1	0	0	$\phi \times 5$ (divided by 5)
0	1	0	1	$\phi \times 6$ (divided by 6)
0	1	1	0	$\phi \times 7$ (divided by 7)
0	1	1	1	$\phi \times 8$ (divided by 8)
...
1	1	1	1	$\phi \times 16$ (divided by 16)

(ϕ is the period of the source clock.)

- Initialized to "0000_B" by reset (INIT).
- Read and write are possible.

Set "1111_B" (divided by 16) if not using the external bus interface.

[bit3 to bit0] (reserved bit)

- Initialized to "0000_B" by reset (INIT).
- Always write "0000_B" to these bits.

3.11.8 Peripheral Circuit Functions in the Clock Controller

The following describes the peripheral circuit functions in the clock controller.

■ Time-base Counter

The clock controller includes a 26-bit time-base counter which runs on the source clock.

In addition to generating the oscillation stabilization wait time (see Section "3.10.4 Oscillation Stabilization Wait Time"), the time-base counter is used for the following purposes.

- Watchdog timer

The watchdog timer is used to detect system runaway and measures the bit output of the time-base counter.

- time-base timer

Uses the output of the time-base counter to generate interval interrupts.

Hereafter, these functions are explained.

● Watchdog timer

The watchdog timer uses the output of the time-base counter to detect program runaway.

If postponement of the watchdog reset is not generated between the intervals that have been set, due to a program runaway or such like, the set initialization reset (INIT) request is generated as a watchdog reset.

[Starting the watchdog timer and setting the period]

The watchdog timer is activated by first writing a period setting value to the WT1 and WT0 bits in the reset source register/watchdog timer control register (RSRR) after reset (RST). In this case, the interval for the watchdog timer is set by the bit9 and bit8: WT1 and WT0 bits. For the interval setting, only the time that has been set through this first writing will be valid, and all other writings after that will be ignored.

[Generation of a watchdog reset]

The flag for generating watchdog resets is set by the falling edge of the time-base counter output for the interval that has been set. If the flag is set when the 2nd falling edge is detected, a set initialization reset (INIT) request is generated as the watchdog reset.

[Stopping watchdog timer]

Once the watchdog timer is activated, the watchdog timer cannot be stopped until an operation initialization reset (RST) is generated.

Under the following status in which an operation initialization reset (RST) is generated, the watchdog timer is stopped and does not function until activated by a re-program operation.

- State of operation initialization reset (RST)
- State of set initialization reset (INIT)
- State of system initialization reset (SINIT)
- State of oscillation stabilization wait reset (RST)

[Temporarily halting the watchdog timer (automatically postpone generation)]

The watchdog timer initializes the flag to once generate a watchdog reset when the program operation of the CPU is stopped and postpones generation of a watchdog reset. The stop of the program operation concretely shows the following operations.

- Sleep state
- Stop state
- Oscillation stabilization wait RUN state
- During DMA transfer to the D-bus (data bus)
- During a break when using the emulator-debugger

When the time-base counter is cleared, the flag for generating watchdog resets is simultaneously initialized, and generation of a watchdog reset will be postponed.

Note:

Clear the time-base counter immediately before reading standby control register (STCR) after setting STOP mode bit/SLEEP mode bit, when the mode shifts to sleep mode/stop mode after starting the watchdog timer. The program example is described as follows;

● Sample program

Standby (stop or sleep) mode shift processing

```
-----
// -- Write STCR
LDI    #_STCR,R0          // STCR register (0481H)
LDI    #value_of_standby, R1 // value_of_standby is a write data to STCR
STB    R1,@R0             // Write to STCR
// -- Write CTBR
LDI    #_CTBR,R2          // CTBR register (0483H)
LDI    #0xA5,R1           // Clear command (1)
STB    R1,@R2             // Write A5H to CTBR
LDI    #0x5A,R1           // Clear command (2)
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)

LDUB    @R0,R1            // Read STCR (Start the synchronous stand-by shifting)
LDUB    @R0,R1            // Read dummy STCR
NOP                                           // NOP × 5 for adjust timing
NOP
NOP
NOP
NOP
-----
```

● time-base timer

The time-base timer uses the output of the time-base counter to generate interval interrupts.

The timer is suitable for measuring relatively long times, up to $\{\text{source clock} \times 2^{26}\}$ cycles such as for the main PLL lock wait time.

The time-base timer interruption request is generated when the falling edge of the output for the time-base counter that supports the set interval is detected.

[Setting of time-base timer at startup and intervals]

The time-base timer sets the interval using the bit13 to bit11: TBC2, TBC1, and TBC0 bits of the time-base counter control register (TBCR).

As the falling edge of the output for the time-base counter that supports the set interval is always detected, after the interval is set, firstly clear the bit15: TBIF bit, and then enable interruption request output by setting "1" as the bit14: TBIE bit.

Before changing the interval time, disable the interrupt request output by setting "0" as the bit14: TBIE bit.

As the time-base counter always counts without being influenced by these settings, clear the time-base counter before enabling interruption in order to get accurate interval interruption times. If this is not done, an interrupt request may be generated immediately after interrupts are enabled.

[Clearness of time-base counter by program]

Writing "A5_H" and then "5A_H" to the time-base counter clear register (CTBR) clears all bits of the time-base counter to "0" immediately after writing "5A_H". Although there is no limit on the length of time between writing "A5_H" and "5A_H", if you write a value other than "5A_H" after writing "A5_H", the counter is not cleared the next time you write "5A_H" unless you first write "A5_H" again.

The flag for generating watchdog resets is simultaneously initialized when this time-base counter is cleared, and generation of watchdog resets will be postponed.

[Clearness of time-base counter by state of device]

The time-base counter is cleared all bits to "0" simultaneously when the device goes to the following states.

- Stop state
- State of set initialization reset (INIT)
- State of system initialization reset (SINIT)

In particular, under the stop status, the time-base counter is used to measure oscillation stabilization wait times, so an interval interruption of the time-base timer may be generated unintentionally. Thus, before setting the stop mode, disable time-base timer interruption, and try not to use the time-base timer.

For statuses other than that, time-base timer interruption is automatically disabled as operation initialization reset (RST) is generated.

Note:

Clear the time-base counter immediately before reading standby control register (STCR) after setting STOP mode bit/SLEEP mode bit, when the mode shifts to sleep mode/stop mode after starting the watchdog timer. The program example is described as follows;

● Sample program

Standby (stop or sleep) mode shift processing

```
-----
// -- Write STCR
LDI    #_STCR,R0          // STCR register (0481H)
LDI    #value_of_standby, R1 // value_of_standby is a write data to STCR
STB    R1,@R0             // Write to STCR
// -- Write CTBR
LDI    #_CTBR,R2          // CTBR register (0483H)
LDI    #0xA5,R1           // Clear command (1)
STB    R1,@R2             // Write A5H to CTBR
LDI    #0x5A,R1           // Clear command (2)
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)

LDUB   @R0,R1             // Read STCR (Start the synchronous stand-by shifting)
LDUB   @R0,R1             // Read dummy STCR
NOP    // NOP × 5 for adjust timing
NOP
NOP
NOP
NOP
-----
```

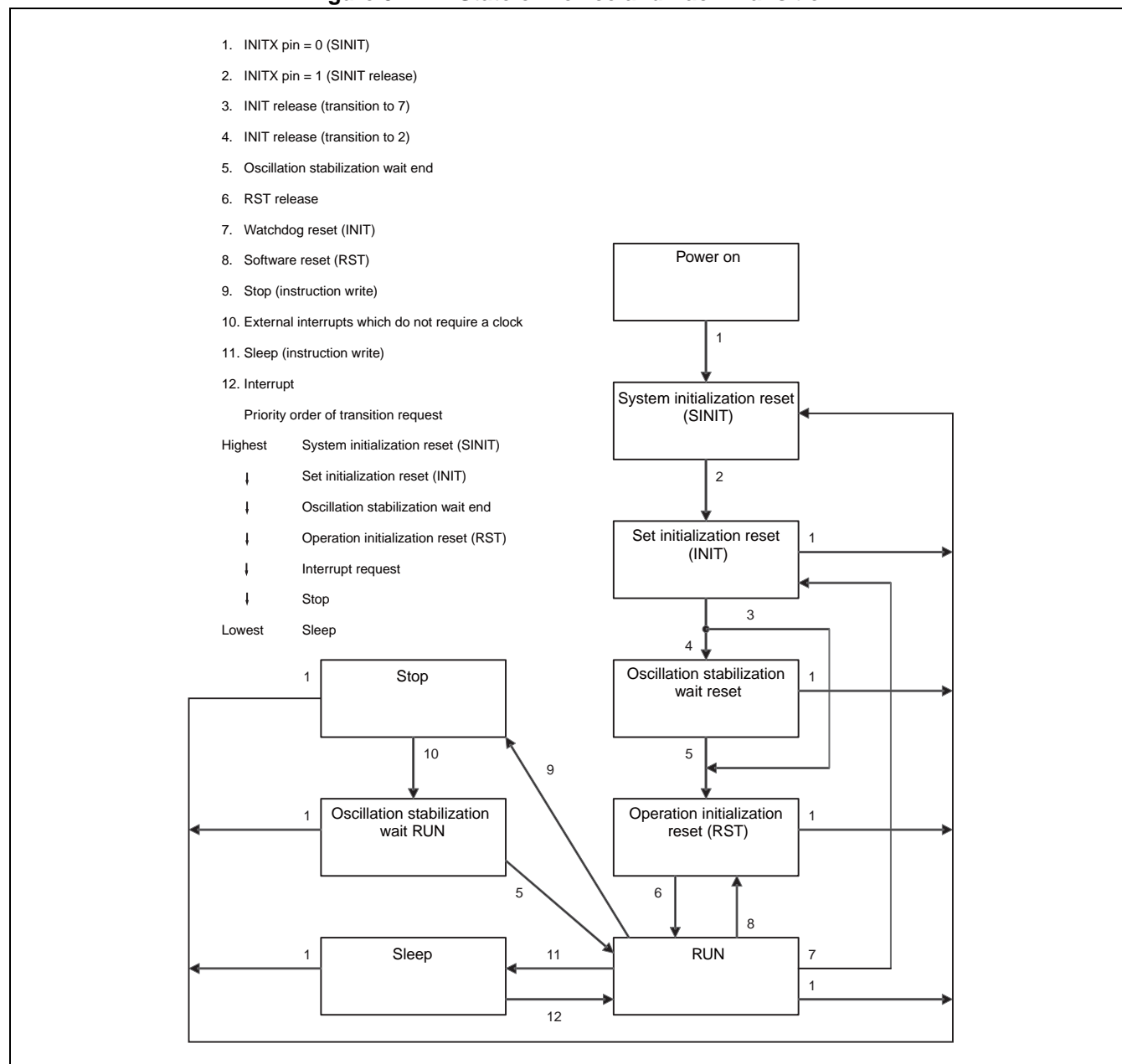
3.12 Device State Control

This section describes the various device states and how these are controlled.

■ State of Device and Each Transition

The figure below shows the state transitions for this device.

Figure 3.12-1 State of Device and Each Transition



The following has MB91470/480 series operation states.

● **State of RUN (normal operation)**

This is the program execution state.

All internal clocks are supplied, and all the circuits are operable.

Each status transition request is accepted, but when synchronous reset mode is selected, the status transition operation for normal reset mode cases and for some requests is different. See "■ Synchronous Reset Operation" in Section "3.10.5 Reset Operation Mode" for details.

● **Sleep state**

The program halts in this state. The device is set to this state by program operation.

Only the program execution of CPU stops, and the circuit in the surrounding is operable. All internal memory and internal and external buses halt unless requested by the DMA controller.

The device recovers from this state when a valid interrupt request occurs and goes to the RUN state (normal operation).

The device changes to the system initialization reset (SINIT) state when a system initialization reset (SINIT) request occurs.

● **Stop state**

It is a stopped state of the device. The device is set to this state by program operation.

All internal circuits stop. All internal clocks halt. The oscillation circuit and main PLL can also be set to halt in this state. Moreover, an external terminal can be made uniform high impedance by setting. (A part of pin is excluded)

The device changes to the oscillation stabilization wait RUN state when certain interrupt requests occur (interrupt requests able to be generated while the clock is halted).

The device changes to the system initialization reset (SINIT) state when a system initialization reset (SINIT) request occurs.

● **Oscillation stabilization wait RUN state**

It is a stopped state of the device. The device changes to this state after recovering from stop mode.

All internal circuits except the clock generation control unit (time-base counter and device status control unit) are stopped. All internal clocks halt, but the oscillation circuit and the main PLL (if enabled) continue to operate.

The high impedance control of an external terminal in the state etc. of the stop is released.

The device goes to the RUN state (normal operation) after the specified oscillation stabilization wait time elapses.

The device changes to the system initialization reset (SINIT) state when a system initialization reset (SINIT) request occurs.

● **Oscillation stabilization wait reset (RST) state**

It is a stopped state of the device. The device goes to this state after recovering from a set initialization reset (INIT) status.

All internal circuits except the clock generation control unit (time-base counter and device status control unit) are stopped. All internal clocks halt, but the oscillation circuit continues to operate.

Operation initialization reset (RST) is output to an internal circuit.

The device goes to the Operation initialization reset (RST) state after the specified oscillation stabilization wait time elapses.

The device changes to the system initialization reset (SINIT) state when a system initialization reset (SINIT) request occurs.

● State of operation initialization reset (RST)

The program is being initialized. Transits by receiving the operation initialization reset (RST) request or ending the oscillation stabilization wait reset (RST) status.

The program execution of CPU stops, and the program counter is initialized. The circuit in the surrounding is initialized excluding part. All internal clocks, the oscillation circuit, and the main PLL (if enabled) operate.

Operation initialization reset (RST) is output to an internal circuit.

Transits to the RUN status (normal operation) by diminishing the operation initialization reset (RST) request, and operation initialization reset sequence is executed.

The device changes to the system initialization reset (SINIT) state when a system initialization reset (SINIT) request occurs.

● State of set initialization reset (INIT)

The settings are being initialized. The device changes to this state on receiving a set initialization reset (INIT) request.

The program execution of CPU stops, and the program counter is initialized. All the circuits in the surrounding are initialized. The oscillation circuit operates, but the main PLL is halted. All internal clocks and oscillation circuits operate.

Outputs a set initialization reset (INIT) and operation initialization reset (RST) to internal circuits.

This status is cancelled by diminishing the set initialization reset (INIT) request, and transits to the oscillation stabilization wait reset (RST) status or the operation initialization reset (RST) status. Then the operation initialization reset sequence is executed.

● State of system initialization reset (SINIT)

The system settings are being initialized. The device changes to this state on receiving a system initialization reset (SINIT) request.

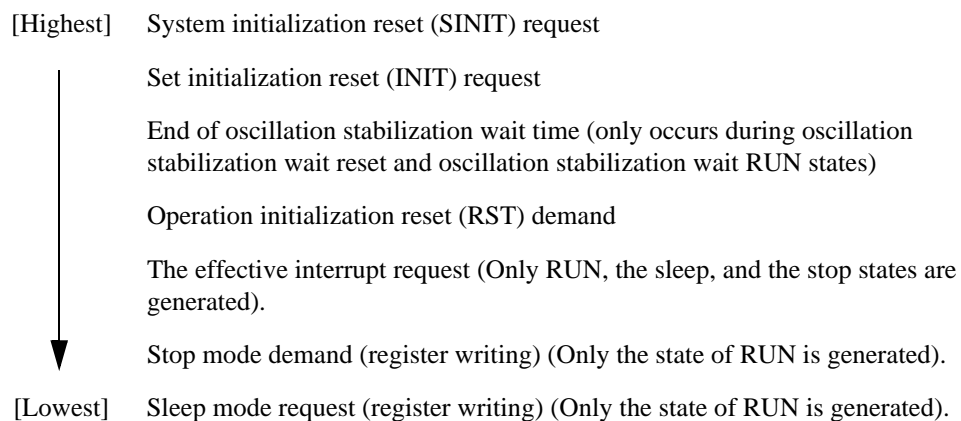
The program execution of CPU stops, and the program counter is initialized. All the circuits in the surrounding are initialized. The oscillation circuit operates, but the main PLL and all internal clocks are halted.

Outputs a system initialization reset (SINIT), set initialization reset (INIT), and operation initialization reset (RST) to internal circuits.

This status is canceled by diminishing the system initialization reset (SINIT) request, and transits to the set initialization reset (INIT) status.

● Priority level of each state transition demand

In any state, each state transition demand conforms to the following priority levels. However, as certain requests only occur in specific states, these are only meaningful in those states.



■ Low-power Consumption Mode

The following describes the low-power consumption modes available on MB91470/480 series and how to use them.

MB91470/480 series has the following low-power consumption modes.

- Sleep mode
The device is set to sleep mode by setting to a register.
- Stop mode
The device is set to stop mode by setting to a register.

The following describes each mode.

● Sleep mode

Writing "1" to bit6: SLEEP bit of standby control register (STCR) sets sleep mode and changes the device to the sleep state. The device remains in the sleep state until something happens that causes the device to recover from the sleep state.

See also "■State of Device and Each Transition ●Sleep state" in Section "3.12 Device State Control" for details about sleep state.

[Transition to sleep mode]

If synchronous standby mode (set using bit8: SYNCSC bit of the time-base counter control register (TBCR)) is used when setting the device to sleep mode, always use the following sequence.

```
-----  
// -- Write STCR  
LDI    #_STCR,R0          // STCR register (0481H)  
LDI    #value_of_standby,R1 // value_of_standby is a write data to STCR  
STB    R1,@R0             // Write to STCR  
// -- Write CTBR  
LDI    #_CTBR,R2          // CTBR register (0483H)  
LDI    #0xA5,R1           // Clear command (1)  
STB    R1,@R2             // Write A5H to CTBR  
LDI    #0x5A,R1           // Clear command (2)  
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)  
  
LDUB   @R0,R1             // Read STCR (Start the synchronous stand-by shifting)  
LDUB   @R0,R1             // Read dummy STCR  
NOP  
NOP // NOP × 5 for adjust timing  
NOP  
NOP  
NOP  
NOP  
-----
```

If "1" is written to both this bit and bit7: STOP bit of the standby control register (STCR), bit7: STOP bit has precedence and the device goes to the stop state.

[Circuits that halt during sleep state]

- Program execution by the CPU
- Bit search module *
- Various internal memory *
- Internal and external buses *

*: The circuits operate if DMA transfer occurs.

[Circuits that do not halt during sleep state]

- Oscillation circuit
- Main PLL (if enabled)
- Clock generation control unit
- Interrupt controller
- Peripheral circuit
- DMA controller
- Multiplication and addition calculator
- On chip Debug Support Unit (DSU)

[Events that recover the device from sleep state]

- Generation of a valid interrupt request

The device recovers from sleep mode when an interrupt request occurs that is not disabled interrupt ("1111_B") in the ICR register and changes to the RUN state (normal operation). In this case, you should set the I flag in the CPU's PS register to "1" to enable interrupt acceptance and cause the interrupt handler to be executed on recovering from sleep mode.

The device does not recover from sleep mode if an interrupt request occurs that is disabled interrupt ("1111_B") in the ICR register.

- Generation of a system initialization reset (SINIT) request

The device always goes to the system initialization reset (SINIT) state unconditionally when a system initialization reset (SINIT) request occurs.

Note: See "■State of Device and Each Transition ●Priority level of each state transition demand" in Section "3.12 Device State Control" for details of the priority order for the different types of trigger.

[Synchronous standby operation]

Synchronous standby operation is enabled if bit8: SYNCS bit of the time-base counter control register (TBCR) is set to "1". Transition to the sleep state is not caused only by a write to the SLEEP bit. Then, transition to the sleep state occurs by reading the STCR register.

To enter the sleep mode, be sure to use the sequence in [Transition to sleep mode].

Note:

Clear the time-base counter immediately before reading standby control register (STCR) after setting STOP mode bit/SLEEP mode bit, when the mode shifts to sleep mode/stop mode after starting the watchdog timer. The program example is described as follows;

● Sample program

Standby (stop or sleep) mode shift processing

```
-----  
// -- Write STCR  
LDI    #_STCR,R0          // STCR register (0481H)  
LDI    #value_of_standby, R1 // value_of_standby is a write data to STCR  
STB    R1,@R0             // Write to STCR  
// -- Write CTBR  
LDI    #_CTBR,R2          // CTBR register (0483H)  
LDI    #0xA5,R1           // Clear command (1)  
STB    R1,@R2             // Write A5H to CTBR  
LDI    #0x5A,R1           // Clear command (2)  
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)  
  
LDUB   @R0,R1             // Read STCR (Start the synchronous stand-by shifting)  
LDUB   @R0,R1             // Read dummy STCR  
NOP    // NOP × 5 for adjust timing  
NOP  
NOP  
NOP  
NOP  
-----
```

● Stop mode

Writing "1" to bit7: STOP bit of the standby control register (STCR) sets stop mode and changes the device to the stop state. The device remains in the stop state until something happens that causes the device to recover from the stop state.

See also "■State of Device and Each Transition ●Stop mode" in Section "3.12 Device State Control" for details about the stop state.

[Transition to stop mode]

If synchronous standby mode (set using bit8: SYNCs bit of the time-base counter control register (TBCR)) is used when setting the device to stop mode, always use the following sequence.

```
-----
// -- Write STCR
LDI    #_STCR,R0          // STCR register (0481H)
LDI    #value_of_standby,R1 // value_of_standby is a write data to STCR
STB    R1,@R0             // Write to STCR

// -- Write CTBR
LDI    #_CTBR,R2          // CTBR register (0483H)
LDI    #0xA5,R1           // Clear command (1)
STB    R1,@R2             // Write A5H to CTBR
LDI    #0x5A,R1           // Clear command (2)
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)

LDUB   @R0,R1             // Read STCR (Start the synchronous stand-by shifting)
LDUB   @R0,R1             // Read dummy STCR
NOP                    // NOP × 5 for adjust timing
NOP
NOP
NOP
NOP
NOP
-----
```

If "1" is written to both this bit and bit6: SLEEP bit of the standby control register (STCR), bit7: STOP bit has precedence and the device goes to the stop state.

[Circuits that halt during stop state]

All circuits halt except the following.

[Circuits that do not halt during stop state]

- Oscillation circuits that are not set to halt

The oscillation circuit for the main clock does not halt during the stop state if bit0: OSCD1 bit in standby control register (STCR) is set to "0".

- Main PLL if enabled and if connected to an oscillation circuit that is not set to halt

The PLL for the main clock does not halt during the stop state if bit0: OSCD1 bit in standby control register (STCR) is set to "0" and bit10: PLL1EN bit of the clock source control register (CLKR) is set to "1".

[Pin high-impedance control during the stop state]

Pin outputs go to the high-impedance state during the stop state if bit5: HIZ bit of the standby control register (STCR) is set to "1". The pins to which this control applies are listed in "APPENDIX C Pin States in Each CPU State".

If bit5: HIZ bit of the standby control register (STCR) is set to "0" during the stop state, pin outputs maintain the values they had prior to the device changing to the stop state. For details, refer to "APPENDIX C Pin States in Each CPU State".

[Events that recover the device from stop mode]

- Generation of certain valid interrupt requests (clock is not required).

Only some enabled external interrupt and the NMI input pin are valid.

The device recovers from stop mode when an interrupt request occurs that is not disabled interrupt ("1111_B") in the ICR register and changes to the oscillation stabilization wait RUN state. In this case, you should set the I flag in the CPU's PS register to "1" to enable acceptance of interrupts and cause the interrupt handler to be executed on recovering from stop mode.

The device does not recover from stop mode if an interrupt request occurs that is disabled interrupt ("1111_B") in the ICR register.

- Generation of a system initialization reset (SINIT) request

The device always goes to the system initialization reset (SINIT) state unconditionally when a system initialization reset (SINIT) request occurs.

Note: See "■State of Device and Each Transition ●Priority level of each state transition demand" in Section "3.12 Device State Control" for details of the priority order for the different types of trigger.

[Source clock selection in stop mode]

Always set the clock source to the main clock divided by two before setting stop mode. See Section "3.11 Clock Generation Control", particularly Section "3.11.2 PLL Control" for details.

The same restrictions as in normal operation apply to the setting of the divide ratio.

[Synchronous standby operation]

Synchronous standby operation is enabled if bit8: SYNCS bit of the time-base counter control register (TBCR) is set to "1". In this case, simply writing to the STOP bit will not change the device to the stop state. Then, transition to the stop state occurs by reading the STCR register.

When using stop mode, always use the sequence described in [Transition to stop mode].

Note:

Clear the time-base counter immediately before reading standby control register (STCR) after setting STOP mode bit/SLEEP mode bit, when the mode shifts to sleep mode/stop mode after starting the watchdog timer. The program example is described as follows;

● Sample program

Standby (stop or sleep) mode shift processing

```
-----
// -- Write STCR
LDI    #_STCR,R0          // STCR register (0481H)
LDI    #value_of_standby, R1 // value_of_standby is a write data to STCR
STB    R1,@R0             // Write to STCR
// -- Write CTBR
LDI    #_CTBR,R2          // CTBR register (0483H)
LDI    #0xA5,R1           // Clear command (1)
STB    R1,@R2             // Write A5H to CTBR
LDI    #0x5A,R1           // Clear command (2)
STB    R1,@R2             // Write 5AH to CTBR (time-base counter clear)

LDUB   @R0,R1             // Read STCR (Start the synchronous stand-by shifting)
LDUB   @R0,R1             // Read dummy STCR
NOP    // NOP × 5 for adjust timing
NOP
NOP
NOP
NOP
-----
```


CHAPTER 4

EXTERNAL BUS INTERFACE

The external bus interface controller controls the interfaces with the internal bus for LSI and with external memory and I/O devices. This chapter explains each function of the external bus interface.

- 4.1 Features of the External Bus Interface
- 4.2 External Bus Interface Registers
- 4.3 Chip Select Area
- 4.4 Endian and Bus Access
- 4.5 Ordinary Bus Interface
- 4.6 Address/Data Multiplex Interface
- 4.7 Procedure for Setting a Register

4.1 Features of the External Bus Interface

This section describes the features of the external bus interface.

■ Features of the External Bus Interface

- Addresses of up to 16 bits can be output.
 - Various kinds of external memory (8-bit/16-bit) can be directly connected and multiple access timings can be mixed and controlled.
 - Asynchronous SRAM and asynchronous ROM/Flash memory (multiple write strobe method or byte enable method)
 - Address/data multiplex bus (8-bit/16-bit width only)
 - Synchronous memory (such as ASIC built-in memory)
-

Note:

Synchronous SRAM cannot be directly connected.

- Three independent banks (chip select areas) can be set, and chip select corresponding to each bank can be output.
 - The size of each area can be set in multiples of 64 Kbytes (64 Kbytes to 2 GB for each chip select area).
 - An area can be set at any location in the logical address space (Boundaries may be limited depending on the size of the area.)
 - In each chip select area, the following functions can be set independently:
 - Enabling and disabling of the chip select area (Disabled areas cannot be accessed)
 - Setting of the access timing type to support various kinds of memory
 - Detailed access timing setting (individual setting of the access type such as the wait cycle)
 - Setting of the data bus width (8-bit/16-bit)
 - Setting for the order of bytes (big or little endian)
-

Note:

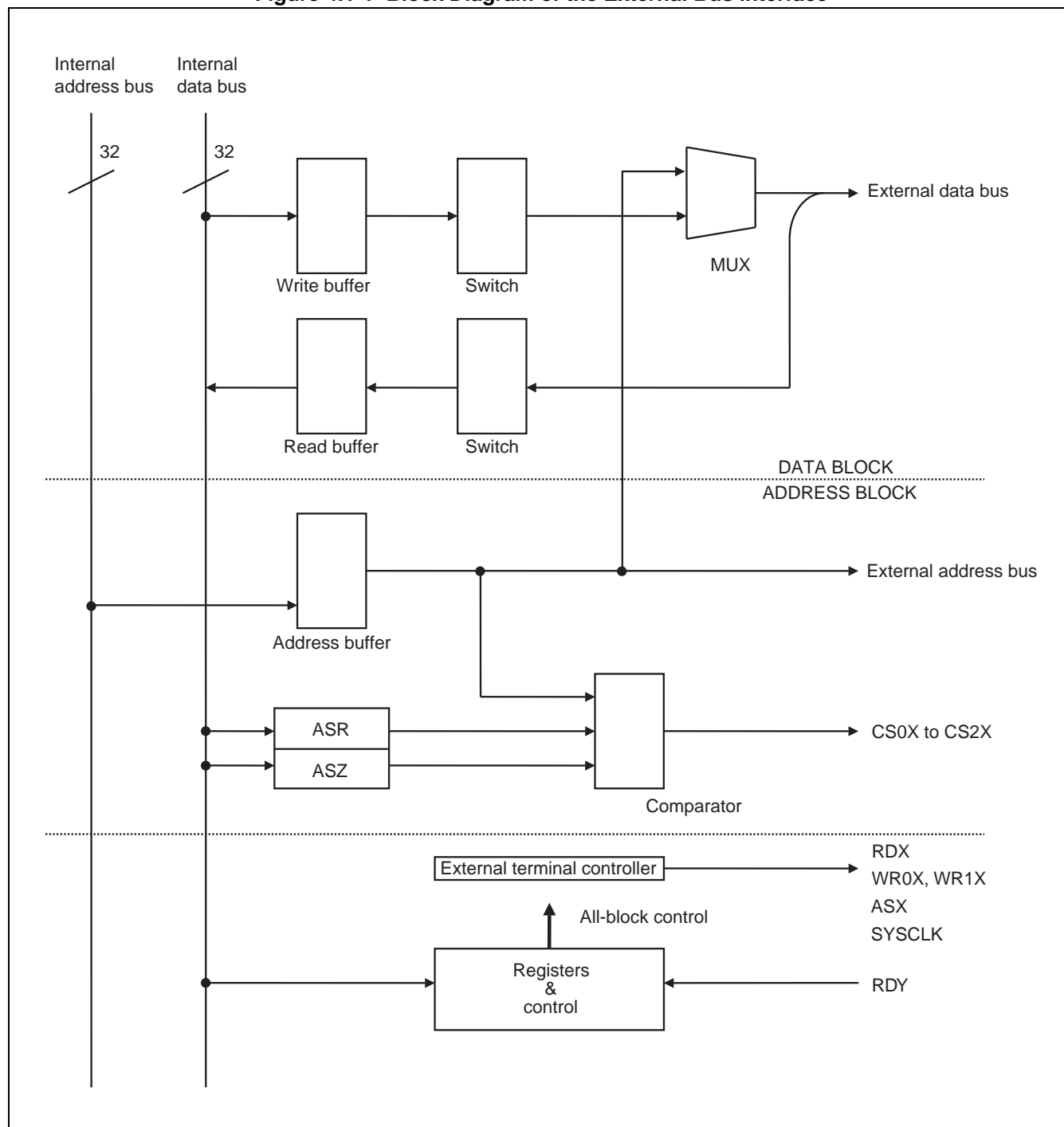
Only big endian can be set for the CS0X area.

- Setting of write disable (read only area)
- A different detailed timing can be set for each access timing type.
 - For the same type of access timing, a different setting can be made in each chip select area.
 - Auto-wait can be set to up to 15 cycles (asynchronous SRAM, ROM, Flash, and I/O area).
 - The bus cycle can be extended by external RDY input (asynchronous SRAM, ROM, Flash, and I/O area).
 - Various kinds of idle/recovery cycles and setting delays can be inserted.

■ Block Diagram of the External Bus Interface

Figure 4.1-1 shows the block diagram of the external bus interface.

Figure 4.1-1 Block Diagram of the External Bus Interface



■ I/O Pins

The I/O pins are external bus interface pins.

[Ordinary bus interface]

A15 to A00, D31 to D16 (A15 to A00)

CS0X, CS1X, CS2X

SYSCLK

ASX

RDX

WR0X, WR1X,

RDY

■ List of External Bus Interface Registers

The configuration of the external bus interface registers is shown below:

Address	bit31	bit24	bit23	bit16	bit15	bit8	bit7	bit0
00000640 _H	ASR0				ACR0			
00000644 _H	ASR1				ACR1			
00000648 _H	ASR2				ACR2			
0000064C _H	Reserved				Reserved			
00000650 _H	Reserved				Reserved			
00000654 _H	Reserved				Reserved			
00000658 _H	Reserved				Reserved			
0000065C _H	Reserved				Reserved			
00000660 _H	AWR0				AWR1			
00000664 _H	AWR2				Reserved			
00000668 _H	Reserved				Reserved			
0000066C _H	Reserved				Reserved			
00000670 _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
00000674 _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
00000678 _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
0000067C _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
00000680 _H	CSER	Reserved		Reserved	Reserved	Reserved	Reserved	
00000684 _H	Reserved				Reserved	Reserved		
00000688 _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
0000068C _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
000007F8 _H	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved	
000007FC _H	Reserved	(MODR)		Reserved	Reserved	Reserved	Reserved	

Reserved indicates a reserved register. For a rewrite, be sure to set 0.

MODR cannot be accessed from user programs.

4.2 External Bus Interface Registers

This section describes the registers of the external bus interface.

■ Register Overview of External Bus Interface

The following registers are used by the external bus interface:

- Area Select Registers (ASR0 to ASR2)
- Area Configuration Registers (ACR0 to ACR2)
- Area Wait Registers (AWR0 to AWR2)
- Chip Select Enable Register (CSER)

4.2.1 Area Select Registers (ASR0 to ASR2)

This section describes the area select registers in detail.

■ Configuration of Area Select Registers (ASR0 to ASR2)

The configuration of ASR0 to ASR2 is shown below:

									Initial value	
ASR0	bit15	bit14	bit13	bit12	...	bit2	bit1	bit0	At INIT	At RST
00000640 _H	A31	A30	A29	A18	A17	16	0000 _H	0000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ASR1	bit15	bit14	bit13	bit12	...	bit2	bit1	bit0		
00000644 _H	A31	A30	A29	A18	A17	16	XXXX _H	XXXX _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ASR2	bit15	bit14	bit13	bit12	...	bit2	bit1	bit0		
00000648 _H	A31	A30	A29	A18	A17	16	XXXX _H	XXXX _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

R/W: Readable/writable

The area select registers (ASR0 to ASR2) specify the start address of each chip select area for CS0X to CS2X.

The start address can be specified in the 16 high-order bits from A[31:16]. Each chip select area starts with the address set in these registers and covers the range set by bit ASZ[3:0] of registers ACR0 to ACR2.

The boundary of each chip select area obeys the setting for bit ASZ[3:0] of registers ACR0 to ACR2. For example, if an area of 1M byte is set by bit ASZ[3:0], the four low-order bits of registers ASR0 to ASR2 are ignored and only bit A[31:20] is valid.

The ASR0 register is initialized to "0000_H" by INIT and RST. ASR1 to ASR2 are not initialized by INIT and RST, and are therefore undefined. After starting LSI operation, be sure to set the corresponding ASR register before enabling each chip select area with the CSER register.

MB91470/480 Series**4.2.2 Area Configuration Registers (ACR0 to ACR2)**

This section describes the area configuration registers in detail.

■ Configuration of Area Configuration Registers (ACR0 to ACR2)

The configuration of ACR0 to ACR2 is shown below:

									Initial value	
									At INIT	At RST
ACR0H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8		
00000642 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	-	-	1111xx-- _B	1111**-- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ACR0L	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		
00000643 _H	-	-	WREN	0	TYP3	TYP2	TYP1	TYP0	--000000 _B	--000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ACR1H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8		
00000646 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	-	-	xxxxxx-- _B	xxxxxx-- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ACR1L	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		
00000647 _H	-	-	WREN	LEND	TYP3	TYP2	TYP1	TYP0	--xxxxxx _B	--xxxxxx _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ACR2H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8		
0000064A _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	-	-	xxxxxx-- _B	xxxxxx-- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ACR2L	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		
0000064B _H	-	-	WREN	LEND	TYP3	TYP2	TYP1	TYP0	--xxxxxx _B	--xxxxxx _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

R/W: Readable/writable

ACR0 to ACR2 (area configuration registers 0 to 2) are used to set the functions of each chip select area.

[bit15 to bit12] ASZ3 to ASZ0 (Area Size Bits [3:0])

These bits set the size of each chip select area as indicated in Table 4.2-1.

Table 4.2-1 Size of Each Chip Select Area as Set with the Area Size Bits

ASZ3	ASZ2	ASZ1	ASZ0	Size of each chip select area
0	0	0	0	64 Kbytes (00010000 _H byte, ASR A[31:16] bits are valid)
0	0	0	1	128 Kbytes (00020000 _H byte, ASR A[31:17] bits are valid)
0	0	1	0	256 Kbytes (00040000 _H byte, ASR A[31:18] bits are valid)
0	0	1	1	512 Kbytes (00080000 _H byte, ASR A[31:19] bits are valid)
0	1	0	0	1M byte (00100000 _H byte, ASR A[31:20] bits are valid)
0	1	0	1	2M bytes (00200000 _H byte, ASR A[31:21] bits are valid)
0	1	1	0	4M bytes (00400000 _H byte, ASR A[31:22] bits are valid)
0	1	1	1	8M bytes (00800000 _H byte, ASR A[31:23] bits are valid)
1	0	0	0	16M bytes (01000000 _H byte, ASR A[31:24] bits are valid)
1	0	0	1	32M bytes (02000000 _H byte, ASR A[31:25] bits are valid)
1	0	1	0	64M bytes (04000000 _H byte, ASR A[31:26] bits are valid)
1	0	1	1	128M bytes (08000000 _H byte, ASR A[31:27] bits are valid)
1	1	0	0	256M bytes (10000000 _H byte, ASR A[31:28] bits are valid)
1	1	0	1	512M bytes (20000000 _H byte, ASR A[31:29] bits are valid)
1	1	1	0	1024M bytes (40000000 _H byte, ASR A[31:30] bits are valid)
1	1	1	1	2048M bytes (80000000 _H byte, ASR A[31] bit is valid)

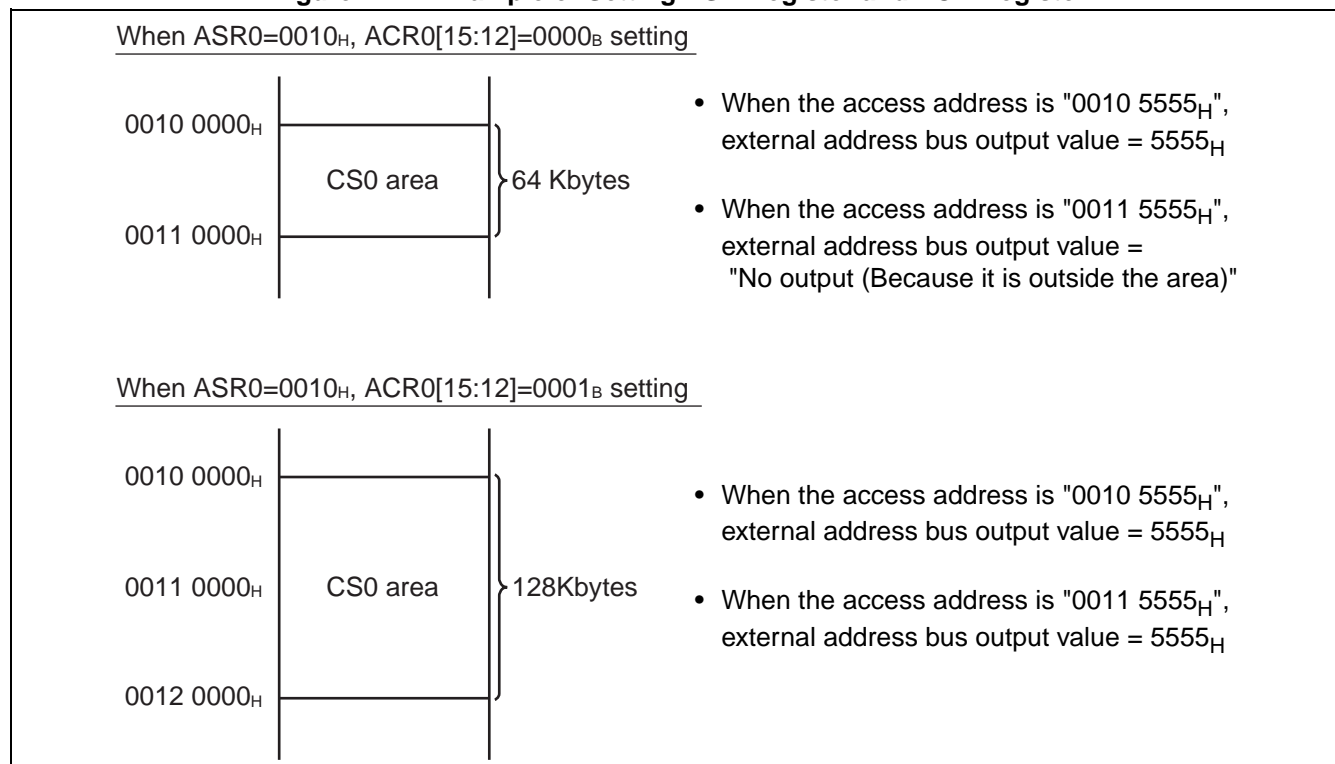
ASZ[3:0] is used to set the size of each area by modifying the number of bits for address comparison to a value different from ASR. Thus, an ASR contains bits that are not compared.

Bit ASZ[3:0] of ACR0 is initialized to "1111_B" (0F_H) by RST. Despite this setting, however, the CS0 area just after RST is executed is specially set from 00000000_H to FFFFFFFF_H (setting of entire area). The entire-area setting is reset after the first write to ACR0 and an appropriate size is set as indicated in Table 4.2-1.

Note:

Because the external address bus width is 16 bits in this device, the setting to be ASZ [3:0] \geq "0001_B" doesn't have the meaning. When the setting to be ASZ [3:0] \geq "0001_B" is executed, it becomes mirror arrangement shown in Figure 4.2-1.

Figure 4.2-1 Example of Setting ASR Register and ACR Register



[bit11, bit10] DBW1, DBW0 (Data Bus Width [1:0])

These bits set the data bus width of each chip select area as indicated in the following table:

DBW1	DBW0	Data bus width
0	0	8 bits (byte access)
0	1	16 bits (halfword access)
1	0	Setting disabled
1	1	Setting disabled

The same values as those of the WTH bits of the mode vector are written automatically to bit DBW[1:0] of ACR0 during the reset sequence.

[bit9 to bit6] Reserved: Reserved bits

Be sure to set "0000_B".

[bit5] WREN (WRite ENable)

This bit enables or disables writing to each chip select area.

WREN	Write enable/disable
0	Writing is disabled
1	Writing is enabled

If an area for which write operations are disabled is accessed for a write operation from the internal bus, the access is ignored and no external access at all is performed.

Set the WREN bit of areas for which write operations are required, such as data areas, to 1.

[bit4] LEND (Little ENDian select)

This bit sets the byte ordering of each chip select area.

LEND	Byte ordering
0	Big endian
1	Little endian

Be sure to set the LEND bit of ACR0 to 0. CS0 area supports only the big endian method.

[bit3 to bit0] TYP3 to TYP0 (TYPE select)

These bits set the access type of each chip select area as indicated in the following table:

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM and I/O)
	1	x	x	Address data multiplex access (8/16-bit bus width only)
	x	x	0	Disable WAIT insertion by the RDY pin.
		x	1	Enable WAIT insertion by the RDY pin.
		0	x	Setting disabled
		1	x	Setting disabled
1	0	0	0	Setting disabled
			1	Setting disabled
	0	1	0	Setting disabled
	0	1	1	Setting disabled
	1	0	0	Setting disabled
	1	0	1	Setting disabled
	1	1	0	Setting disabled
	1	1	1	Mask area setting (The access type is the same as that of the overlapping area) *

Set the access type as the combination of all bits.

*: CS area mask setting function

If you want to set an area some of whose operation settings are changed for a certain CS area (referred to as the base setting area), you can set TYP3 to TYP0 of ACR in another CS area to 1111_B so that the area can function as a mask setting area.

If you do not use the mask setting function, disable any overlapping area settings for multiple CS areas.

Access operations to the mask setting area are as follows:

- CSX corresponding to a mask setting area is not asserted.
- CSX corresponding to a base setting area is not asserted.
- For the following ACR settings, the settings on the mask setting area side are valid:
 - bit11, bit10: DBW[1:0]: Bus width setting
 - bit5: WREN: Write-enable setting
Note: For this setting only, a setting that is different from the base setting area is disabled.
 - bit4: LEND: Little endian setting
- For the following ACR setting, the setting on the base setting area side is valid:
 - bit3 to bit0: TYP[3:0]: Access type setting
- For the AWR settings, the settings on the mask setting area side are valid.

A mask setting area can be set for only part of another CS area (base setting area). You cannot set a mask setting area for an area without a base setting area. In addition, the mask setting area must not be duplicated. Be careful when setting the ASR and ACR:ASZ[3:0] bits.

Note:

The following restrictions apply to bits [3:0] TYP[3:0]:

- A write-enable setting cannot be implemented by a mask.
 - Write-enable settings in the base CS area and the mask setting area must be identical.
 - If write operations to a mask setting area are disabled, the area is not masked and operates as a base CS area.
 - If write operations to the base CS area are disabled but are enabled to the mask setting area, the area has no base, resulting in malfunctions.
-

MB91470/480 Series**4.2.3 Area Wait Registers (AWR0 to AWR2)**

This section describes the area wait registers in detail.

■ Configuration of Area Wait Registers (AWR0 to AWR2)

The configuration of AWR0 to AWR2 is shown below:

AWR0H								Initial value		
Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	At INIT	At RST
00000660 _H	W15	W14	W13	W12	-	-	-	-	0111---- _B	0111---- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
AWR0L										
Address	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16		
00000661 _H	W07	W06	W05	W04	-	W02	W01	W00	1111-111 _B	1111-011 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
AWR1H										
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8		
00000662 _H	W15	W14	W13	W12	-	-	-	-	XXXX---- _B	XXXX---- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
AWR1L										
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0		
00000663 _H	W07	W06	W05	W04	-	W02	W01	W00	XXXX-XXX _B	XXXX-XXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
AWR2H										
Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24		
00000664 _H	W15	W14	W13	W12	-	-	-	-	XXXX---- _B	XXXX---- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
AWR2L										
Address	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16		
00000665 _H	W07	W06	W05	W04	-	W02	W01	W00	XXXX-XXX _B	XXXX-XXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
R/W: Readable/writable										

R/W: Readable/writable

AWR0 to AWR2 specify various kinds of waits for each chip select area.

The function of each bit changes according to the access type (bits TYP3 to TYP0) setting of the ACR0 to ACR2 registers.

■ Normal Access and Address/Data Multiplex Access

A chip select area specified using the following settings for the access type (bit TYP[3:0]) of registers ACR0 to ACR2 operates as an area for normal access or address/data multiplex access.

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM, I/O, and single/page)
0	1	x	x	Address data multiplex access (8/16-bit bus width only)

The following lists the functions of each AWR0 to AWR2 bit for a normal access or address/data multiplex access area. Since the initial values of registers other than AWR0 are undefined, set them to their initial values before enabling each area with the CSER register.

[bit15 to bit12] W15 to W12 (First Access Wait Cycle)

These bits set the number of auto-wait cycles to be inserted into the first access cycle of each cycle. Except for the burst access cycles, only this wait setting is used.

The initial value of the CS0 area is set to 7 (wait). The initial values of the other areas are undefined.

W15	W14	W13	W12	First access wait cycle
0	0	0	0	Auto-wait cycle 0
0	0	0	1	Auto-wait cycle 1
...				...
1	1	1	1	Auto-wait cycle 15

[bit11 to bit8] Reserved: Reserved bits

Be sure to set "1111_B".

[bit7, bit6] W07, W06 (Read -> Write Idle Cycle)

The read -> write idle cycle is set to prevent collision of read data and write data on the data bus when a write cycle follows a read cycle. During an idle cycle, all chip select signals are negated and the data terminals maintain the high impedance state. If a write cycle follows a read cycle or an access operation to another chip select area occurs after a read cycle, the specified idle cycle is inserted.

W07	W06	Read -> write idle cycles
0	0	0 cycle
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

[bit5, bit4] W05, W04 (Write Recovery Cycle)

The write recovery cycle is set if a device that limits the access period after write access is to be controlled. During a write recovery cycle, all chip select signals are negated and the data pins maintain the high impedance state.

If the write recovery cycle is set to 1 or more, a write recovery cycle is always inserted after write access.

W05	W04	Write recovery cycles
0	0	0 cycle
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

[bit3] Reserved: Reserved bit

Be sure to set "1".

[bit2] W02 (Address -> CSX Delay)

The address -> CSX delay setting is made when a certain type of setup is required for the address when CSX falls or CSX edges are needed for successive accesses to the same chip select area.

Set the address and set the delay from ASX output to CS0X to CS2X output.

W02	Address -> CSX delay
0	No delay
1	Delay

If no delay is selected by setting 0, assertion of CS0X to CS2X starts at the same timing that ASX is asserted. If, at this point, successive accesses are made to the same chip select area, assertion of CS0X to CS2X without change between two access operations may continue.

If delay is specified by selecting 1, assertion of CS0X to CS2X starts when the external memory clock SYSCLK output rises. If, at this point, successive accesses are made to the same chip select area, CS0X to CS2X are negated at a timing between two access operations.

If CSX delay is selected, one setup cycle is inserted before asserting the read/write strobe after assertion of the delayed CSX (operation is the same as the CSX -> RDX/WR0X, WR1X setup setting of W01).

[bit1] W01 (CSX -> RDX/WR0X, WR1X Setup Extension Cycle: CSX -> RDX/WR0X, WR1X setup)

The CSX -> RDX/WR0X, WR1X setup extension cycle is set to extend the period before the read/write strobe is asserted after CSX is asserted. At least one setup extension cycle is inserted before the read/write strobe is asserted after CSX is asserted.

W01	CSX -> RDX/WR0X, WR1X setup extension cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting 0, RDX/WR0X, WR1X are output at the earliest when external memory clock SYSCLK output rises just after CSX is asserted. RDX/WR0X, WR1X may be delayed one cycle or more depending on the internal bus state.

If 1 cycle is selected by setting 1, RDX/WR0X, WR1X are always output 1 cycle or more later.

When successive accesses are made within the same chip select area without negating CSX, a setup extension cycle is not inserted. If a setup extension cycle for determining the address is required, set the W02 bit and insert the address -> CSX delay. Since CSX is negated for each access operation, the setup extension cycle is enabled.

If the CSX delay set by W02 is inserted, this setup cycle is always enabled regardless of the setting of the W01 bit.

[bit0] W00 (RDX/WR0X, WR1X -> CSX Hold Extension Cycle: RDX/WR0X, WR1X -> CSX Hold Cycle)

The RDX/WR0X, WR1X -> CSX hold extension cycle is set to extend the period before negating CSX after the read/write strobe is negated. One hold extension cycle is inserted before CSX is negated after the read/write strobe is negated.

W00	RDX/WR0X, WR1X -> CS hold extension cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting 0, CS0X to CS2X are negated when the hold delay has elapsed on the rising edge of external memory clock SYSCLK output after RDX/WR0X, WR1X are negated.

If 1 cycle is selected by setting 1, CS0X to CS2X are negated one cycle later.

When making successive accesses within the same chip select area without negating CSX, the hold extension cycle is not inserted. If a hold extension cycle for determining the address is required, set the W02 bit and insert the address -> CSX delay. Since CSX is negated for each access operation, this hold extension cycle is enabled.

MB91470/480 Series**4.2.4 Chip Select Enable Register (CSER)**

This section describes the chip select enable register in detail.

■ Configuration of the Chip Select Enable Register (CSER)

The configuration of CSER is shown below:

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value	
	At INIT	At RST								
00000680 _H	-	-	-	-	-	CSE2	CSE1	CSE0	-----001 _B	-----001 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

R/W: Readable/writable

This register enables or disables each chip select area.

[bit31 to bit27] Reserved: Reserved bits

Be sure to set "0000_B".

[bit26 to bit24] CSE[3:0] (Chip Select Area Enable: Chip select enable 0 to 3)

These bits are the chip select area enable bits for CS0X to CS2X.

The initial value is "00000001_B", which enables only the CS0 area.

When 1 is written, a chip select area operates according to the settings of ASR0 to ASR2, ACR0 to ACR2, and AWR0 to AWR2.

Before setting this register, be sure to make all settings required for the corresponding chip select areas.

CSE[2:0]	Area control
0	Disable
1	Enable

Table 4.2-2 lists the CSE bits and corresponding CSXs.

Table 4.2-2 CSE Bits and Corresponding CSXs

CSE bit	Corresponding CSX
bit24:CSE0	CS0X
bit25:CSE1	CS1X
bit26:CSE2	CS2X

4.3 Chip Select Area

In the external bus interface, a total of three chip select areas can be set.

The address space of each area can be placed, in units of a minimum of 64 Kbytes, anywhere in the 4 GB space using ASR0 to ASR2 (Area Select Registers) and ACR0 to ACR2 (Area Configuration Registers).

When bus access is made to an area specified by these registers, the corresponding chip select signals (CS0X to CS2X) are activated (L output) during the access cycle.

■ Example of Setting ASRs and ASZ[3:0]

1. ASR1=0005_H ACR1 --> ASZ[3:0]=0000_B: Chip select area 1 is assigned to 00050000_H to 0005FFFF_H.
2. ASR2=0011_H ACR2 --> ASZ[3:0]=0100_B: Chip select area 2 is assigned to 00100000_H to 001FFFFF_H.

Since at this point 1M byte is set ACR for bit ASZ[3:0], the unit for boundaries 1M byte and ASR2[19:16] is ignored.

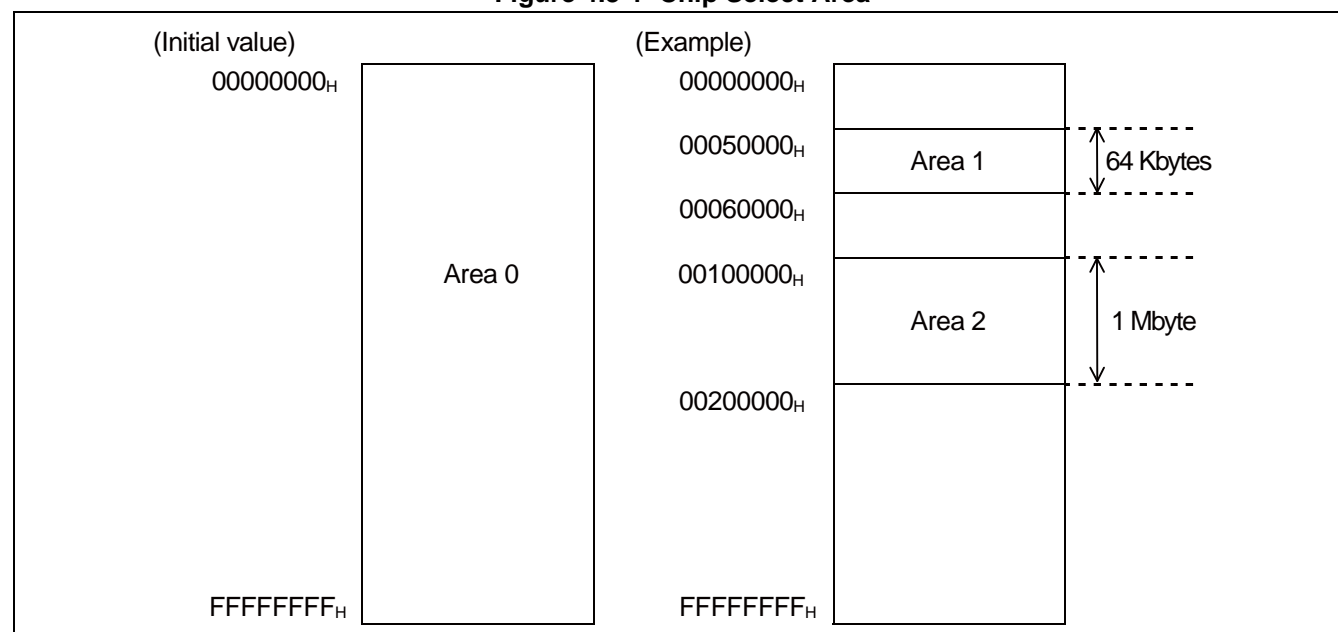
Before there is any writing to ACR0 after a reset, 00000000_H to FFFFFFFF_H is assigned to chip select area 0.

Note:

Set the chip select areas so that there is no overlap.

Figure 4.3-1 shows the chip select area.

Figure 4.3-1 Chip Select Area



4.4 Endian and Bus Access

This section describes endian and bus access.

■ Overview of Endian

Except for specific areas, FR family devices enable switching between the big endian and little endian methods for each chip select.

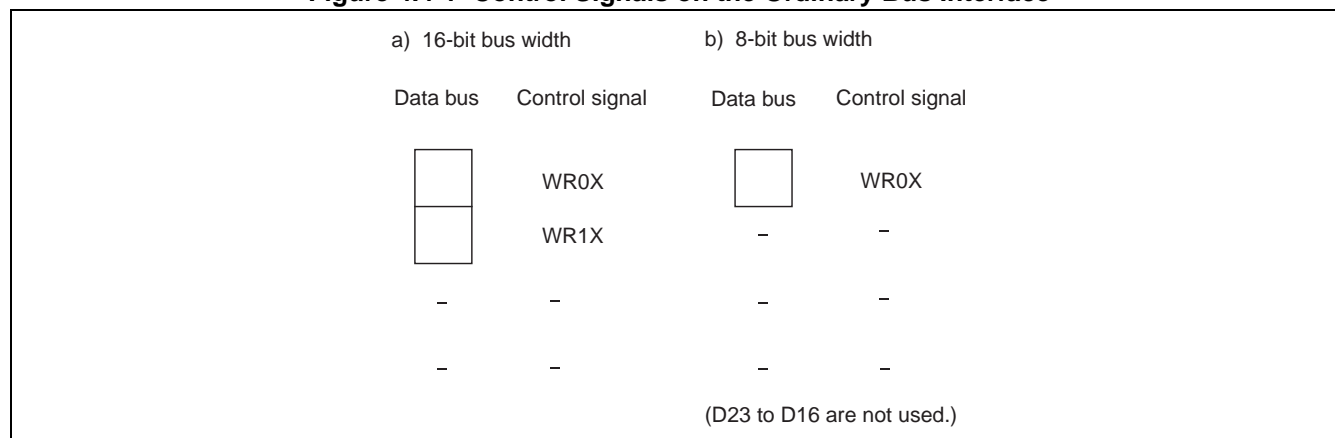
4.4.1 Relationship between Data Bus Widths and Control Signals

There is a one-to-one correspondence between the WR[1:0]X control signals and the location of the bytes on the data bus regardless of the data bus width. This section summarizes the location of the bytes on the data bus used for the specified data bus width and the corresponding control signal for each bus mode.

■ Control Signals on the Ordinary Bus Interface

Figure 4.4-1 shows the 16-bit bus width and 8-bit bus width control signals on the ordinary bus interface.

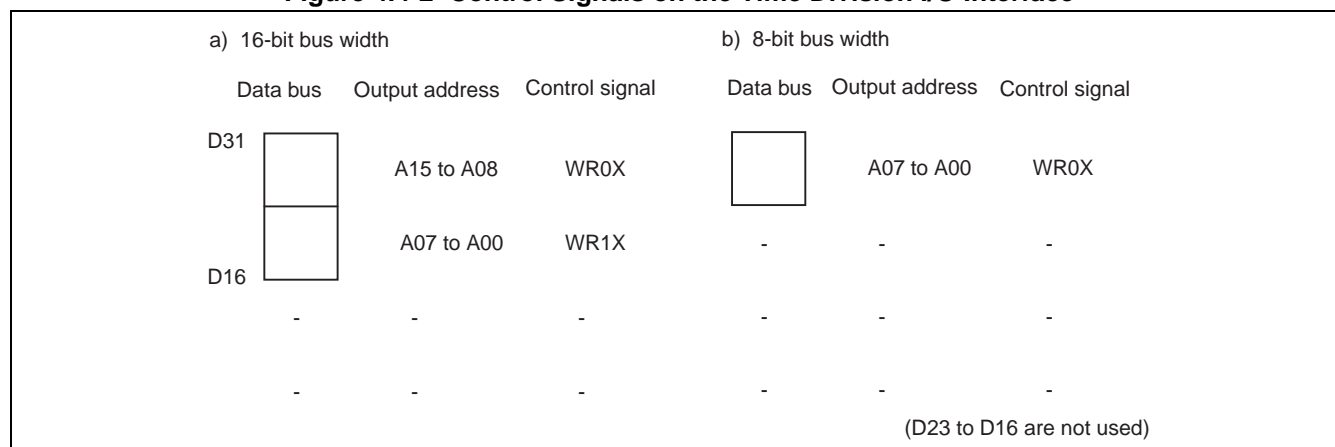
Figure 4.4-1 Control Signals on the Ordinary Bus Interface



■ Control Signals on the Time Division I/O Interface

Figure 4.4-2 shows the 16-bit bus width and 8-bit bus width control signals on the time division I/O interface.

Figure 4.4-2 Control Signals on the Time Division I/O Interface



MB91470/480 Series

4.4.2 Big Endian Bus Access

Except for the CS0 area, the FR family can switch between the big endian method and little endian method for each chip select area. When the LEND bit of the ACR register is set to 0, the chip select area is treated as big endian.

Normally, the FR family executes external bus access using big endian.

■ Big Endian Data Format

Figure 4.4-3 shows the relationship between the internal register and external data bus based on the data format of word access (when the LD and ST instructions are executed.)

Figure 4.4-3 Word Access (When LD and ST Instructions Executed)

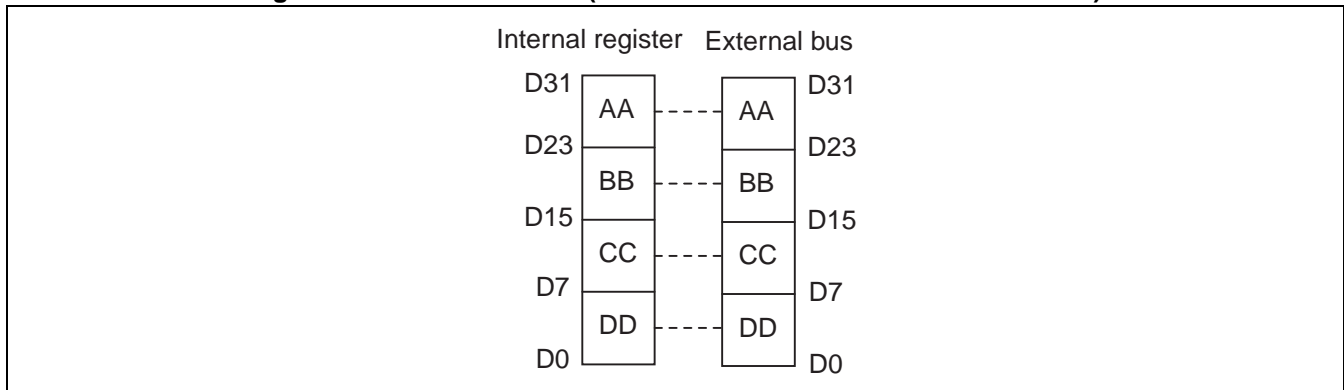


Figure 4.4-4 shows the relationship between the internal register and external data bus based on the data format of halfword access (when the LDUH and STH instructions are executed).

Figure 4.4-4 Halfword Access (When LDUH and STH Instructions Executed)

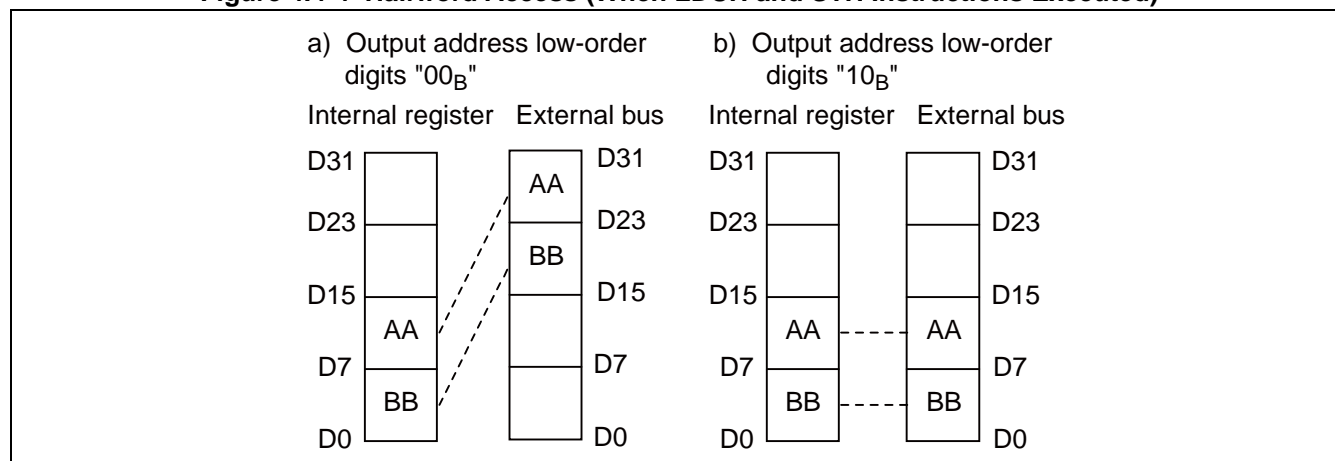
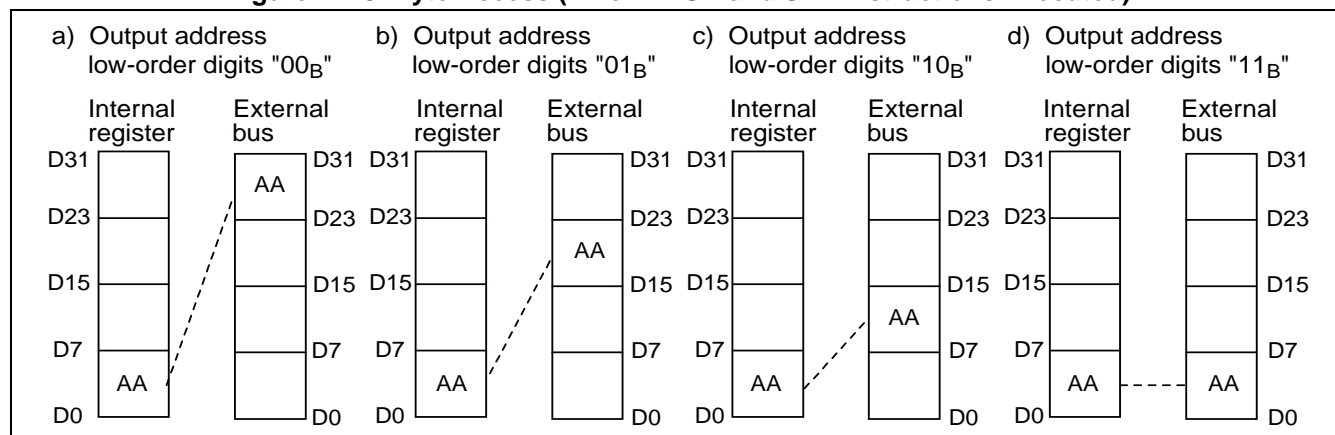


Figure 4.4-5 shows the relationship between the internal register and external data bus based on the data format of byte access (when the LDUB and STB instructions are executed).

Figure 4.4-5 Byte Access (When LDUB and STB Instructions Executed)



MB91470/480 Series

■ Data Bus Width of Big Endian

Figure 4.4-6 shows the data bus width for a bus width of 16 bits.

Figure 4.4-6 Data Bus Width for 16-bit Bus Width

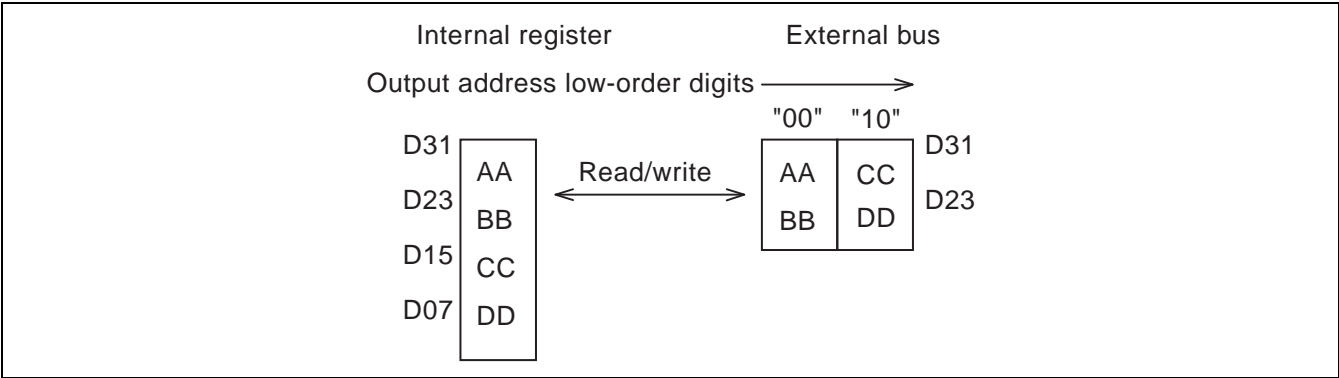
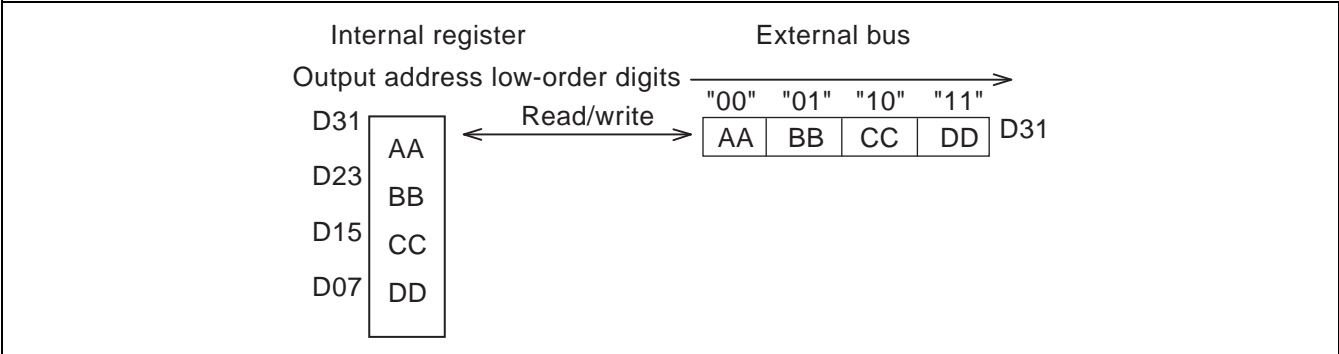


Figure 4.4-7 shows the data bus width for a bus width of 8 bits.

Figure 4.4-7 Data Bus Width for 8-bit Bus Width




External Bus Access



For big endian for external bus access, the following items are arranged as illustrated later for bus widths of 16 and 8 bits and for word, halfword, and byte access:

- Access byte location
- Program address and output address
- Bus access count

PA1/PA0 : Two low-order bits of the address specified by the program

Output A1/A0: Two low-order bits of the output address

 : Location of initial byte of the output address

 +  : Data byte location to be accessed

(1) to (4): Bus access count

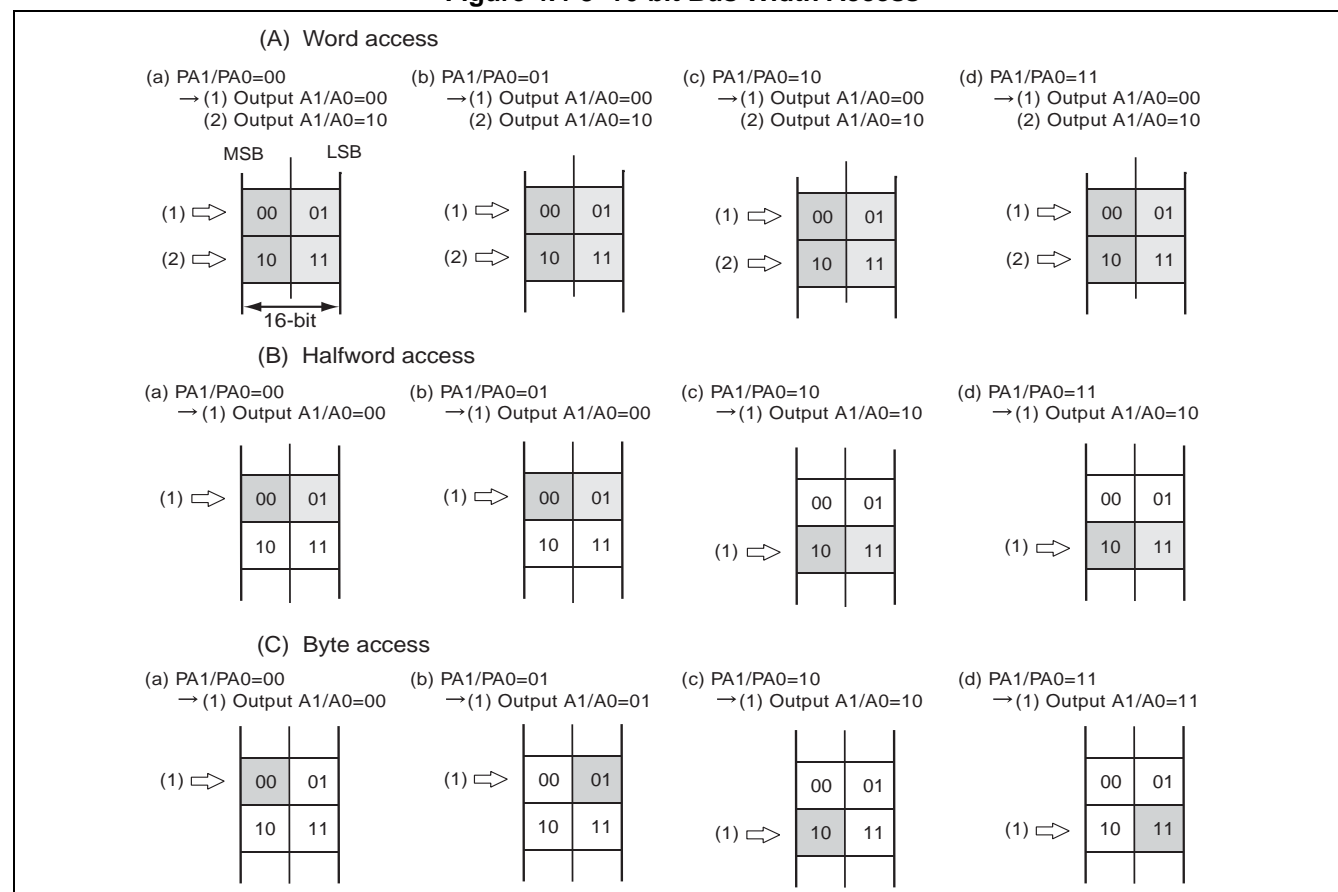
The FR family does not detect misalignment errors.

Therefore, for word access, the lower two bits of the output address are always 00 regardless of whether 00, 01, 10, or 11 is specified as the lower two bits by the program. For halfword access, the lower two bits of the output address are 00 if the lower two bits specified by the program are 00 or 01, and are 10 if 10 or 11.

16-bit bus width

Figure 4.4-8 shows the access operations for a bus width of 16 bits.

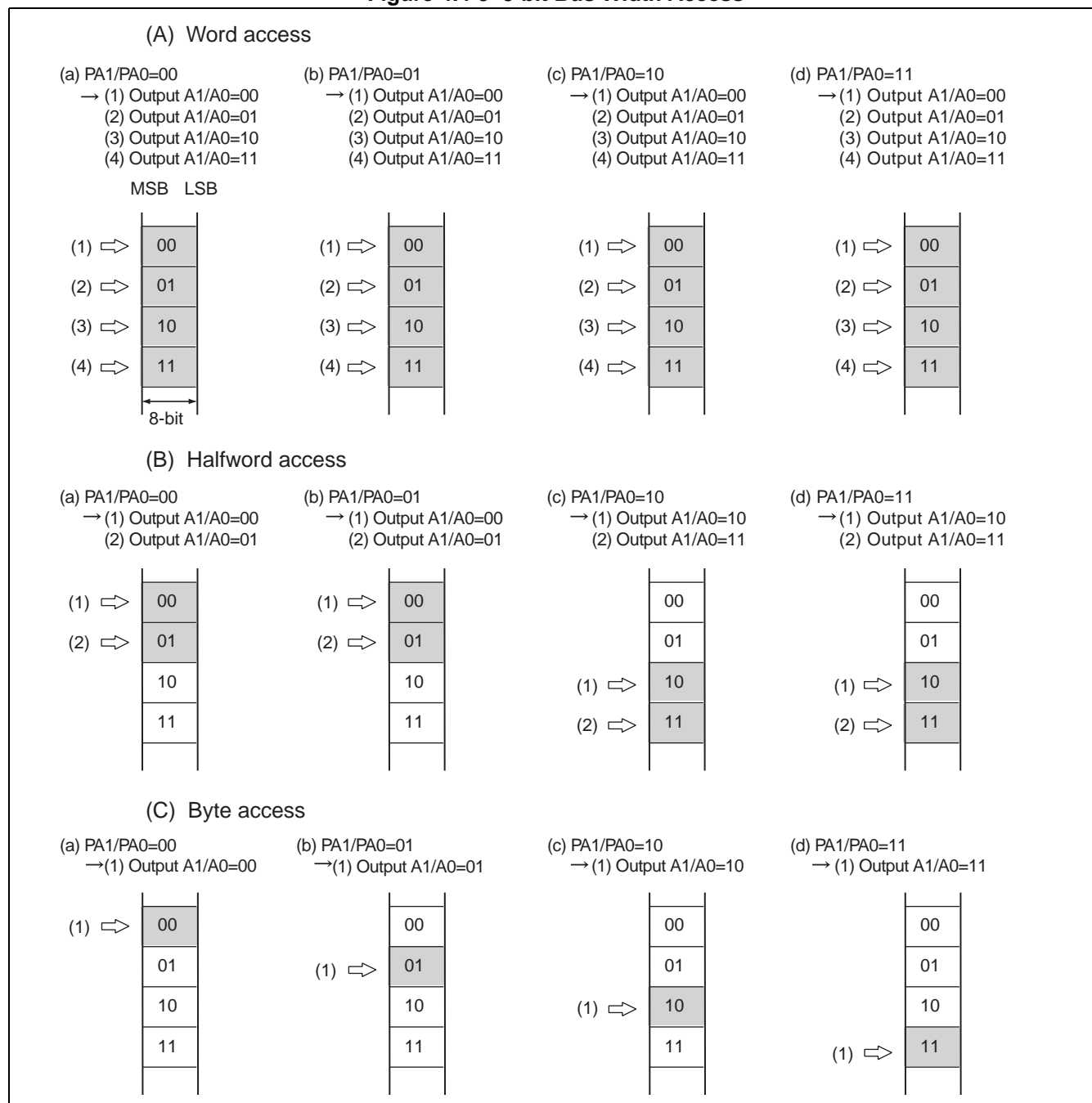
Figure 4.4-8 16-bit Bus Width Access



● 8-bit bus width

Figure 4.4-9 shows the access operations for bus width of 8 bits.

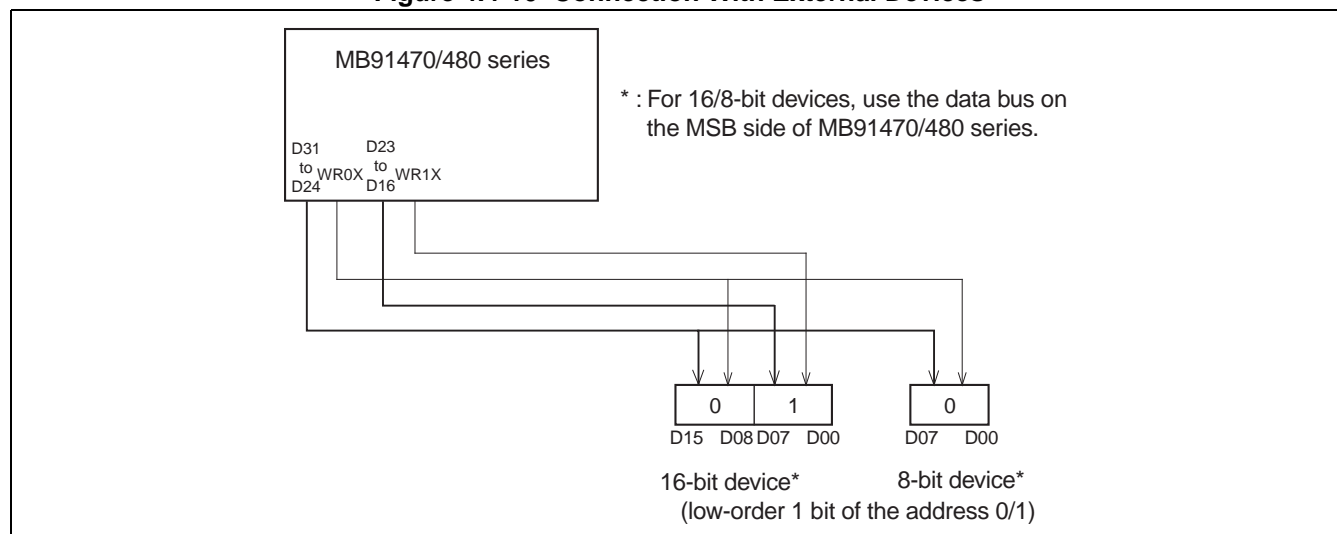
Figure 4.4-9 8-bit Bus Width Access



■ **Example of Connection with External Devices**

Figure 4.4-10 shows an example of connection between the LSI and external devices.

Figure 4.4-10 Connection With External Devices



MB91470/480 Series

4.4.3 Little Endian Bus Access

Except for the CS0 area, the FR family can switch between the big endian method and little endian method for each chip select area. When the LEND bit of the ACR register is set to 1, the chip select area is treated as little endian.

■ Overview of Little Endian Method

Little endian bus access of the FR family is implemented by using the bus access operation used for the big endian method. Basically, the order of output addresses and control signal output are the same as those for the big endian method and the locations of the bytes on the data bus are swapped in accordance with the bus width.

Note that, when a connection is made, the big endian area and the little endian area must be kept physically separate.

- The order of addresses that are output is the same for little endian and big endian.
- Word access

The byte data on the MSB side for big endian address $A[1:0]=00_H$ becomes byte data on the LSB side when the little endian method is used.

For a word access, the locations of all four bytes in the word are reversed.

- Halfword access

The byte data on the MSB side for the big endian address $A[0]$ becomes byte data on the LSB side when the little endian method is used.

For halfword access, the locations of two bytes in the halfword are reversed.

- Byte access

There is no difference between little endian and big endian.

- There is no difference between the little and big endian methods regarding the data bus and control signals used for bus widths of 16 and 8 bits.

[Restrictions on the Little Endian Area]

- Do not place any instruction code in a little endian area.

■ Little Endian Data Format

Figure 4.4-11 shows the relationship between the internal register and external data bus based on the data format of word access (when the LD and ST instructions are executed).

Figure 4.4-11 Word Access (When LD and ST Instructions Executed)

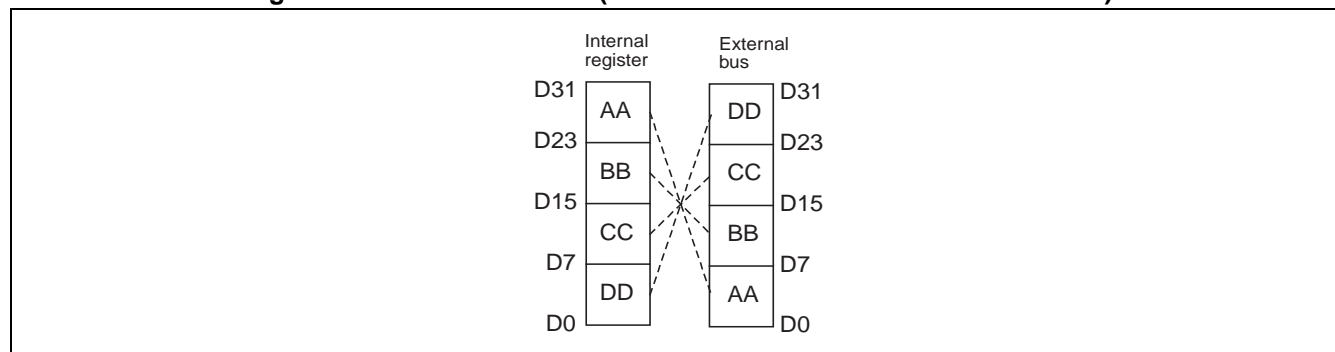


Figure 4.4-12 shows the relationship between the internal register and external data bus based on the data format of halfword access (when the LDUH and STH instructions are executed).

Figure 4.4-12 Halfword Access (When LDUH and STH Instructions Executed)

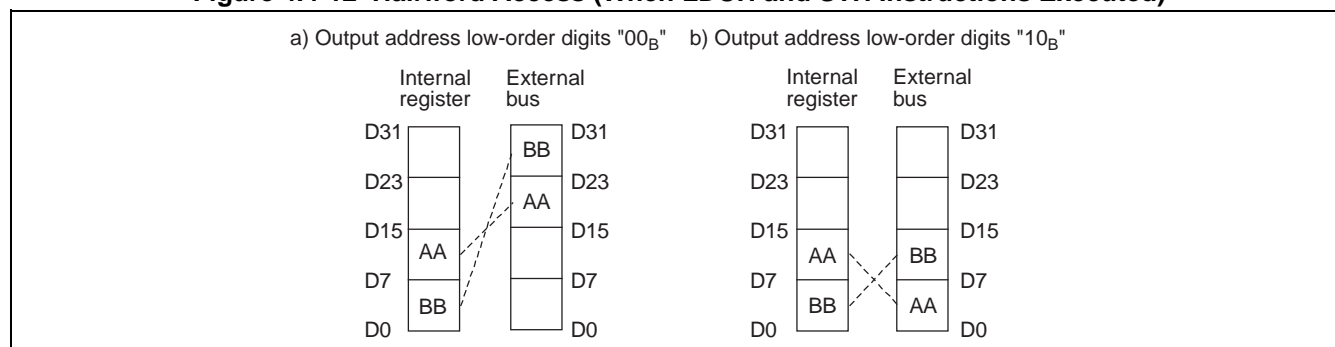
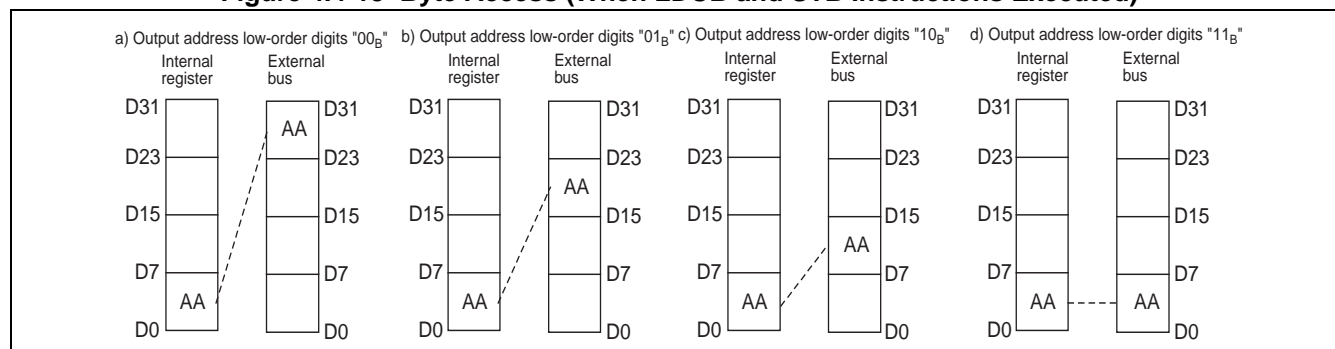


Figure 4.4-13 shows the relationship between the internal register and external data bus based on the data format of byte access (when the LDUB and STB instructions are executed).

Figure 4.4-13 Byte Access (When LDUB and STB Instructions Executed)



■ Data Bus Width of Little Endian

Figure 4.4-14 shows the data bus width for a bus width of 16 bits.

Figure 4.4-14 Data Bus Width for 16-bit Bus Width

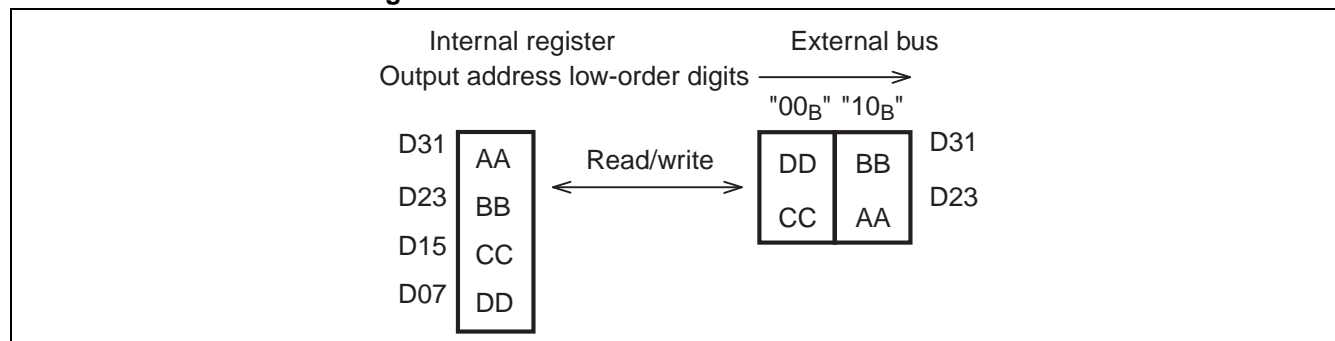
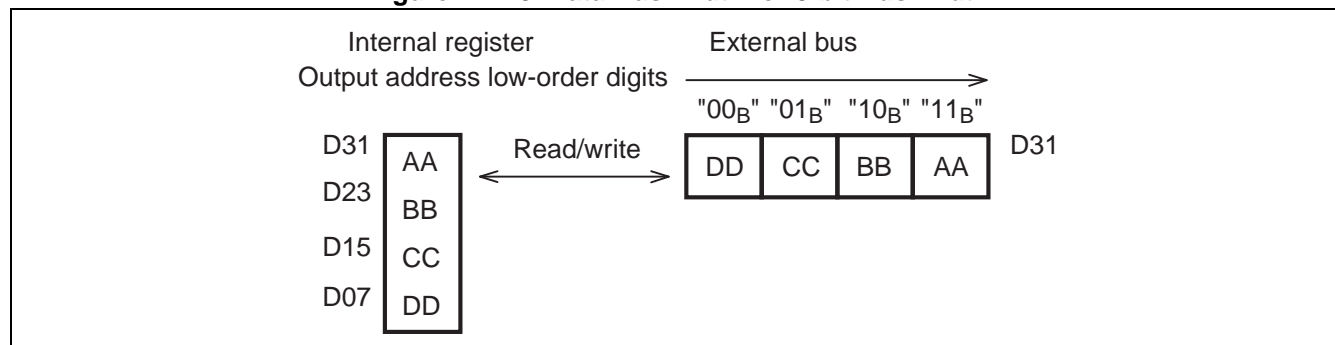


Figure 4.4-15 shows the data bus width for a bus width of 8 bits.

Figure 4.4-15 Data Bus Width for 8-bit Bus Width

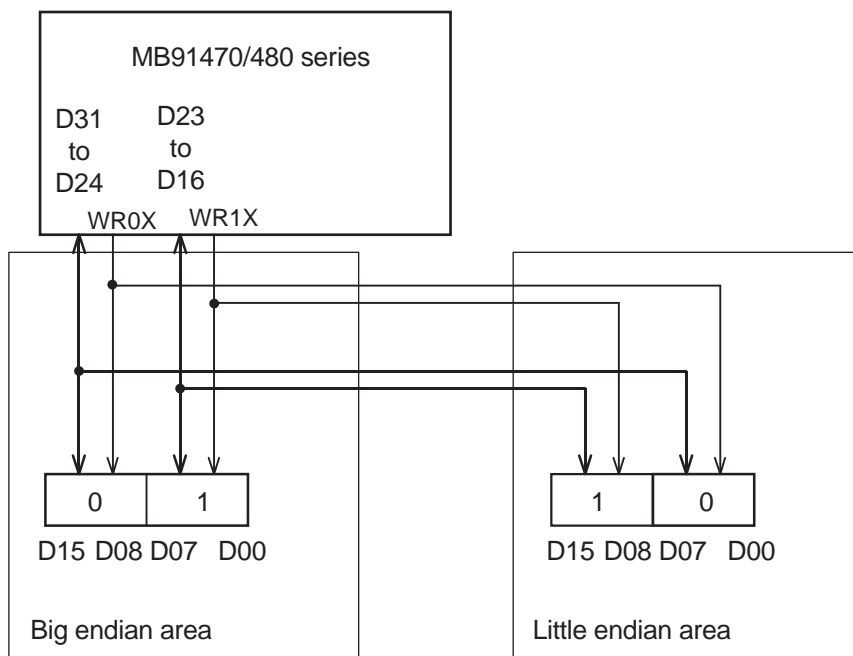


■ **Connection between MB91470/480 Series and the Endian Areas**

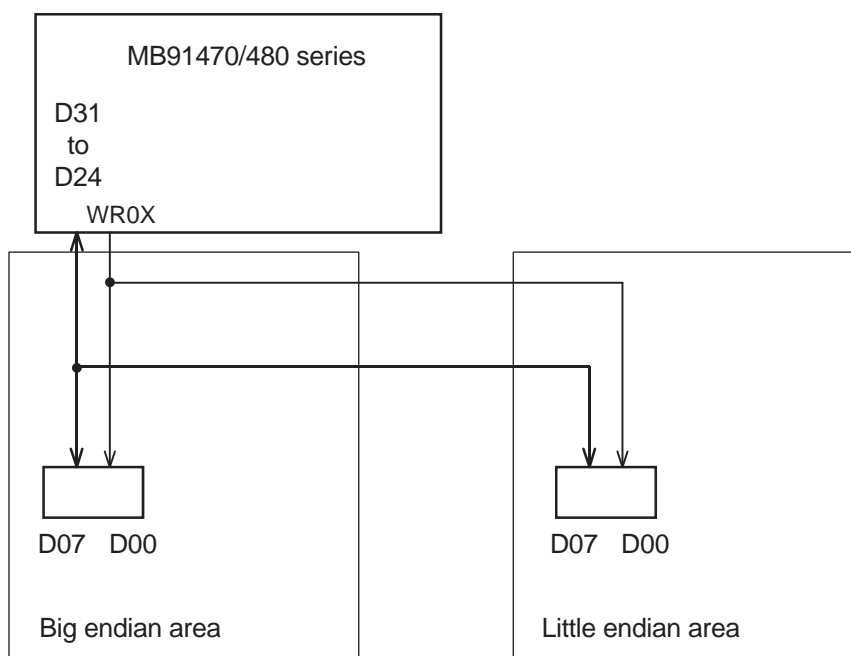
Figure 4.4-16 shows connection between MB91470/480 series and endian.

Figure 4.4-16 Connection between MB91470/480 Series and Endian

● 16-bit bus width



● 8-bit bus width



MB91470/480 Series

4.4.4 External Access

This section describes the relationship between the internal register and external data bus based on the endian mode and the bus width.

Word Access

External access using word access is shown below:

	Big endian mode	Little endian mode
16-bit bus width	<div><div>Internal register</div><div>External terminal</div><div>Control terminal</div><div>Address: "0" "2"</div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>WR0X</div><div>WR1X</div><div>—</div><div>—</div></div><div><div>(1)</div><div>(2)</div></div></div>	<div><div>Internal register</div><div>External terminal</div><div>Control terminal</div><div>Address: "0" "2"</div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>D31</div><div>DD</div><div>CC</div><div>BB</div><div>AA</div><div>D00</div></div><div><div>WR0X</div><div>WR1X</div><div>—</div><div>—</div></div><div><div>(1)</div><div>(2)</div></div></div>
8-bit bus width	<div><div>Internal register</div><div>External terminal</div><div>Control terminal</div><div>Address: "0" "1" "2" "3"</div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>WR0X</div><div>—</div><div>—</div><div>—</div></div><div><div>(1)</div><div>(2)</div><div>(3)</div><div>(4)</div></div></div>	<div><div>Internal register</div><div>External terminal</div><div>Control terminal</div><div>Address: "0" "1" "2" "3"</div><div><div>D31</div><div>AA</div><div>BB</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>D31</div><div>DD</div><div>CC</div><div>BB</div><div>AA</div><div>D00</div></div><div><div>WR0X</div><div>—</div><div>—</div><div>—</div></div><div><div>(1)</div><div>(2)</div><div>(3)</div><div>(4)</div></div></div>

■ **Halfword Access**

External access using halfword access is shown below:

	Big endian mode	Little endian mode
16-bit bus width	<p>Internal register External terminal Control terminal</p> <p>Address: "0"</p> <p>(1)</p>	<p>Internal register External terminal Control terminal</p> <p>Address: "0"</p> <p>(1)</p>
	<p>Internal register External terminal Control terminal</p> <p>Address: "2"</p> <p>(1)</p>	<p>Internal register External terminal Control terminal</p> <p>Address: "2"</p> <p>(1)</p>
8-bit bus width	<p>Internal register External terminal Control terminal</p> <p>Address: "0" "1"</p> <p>(1) (2)</p>	<p>Internal register External terminal Control terminal</p> <p>Address: "0" "1"</p> <p>(1) (2)</p>
	<p>Internal register External terminal Control terminal</p> <p>Address: "2" "3"</p> <p>(1) (2)</p>	<p>Internal register External terminal Control terminal</p> <p>Address: "2" "3"</p> <p>(1) (2)</p>

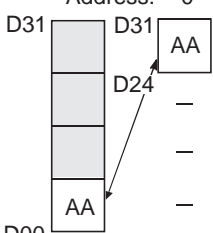
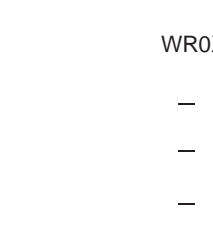
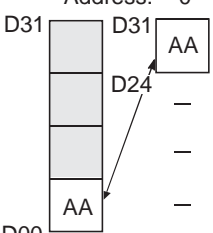
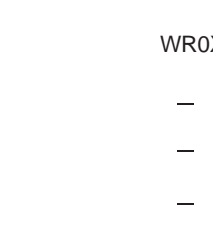
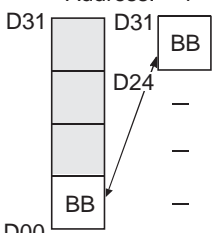
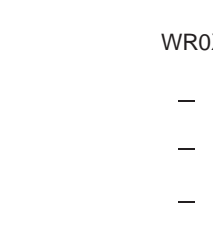
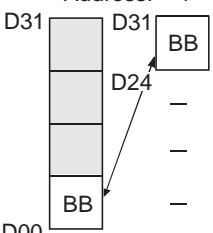
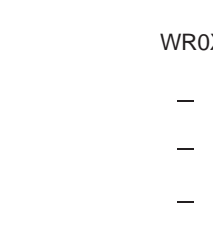
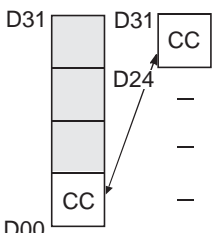
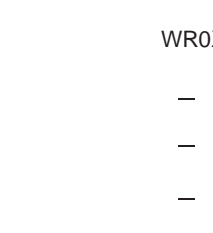
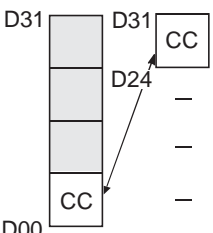
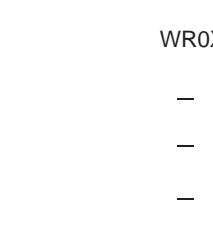
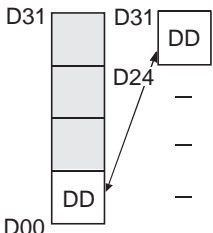
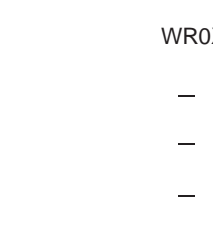
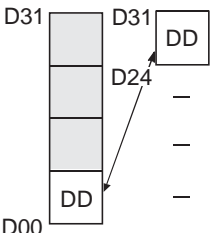
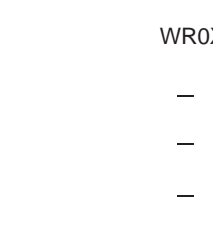
■ Byte Access

External access using byte access is shown below:

	Big endian mode	Little endian mode
16-bit bus width	<p>Internal register Address: "0"</p> <p>D31 D00 AA</p> <p>(1)</p>	<p>Internal register Address: "0"</p> <p>D31 D00 AA</p> <p>(1)</p>
	<p>Internal register Address: "1"</p> <p>D31 D00 BB</p> <p>(1)</p>	<p>Internal register Address: "1"</p> <p>D31 D00 BB</p> <p>(1)</p>
	<p>Internal register Address: "2"</p> <p>D31 D00 CC</p> <p>(1)</p>	<p>Internal register Address: "2"</p> <p>D31 D00 CC</p> <p>(1)</p>
	<p>Internal register Address: "3"</p> <p>D31 D00 DD</p> <p>(1)</p>	<p>Internal register Address: "3"</p> <p>D31 D00 DD</p> <p>(1)</p>

(Continued)

(Continued)

	Big endian mode			Little endian mode		
8-bit bus width	Internal register Address: "0"  D31 D24 D00 AA (1)	External terminal Address: "0"  D31 D24 D00 AA (1)	Control terminal Address: "0" WR0X — — —	Internal register Address: "0"  D31 D24 D00 AA (1)	External terminal Address: "0"  D31 D24 D00 AA (1)	Control terminal Address: "0" WR0X — — —
	Internal register Address: "1"  D31 D24 D00 BB (1)	External terminal Address: "1"  D31 D24 D00 BB (1)	Control terminal Address: "1" WR0X — — —	Internal register Address: "1"  D31 D24 D00 BB (1)	External terminal Address: "1"  D31 D24 D00 BB (1)	Control terminal Address: "1" WR0X — — —
	Internal register Address: "2"  D31 D24 D00 CC (1)	External terminal Address: "2"  D31 D24 D00 CC (1)	Control terminal Address: "2" WR0X — — —	Internal register Address: "2"  D31 D24 D00 CC (1)	External terminal Address: "2"  D31 D24 D00 CC (1)	Control terminal Address: "2" WR0X — — —
	Internal register Address: "3"  D31 D24 D00 DD (1)	External terminal Address: "3"  D31 D24 D00 DD (1)	Control terminal Address: "3" WR0X — — —	Internal register Address: "3"  D31 D24 D00 DD (1)	External terminal Address: "3"  D31 D24 D00 DD (1)	Control terminal Address: "3" WR0X — — —

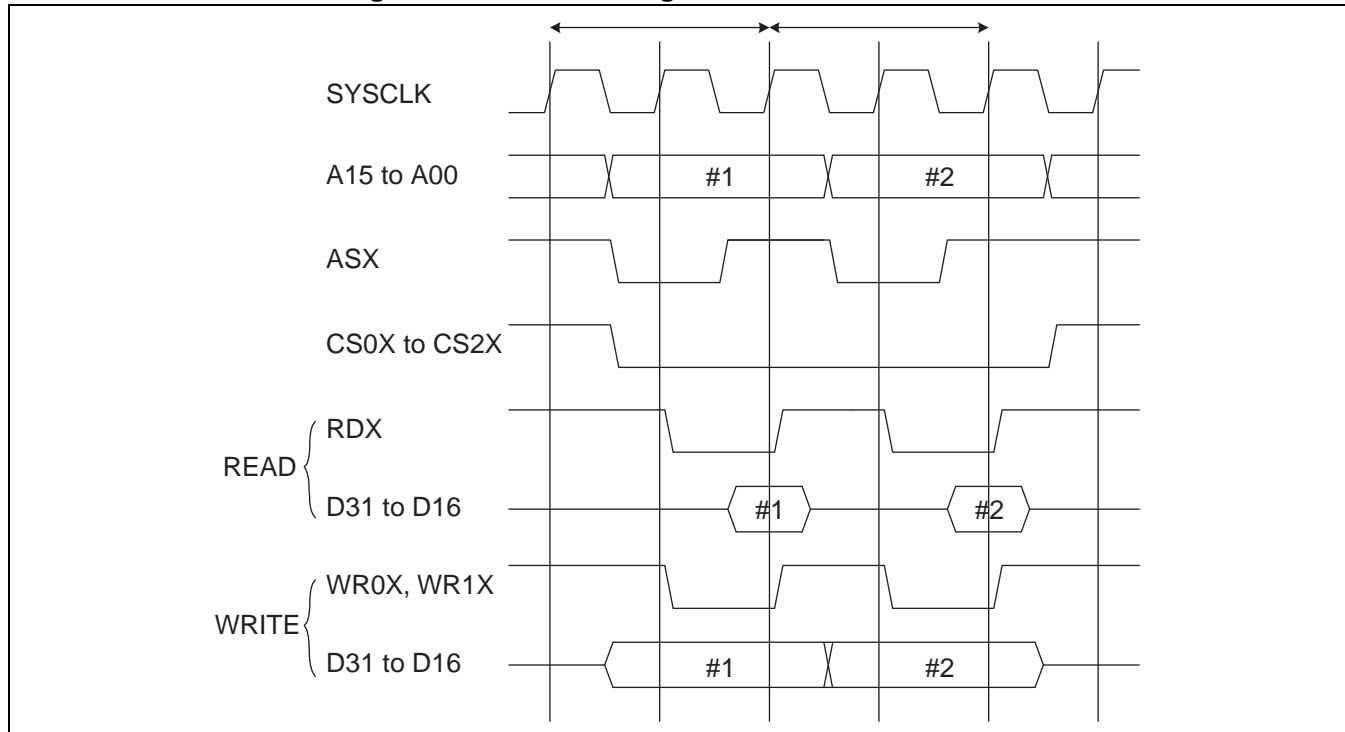
MB91470/480 Series**4.5 Ordinary Bus Interface**

For an ordinary bus interface, the two clock cycles required for both read access and write access become the basic bus cycle.

■ **Basic Timing (For Successive Accesses) (TYP[3:0]=0000_B, AWR=0008_H)**

Figure 4.5-1 shows the basic timing for successive accesses.

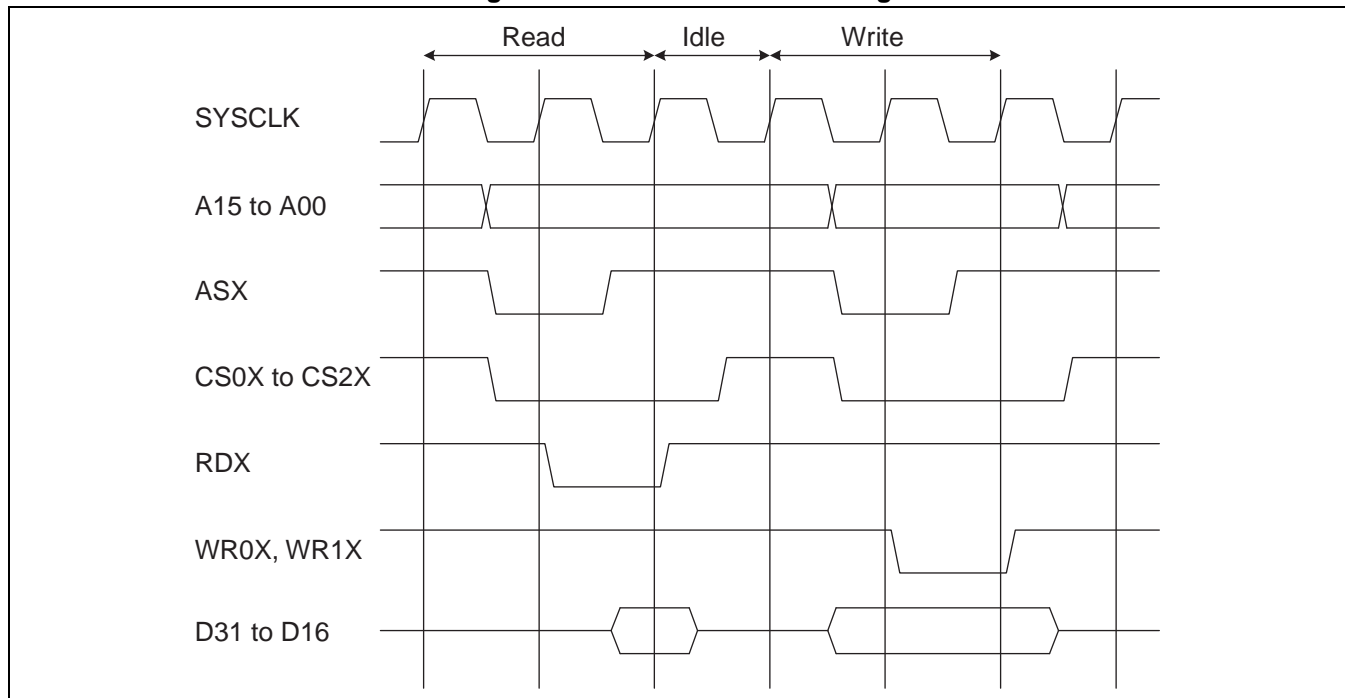
Figure 4.5-1 Basic Timing For Successive Accesses



- ASX is asserted for one cycle in the bus access start cycle.
- A[15:0] continues to output the address of the location of the start byte in word/halfword/byte access from the bus access start cycle to the bus access end cycle.
- If the W02 bit of the AWR[0:2] registers is 0, CS0X to CS2X are asserted at the same timing as ASX. For successive accesses, CS0X to CS2X are not negated. If the W00 bit of the AWR register is 0, CS0X to CS2X are negated after the bus cycle ends. If the W00 bit is 1, CS0X to CS2X are negated one cycle after bus access ends.
- RDX, WR0X, WR1X are asserted from the 2nd cycle of the bus access. Negation occurs after the wait cycle of bits W15 to W12 for the AWR register is inserted. The timing of asserting RDX, WR0X, WR1X can be delayed by one cycle by setting the W01 bit of the AWR register to 1.
- For read access, D[31:16] is read when SYSCLK rises in the cycle in which the wait cycle ended after RDX was asserted.
- For write access, data output to D[31:16] starts at the timing at which WR0X, WR1X are asserted.

■ Read -> Write Timing (TYP[3:0]=0000_B, AWR=0048_H)

Figure 4.5-2 shows the read -> write timing.

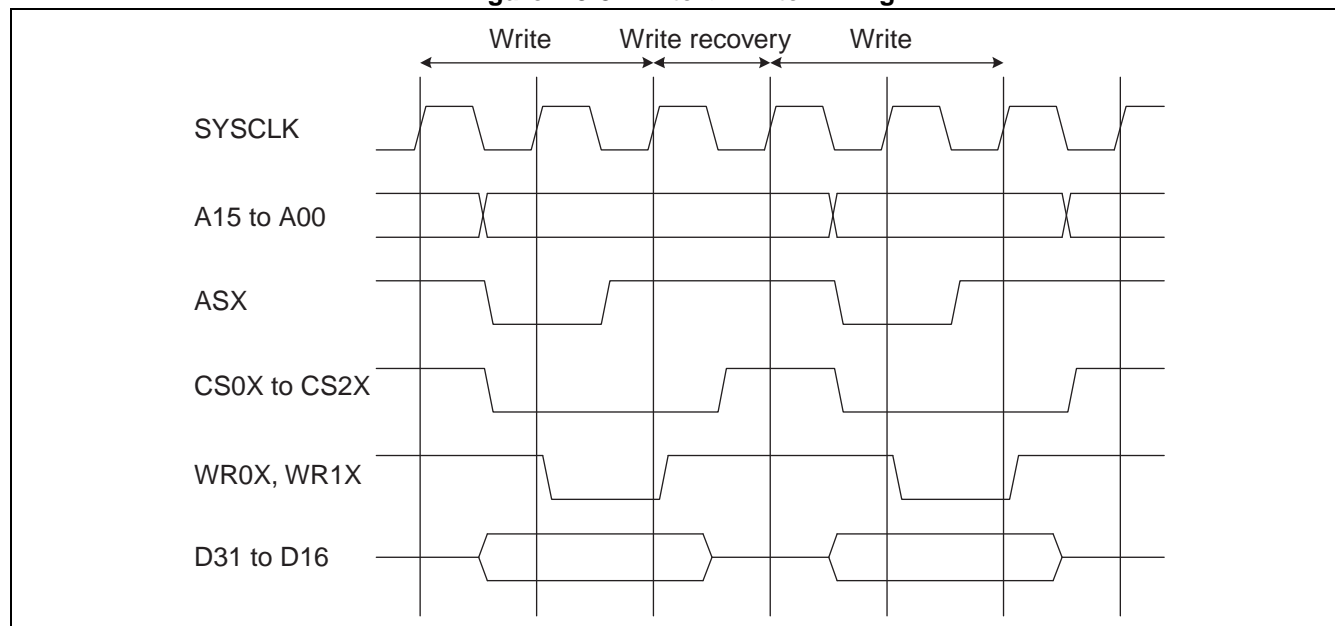
Figure 4.5-2 Read -> Write Timing

- Setting of the W07/W06 bits of the AWR register enables 0 to 3 idle cycles to be inserted.
- Settings in the CS area on the read side are enabled.
- This idle cycle is inserted if the next access after a read access is write access or access to another area.

■ Write -> Write Timing (TYP[3:0]=0000_B, AWR=0018_H)

Figure 4.5-3 shows the write -> write timing.

Figure 4.5-3 Write -> Write Timing

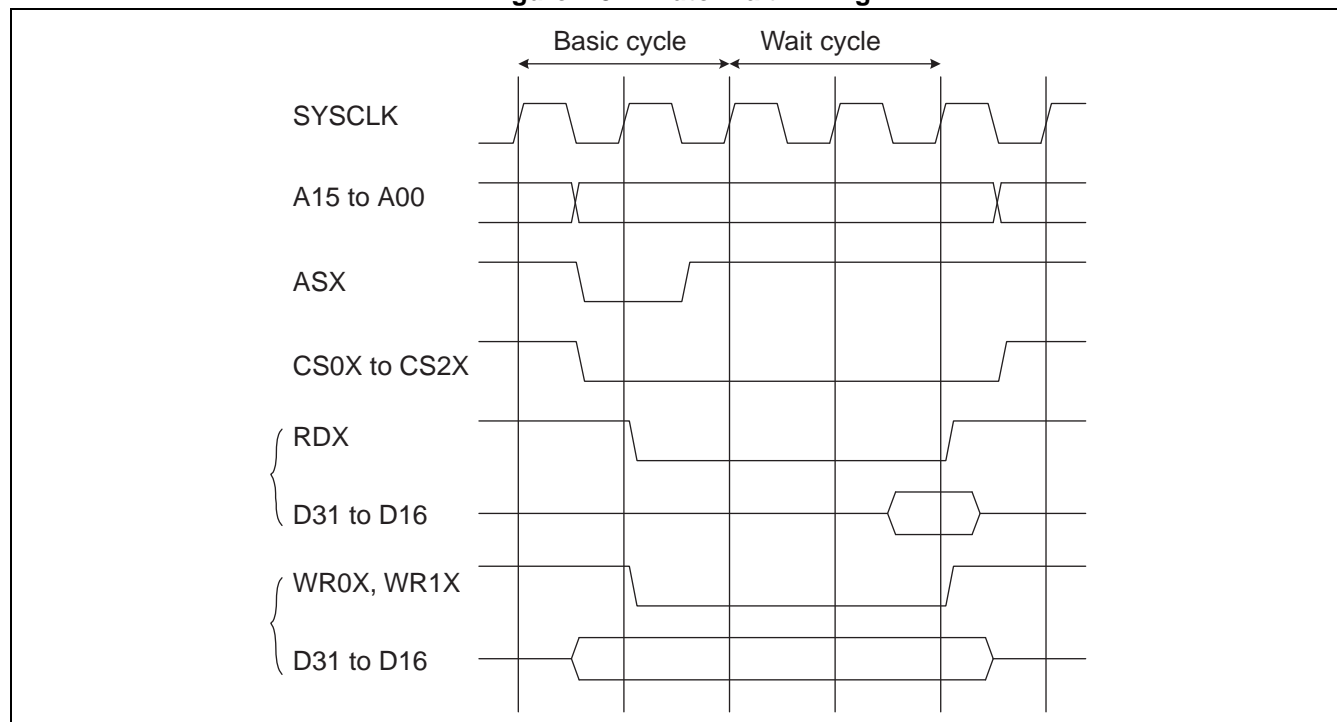


- Setting of the W05/W04 bits of the AWR register enables 0 to 3 write recovery cycles to be inserted.
- After all of the write cycles, recovery cycles are generated.
- Write recovery cycles are also generated if write access is divided into phases for access with a bus width wider than that specified.

■ Auto-Wait Timing (TYP[3:0]=0000_B, AWR=2008_H)

Figure 4.5-4 shows the auto-wait timing.

Figure 4.5-4 Auto-Wait Timing

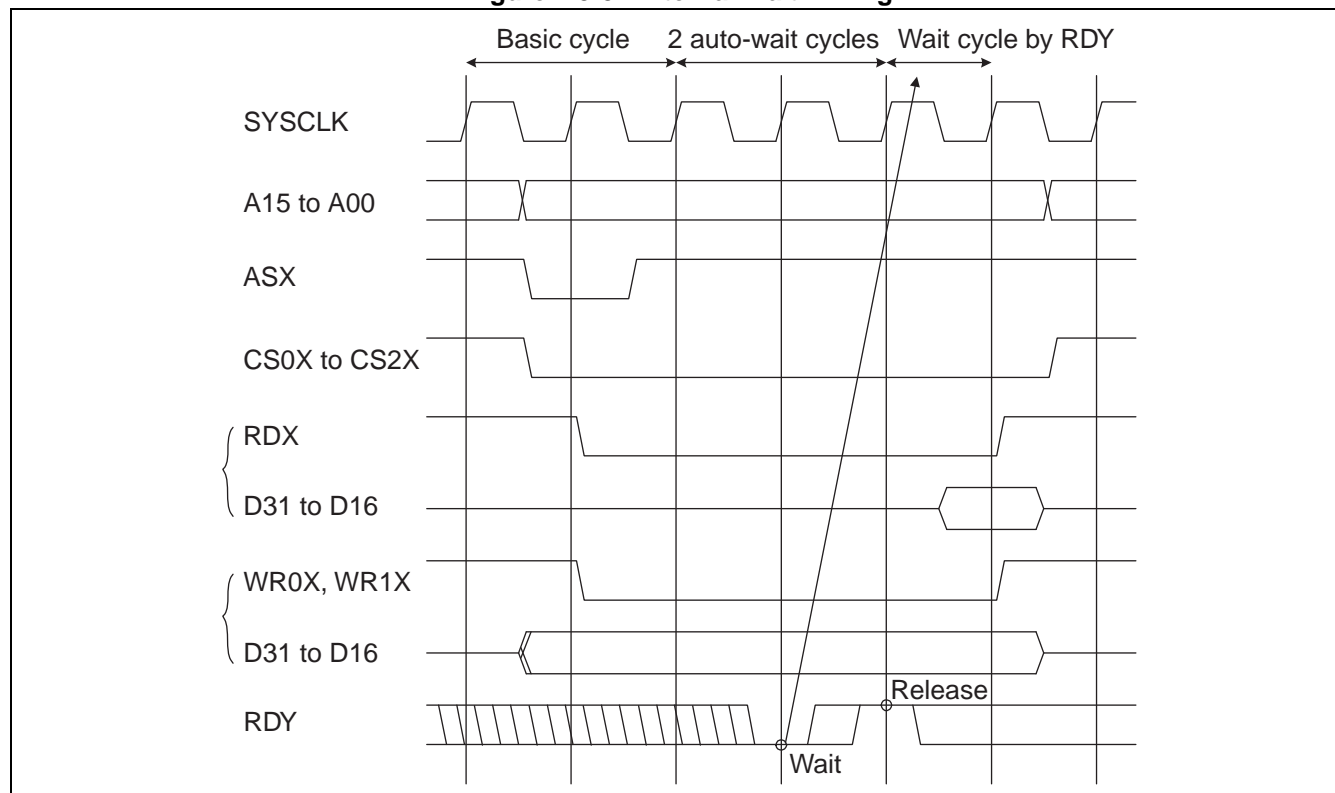


- Setting of the W15-12 bits (first wait cycles) of the AWR register enables 0 to 15 auto-wait cycles to be set.
- In Figure 4.5-4, two auto-wait cycles are inserted, making a total of four cycles for access. If auto-wait is set, the minimum number of bus cycles is 2 cycles + (first wait cycles). For a write operation, the minimum number of bus cycles may be still longer depending on the internal state.

■ External Wait Timing (TYP[3:0]=0001_B, AWR=2008_H)

Figure 4.5-5 shows the external wait timing.

Figure 4.5-5 External Wait Timing

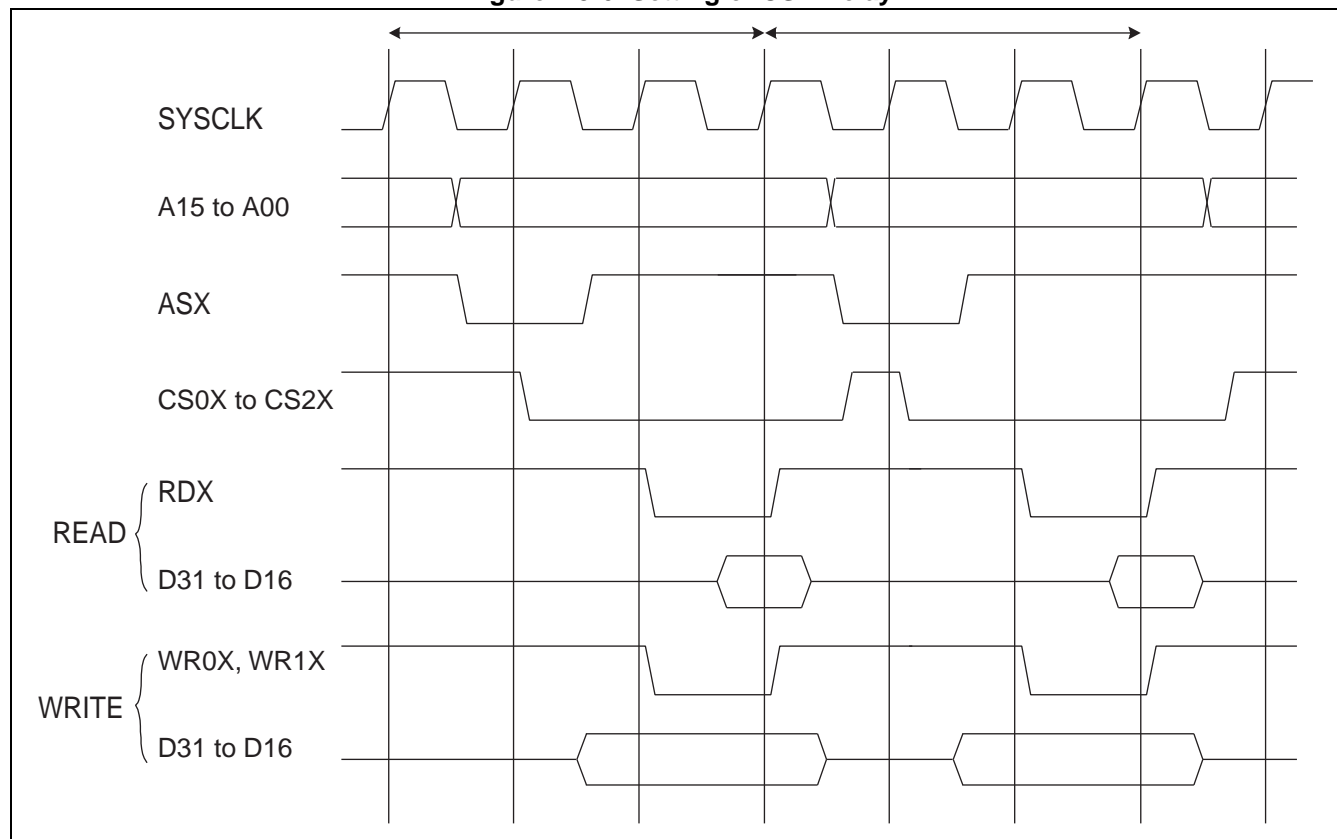


Setting 1 for the TYP0 bit of the ACR register and enabling the external RDY input pin enable external wait cycles to be inserted. In the figure above, because waiting using the auto-wait cycle is enabled, the section of the RDY pin indicated by hatching is disabled. The value of the RDY input pin is decided following the last cycle of the auto-wait cycle. Also, after a wait cycle is completed, the value of the RDY input pin is disabled until the next access cycle starts.

■ **CSX Delay Setting (TYP[3:0]=0000_B, AWR=000C_H)**

Figure 4.5-6 shows setting of a CSX delay.

Figure 4.5-6 Setting of CSX Delay

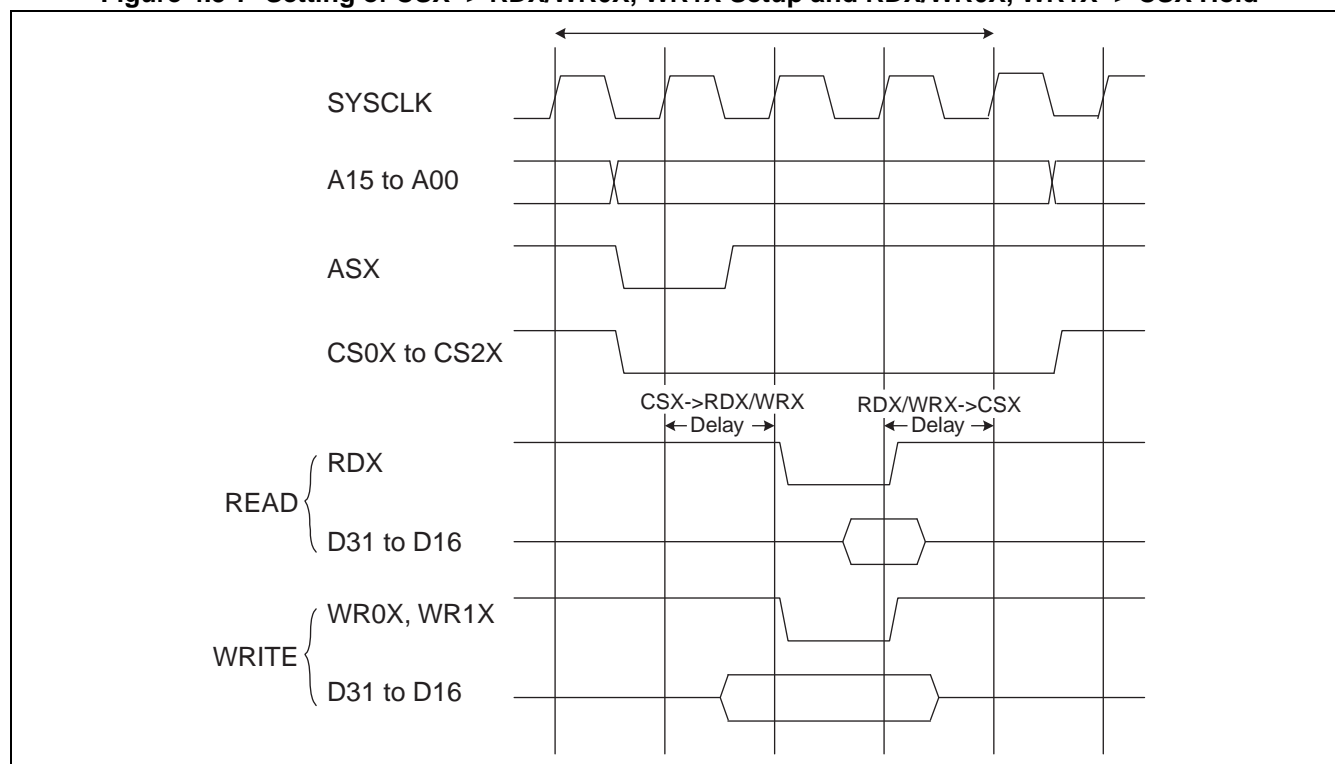


If the W02 bit is 1, assertion starts in the cycle following the cycle in which ASX is asserted. For successive accesses, a negation period is inserted.

■ Setting of CSX -> RDX/WR0X, WR1X Setup and of RDX/WR0X, WR1X -> CSX Hold (TYP[3:0]=0000_B, AWR=000B_H)

Figure 4.5-7 shows setting of the CSX -> RDX/WR0X, WR1X setup and setting of RDX/WR0X, WR1X -> CSX hold.

Figure 4.5-7 Setting of CSX -> RDX/WR0X, WR1X Setup and RDX/WR0X, WR1X -> CSX Hold



- Setting 1 for the W01 bit of the AWR register enables the CSX -> RDX/WR0X, WR1X setup delay to be set. Set this bit to extend the period between chip select assertion and read/write strobe.
- Setting 1 for the W00 bit of the AWR register enables the RDX/WR0X, WR1X -> CSX hold delay to be set. Set this bit to extend the period between read/write strobe negation and chip select negation.
- The CSX -> RDX/WR0X, WR1X setup delay (W01 bit) and RDX/WR0X, WR1X -> CSX hold delay (W00 bit) can be set independently.
- When making successive accesses within the same chip select area without negating the chip select, neither a CSX -> RDX/WR0X, WR1X setup delay nor an RDX/WR0X, WR1X -> CSX hold delay is inserted.
- If a setup cycle for determining the address or a hold cycle for determining the address is needed, set 1 for the address -> CSX delay setting (W02 bit of the AWR register).

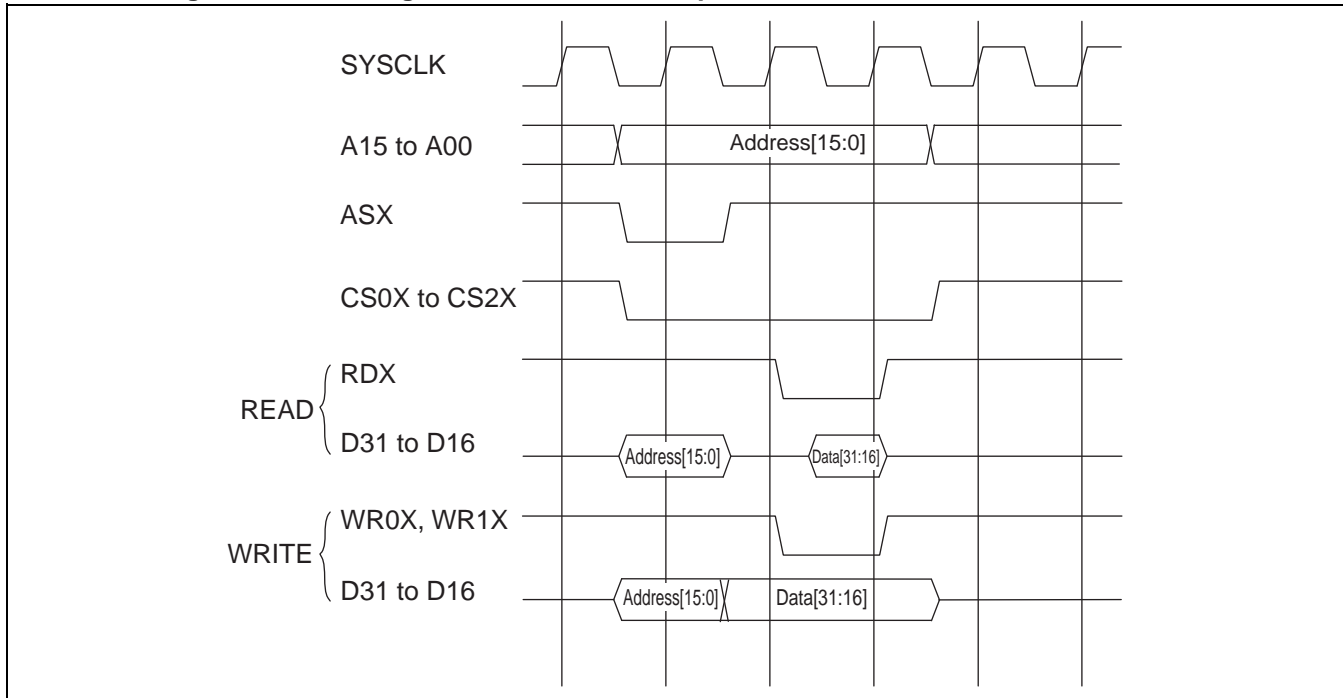
4.6 Address/Data Multiplex Interface

This section describes setting of the address/data multiplex interface.

■ Without External Wait (TYP[3:0] = 0100_B, AWR = 0008_H)

Figure 4.6-1 shows setting of the address/data multiplex interface when there is no external wait.

Figure 4.6-1 Setting of Address/Data Multiplex Interface Without an External Wait

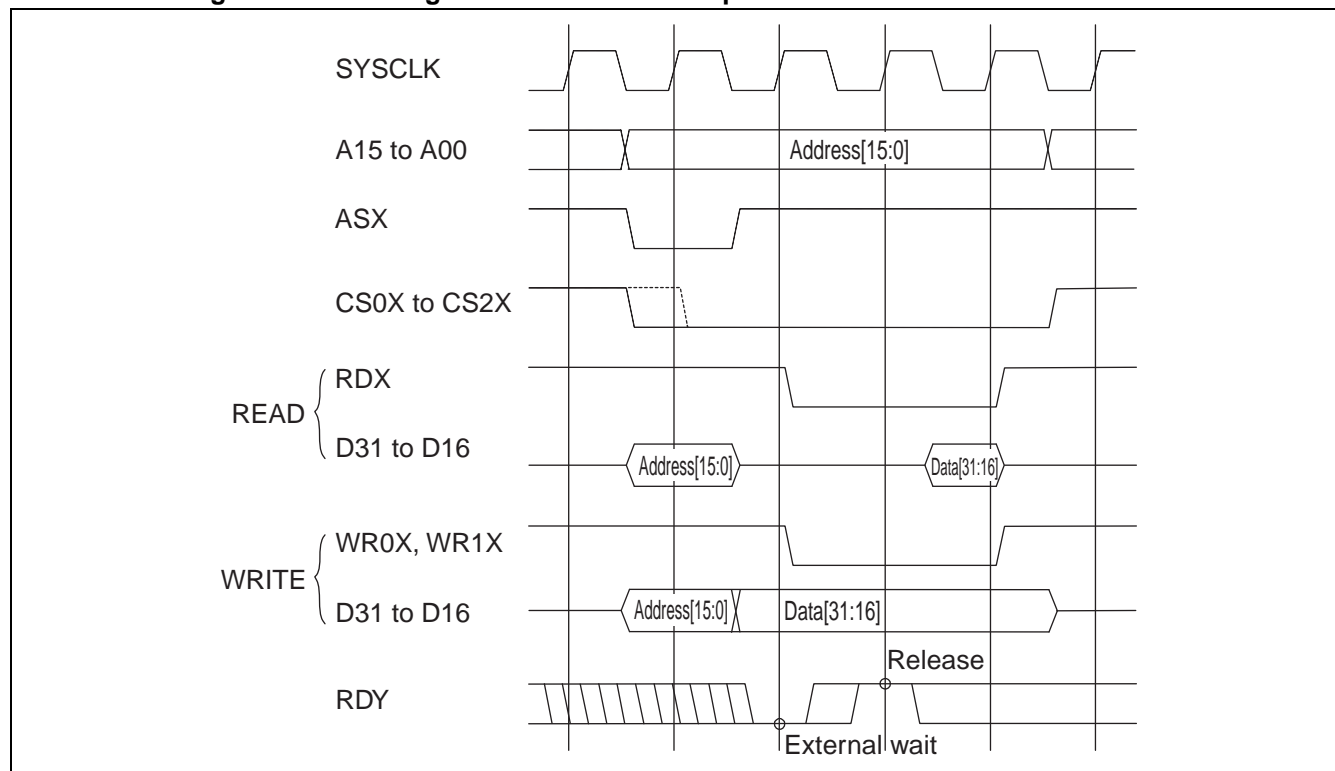


- Making a setting such as TYP[3:0]=01XX_B in the ACR register enables the address/data multiplex interface to be set.
- If the address/data multiplex interface is set, set 8 bits or 16 bits for the data bus width (DBW[1:0] bit). The 32-bit bus width is not supported.
- In the address/data multiplex interface, the total of 3 cycles of 2 address output cycles + 1 data cycle becomes the basic number of access cycles.
- In the address output cycle, ASX is asserted as the output address latch enable signal. However, when CSX → RD0X/WR0X, WR1X setup delay (AWR:W01) is set to 0, the multiplex address output cycle consists of only one cycle as shown in Figure 4.6-1. Since the address cannot be directly latched at the rising edge of ASX, fetch the address at the rising edge for SYSCLK of the cycle in which ASX is asserted (Low).
- As with a normal interface, the address indicating the start of access is outputted to A[15:0] during the time division bus cycle.
- As with the normal interface, auto-wait (AWR:W15 to W12), read → write idle cycle (AWR:W07, W06), write recovery (AWR:W05, W04), address → CSX delay (AWR:W02), CSX → RD0X/WR0X, WR1X setup delay (AWR:W01), and RD0X/WR0X, WR1X → CSX hold delay (AWR:W00) can be set.

■ With External Wait (TYP[3:0]=0101_B, AWR=1008_H)

Figure 4.6-2 shows setting of the address/data multiplex interface with an external wait.

Figure 4.6-2 Setting of Address/Data Multiplex Interface With an External Wait

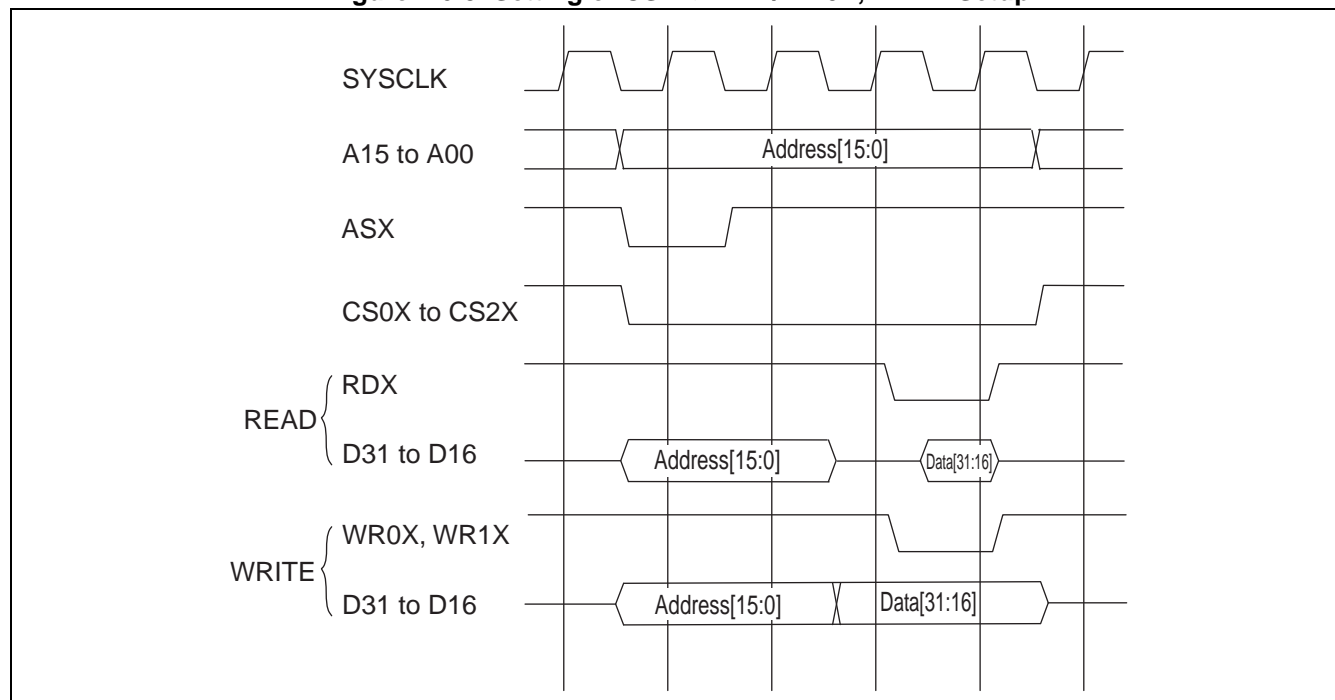


Making a setting such as TYP[3:0]=01X1_B in the ACR register enables RDY input in the address/data multiplex interface.

■ **CSX -> RDX/WR0X, WR1X Setup (TYP[3:0] = 0101_B, AWR=100B_H)**

Figure 4.6-3 shows setting of the CSX -> RDX/WR0X, WR1X setup.

Figure 4.6-3 Setting of CSX -> RDX/WR0X, WR1X Setup



Setting 1 for the CSX -> RDX/WR0X, WR1X setup delay (AWR:W01) enables the multiplex address output cycle to be extended by one cycle as shown in Figure 4.6-3, allowing the address to be latched directly to the rising edge of ASX. Use this setting if you want to use ASX as an ALE (Address Latch Enable) strobe without using SYSCLK.

4.7 Procedure for Setting a Register

Observe the following rules when setting the external bus interface:

■ Procedure for Setting the External Bus Interface

- Before rewriting the contents of a register, be sure to set the CSER register so that the corresponding area is not used (0). If you change the settings while 1 is set, access before and after the change cannot be guaranteed.
- Use the following procedure to change a register:
 - 1) Set 0 for the CSER bit corresponding to the applicable area.
 - 2) Set both ASR and ACR at the same time using word access. When accessing ASR and ACR using half word, set ACR after setting ASR.
 - 3) Set AWR.
 - 4) Set the CSER bit corresponding to the applicable area.
- The CS0X area is enabled after a reset is released. If the area is used as a program area, the register contents need to be rewritten while the CSER bit is 1. In this case, make the settings described in 2), 3) above in the initial state with a low-speed internal clock. Then, switch the clock to a high-speed clock.

CHAPTER 5

I/O PORTS

This chapter outlines the I/O ports and describes the configuration and functions of their registers.

5.1 Overview of I/O Port

5.2 Block Diagrams of I/O Port

5.3 I/O Port Registers

5.1 Overview of I/O Port

This section describes the I/O ports used in the MB91470/480 series.

■ Overview of Ports

MB91470/480 series can use its pins as I/O ports when they are set not to serve for input to or output from their respective external bus interfaces or peripherals.

■ Configuration

The control section of the port comprises the following four registers:

- **PFR: Port Function Register**

This setup register switches the function of a pin to be used between a peripheral output/external bus interface and a general-purpose port.

- **DDR: Data Direction Register**

This setup register switches the direction of data between input and output, when the pin is used as a general-purpose port.

- **PDR: Port Data Register**

This register is used to set data.

- **PCR: Pull-up Control Register**

This setup register is used to enable the pull-up function.

5.2 Block Diagrams of I/O Port

This section describes block diagrams of I/O ports.

■ Block Diagrams of Ports

In MB91470/480 series, four types of ports are available, depending on an external bus interface or peripheral that is also used as the general-purpose port.

● Normal I/O port

This dual-function I/O port has the most basic configuration and also serves as a peripheral input/output. It is composed of PFR, DDR, PDR and PCR.

● External interrupt input I/O port

This I/O port also serves as an external interrupt input and comprises PFR, DDR, PDR, PCR and an external interrupt input enabling signal.

● Analog input I/O port

This I/O port also functions as an analog input and comprises PFR, DDR, PDR, PCR and an analog input enabling signal.

● External bus interface I/O port

This I/O port also functions as an external bus interface and comprises PFR, DDR, PDR, PCR and an external bus interface mode enabling signal.

● Multi-function timer I/O port

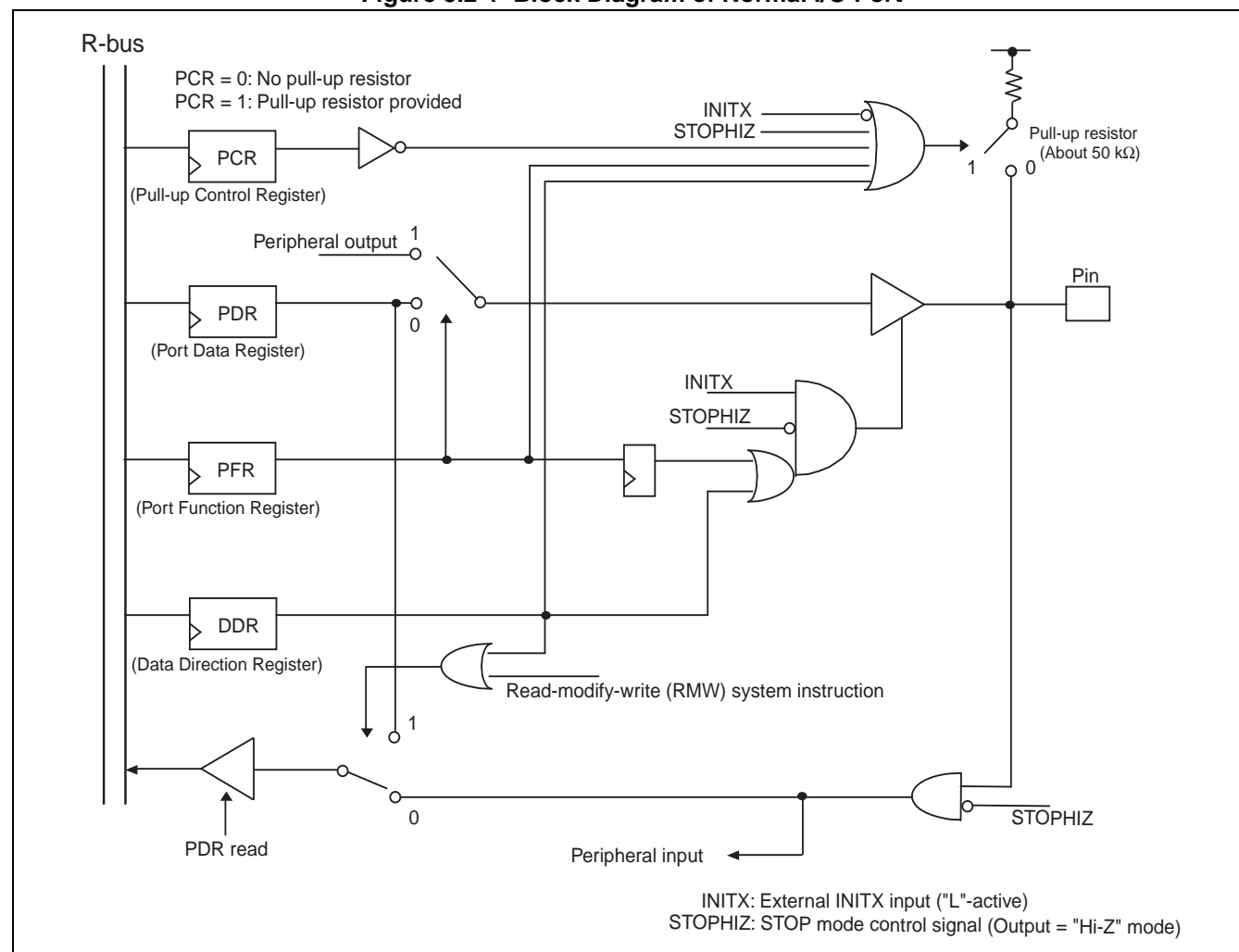
This I/O port also functions as waveform generator output of a multi-function timer (RTO0 to RTO11) and comprises PFR, DDR, PDR, PCR, and a DDTI interrupt flag signal. See Section "12.4.13 Waveform Control Register (SIGCR1/SIGCR2)" for details of the DDTI interrupt flag.

5.2.1 Normal I/O Port

This section describes the block diagram of the normal I/O port.

■ Block Diagram of Normal I/O Port

Figure 5.2-1 Block Diagram of Normal I/O Port



■ Modes of I/O Port

● In port input mode (PFR=0 & DDR=0)

PDR read:	The level at the corresponding external pin is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write	A setting value is written to the PDR.

● In port output mode (PFR=0 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	The PDR value is output to the corresponding external pin.

● In peripheral output mode 1 (PFR=1 & DDR=0)

PDR read:	The output value from the corresponding peripheral is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

● In peripheral output mode 2 (PFR=1 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

■ Setting Value in Pull-up Resistor Control Register

In the following modes, the setting in the pull-up resistor control register is invalid.

- When external INITX input is active (when "L" is selected for input)
- When STOP mode is selected (HIZ=1)
- When peripheral output mode is selected (PFR=1)
- When port output mode is selected (DDR=1)

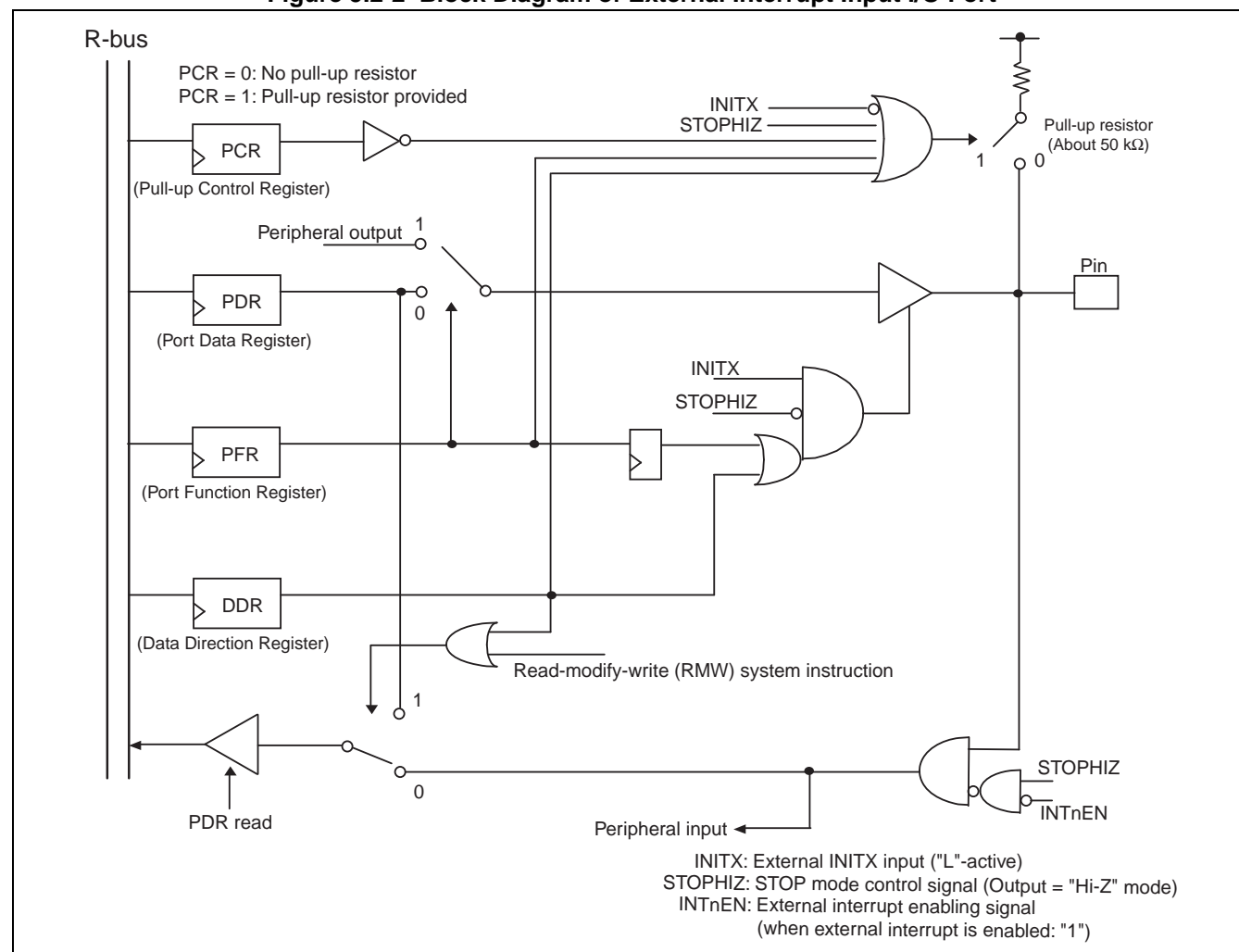
In any mode other than the above, the setting in the pull-up resistor control register has priority.

5.2.2 External Interrupt Input I/O Port

This section describes the block diagram of the external interrupt input I/O port.

■ Block Diagram of External Interrupt Input I/O Port

Figure 5.2-2 Block Diagram of External Interrupt Input I/O Port



■ Modes of I/O Port

Same as the normal I/O port

■ Setting Value in Pull-up Resistor Control Register

Same as the normal I/O port

■ Controlling the Enabling of Input

While the normal I/O port has its input fixed to "L" in STOP mode (HIZ=1), this dual-function port has the respective port enabled to be input even in STOP mode (HIZ=1), when external interrupt input is enabled (INTnEN=1).

Note:

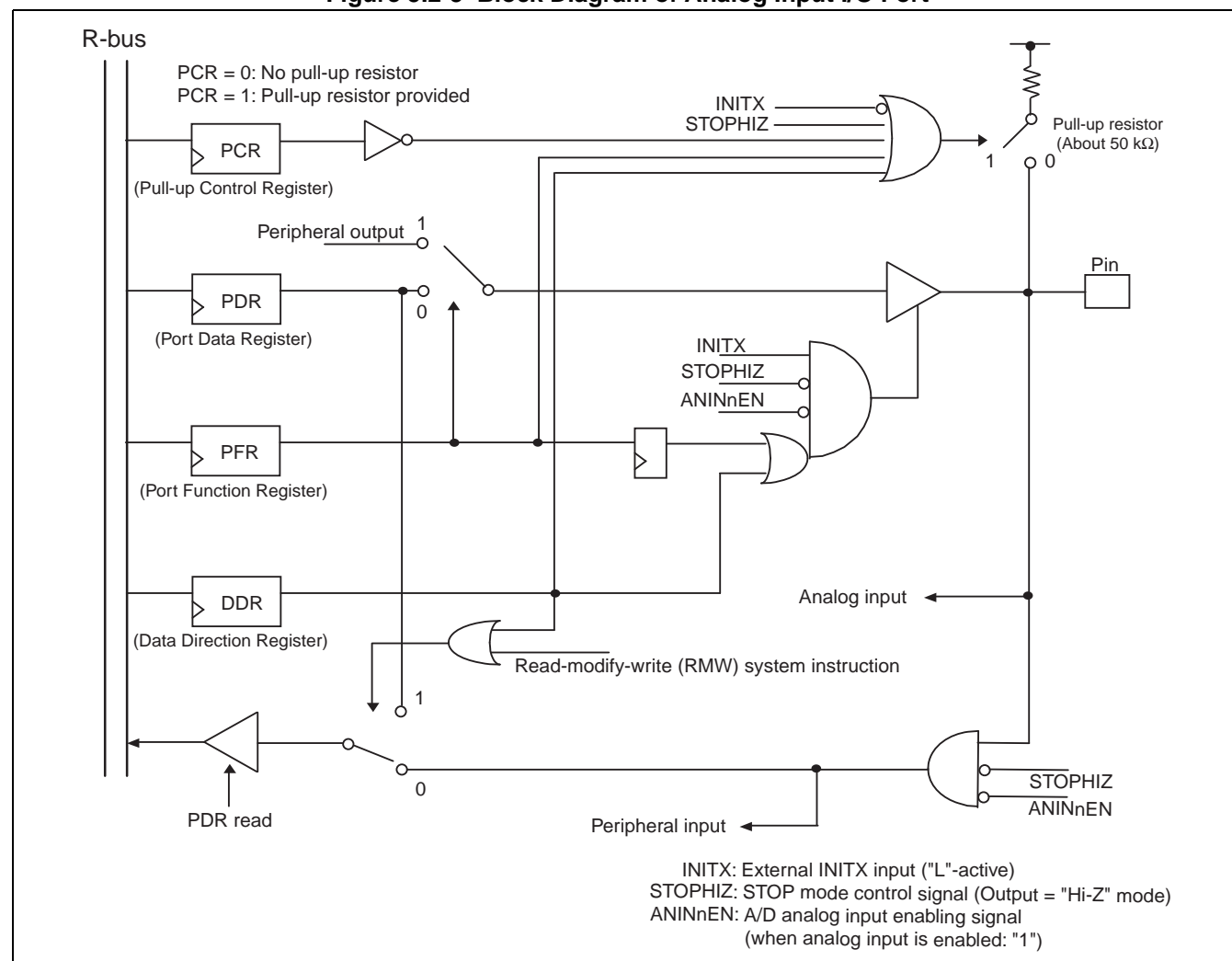
When external interrupt input is enabled in STOP mode (HIZ=1), the respective port can be input but the setting in the pull-up resistor control register is invalid.

5.2.3 Analog Input I/O Port

This section describes the block diagram of the analog input I/O port.

■ Block Diagram of Analog Input I/O Port

Figure 5.2-3 Block Diagram of Analog Input I/O Port



■ Modes of I/O Port

In any mode other than analog input mode (ANINnEN=0), the same specifications as for the normal I/O port apply. In analog input mode (ANINnEN=1), however, the specifications below are followed.

● In port input mode (PFR=0 & DDR=0)

PDR read:	"0" is always read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

● In port output mode (PFR=0 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	The PDR value is output to the corresponding external pin.

● In peripheral output mode 1 (PFR=1 & DDR=0)

PDR read:	"0" is always read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

● In peripheral output mode 2 (PFR=1 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

■ Setting Value in Pull-up Resistor Control Register

Same as the normal I/O port

■ Controlling the Enabling of Input

While the normal I/O port has its input fixed to "L" in STOP mode (HIZ=1), this dual-function port also has its input fixed to "L" even in analog input mode (ANINnEN=1).

Note:

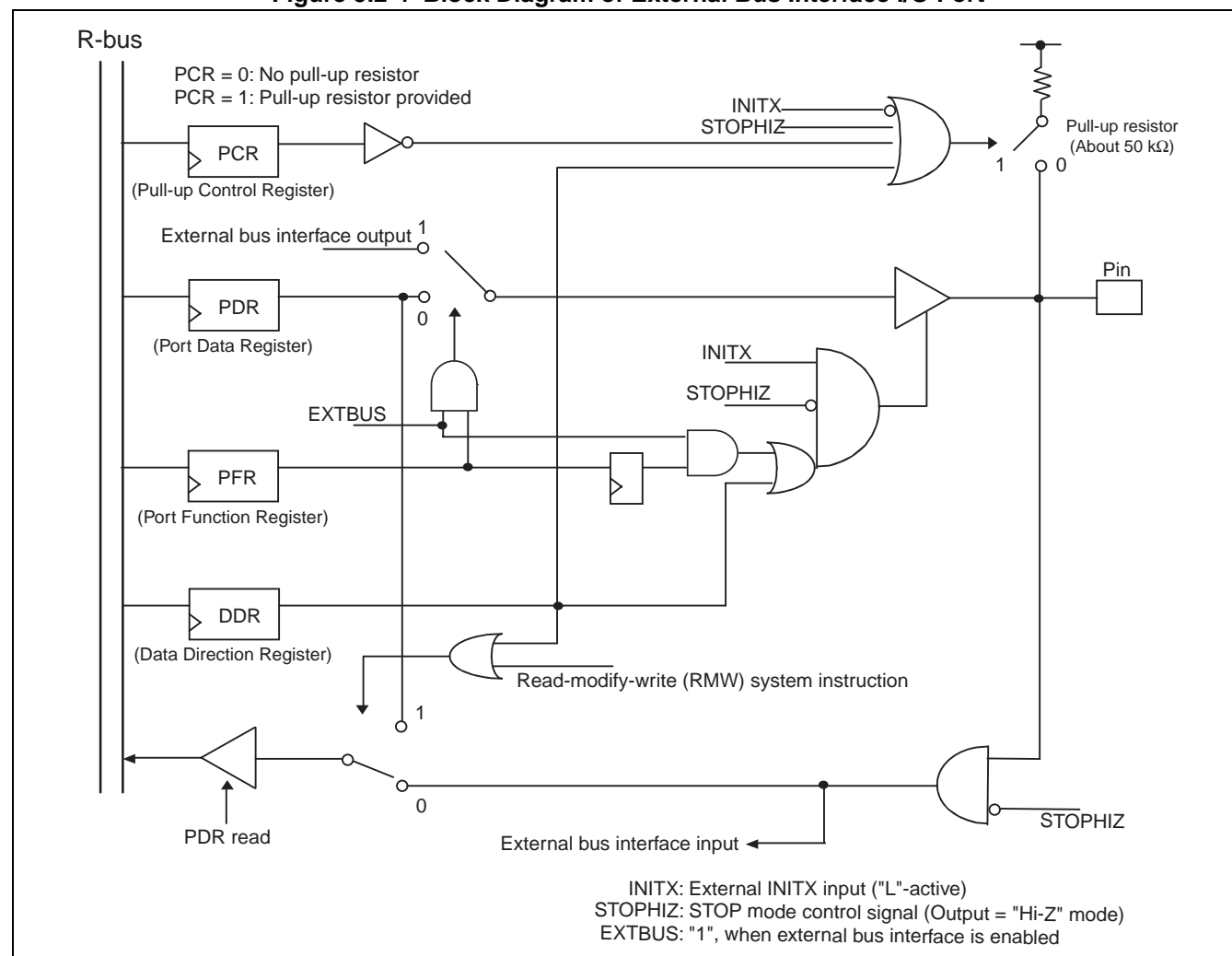
The setting in the pull-up resistor control register is valid even in analog input mode (ANINnEN=1).

5.2.4 External Bus Interface I/O Port

This section describes the block diagram of the external bus interface I/O port.

■ Block Diagram of External Bus Interface I/O Port

Figure 5.2-4 Block Diagram of External Bus Interface I/O Port



■ Modes of I/O Port

- In port input mode (EXTBUS=0 & DDR=0 or EXTBUS=1 & PFR=0 & DDR=0)

PDR read:	The level at the corresponding external pin is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

- In port output mode (EXTBUS=0 & DDR=1 or EXTBUS=1 & PFR=0 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

- In external bus interface output mode 1 (EXTBUS=1 & PFR=1 & DDR=0)

PDR read:	The output value of the respective external bus interface is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

- In external bus interface output mode 2 (EXTBUS=1 & PFR=1 & DDR=1)

PDR read:	The PDR value is read.
PDR read modify write (RMW) system instruction:	The PDR value is read.
PDR write:	A setting value is written to the PDR.

■ Setting Value in Pull-up Resistor Control Register

In the following modes, the setting in the pull-up resistor control register is invalid.

- When external INITX input is active (when "L" is selected for input)
- When STOP mode is selected (HIZ=1)
- When port output mode is selected (DDR=1)

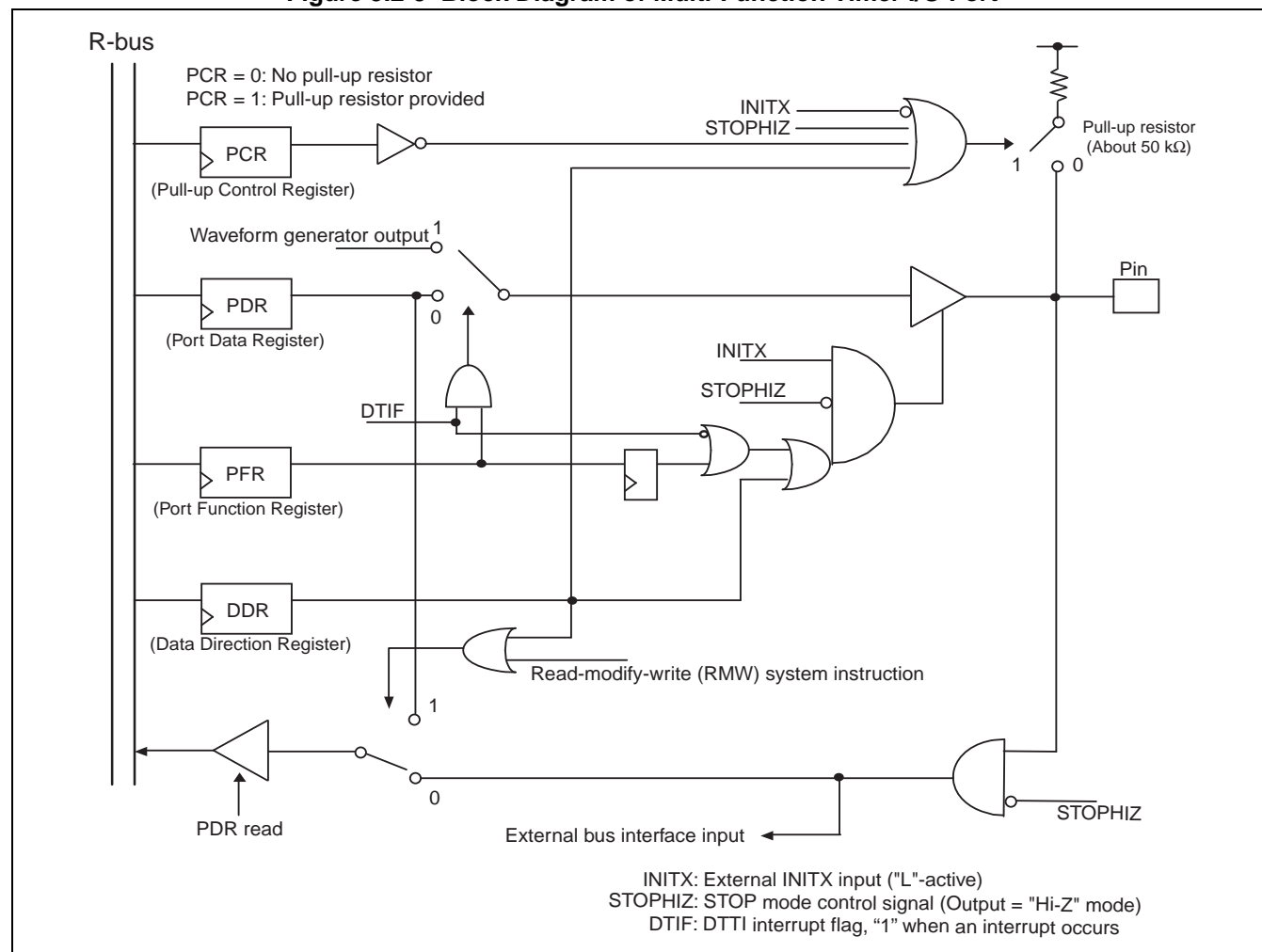
Otherwise, the setting in the pull-up resistor control register has priority.

5.2.5 Multi-Function Timer I/O Port

This section describes the block diagram of the multi-function timer I/O port.

■ Block Diagram of Multi-Function Timer I/O Port

Figure 5.2-5 Block Diagram of Multi-Function Timer I/O Port



■ Modes of I/O Port

- In port input mode (DTIF=1 & DDR=0, or, DTIF=0 & PFR=0 & DDR=0)
 - PDR read: The level at the corresponding external pin is read.
 - PDR read modify write (RMW) system instruction: The PDR value is read.
 - PDR write: A setting value is written to the PDR.
- In port output mode(DTIF=1 & DDR=1, or, DTIF=0 & PFR=0 & DDR=1)
 - PDR read: The PDR value is read.
 - PDR read modify write (RMW) system instruction: The PDR value is read.
 - PDR write: A setting value is written to the PDR.
- In waveform generator output mode 1 (DTIF=0 & PFR=1 & DDR=0)
 - PDR read: The value of the waveform generator output is read.
 - PDR read modify write (RMW) system instruction: The PDR value is read.
 - PDR write: A setting value is written to the PDR.
- In waveform generator output mode 2 (DTIF=0 & PFR=1 & DDR=1)
 - PDR read: The PDR value is read.
 - PDR read modify write (RMW) system instruction: The PDR value is read.
 - PDR write: A setting value is written to the PDR.

■ Setting Value in Pull-up Resistor Control Register

In the following modes, the setting in the pull-up resistor control register is invalid.

- When external INITX input is active (when "L" is selected for input)
- When STOP mode is selected (HIZ=1)
- When port output mode is selected (DDR=1)
- When waveform generator output mode is selected (DTIF=0 & PFR=1)

Otherwise, the setting in the pull-up resistor control register has priority.

5.3 I/O Port Registers

This section explains registers of I/O port.

■ Port Data Registers (PDR: PDR0 to PDR3, PDR5, PDR6, PDR8 to PDRH, PDRJ, PDRL, PDRM, PDRP to PDRS)

PDR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000000 _H	P07	P06	P05	P04	P03	P02	P01	P00	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000001 _H	P17	P16	P15	P14	P13	P12	P11	P10	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR2									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000002 _H	P27	P26	P25	P24	P23	P22	P21	P20	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR3									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000003 _H	P37	P36	P35	P34	P33	P32	P31	P30	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR5									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000004 _H	-	P56	P55	P54	P53	P52	P51	P50	- XXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR6									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000005 _H	-	-	-	-	-	-	P61	P60	----- XX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR8									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000006 _H	P87	P86	P85	P84	P83	P82	P81	P80	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR9									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000007 _H	P97	P96	P95	P94	P93	P92	P91	P90	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

PDRA									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000008 _H	-	-	-	PA4	PA3	PA2	PA1	PA0	- - - XXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRB									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000009 _H	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR C									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000A _H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRD									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000B _H	-	-	-	-	PD3	PD2	PD1	PD0	- - - - XXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRE									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000C _H	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRF									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000D _H	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRG									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000E _H	-	-	PG5	PG4	PG3	PG2	PG1	PG0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRH									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000000F _H	-	-	PH5	PH4	PH3	PH2	PH1	PH0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRJ									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000010 _H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000012 _H	-	-	-	-	-	PL2	PL1	PL0	- - - - - XXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

PDRM

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000013 _H	-	-	-	-	PM3	PM2	PM1	PM0	- - - - XXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PDRP

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000014 _H	-	-	PP5	PP4	PP3	PP2	PP1	PP0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PDRQ

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000015 _H	-	-	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PDRR

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000016 _H	-	-	PR5	PR4	PR3	PR2	PR1	PR0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PDRS

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000017 _H	-	-	PS5	PS4	PS3	PS2	PS1	PS0	- - XXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

PDR0 to PDR3, PDR5, PDR6, PDR8 to PDRH, PDRJ, PDRL, PDRM, PDRP to PDRS are I/O data registers of the I/O port. These are controlled for input/output by their respective DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS, PFR8, PFR9, PFRB, PFRF to PFRH, PFRJ, PFRM, PFRQ, PFRS.

MB91470/480 Series**■ Data Direction Registers (DDR: DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS)**

DDR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000400 _H	P07	P06	P05	P04	P03	P02	P01	P00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000401 _H	P17	P16	P15	P14	P13	P12	P11	P10	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR2									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000402 _H	P27	P26	P25	P24	P23	P22	P21	P20	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR3									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000403 _H	P37	P36	P35	P34	P33	P32	P31	P30	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR5									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000404 _H	-	P56	P55	P54	P53	P52	P51	P50	- 0000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR6									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000405 _H	-	-	-	-	-	-	P61	P60	- - - - - 00 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR8									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000406 _H	P87	P86	P85	P84	P83	P82	P81	P80	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR9									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000407 _H	P97	P96	P95	P94	P93	P92	P91	P90	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDRA									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000408 _H	-	-	-	PA4	PA3	PA2	PA1	PA0	- - - 00000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDRB									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000409 _H	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

DDRC

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040A _H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRD

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040B _H	-	-	-	-	PD3	PD2	PD1	PD0	- - - - 0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRE

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040C _H	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRF

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040D _H	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRG

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040E _H	-	-	PG5	PG4	PG3	PG2	PG1	PG0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRH

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000040F _H	-	-	PH5	PH4	PH3	PH2	PH1	PH0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRJ

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000410 _H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRL

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000412 _H	-	-	-	-	-	PL2	PL1	PL0	- - - - - 000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRM

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000413 _H	-	-	-	-	PM3	PM2	PM1	PM0	- - - - 0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRP

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000414 _H	-	-	PP5	PP4	PP3	PP2	PP1	PP0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

DDRQ

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000415 _H	-	-	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRR

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000416 _H	-	-	PR5	PR4	PR3	PR2	PR1	PR0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

DDRS

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000417 _H	-	-	PS5	PS4	PS3	PS2	PS1	PS0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS control the I/O direction of the corresponding I/O port for each bit.

When PFR = 0 DDR = 0: Port input

■ **Pull-up Control Registers (PCR: PCR0 to PCR3, PCR5, PCR6, PCR8 to PCRH, PCRJ, PCRL, PCRM, PCRP to PCRS)**

PCR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000600 _H	P07	P06	P05	P04	P03	P02	P01	P00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000601 _H	P17	P16	P15	P14	P13	P12	P11	P10	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR2									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000602 _H	P27	P26	P25	P24	P23	P22	P21	P20	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR3									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000603 _H	P37	P36	P35	P34	P33	P32	P31	P30	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR5									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000604 _H	-	P56	P55	P54	P53	P52	P51	P50	- 0000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR6									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000605 _H	-	-	-	-	-	-	P61	P60	- - - - - 00 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR8									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000606 _H	P87	P86	P85	P84	P83	P82	P81	P80	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR9									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000607 _H	P97	P96	P95	P94	P93	P92	P91	P90	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCRA									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000608 _H	-	-	-	PA4	PA3	PA2	PA1	PA0	- - - 00000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCRB									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000609 _H	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

PCRC

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060A _H	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRD

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060B _H	-	-	-	-	PD3	PD2	PD1	PD0	- - - - 0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRE

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060C _H	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRF

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060D _H	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRG

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060E _H	-	-	PG5	PG4	PG3	PG2	PG1	PG0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRH

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000060F _H	-	-	PH5	PH4	PH3	PH2	PH1	PH0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRJ

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000610 _H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRL

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000612 _H	-	-	-	-	-	PL2	PL1	PL0	- - - - - 000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRM

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000613 _H	-	-	-	-	PM3	PM2	PM1	PM0	- - - - 0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRP

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000614 _H	-	-	PP5	PP4	PP3	PP2	PP1	PP0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

PCRQ

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000615 _H	-	-	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRR

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000616 _H	-	-	PR5	PR4	PR3	PR2	PR1	PR0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PCRS

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000617 _H	-	-	PS5	PS4	PS3	PS2	PS1	PS0	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

PCR0 to PCR3, PCR5, PCR6, PCR8 to PCRH, PCRJ, PCRL, PCRM, PCRP to PCRS control the pull-up resistor for the corresponding I/O port.

PCR = 0: No pull-up resistor

MB91470/480 Series**■ Port Function Control Registers (PFR: PFR0 to PFR3, PFR5, PFR6, PFR8, PFR9, PFR0, PFRF to PFRH, PFRJ, PFRM, PFRQ, PFRS)**

PFR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000420 _H	D23	D22	D21	D20	D19	D18	D17	D16	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000421 _H	D31	D30	D29	D28	D27	D26	D25	D24	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR2									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000422 _H	A07	A06	A05	A04	A03	A02	A01	A00	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR3									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000423 _H	A15	A14	A13	A12	A11	A10	A09	A08	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR5									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000424 _H	-	WR1X	WR0X	RDX	ASX	CS2X	CS1X	CS0X	- 1111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR6									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000425 _H	-	-	-	-	-	-	RDY	SYSCLK	- - - - - 11 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR8									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000426 _H	PPG7E	PPG6E	PPG5E	PPG4E	-	-	-	-	0000- - - _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR9									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000427 _H	PPG15E	PPG14E	PPG13E	PPG12E	PPG11E	PPG10E	PPG9E	PPG8E	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFR0C									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000042A _H	-	-	SOT5E	-	SCK5E	SOT4E	-	SCK4E	- - 0- 00- 0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PFRF									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000042D _H	-	-	-	-	-	-	-	CLKP OUTE	- - - - - 0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

PFRG									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000042E _H	-	-	SOT1E	-	SCK1E	SOT0E	-	SCK0E	-- 0- 00- 0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFRH									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000042F _H	-	-	SOT3E	-	SCK3E	SOT2E	-	SCK2E	-- 0- 00- 0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFRJ									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000430 _H	TOUT3E	-	TOUT2E	-	TOUT1E	-	TOUT0E	-	0- 0- 0- 0- 0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFRM									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000433 _H	-	-	-	-	PPG3E	PPG2E	PPG1E	PPG0E	- - - - 0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFRQ									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000435 _H	-	-	RTO5E	RTO4E	RTO3E	RTO2E	RTO1E	RTO0E	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFRS									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000437 _H	-	-	RTO11E	RTO10E	RTO9E	RTO8E	RTO7E	RTO6E	- - 000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

PFR0 to PFR3, PFR5, PFR6, PFR8 to PFR9, PFRJ, PFRM, PFRQ, PFRS control each bit in the output of the corresponding peripheral.

Be sure to write "0" to undefined bits.

The following table lists individual PFR registers, their initial values and functions:

Table 5.3-1 Initial Value and Function of PFR Register (1 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFR0	0	D16	0	General-purpose port
			1	Outputs external data bit16 [Initial value]
	1	D17	0	General-purpose port
			1	Outputs external data bit17 [Initial value]
	2	D18	0	General-purpose port
			1	Outputs external data bit18 [Initial value]
	3	D19	0	General-purpose port
			1	Outputs external data bit19 [Initial value]
	4	D20	0	General-purpose port
			1	Outputs external data bit20 [Initial value]
	5	D21	0	General-purpose port
			1	Outputs external data bit21 [Initial value]
	6	D22	0	General-purpose port
			1	Outputs external data bit22 [Initial value]
PFR1	0	D24	0	General-purpose port
			1	Outputs external data bit24 [Initial value]
	1	D25	0	General-purpose port
			1	Outputs external data bit25 [Initial value]
	2	D26	0	General-purpose port
			1	Outputs external data bit26 [Initial value]
	3	D27	0	General-purpose port
			1	Outputs external data bit27 [Initial value]
	4	D28	0	General-purpose port
			1	Outputs external data bit28 [Initial value]
	5	D29	0	General-purpose port
			1	Outputs external data bit29 [Initial value]
	6	D30	0	General-purpose port
			1	Outputs external data bit30 [Initial value]
	7	D31	0	General-purpose port
			1	Outputs external data bit31 [Initial value]

Table 5.3-1 Initial Value and Function of PFR Register (2 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFR2	0	A00	0	General-purpose port
			1	Outputs external address bit0 [Initial value]
	1	A01	0	General-purpose port
			1	Outputs external address bit1 [Initial value]
	2	A02	0	General-purpose port
			1	Outputs external address bit2 [Initial value]
	3	A03	0	General-purpose port
			1	Outputs external address bit3 [Initial value]
	4	A04	0	General-purpose port
			1	Outputs external address bit4 [Initial value]
	5	A05	0	General-purpose port
			1	Outputs external address bit5 [Initial value]
	6	A06	0	General-purpose port
			1	Outputs external address bit6 [Initial value]
	7	A07	0	General-purpose port
			1	Outputs external address bit7 [Initial value]
PFR3	0	A08	0	General-purpose port
			1	Outputs external address bit8 [Initial value]
	1	A09	0	General-purpose port
			1	Outputs external address bit9 [Initial value]
	2	A10	0	General-purpose port
			1	Outputs external address bit10 [Initial value]
	3	A11	0	General-purpose port
			1	Outputs external address bit11 [Initial value]
	4	A12	0	General-purpose port
			1	Outputs external address bit12 [Initial value]
	5	A13	0	General-purpose port
			1	Outputs external address bit13 [Initial value]
	6	A14	0	General-purpose port
			1	Outputs external address bit14 [Initial value]
	7	A15	0	General-purpose port
			1	Outputs external address bit15 [Initial value]

Table 5.3-1 Initial Value and Function of PFR Register (3 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFR5	0	CS0X	0	General-purpose port
			1	Outputs external chip select 0 [Initial value]
	1	CS1X	0	General-purpose port
			1	Outputs external chip select 1 [Initial value]
	2	CS2X	0	General-purpose port
			1	Outputs external chip select 2 [Initial value]
	3	ASX	0	General-purpose port
			1	Outputs external address strobe [Initial value]
	4	RDX	0	General-purpose port
			1	Outputs external read strobe [Initial value]
	5	WR0X	0	General-purpose port
			1	Outputs external write strobe [Initial value] Corresponding to external data bus bit31 to bit24
	6	WR1X	0	General-purpose port
			1	Outputs external write strobe [Initial value] Corresponding to external data bus bit23 to bit16
PFR6	0	SYSCLK	0	General-purpose port
			1	Outputs external clock [Initial value]
	1	RDY	0	General-purpose port
			1	Inputs external ready [Initial value]
PFR8	4	PPG4E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 4
	5	PPG5E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 5
	6	PPG6E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 6
	7	PPG7E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 7

Table 5.3-1 Initial Value and Function of PFR Register (4 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFR9	0	PPG8E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 8
	1	PPG9E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 9
	2	PPG10E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 10
	3	PPG11E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 11
	4	PPG12E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 12
	5	PPG13E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 13
PFRF	0	SCK4E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface4
	2	SOT4E	0	General-purpose port [Initial value]
			1	Outputs data for multi-function serial interface4
	3	SCK5E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface5
PFRF	0	CLKP OUTE	0	General-purpose port [Initial value]
			1	Outputs clock for clock monitor
PFRG	0	SCK0E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface0
	2	SOT0E	0	General-purpose port [Initial value]
			1	Outputs data for multi-function serial interface0
	3	SCK1E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface1
PFRG	5	SOT1E	0	General-purpose port [Initial value]
			1	Outputs data for multi-function serial interface1

Table 5.3-1 Initial Value and Function of PFR Register (5 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFRH	0	SCK2E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface2
	2	SOT2E	0	General-purpose port [Initial value]
			1	Outputs data for multi-function serial interface2
	3	SCK3E	0	General-purpose port [Initial value]
			1	Outputs clock for multi-function serial interface3
PFRJ	1	TOUT0E	0	General-purpose port [Initial value]
			1	Outputs data for base timer 0
	3	TOUT1E	0	General-purpose port [Initial value]
			1	Outputs data for base timer 1
	5	TOUT2E	0	General-purpose port [Initial value]
			1	Outputs data for base timer 2
PFRM	0	PPG0E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 0
	1	PPG1E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 1
	2	PPG2E	0	General-purpose port [Initial value]
			1	Outputs PPG timer 2
PFRQ	0	RTO0E	0	General-purpose port [Initial value]
			1	Outputs waveform 0 of waveform generator 0
	1	RTO1E	0	General-purpose port [Initial value]
			1	Outputs waveform 1 of waveform generator 0
	2	RTO2E	0	General-purpose port [Initial value]
			1	Outputs waveform 2 of waveform generator 0
	3	RTO3E	0	General-purpose port [Initial value]
			1	Outputs waveform 3 of waveform generator 0
	4	RTO4E	0	General-purpose port [Initial value]
			1	Outputs waveform 4 of waveform generator 0
	5	RTO5E	0	General-purpose port [Initial value]
			1	Outputs waveform 5 of waveform generator 0

Table 5.3-1 Initial Value and Function of PFR Register (6 / 6)

Register Name	Bit	Bit Name	Setting value	Function
PFRS	0	RTO6E	0	General-purpose port [Initial value]
			1	Outputs waveform 6 of waveform generator 1
	1	RTO7E	0	General-purpose port [Initial value]
			1	Outputs waveform 7 of waveform generator 1
	2	RTO8E	0	General-purpose port [Initial value]
			1	Outputs waveform 8 of waveform generator 1
	3	RTO9E	0	General-purpose port [Initial value]
			1	Outputs waveform 9 of waveform generator 1
	4	RTO10E	0	General-purpose port [Initial value]
			1	Outputs waveform 10 of waveform generator 1
	5	RTO11E	0	General-purpose port [Initial value]
			1	Outputs waveform 11 of waveform generator 1

Notes:

- The settings in PFR0 to PFR3, PFR5 and PFR6 are only valid in bus mode 1 or 2. In bus mode 0, the respective ports are always used as general-purpose ports, regardless of the setting value of PFR.
- Regardless of the settings in PFRB to PFRE, the respective ports are always used as analog inputs, when analog input has been enabled.
- PPG are also used as INT4 to INT15, external interrupts. Therefore, make sure to disable the corresponding external interrupt input before setting the respective PFR8 or PFR9, when enabling PPG to be output.
- The settings in PFRQ and PFRS are invalid when the DTIF (DTTI interrupt flag) in the waveform generator 0/1 is 1, and the respective ports are always used as general-purpose ports.

CHAPTER 6

INTERRUPT CONTROLLER

This chapter explains the overview of the interrupt controller, the configuration and functions of registers, and interrupt controller operation.

- 6.1 Overview of Interrupt Controller
- 6.2 Interrupt Controller Registers
- 6.3 Block Diagram of Interrupt Controller
- 6.4 Register Details Explanation of Interrupt Controller
- 6.5 Operation of Interrupt Controller

6.1 Overview of Interrupt Controller

The interrupt controller manages interrupt reception and arbitration processing.

■ Hardware Configuration of the Interrupt Controller

The interrupt controller consists of the following register and circuit.

- ICR register
- Interrupt priority judgment circuit
- Interrupt level and interrupt number (vector) generation unit
- Unit for generating HOLD request cancel requests

■ Major Functions

The main functions of this module are as follows.

- Detection of NMI requests and interrupt requests
- judgment of priorities (based on interrupt level and number)
- Pass the interrupt level for the interrupt selected by judgment result (to the CPU)
- Pass the interrupt number for the interrupt selected by judgment result (to the CPU)
- Send a stop mode recovery notification (to the CPU) if an NMI or interrupt with a level other than "1111_B" occurs.
- Generates the HOLD request cancel request to the DMAC.

6.2 Interrupt Controller Registers

Figure 6.2-1 shows the interrupt controller registers.

■ Interrupt Controller Registers

Figure 6.2-1 Interrupt Controller Registers

ICR00										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000440 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR01										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000441 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR02										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000442 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR03										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000443 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR04										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000444 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR05										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000445 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR06										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000446 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR07										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000447 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR08										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000448 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR09										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000449 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		
ICR10										
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
0000044A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B	
				R	R/W	R/W	R/W	R/W		

(Continued)

(Continued)

ICR11									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000044B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR12									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000044C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR13									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000044D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR14									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000044E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR15									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000044F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR16									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000450 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR17									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000451 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR18									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000452 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR19									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000453 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR20									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000454 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR21									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000455 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR22									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000456 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	
ICR23									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000457 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	

(Continued)

(Continued)

ICR24								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000458 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR25								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000459 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR26								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR27								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR28								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR29								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR30								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR31								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000045F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR32								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000460 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR33								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000461 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR34								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000462 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR35								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000463 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W
ICR36								
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00000464 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0
				R	R/W	R/W	R/W	R/W

(Continued)

(Continued)

ICR37									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000465 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR38									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000466 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR39									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000467 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR40									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000468 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR41									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000469 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR42									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR43									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR44									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR45									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR46									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
ICR47									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000046F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
				R	R/W	R/W	R/W	R/W	
HRCL									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000045 _H	MHALTI	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	0--1111 _B
	R/W			R	R/W	R/W	R/W	R/W	

R/W: Readable/writable

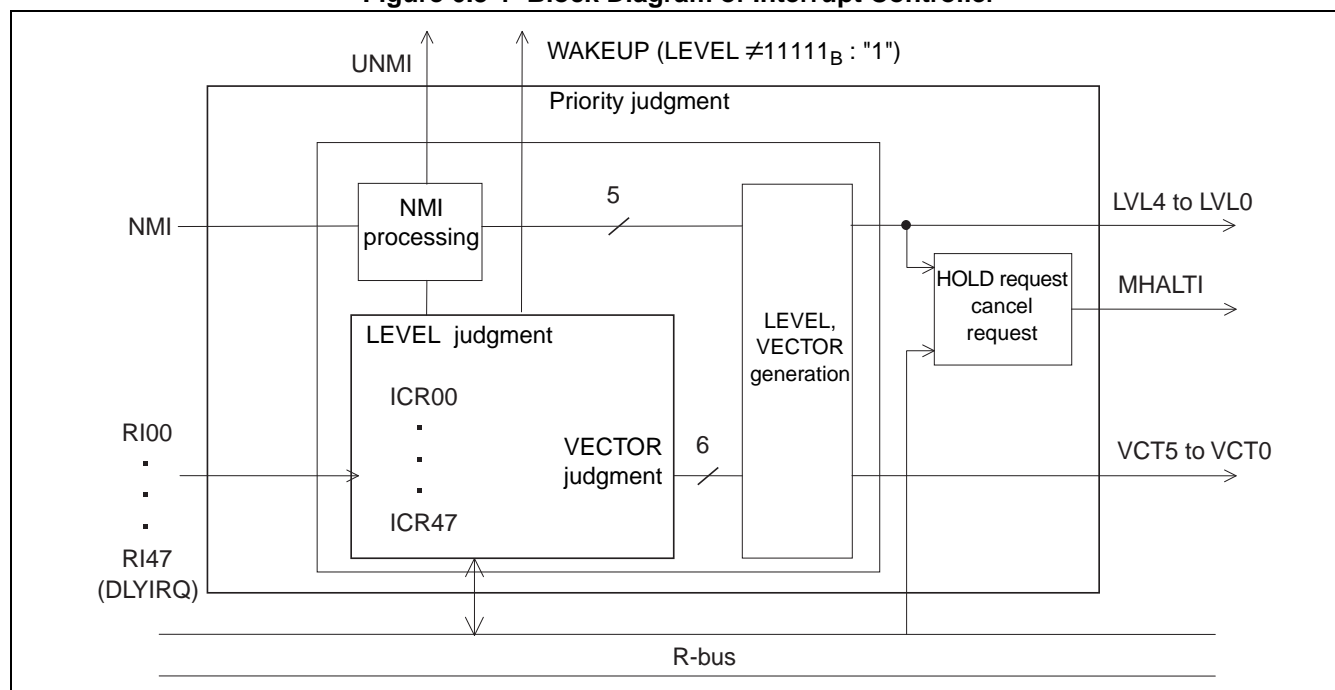
R: Read only

6.3 Block Diagram of Interrupt Controller

Figure 6.3-1 shows the block diagram of the interrupt controller.

■ Block Diagram of the Interrupt Controller

Figure 6.3-1 Block Diagram of Interrupt Controller



6.4 Register Details Explanation of Interrupt Controller

This section explains the registers used by the interrupt controller in detail.

■ Interrupt Control Register (ICR)

ICR00 to ICR47									
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ch.0 000440 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
to				R	R/W	R/W	R/W	R/W	
ch.47 00046F _H									
R/W: Readable/writable									
R: Read only									

The interrupt control register. One register is provided for each interrupt input to set the interrupt level for the corresponding interrupt request.

[bit4 to bit0] ICR4 to ICR0

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 this register is greater than or equal to the level mask value set in the ILM register of CPU.

Initialized to "1111_B" by reset.

Table 6.4-1 lists the correspondence between the interrupt level and the available interrupt level setting bits.

Table 6.4-1 Correspondence between Interrupt Levels and Available Interrupt Level Setting Bit Settings

ICR4	ICR3	ICR2	ICR1	ICR0	Interrupt level	
0	0	0	0	0	0	System reserved
0	1	1	1	0	14	
0	1	1	1	1	15	NMI
1	0	0	0	0	16	Maximum permitted level setting (High) ↑ ↓ (Low)
1	0	0	0	1	17	
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	
1	1	1	1	1	31	Interrupt disabled

ICR4 is fixed at "1". Writing "0" is not permitted.

■ Hold Request Cancel Level Register (HRCL)

HRCL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000045 _H	MHALTI	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	0--11111 _B
	R/W			R	R/W	R/W	R/W	R/W	
R/W: Readable/writable									
R: Read only									

The level setting register used to generate requests to cancel a hold request.

[bit7] MHALTI

The MHALTI bit causes DMA transfers to be halted by an NMI request. The bit is set to "1" by an NMI request and is cleared by writing "0". Always clear this bit at the end of the NMI routine in the same way as for standard interrupt routines.

[bit4 to bit0] LVL4 to LVL0

These set the interrupt level for generating a request to the bus master to cancel a hold request.

If an interrupt request with a higher-priority level than the interrupt level set in this register occurs, a request to cancel the hold request is passed to the bus master.

The LVL4 bit is fixed at "1". Writing "0" is not permitted.

6.5 Operation of Interrupt Controller

This section explains the operation of the interrupt controller.

■ Determining the Priority

This module selects the highest priority interrupt amongst any interrupt factors that occur simultaneously and outputs the interrupt level and interrupt number for the interrupt factor to the CPU.

The criteria for determining the priority of interrupt factor are as follows.

(1) NMI

(2) Interrupt factor that satisfies the following conditions

- Interrupt level is other than 31. (31 is interrupt disabled)
- Interrupt factor with lowest interrupt level value.

Of these, the interrupt factor with the lowest interrupt number.

If no interrupt factor is selected by the above criteria, 31 (1111_B) is output as the interrupt level. The interrupt number in this case is undefined.

"APPENDIX B Interrupt Vector" lists the relationship among the interrupt factor, interrupt number, and interrupt level.

■ NMI (Non Maskable Interrupt)

The NMI has the highest priority of all the interrupt factors handled by this module.

Accordingly, the NMI is always selected if it occurs at the same time as another interrupt factor.

- The following information is passed to the CPU when an NMI occurs.

Interrupt level : 15 ("0111_B")

Interrupt number : 15 ("000111_B")

- NMI detection

The external interrupts/NMI module sets and detects NMIs. In this module, an NMI request only generates the interrupt level, interrupt number, and MHALTI.

- Halt on DMA transfer by NMI

When an NMI request occurs, the MHALTI bit in the HRCL register goes to "1" to halt DMA transfer. To release the halt on DMA transfer, clear the MHALTI bit to "0" at the end of the NMI routine.

■ Hold Request Cancel Request

If you want to process high priority interrupts during a CPU hold (during DMA transfer), the module that generated the hold request needs to cancel the request. Set the interrupt level at which a request to cancel is to be generated in the HRCL register.

● Generation criteria

If an interrupt factor with a higher-priority level than the level set in the HRCL register occurs, a request to cancel the hold request is passed to the DMAC.

If interrupt level in HRCL register > level of interrupt after the priority judgment, then do generate cancel request.

If interrupt level in HRCL register ≤ level of interrupt after the priority judgment, then do not generate cancel request.

The cancel request remains active until the interrupt factor that generated the cancel request is cleared and therefore no DMA transfer occurs during this time. Always clear the associated interrupt factor.

As the MHALTI bit in the HRCL register goes to "1" when an NMI is used, the cancel request is active.

● Possible levels

The values able to be set in the HRCL register are "10000_B" to "11111_B", the same as the ICR.

If "11111_B" is set, a cancel request is generated for all interrupt levels. If "10000_B" is set, a cancel request is only generated for an NMI.

Table 6.5-1 shows the interrupt level settings for generating a request to cancel a hold request.

Table 6.5-1 Interrupt Level Settings That Generate a Hold Request Cancel Request

HRCL register	Interrupt levels that generate a cancel request
16	NMI only
17	NMI, interrupt level 16
18	NMI, interrupt levels 16 and 17
to	to
31	NMI, interrupt levels 16 to 30 [Initial value]

Once reset, DMA transfer is halted for all interrupt levels. As this means that no DMA transfer will be performed when an interrupt occurs, set the required value in the HRCL register.

■ Recovery from a Standby Mode (Stop or Sleep)

The function for using an interrupt request to recover from stop mode is performed by this module.

If one or more interrupt requests from a peripheral including NMI (with interrupt level other than "11111_B") occur, a request to recover from stop mode is sent to the clock controller.

As the priority judgment unit restarts operation once the clock supply starts after recovery from stop mode, the CPU is able to execute instructions until the priority judgment unit produces a result.

The same operation occurs when recovering from sleep mode.

Access to the registers in this module remains possible even in sleep mode.

Notes:

- The device also recovers from stop mode when an NMI request occurs. However, apply a valid input level to the NMIX pin in stop mode.
 - Set the interrupt level for interrupt factors that you do not want to cause the device to recover from stop or sleep mode to "11111_B" in the corresponding peripheral control register.
-

■ Example of Using the Function to Generate a Request to Cancel a Hold Request (HRCR)

If you want the CPU to perform high-priority processing during DMA transfer, you need to cancel the hold state by requesting the DMA to cancel its hold request. This uses an interrupt to cause the DMA to cancel its hold request and to give priority to CPU operation.

● Control register

- HRCL (Hold request cancel level setting register): this module

If an interrupt with a higher-priority level than the interrupt level set in this register occurs, a request to cancel the hold request is passed to the DMA. Sets the level to use as the criterion.

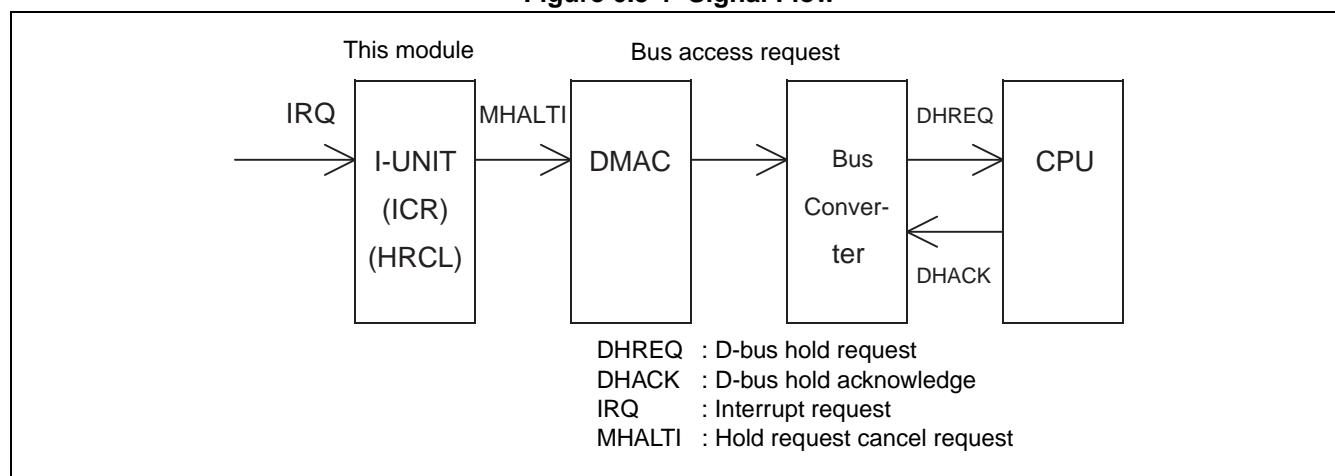
- ICR: this module

Set an interrupt level with a higher priority than the level set in the HRCL register in the ICRs of the interrupt factors you want to use.

● Hardware configuration

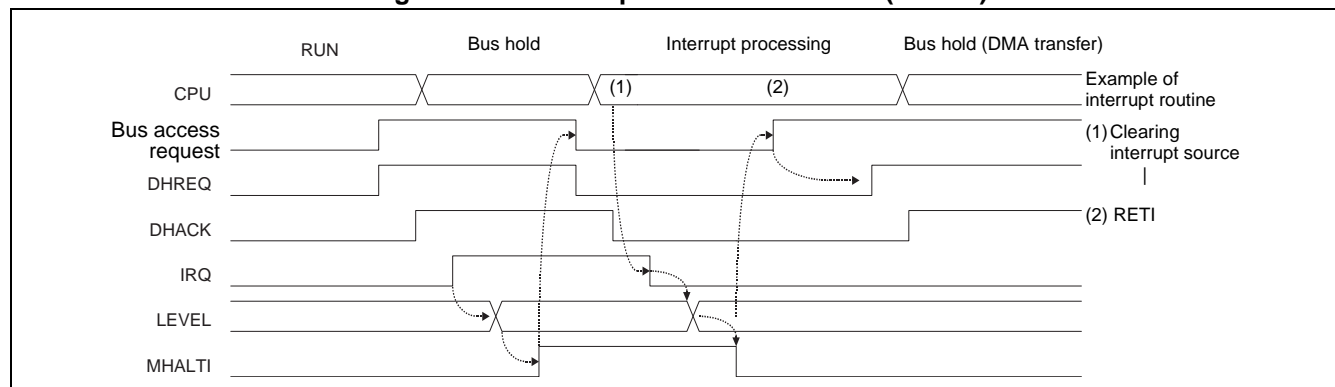
The signal flow is shown below.

Figure 6.5-1 Signal Flow



● Sequence

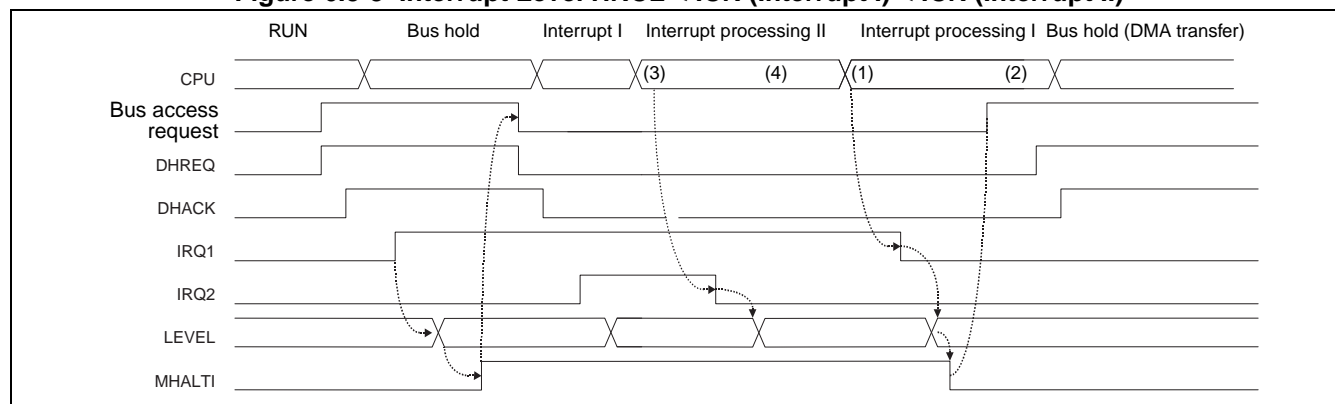
Figure 6.5-2 Interrupt Level HRCL < ICR (LEVEL)



When an interrupt request occurs and the interrupt level changes, the MHALTI signal to the DMA goes active if the new level has a higher priority than the level set in the HRCL register. This causes the DMA to halt access requests and the CPU to recover from the hold state and start processing the interrupt.

The diagram below shows the case when multiple interrupts occur.

Figure 6.5-3 Interrupt Level HRCL < ICR (Interrupt I) < ICR (Interrupt II)



Example of interrupt routine

(1), (3) Clearing interrupt source

|

(2), (4) RETI

The above example shows the case when a higher priority interrupt occurs during execution of interrupt routine I.

DHREQ becomes low while the interrupt with an interrupt level higher than the interrupt level set in the HRCL register is present.

Note:

Take note of the relationship between the interrupt levels set in the HRCL register and ICRs.

CHAPTER 7

EXTERNAL INTERRUPT AND NMI CONTROLLER

This chapter describes the overview of the external interrupt and NMI controller, the configuration and functions of registers, and operation of the external interrupt and NMI controller.

- 7.1 Overview of External Interrupt/NMI Controller
- 7.2 Registers of External Interrupt/NMI Controller
- 7.3 Operation of External Interrupt/NMI Controller

7.1 Overview of External Interrupt/NMI Controller

The external interrupt controller is a block that controls external interrupt requests input to NMIX and INT0 to INT15.

For external interrupt input, "H level", "L level", "rising edge", or "falling edge" can be selected as the level of a request to be detected.

■ Register List

Following figure shows the register list of the external interrupt/NMI controller.

External interrupt source register

EIRR0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000040 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EIRR1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000154 _H	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Interrupt enable register

ENIR0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000041 _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ENIR1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000155 _H	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

External interrupt request level setting register

ELVR0

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
00000042 _H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ELVR1

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
00000156 _H	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ELVR0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000043 _H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ELVR1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000157 _H	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

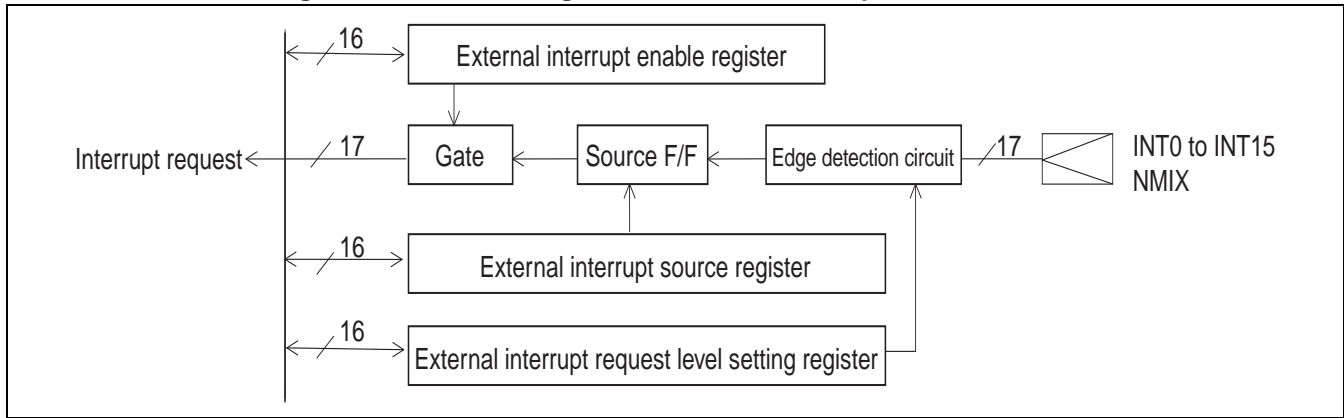
R/W: Readable/writable

MB91470/480 Series

■ Block Diagram of External Interrupt/NMI Controller

Figure 7.1-1 shows a block diagram of external interrupt/NMI controller.

Figure 7.1-1 Block Diagram of External Interrupt/NMI Controller



7.2 Registers of External Interrupt/NMI Controller

This section explains the configuration and functions of registers used by external interrupt/NMI controller.

■ Interrupt Enable Register (ENIR (ENIR0, ENIR1): ENable Interrupt Request Register)

ENIR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000041 _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ENIR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000155 _H	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable									

ENIR register controls mask of the external interrupt request output. Output for an interrupt request is enabled based on the bit in this register to which "1" has been written (INT0 enable is controlled by EN0), and the interrupt request is output to the interrupt controller. The pin corresponding to the bit to which "0" is written holds the interrupt source but does not generate a request to the interrupt controller.

No enable bit exists for NMI.

■ External Interrupt Source Register (EIRR (EIRR0, EIRR1): External Interrupt Request Register)

EIRR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000040 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
EIRR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000154 _H	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable									

EIRR register is a register that shows a corresponding external interrupt request exists when reading, and that clears a content of the flip-flop showing this request when writing.

If the read value of this EIRR register is "1", there is an external interrupt request at the pin corresponding to this bit. Write "0" to this register to clear the request flip-flop of the corresponding bit.

Writing "1" to this register is invalid.

"1" is read in a read operation of the read modify write (RMW) instruction.

The flag for NMI cannot be accessed by a user.

MB91470/480 Series

■ External Interrupt Request Level Setting Register (ELVR (ELVR0, ELVR1): External LeVel Register)

ELVR0									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
00000042 _H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ELVR1									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
00000156 _H	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ELVR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000043 _H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ELVR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
00000157 _H	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable

ELVR is a register to select request detections. In ELVR, two bits each are assigned to INT0 to INT15, which results in the settings shown in table below. When each bit of the EIRR is cleared while the level is in the request input level, an appropriate bit is set again as long as the input is at active level.

Table 7.2-1 Assignment of ELVR

LBx	LAx	Operation
0	0	"L" level indicates the existence of a request
0	1	"H" level indicates the existence of a request
1	0	A rising edge indicates the existence of a request
1	1	A falling edge indicates the existence of a request

Detection level of NMI is always a falling edge level.

Also, when using NMI to return from the stop state, detection level is "L" level.

Note:

If the external interrupt request level is changed, it may cause the internal interrupt. So, it is necessary to clear the external interrupt cause register (EIRR) after reading the external interrupt request level register.

Before clearing the external interrupt cause register, read out the external interrupt request register for writing clear.

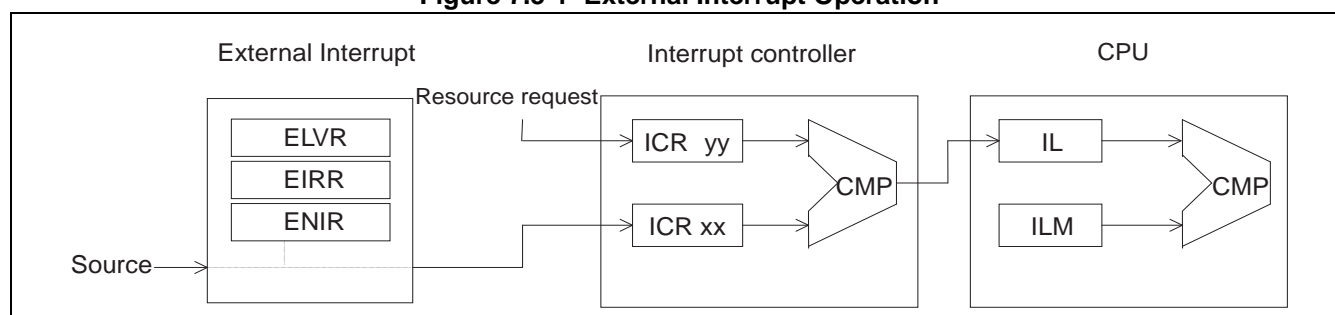
7.3 Operation of External Interrupt/NMI Controller

If, after a request level and an enable register are defined, a request defined in the ELVR register is input to the corresponding pin, this module generates an interrupt request signal to the interrupt controller. For simultaneous interrupt requests from resources, the interrupt controller determines the interrupt request with the highest priority and generates an interrupt for it.

■ Operation of an External Interrupt

Figure 7.3-1 shows the external interrupt operation.

Figure 7.3-1 External Interrupt Operation



■ Return from Standby

Be sure to disable the channel that is not used before entering to standby.

■ Operating Procedure for an External Interrupt

Set up a register located inside the external interrupt controller as follows:

1. Set that general-purpose I/O port as an input port which also serves as a pin to be used as an external interrupt input.
2. Disable the target bit in the interrupt enable register (ENIR).
3. Set the target bit in the external interrupt request level setting register (ELVR).
4. Read the external interrupt request level setting register (ELVR).
5. Clear the target bit in the external interrupt source register (EIRR).
6. Enable the target bit in the interrupt enable register (ENIR).
 (Simultaneous writing of 16-bit data is supported for steps 5. and 6.)

Before setting a register in this module, you must disable the enable register. In addition, before enabling the enable register, you must clear the interrupt source register. This procedure is required to prevent an interrupt source from occurring by mistake while a register is being set or an interrupt is enabled.

■ External Interrupt Request Level

- If the request level is an edge request, a pulse width of at least peripheral clock is required to detect an edge.
- If the request input level is a level setting, a request input is entered from outside and is then cancelled, the request to the interrupt controller remains active because a source holding circuit exists internally. When the request input level is a level setting, pulse widths must be more than 3 machine cycles. Moreover, even if the factor register is cleared, the interrupt request to the interrupt controller keeps being generated as long as the interrupt input terminal maintains the active level.

The external interrupt source register must be cleared to cancel a request to the interrupt controller.

Figure 7.3-2 Clearing the External Interrupt Source Register when a Level is Set

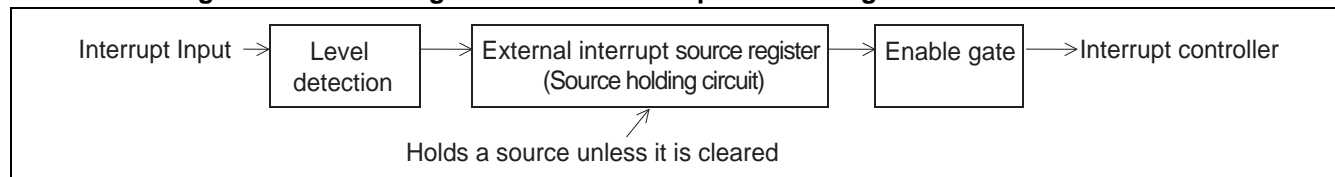
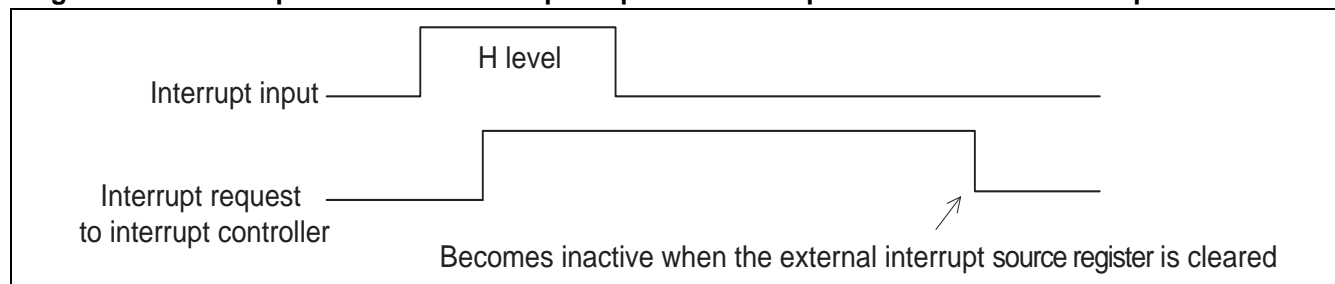


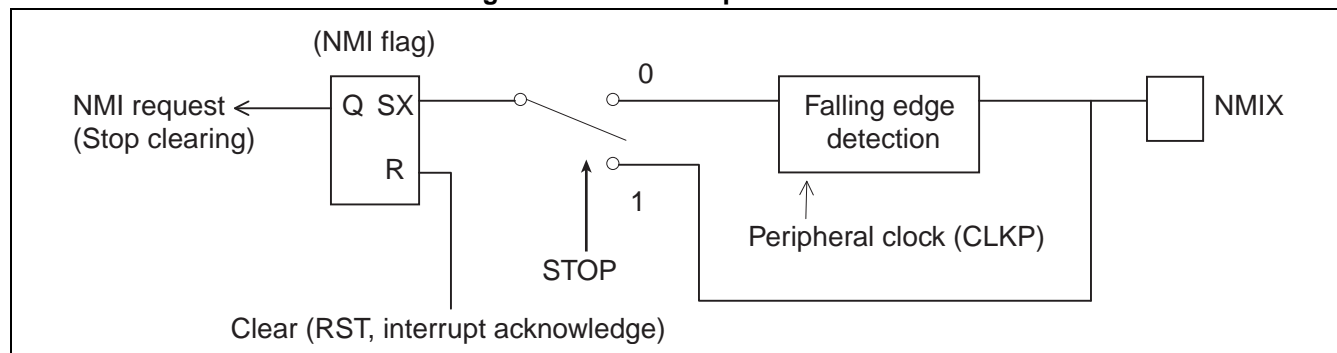
Figure 7.3-3 Interrupt Source and Interrupt Request to Interrupt Controller when Interrupts are Enabled



■ NMI

- An NMI has the highest level among the user interrupts and cannot be masked.
However, as an exception, when NMI is activated without setting ILM, NMI source is detected but CPU will not accept the NMI request. At this time, the NMI source will be held until ILM is set to be accepted by NMI. For this reason, use NMI after resetting and setting ILM value to 16 or higher. Also, since an internal source flag of NMI cannot be accessed from CPU, keep NMIX pin to "H" level after reset.
- An NMI is accepted under the following conditions:
Normal state:Falling edge
STOP state:"L" level
- An NMI can be used to clear stop mode. Inputting the "L" level in the stop state clears the stop state and causes the oscillation stabilization wait time to start.
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. Note that this bit is not readable or writable.

Figure 7.3-4 NMI Request Detector

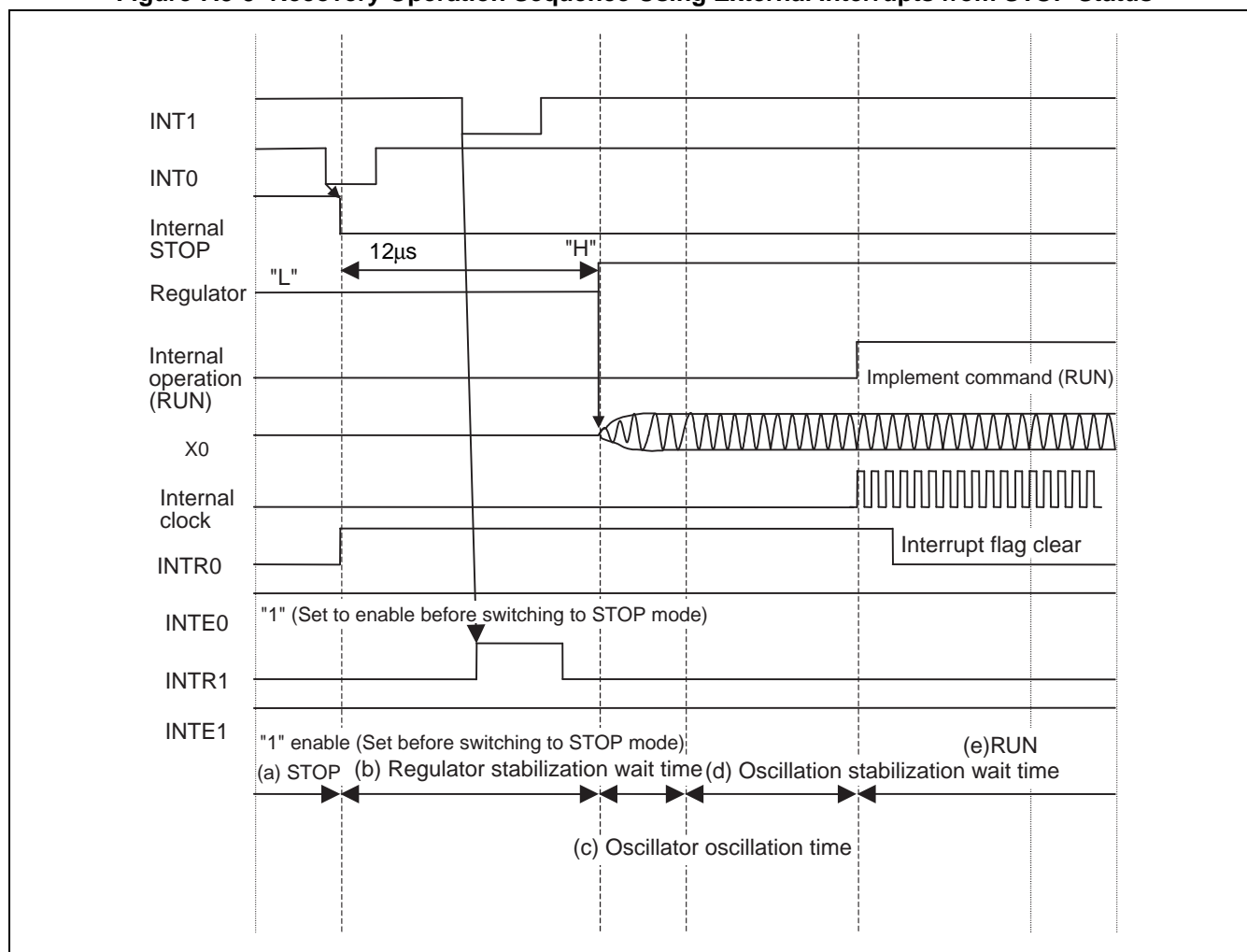


■ Notes If Restoring from STOP Status Performed Using an External Interrupt

During STOP status, external interrupt signals that are first entered to the INT terminal are entered asynchronously, to enable recovery from the STOP status. The period from that STOP being released to the passage of oscillation stabilization wait time contains a period that cannot identify the other external interrupt signal inputs (Period b+c+d for Figure 7.3-5.). To synchronize external interrupt signals after the STOP has been released with the internal clock, while the clock is not stable, interrupt request cannot be stored.

Consequently, if sending external interrupt inputs after the STOP has been released, input external interrupt signals after the oscillation stabilization wait time has elapsed.

Figure 7.3-5 Recovery Operation Sequence Using External Interrupts from STOP Status



■ Recovery Operations from STOP Status

The STOP recovery operation using external interrupts from existing circuits is performed as described below.

● Processing before changing to STOP status

• External Interrupt Process Configuration

It is necessary to set the external interrupt input process to release STOP status before the device transits to STOP status. These configuration are made using the PFR register (Port Function Register) and ENIR register (ENable Interrupt Register). Under normal conditions (i.e., any status other than STOP), the interrupt input process is authorized, so there is no need for special recognition. In STOP status, however, the input path is controlled by the PFR register value.

Pin name used for STOP release	Setting registers and bit
P97/INT15/PPG15	Set PFR9 bit7 to "0".
P96/INT14/PPG14	Set PFR9 bit6 to "0".
P95/INT13/PPG13	Set PFR9 bit5 to "0".
P94/INT12/PPG12	Set PFR9 bit4 to "0".
P93/INT11/PPG11	Set PFR9 bit3 to "0".
P92/INT10/PPG10	Set PFR9 bit2 to "0".
P91/INT9/PPG9	Set PFR9 bit1 to "0".
P90/INT8/PPG8	Set PFR9 bit0 to "0".
P87/INT7/PPG7	Set PFR8 bit7 to "0".
P86/INT6/PPG6	Set PFR8 bit6 to "0".
P85/INT5/PPG5	Set PFR8 bit5 to "0".
P84/INT4/PPG4	Set PFR8 bit4 to "0".

• External Interrupt Inputs

If recovering from STOP status, the external interrupt signals are asynchronous and send the input signal. When this interrupt signal is enabled, the internal STOP signal is immediately turned OFF. At the same time, the external interrupt circuit is switched so as to synchronize other level interrupt inputs.

● Oscillator Oscillation Time

After the regulator stabilization wait time has ended, the clock will start to oscillate. The oscillator oscillation time depends on the used oscillator.

● Oscillation Stabilization Wait Time

After the oscillator oscillation time, an oscillation stabilization wait time is taken inside the device. The oscillation stabilization wait time is specified by bits OS1 and OS0 on the standby control register. After the oscillation stabilization wait time has ended, the internal clock is supplied, and in addition to the activation of interrupt command operations from the external interrupt, it also becomes possible to receive external interrupt requests other than the recovery from STOP request.

CHAPTER 8

REALOS-RELATED HARDWARE

The REALOS-related hardware is used by the real-time OS. Accordingly, these functions cannot be used by user programs if using REALOS.

This chapter explains the overview of the delayed interrupt module and bit search module, the configuration and functions of the registers, and the operation of the delayed interrupt module and bit search module.

8.1 Delayed Interrupt Module

8.2 Bit Search Module

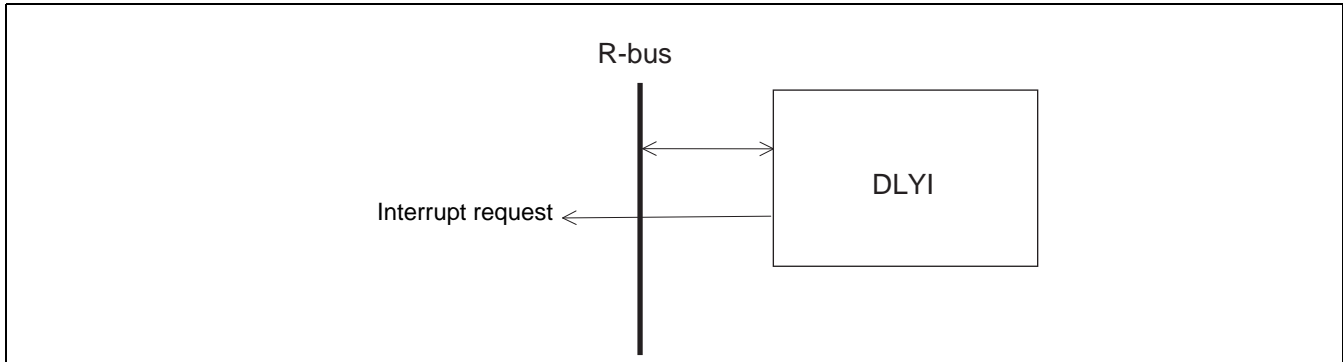
8.1 Delayed Interrupt Module

The delayed interrupt module is used to generate the interrupt for task switching. An interrupt request to the CPU can be generated and cleared by software using this module.

■ Delayed Interrupt Module Registers

DICR										
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000044 _H	-	-	-	-	-	-	-	DLYI	-----0 _B	
								R/W		
R/W: Readable/writable										

■ Block Diagram of Delayed Interrupt Module



■ Register Details Explanation

● DICR (Delayed Interrupt Control Register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
00000044 _H	-	-	-	-	-	-	-	DLYI	-----0 _B	
								R/W		
R/W: Readable/writable										

This register controls the delayed interrupt.

[bit0] DLYI

DLYI	Description
0	Delayed interrupt source cleared or no request present [Initial value]
1	Delayed interrupt source occurred

This bit controls generation and clearing of the interrupt source.

■ Operation Explanation

The delayed interrupt is used to generate the interrupt for task switching. An interrupt request to the CPU can be generated and cleared by software using this function.

● Interrupt number

The delayed interrupt is assigned to the interrupt source corresponding to the highest interrupt number.

On MB91470/480 series, the delayed interrupt has interrupt number 63 (3F_H).

● DLYI Bit of DICR

Writing "1" to this bit generates a delayed interrupt source. Similarly, writing "0" to this bit clears the delayed interrupt source.

This bit functions like a standard interrupt source flag and should be cleared in the interrupt routine at the same time as performing task switching.

8.2 Bit Search Module

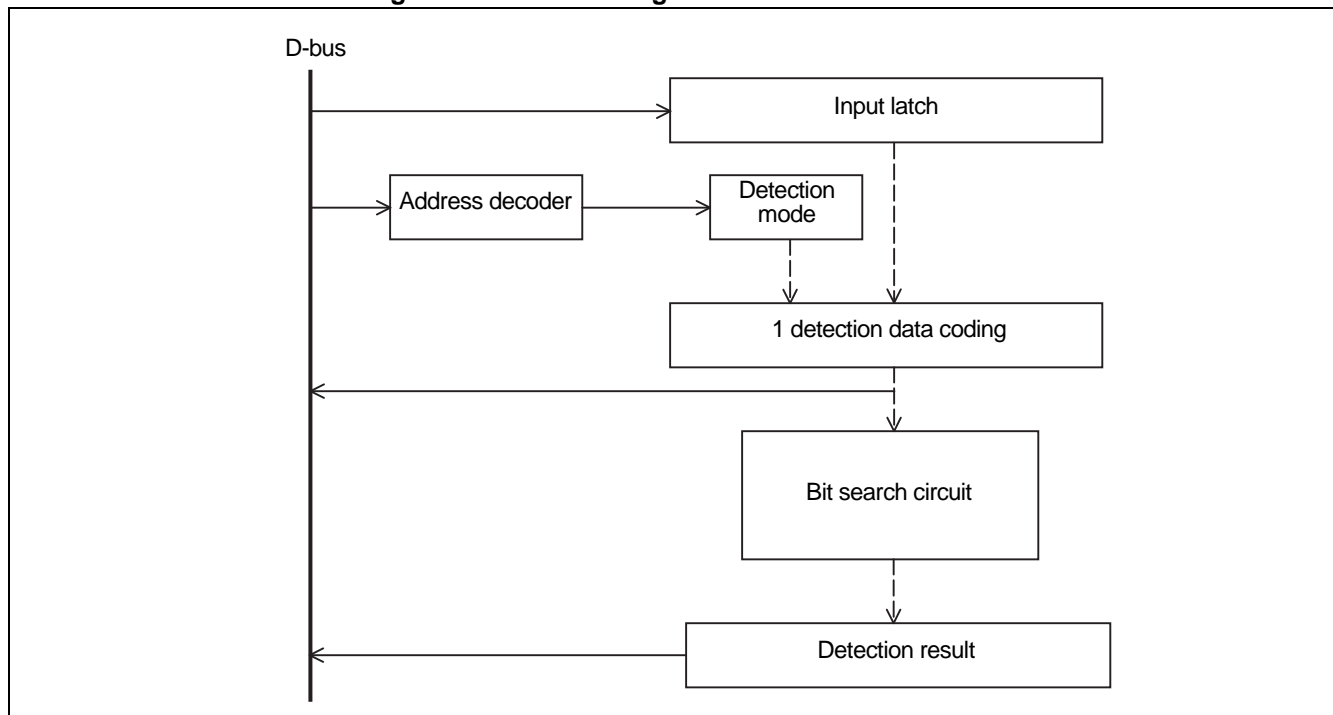
Searches for a zero, one, or change point in the data written to the input register and returns the detected bit position.

■ Bit Search Module Registers

	bit 31	bit0	
Address: 000003F0 _H	BSD0		Zero-detect data register
Address: 000003F4 _H	BSD1		One-detect data register
Address: 000003F8 _H	BSDC		Change point detection data register
Address: 000003FC _H	BSRR		Detection result register

■ Block Diagram of Bit Search Module

Figure 8.2-1 Block Diagram of Bit Search Module



MB91470/480 Series

■ Register Details Explanation

● Zero-detect data register (BSD0)

Address	bit31	bit0
000003F0 _H		
Attribute	→ Write only	
Initial value	→ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B	

Detects "0" in the value written.

The initial value by reset is irregular. The read value is undefined.

Use a 32-bit data transfer instruction to transfer the data

(Do not use 8-bit or 16-bit data transfer instructions).

● One-detect data register (BSD1)

Address	bit31	bit0
000003F4 _H		
Attribute	→ Readable/Writable	
Initial value	→ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B	

Use a 32-bit data transfer instruction to transfer the data

(Do not use 8-bit or 16-bit data transfer instructions).

- Writing

Detects "1" in the value written.

- Reading

Reads the data to enable the internal state of the bit search module to be saved. This is used to save and restore the original state when the bit search module is used by an interrupt handler or similar.

Saving and restoring can be performed using only the one-detect data register even if data is written to the 0-detect or change point detection data registers.

The initial value by reset is irregular.

● Change point detection data register (BSDC)

Address	bit31	bit0
000003F8 _H		
Attribute	→ Write only	
Initial value	→ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B	

Detects the change point in the value written.

The initial value by reset is irregular.

The read value is undefined.

Use a 32-bit data transfer instruction to transfer the data

(Do not use 8-bit or 16-bit data transfer instructions).

● Detection result register (BSRR)

Reads the result of the zero-detect, one-detect, or change point detect operation.

Which result is read from this register is determined by which data register was written to most recently.

Address	bit31	bit0
000003FC _H		
Attribute	→ Read only	
Initial value	→ XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B	

■ Operation Explanation

● 0 detection

The module scans data written to the zero-detect data register from the MSB to the LSB and returns the position of the first "0".

The result is obtained by reading the detection result register.

Table 8.2-1 shows the relationship between the returned value and the detected position.

If no "0" exists (if the value is "FFFFFFFF_H"), 32 is returned as the search result.

[Execution example]

Write data		Read data (decimal)
111111111111111110000000000000 _B (FFFFF000 _H)	→	20
11111000010010011110000010101010 _B (F849E0AA _H)	→	5
100000000000001010101010101010 _B (8002AAAA _H)	→	1
111111111111111111111111111111 _B (FFFFFFFF _H)	→	32

● 1 detection

The module scans data written to the one-detect data register from the MSB to the LSB and returns the position of the first "1".

The result is obtained by reading the detection result register.

Table 8.2-1 shows the relationship between the returned value and the detected position.

If no "1" exists (if the value is "00000000_H"), 32 is returned as the search result.

[Execution example]

Write data		Read data (decimal)
001000000000000000000000000000 _B (20000000 _H)	→	2
00000001001000110100010101100111 _B (01234567 _H)	→	7
000000000000001111111111111111 _B (0003FFFF _H)	→	14
0000000000000000000000000000001 _B (00000001 _H)	→	31
000000000000000000000000000000 _B (00000000 _H)	→	32

MB91470/480 Series

● Change point detection

The data written to the change point detection data register is scanned from bit30 to the LSB and compared with the MSB. The position of the first bit with a value different to the MSB is returned.

The result is obtained by reading the detection result register.

Table 8.2-1 shows the returned value and detected position.

If no change point exists, 32 is returned as the search result.

The change point detection function never returns a result of zero.

[Execution example]

Write data		Read data (decimal)
00100000000000000000000000000000 _B (20000000 _H)	→	2
00000001001000110100010101100111 _B (01234567 _H)	→	7
00000000000000111111111111111111 _B (0003FFFF _H)	→	14
00000000000000000000000000000001 _B (00000001 _H)	→	31
00000000000000000000000000000000 _B (00000000 _H)	→	32
11111111111111111111000000000000 _B (FFFFFF00 _H)	→	20
11111000010010011110000010101010 _B (F849E0AA _H)	→	5
10000000000000101010101010101010 _B (8002AAAA _H)	→	1
11111111111111111111111111111111 _B (FFFFFFFF _H)	→	32

Table 8.2-1 Bit Position and Return Value (Decimal)

Detected bit position	Return value	Detected bit position	Return value	Detected bit position	Return value	Detected bit position	Return value
31	0	23	8	15	16	7	24
30	1	22	9	14	17	6	25
29	2	21	10	13	18	5	26
28	3	20	11	12	19	4	27
27	4	19	12	11	20	3	28
26	5	18	13	10	21	2	29
25	6	17	14	9	22	1	30
24	7	16	15	8	23	0	31
						Not exist	32

■ Backup/Restore Processing

When it is necessary to backup and restore the internal state of the bit search module such as when using the bit search module in an interrupt handler, always use the following procedure.

- (1) Read the one-detect data register and save the value. (backup)
- (2) Use the bit search module.
- (3) Write the data saved in step (1) to the one-detect data register (restore).

This ensures that the value returned when the detection result register is next read will be based on the value written to the bit search module prior to step (1). This procedure will correctly restore the result, even if the data register written to previously was the zero-detect or change point detect data register.

CHAPTER 9

16-BIT RELOAD TIMER

This chapter describes the overview of the reload timer, the configuration and functions of registers, and the reload timer operation.

- 9.1 Overview of 16-bit Reload Timer
- 9.2 16-bit Reload Timer Register
- 9.3 Operation of 16-bit Reload Timer

9.1 Overview of 16-bit Reload Timer

The 16-bit reload timer consists of a 16-bit down-counter, 16-bit reload register, internal count, clock generation prescaler, and control register.

■ Overview of 16-bit Reload Timer

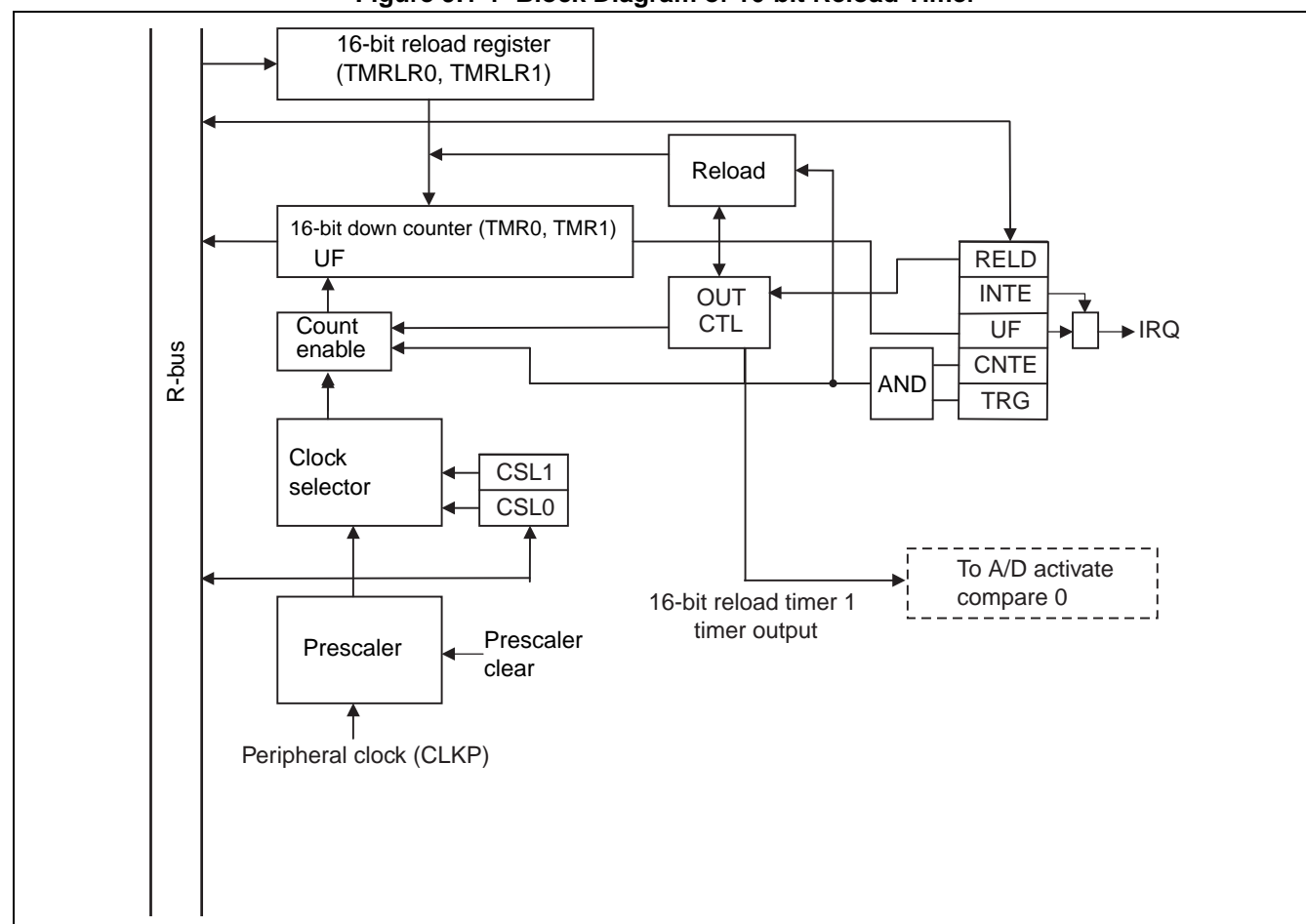
The 16-bit reload timer consists of a 16-bit down-counter, 16-bit reload register, internal count, clock generation prescaler, and control register.

The clock source can be selected from three internal clocks (peripheral clock (CLKP) divided by 2, 8, 32) .

■ Block Diagram of the 16-bit Reload Timer

Figure 9.1-1 shows the block diagram of the 16-bit reload timer.

Figure 9.1-1 Block Diagram of 16-bit Reload Timer



The only 16-bit reload timer 1 timer output can be used as the activation cause of A/D converter. The following A/D converter is activated.

- MB91470 series: 12-bit A/D converter 4
- MB91480 series: 10-bit A/D converter 1

MB91470/480 Series

9.2 16-bit Reload Timer Register

This section describes the configuration and functions of the 16-bit reload timer registers.

■ List of 16-bit Reload Timer Registers

TMCSR0, TMCSR1 high byte

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 004E _H 0000 0056 _H	—	—	—	—	CSL1	CSL0	—	—	---00-- _B
	—	—	—	—	R/W	R/W	—	—	

TMCSR0, TMCSR1 low byte

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000 004F _H 0000 0057 _H	—	—	—	RELD	INTE	UF	CNTE	TRG	---00000 _B
	—	—	—	R/W	R/W	R/W	R/W	R/W	

TMR0, TMR1

Address	bit15	bit0	Initial value
0000 004A _H 0000 0052 _H	<div></div>		XXXX _H
	R		

TMRLR0, TMRLR1

Address	bit15	bit0	Initial value
0000 0048 _H 0000 0050 _H			XXXX _H
	W		

R/W: Readable/writable

R: Read only

W: Write only

9.2.1 Control Status Register (TMCSR)

The control status register (TMCSR) controls the operation mode and interrupt of the 16-bit reload timer.

■ Bit Configuration of Control Status Register (TMCSR)

TMCSR0, TMCSR1 high byte									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 004E _H	—	—	—	—	CSL1	CSL0	—	—	----00-- _B
0000 0056 _H	—	—	—	—	R/W	R/W	—	—	

TMCSR0, TMCSR1 low byte									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000 004F _H	—	—	—	RELD	INTE	UF	CNTE	TRG	---00000 _B
0000 0057 _H	—	—	—	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

[bit15 to bit12] Reserved: Reserved bits

Reading of these bits always returns "0000_B".

[bit11, bit10] CSL1,CSL0: Count source select bits

These are the count source selection bits. The count source can select the internal clock. The available count sources are as follows.

CSL1	CSL0	Count Source (ϕ : Peripheral clock)		$\phi=40\text{MHz}$	$\phi=20\text{MHz}$
0	0	Internal clock	$\phi/2^1$ [Initial value]	50ns	100ns
0	1	Internal clock	$\phi/2^3$	200ns	400ns
1	0	Internal clock	$\phi/2^5$	800ns	1.6 μs
1	1	Setting prohibited		-	-

[bit9 to bit7] Reserved: Reserved bits

Please set to "000_B".

[bit6, bit5] Reserved: Reserved bits

Reading of these bits always returns "0".

[bit4] RELD: Reload enable bit

The reload enable bit. Setting "1" sets reload mode. In this case, the contents of the reload register are loaded to the counter and the count continues when the counter underflows from "0000_H" to "FFFF_H".

Setting "0" sets one-shot mode. In this case, the count halts when the counter underflows from "0000_H" to "FFFF_H".

[bit3] INTE: Interrupt enable bit

The interrupt request enable bit. If this bit is set to "1", an interrupt request is generated when the UF bit is "1". No interrupt request is generated if this bit is set to "0".

[bit2] UF: Underflow interrupt flag

The timer interrupt request flag. This bit is set to "1" when the counter underflows from "0000_H" to "FFFF_H". Writing "0" to this bit clears it.

Writing "1" to this bit has no meaning.

Read modify write (RMW) instructions always read the bit as "1".

[bit1] CNTE: Count enable bit

The count enable bit for the timer. Writing "1" to this bit sets the timer to wait for a start trigger. Writing "0" halts the count.

[bit0] TRG: Trigger bit

The software trigger bit. Writing "1" to this bit generates a software trigger which loads the contents of the reload register to the counter and starts the count.

Writing "0" to this bit has no meaning. Reading value is always "0".

The trigger input in this register is only meaningful when CNTE=1. The trigger has no effect if CNTE=0.

Note:

Only modify the bits other than UF, CNTE, and TRG when CNTE = 0.

9.2.2 16-bit Timer Register (TMR)

The 16-bit timer register (TMR) is used to read the count value of the 16-bit timer.

■ Bit Configuration of 16-bit Timer Register (TMR)

TMR0, TMR1			
Address	bit15	bit0	Initial value
0000 004A _H			XXXX _H
0000 0052 _H			
	R		
R: Read only			

The count value of the 16-bit timer can be read from this register. The initial value is undefined. Always use a 16-bit data transfer instruction to read this register.

9.2.3 16-bit Reload Register (TMRLR)

16-bit reload register (TMRLR) stores the initial value of the counter.

■ Bit Configuration of 16-bit Reload Register (TMRLR)

TMRLR0, TMRLR1			
Address	bit15	bit0	Initial value
0000 0048 _H	<div></div>		XXXX _H
0000 0050 _H			
W			
W: Write only			

This register stores the initial value of the count. The initial value is irregular. Always use a 16-bit data transfer instruction to write to this register.

9.3 Operation of 16-bit Reload Timer

This section explains the internal clock operation and underflow operation of the reload timer.

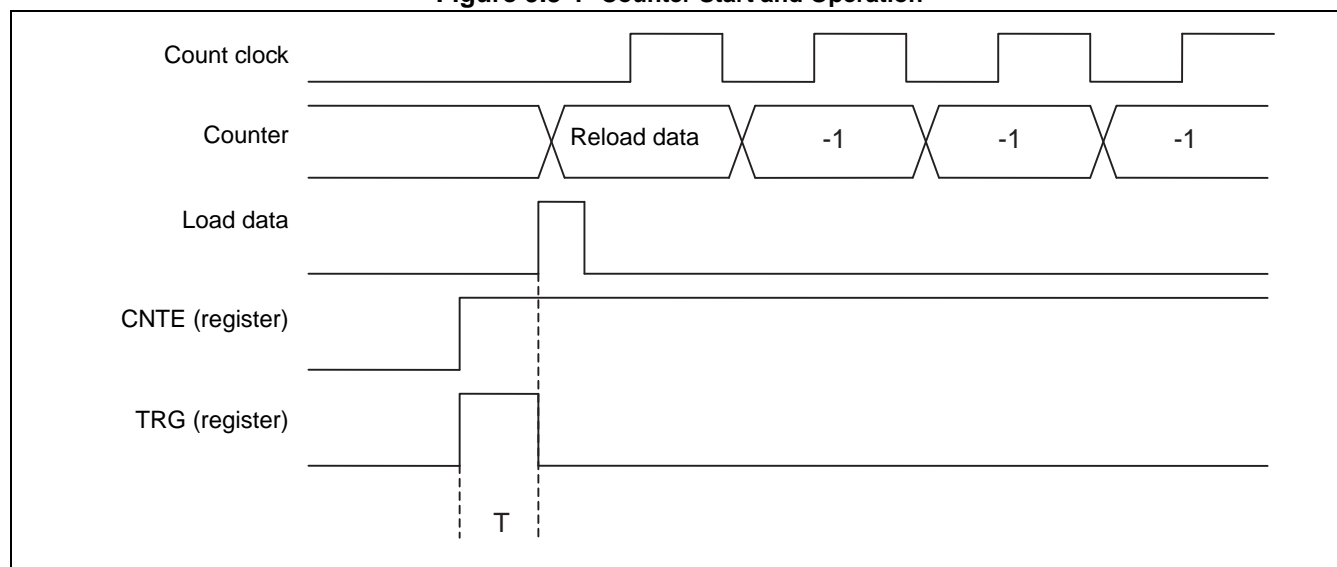
■ Internal Clock Operation

When the timer is driven by the divided internal clock, the count source can be selected from the peripheral clock divided by 2, 8 or 32.

If you wish to enable and start the count at the same time, write "1" to both the CNTE bit and TRG bit in the control status register. The TRG bit trigger input always functions regardless of the operation mode, provided the timer is enabled (CNTE = 1).

The time between input of a counter start trigger and the data in the reload register being loaded to the counter is T (T: cycle of peripheral clock).

Figure 9.3-1 Counter Start and Operation

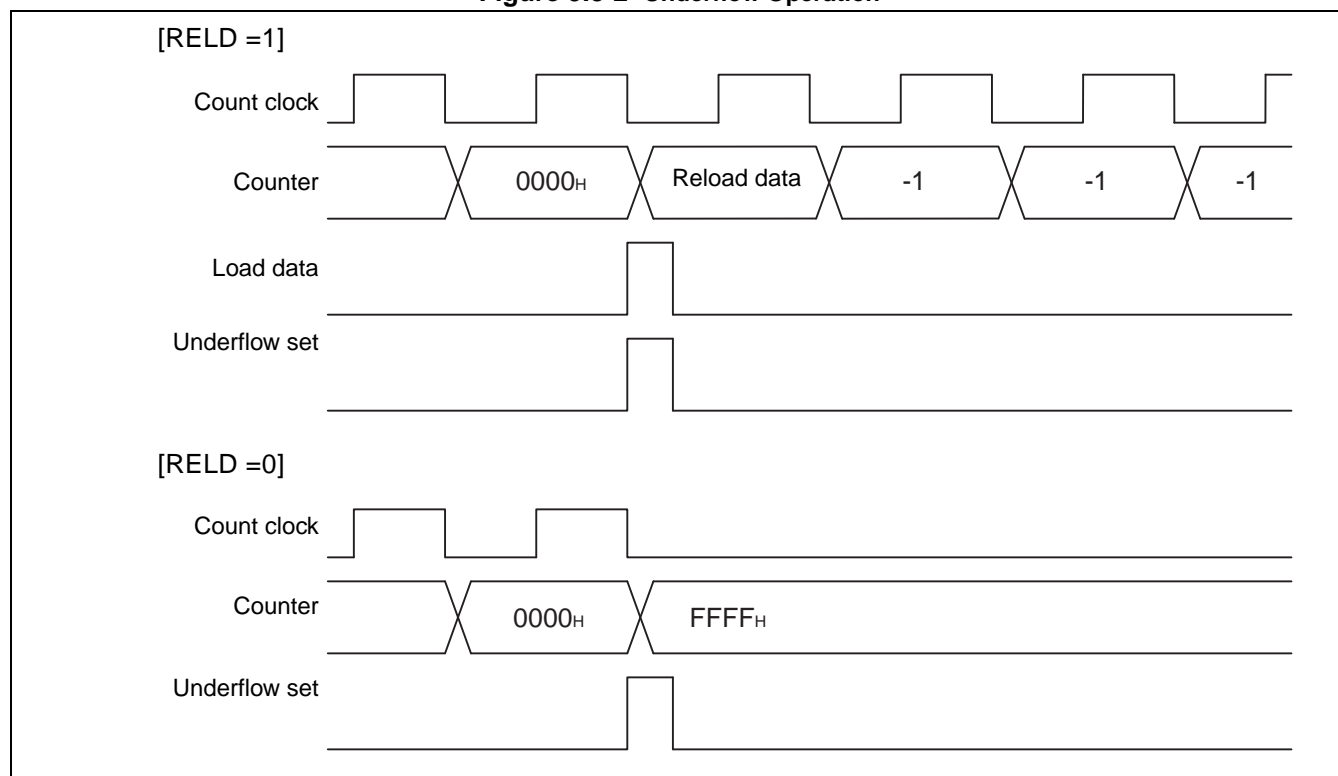


■ Underflow Operation

An underflow on this timer is defined as when the counter goes from "0000_H" to "FFFF_H". Accordingly, an underflow occurs after (reload register setting value + 1) counts.

If the RELD bit in the control status register is "1" when the underflow occurs, the contents of the reload register are loaded to the counter and the count continues. If the RELD bit is "0", the counter halts at "FFFF_H".

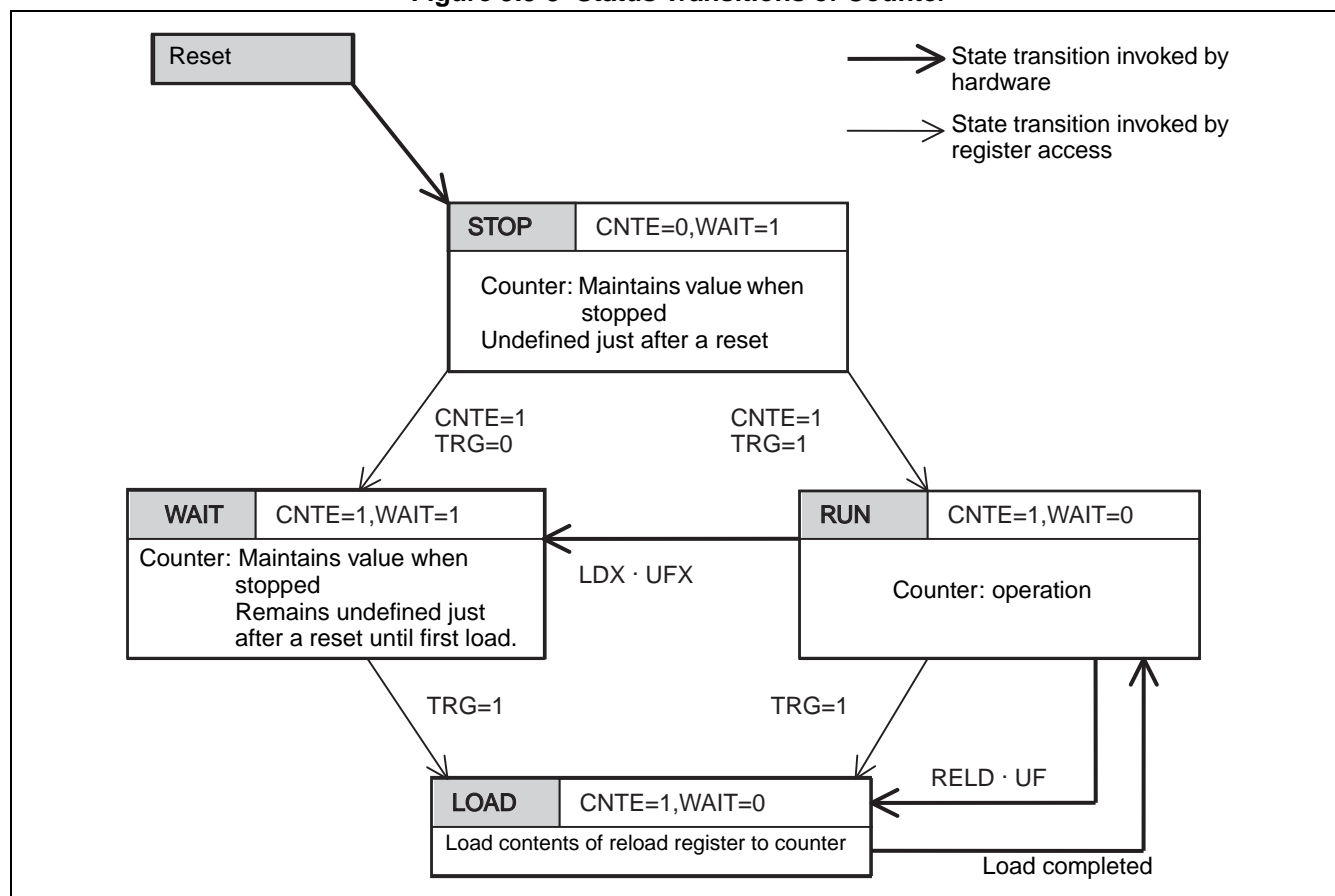
Figure 9.3-2 Underflow Operation



■ Counter Operation States

The counter state is determined by the CNTE bit in the control status register and by the internal WAIT signal. The states that can be set include the stop state, when CNTE=0 and WAIT=1 (STOP state); the startup trigger wait state, when CNTE=1 and WAIT=1 (WAIT state); and the operation state, when CNTE=1 and WAIT=0 (RUN state).

Figure 9.3-3 Status Transitions of Counter



■ Precautions

- Operation of the internal prescaler is enabled when a trigger (software trigger or external trigger) occurs while bit1 of the control status register (timer enable: CNTE) is set to "1".
- If the interrupt request flag is set and cleared at the same timing, the flag set operation has precedence and the clear operation is ignored.
- If writing to the 16-bit timer reload register occurs at the same time as a reload timing, the old data is loaded to the counter and the new data is not loaded to the counter until the next reload timing.
- If a 16-bit timer register load occurs at the same time as a count, the load (reload) operation has precedence.

CHAPTER 10

TIMING GENERATOR

This chapter explains the overview of the timing generator, the configuration and functions of registers, and operation of the timing generator.

- 10.1 Overview of Timing Generator
- 10.2 Block Diagram of Timing Generator
- 10.3 Registers of Timing Generator
- 10.4 Operation of Timing Generator

10.1 Overview of Timing Generator

The timing generator is to activate the delay synchronization for multiple PPG timers.
The MB91470 series has one timing generator.
The MB91480 series has two timing generators.

■ Configuration of Timing Generator

- This generator comprises an 8-bit counter, control register, compare registers, compare circuits and a prescaler.
- It can activate the delay synchronization for 4 channels of PPG.
- Four counter operation clocks (peripheral clock(CLKP)/2, peripheral clock(CLKP)/8, peripheral clock(CLKP)/32 and peripheral clock(CLKP)/64) are available for selection.
- The amount of delay can be set by setting four compare registers corresponding to each PPG channel.

■ Differences between Timing Generators 0 and 1

● Timing generator 0

- Compare registers: : COMP0/COMP2/COMP4/COMP6
- Corresponding PPG channels : ch.0/ch.2/ch.4/ch.6

● Timing generator 1

- Compare registers : COMP1/COMP3/COMP5/COMP7
- Corresponding PPG channels : ch.8/ch.10/ch.12/ch.14

10.2 Block Diagram of Timing Generator

This section shows block diagrams of the timing generators.

Figure 10.2-1 Block Diagram (Timing Generator 0)

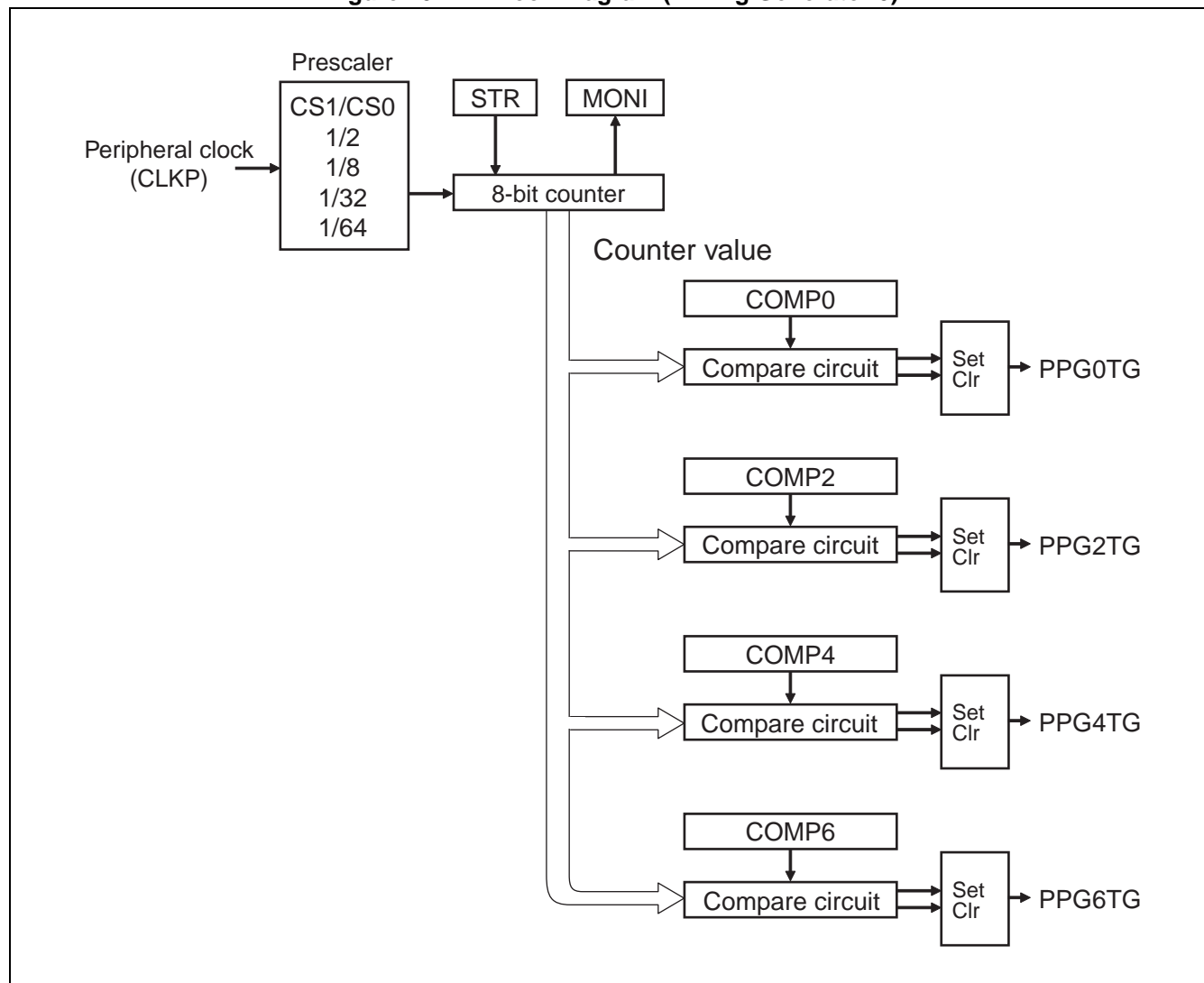
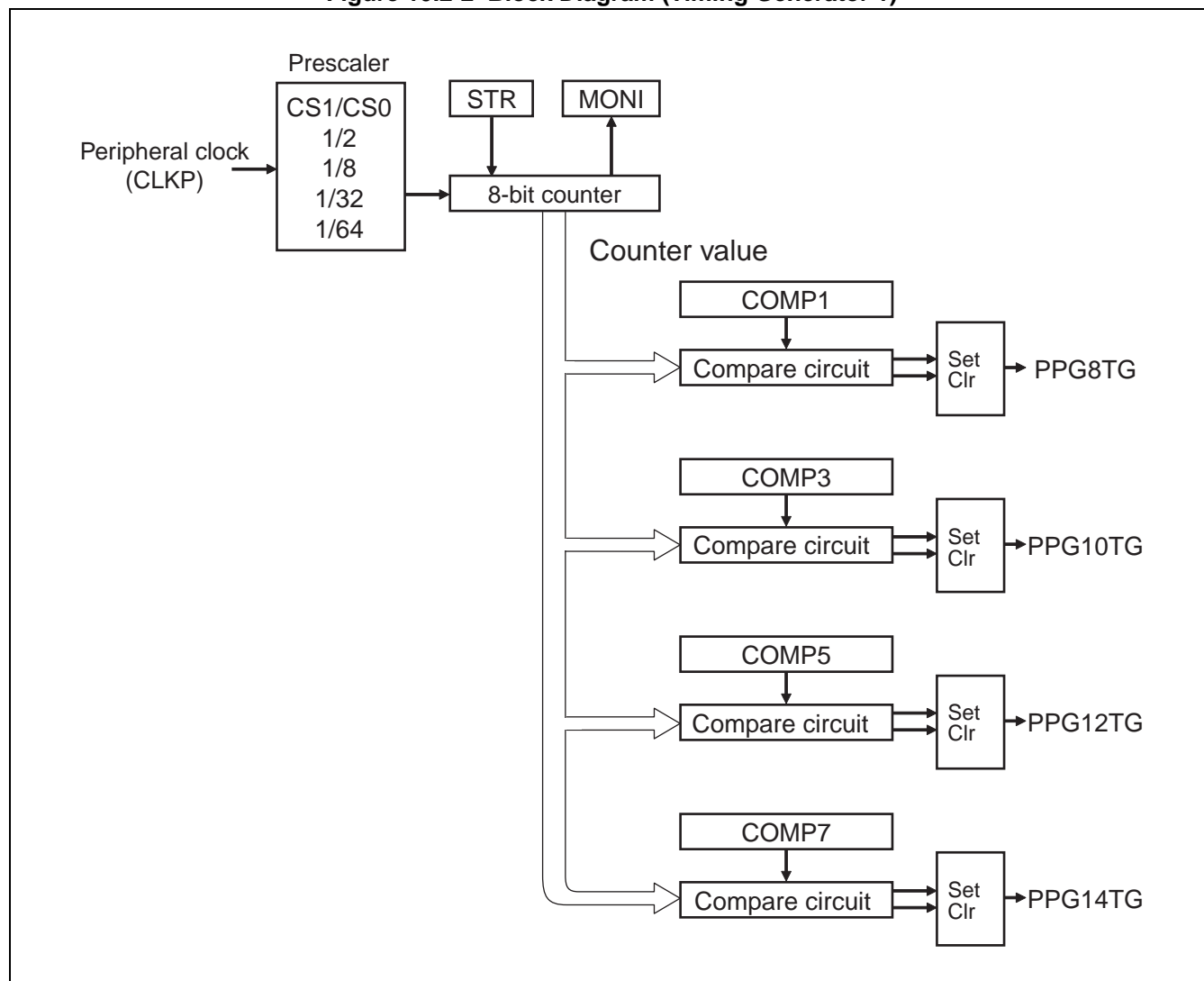


Figure 10.2-2 Block Diagram (Timing Generator 1)



MB91470/480 Series

10.3 Registers of Timing Generator

This section describes the registers of each timing generator.

■ Registers of Timing Generator 0

Control register 0: TTCR0

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value
0000 0144 _H	TRG6O	TRG4O	TRG2O	TRG0O	CS1	CS0	MONI	STR	11110000 _B
	W	W	W	W	R/W	R/W	R	W	

Compare register 0: COMP0

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value
0000 0148 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 2: COMP2

Address	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	Initial value
0000 0149 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 4: COMP4

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 014A _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 6: COMP6

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000 014B _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable

■ **Registers of Timing Generator 1**

Control register 1: TTCR1

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value
0000 014C _H	TRG7O	TRG5O	TRG3O	TRG1O	CS1	CS0	MONI	STR	11110000 _B
	W	W	W	W	R/W	R/W	R	W	

Compare register 1: COMP1

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value
0000 0150 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 3: COMP3

Address	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	Initial value
0000 0151 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 5: COMP5

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 0152 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare register 7: COMP7

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0000 0153 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable

MB91470/480 Series

10.3.1 Timing Generator Control Register (TTCR0/TTCR1)

The timing generator control register (TTCR0/TTCR1) is used to check the status of PPG trigger clear, timer prescaler and 8-bit counter as well as to control their operations.

■ Timing Generator Control Register (TTCR0/TTCR1)

Timing generator control register 0: TTCR0									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 0144 _H	TRG6O	TRG4O	TRG2O	TRG0O	CS1	CS0	MONI	STR	11110000 _B
	W	W	W	W	R/W	R/W	R	W	

Timing generator control register 1: TTCR1									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0000 014C _H	TRG7O	TRG5O	TRG3O	TRG1O	CS1	CS0	MONI	STR	11110000 _B
	W	W	W	W	R/W	R/W	R	W	

R/W : Readable/writable
R : Read only
W : Write only

[bit15 to bit12] TRG6O/TRG4O/TRG2O/TRG0O, TRG7O/TRG5O/TRG3O/TRG1O: PPG Trigger clear bits

Writing "0" to these bits clears the PPG start trigger to be output. The correspondence with trigger of the bits is shown below.

[Timing generator 0]	[Timing generator 1]
TRG0O: PPG0TG	TRG1O: PPG8TG
TRG2O: PPG2TG	TRG3O: PPG10TG
TRG4O: PPG4TG	TRG5O: PPG12TG
TRG6O: PPG6TG	TRG7O: PPG14TG

Read values of this register are always "1".

[bit11, bit10] CS1, CS0: Count clock selection bits

The operation clock of the 8-bit counter is selected as follows:

CS1	CS0	Clock source
0	0	Peripheral clock (CLKP) / 2 (50 ns @40 MHz)[Initial value]
0	1	Peripheral clock (CLKP) / 8 (200 ns @40 MHz)
1	0	Peripheral clock (CLKP) /32 (800 ns @40 MHz)
1	1	Peripheral clock (CLKP) /64 (1.6 μ s @40 MHz)

[bit9] MONI: 8-bit counter operating monitor bit

The operation of the 8-bit counter is selected as follows:

MONI	Status of 8-bit counter
0	Stopping counter [Initial value]
1	Operation counter

Writing value has no meaning.

[bit8] STR: 8-bit counter operation enable bit

The operation of the 8-bit counter is selected as follows:

STR	Operations of 8-bit counter
0	Has no meaning [Initial value]
1	Start counter operation

Read value is always "0".

Writing "0" has no meaning.

MB91470/480 Series**10.3.2 Compare Register (COMP0/COMP2/COMP4/COMP6, COMP1/COMP3/COMP5/COMP7)**

The compare registers (COMP0/COMP2/COMP4/COMP6, COMP1/COMP3/COMP5/COMP7) are used to set PPG start signals. When the 8-bit counter matches with the value of one of the compare registers, the corresponding PPG start signal is set.

■ Compare Register (COMP0/COMP2/COMP4/COMP6, COMP1/COMP3/COMP5/COMP7)

Compare Register 0 : COMP0

Address	bit	31	30	29	28	27	26	25	24	Initial value
0000 0148 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 2 : COMP2

Address	bit	23	22	21	20	19	18	17	16	Initial value
0000 0149 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 4 : COMP4

Address	bit	15	14	13	12	11	10	9	8	Initial value
0000 014A _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 6 : COMP6

Address	bit	7	6	5	4	3	2	1	0	Initial value
0000 014B _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 1 : COMP1

Address	bit	31	30	29	28	27	26	25	24	Initial value
0000 0150 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 3 : COMP3

Address	bit	23	22	21	20	19	18	17	16	Initial value
0000 0151 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 5 : COMP5

Address	bit	15	14	13	12	11	10	9	8	Initial value
0000 0152 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare Register 7 : COMP7

Address	bit	7	6	5	4	3	2	1	0	Initial value
0000 0153 _H		D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/Writable

D7 to D0 : Compare value setting bits

Notes:

- When the value of the compare register is "00000000_B", the PPG start signal won't be set.
 - Make sure to rewrite the register while the 8-bit counter is stopped.
-

10.4 Operation of Timing Generator

This section describes the operation of the timing generator.

■ Operation of Prescaler

This operation sets clock that is the count clock for the 8-bit counter divided by the peripheral clock (CLKP).

■ 8-bit Counter

- The 8-bit counter starts counting the count clock from the prescaler by setting the STR bit.
- The 8-bit counter starts counting up and stops the counting with overflow.
- To start counter during counting is ignored.
- "1" is read to the MONI bit while the 8-bit counter is counting. When stopped, "0" is read.
- The count value of the 8-bit counter is input to comparators.

Figure 10.4-1 Operation/stop Timing of 8-bit Counter

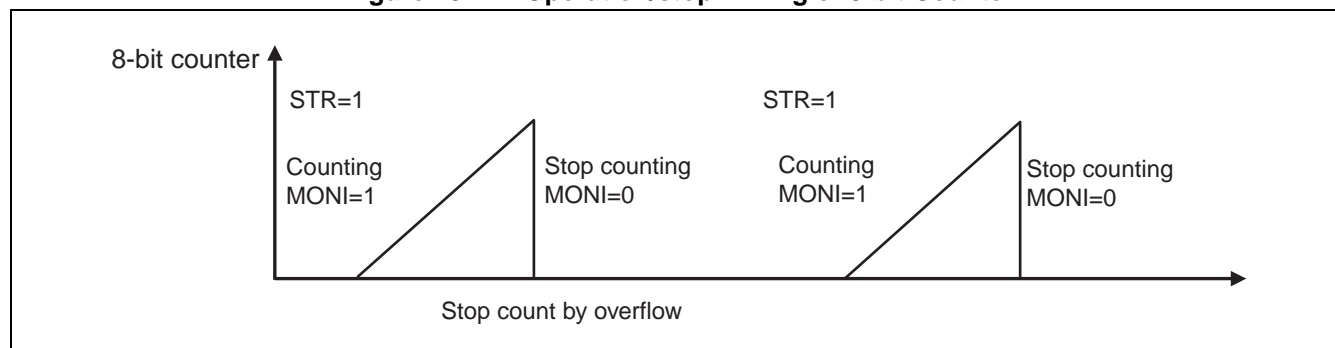
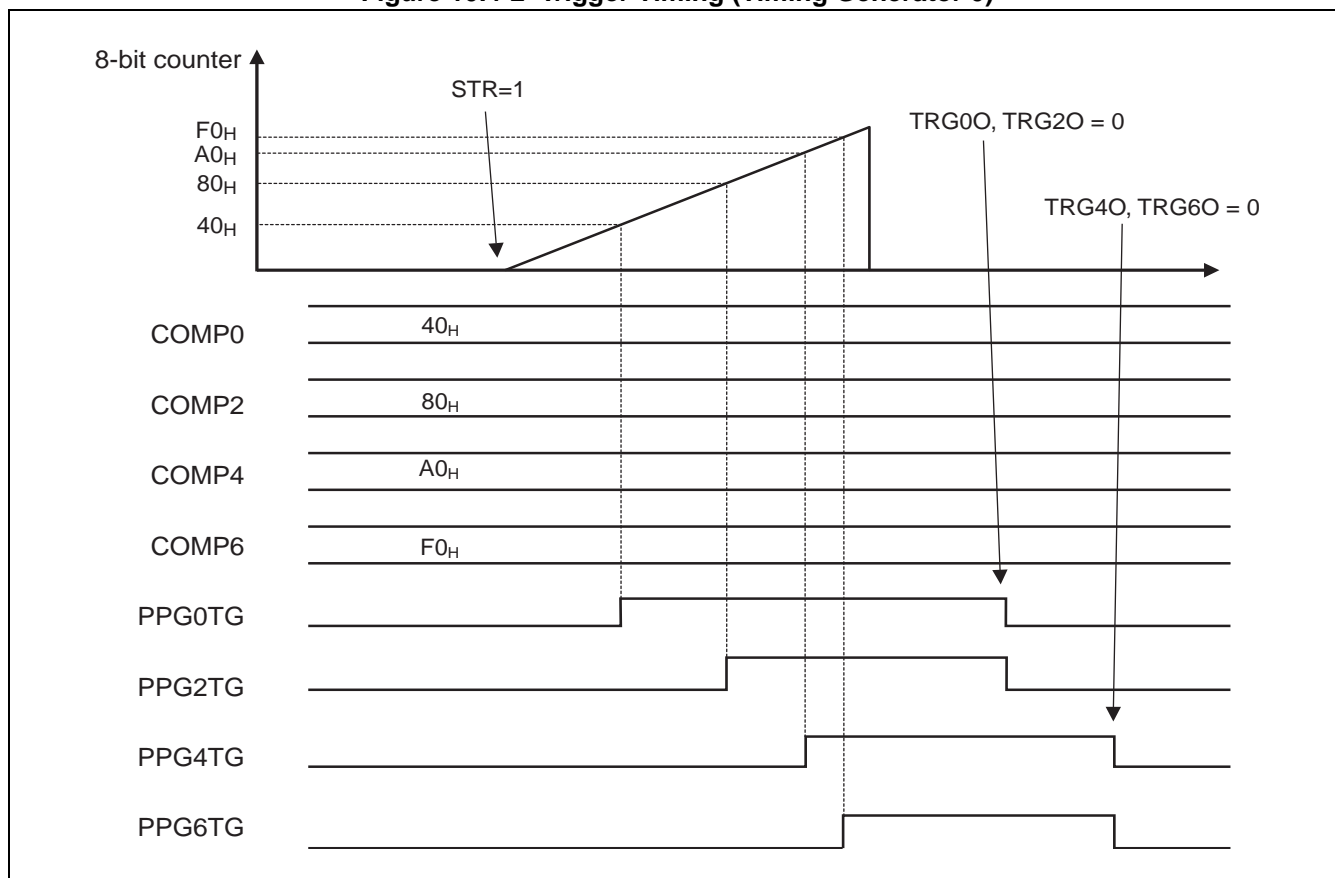


Figure 10.4-2 Trigger Timing (Timing Generator 0)



CHAPTER 11

PPG

This chapter explains the overview of the PPG, the configuration and functions of registers, and the PPG operation.

- 11.1 Overview of PPG
- 11.2 Block Diagram of PPG
- 11.3 Registers of PPG
- 11.4 Operation Explanation of PPG

11.1 Overview of PPG

The PPG is an 8-bit reload timer module that can be used as a PPG output to output pulses controlled by the timer operation.

In MB91470 series, the hardware consists of 8 8-bit down counters, 16 8-bit reload registers, control register, 8 external pulse outputs, and 8 interrupt output. It is an 8ch 8-bit PPG or a 4ch 16-bit PPG.

In MB91480 series, the hardware consists of 16 8-bit down counters, 32 8-bit reload registers, control register, 10 external pulse outputs, and 16 interrupt output. It is a 16ch 8-bit PPG or an 8ch 16-bit PPG.

■ Functions of PPG

- 4 PPG operation
Mode is supported.
- PPG output operation
Outputs a pulse waveform with arbitrary period and duty ratio.
Can also be used in conjunction with an external circuit to form a D/A converter.
- Output inversion function
The PPG output value can be inverted.

■ PPG Mode

- 8-bit PPG output independent operation mode
Can operate as an independent PPG output.
- 16-bit PPG output operation mode
1 channel 16-bit PPG output can be operated.
- 8 + 8-bit PPG output operation mode
With setting the $ch(n + 1)$ output as the $ch(n)$ clock input, the 8-bit PPG output in any cycle can be operated ($n = 0, 2, 4, 6, 8 *$).
- 16 + 16-bit PPG output operation mode
This mode sets the 16-bit prescaler output, $ch(n + 3) + ch(n + 2)$ as a clock input for the 16-bit PPG, $ch(n + 1) + ch(n)$. ($n = 0, 4, 8 *$)

*: MB91480 series only.

■ PPG Channels Corresponding to Each Mode

PPG channel	8-bit mode	8+8-bit mode	16-bit mode	16+16-bit mode
PPG0	PPG0	PPG0+PPG1	PPG0	PPG0+PPG2
PPG1	PPG1			
PPG2	PPG2	PPG2+PPG3	PPG2	
PPG3	PPG3			
PPG4	PPG4	PPG4+PPG5	PPG4	PPG4+PPG6
PPG5	PPG5			
PPG6	PPG6	PPG6+PPG7	PPG6	
PPG7	PPG7			
PPG8	PPG8	PPG8+PPG9	PPG8	PPG8+PPG10
PPG9	PPG9			
PPG10	PPG10	PPG10+PPG11	PPG10	
PPG11	PPG11			
PPG12	PPG12	PPG12+PPG13	PPG12	PPG12+PPG14
PPG13	PPG13			
PPG14	PPG14	PPG14+PPG15	PPG14	
PPG15	PPG15			

11.2 Block Diagram of PPG

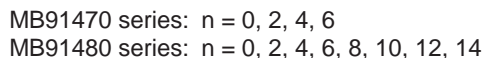
■ Block Diagram of the 8-bit PPG ch.0, ch.2, ch.4, ch.6, ch.8, ch.10, ch.12 and ch.14

ch. (n+1) borrow
Peripheral clock (CLKP) 64-divided
Peripheral clock (CLKP) 16-divided
Peripheral clock (CLKP) 4-divided

PPG output latch

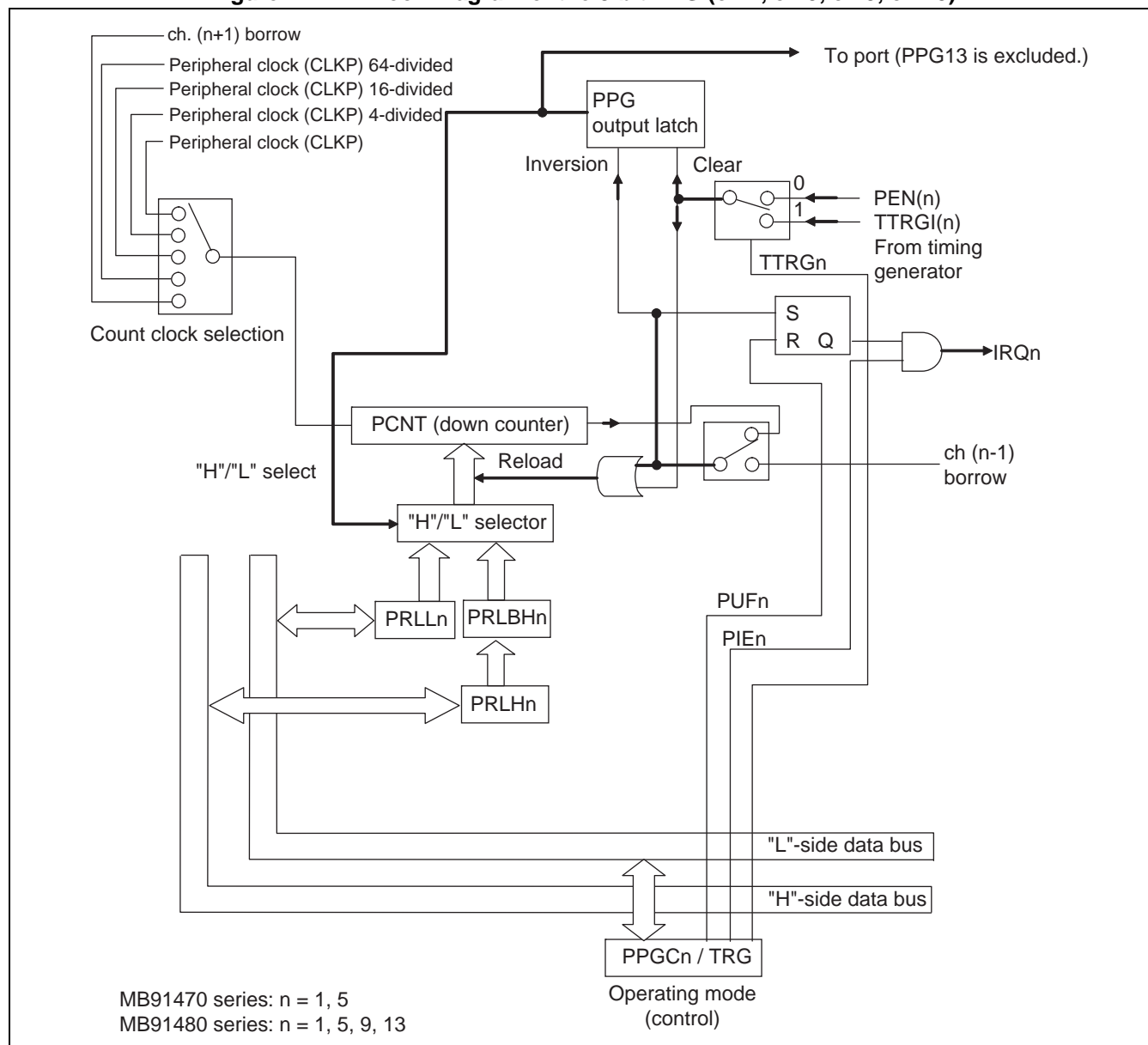
To port (PPG10/12/14 is excluded.)
To multi-function timer 0 (PPG0/2/4)
To multi-function timer 1 (PPG8/10/12)*

*: Not available in MB91470 series.



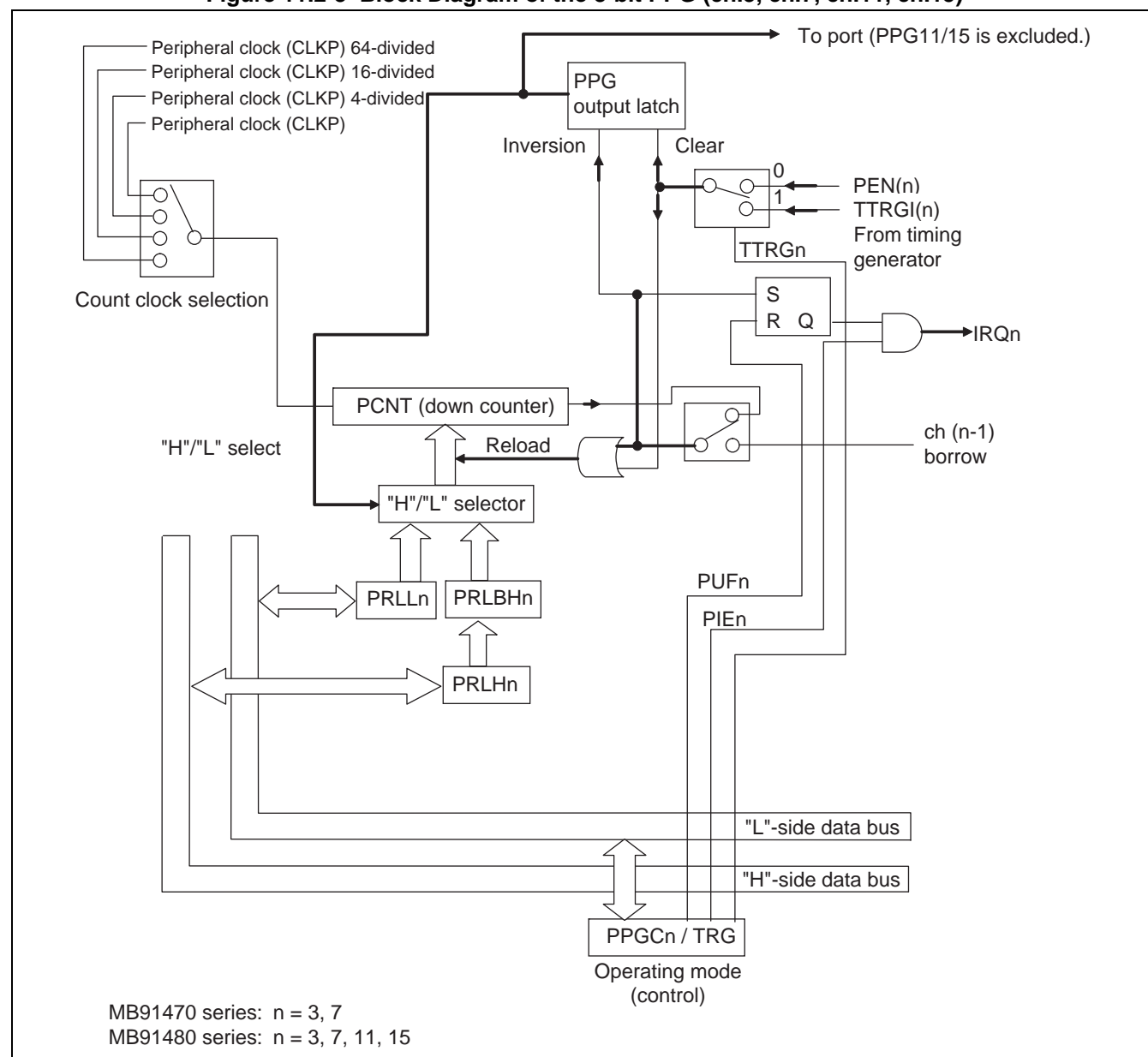
■ Block Diagram of the 8-bit PPG ch.1, ch.5, ch.9 and ch.13

Figure 11.2-2 Block Diagram of the 8-bit PPG (ch.1, ch.5, ch.9, ch.13)

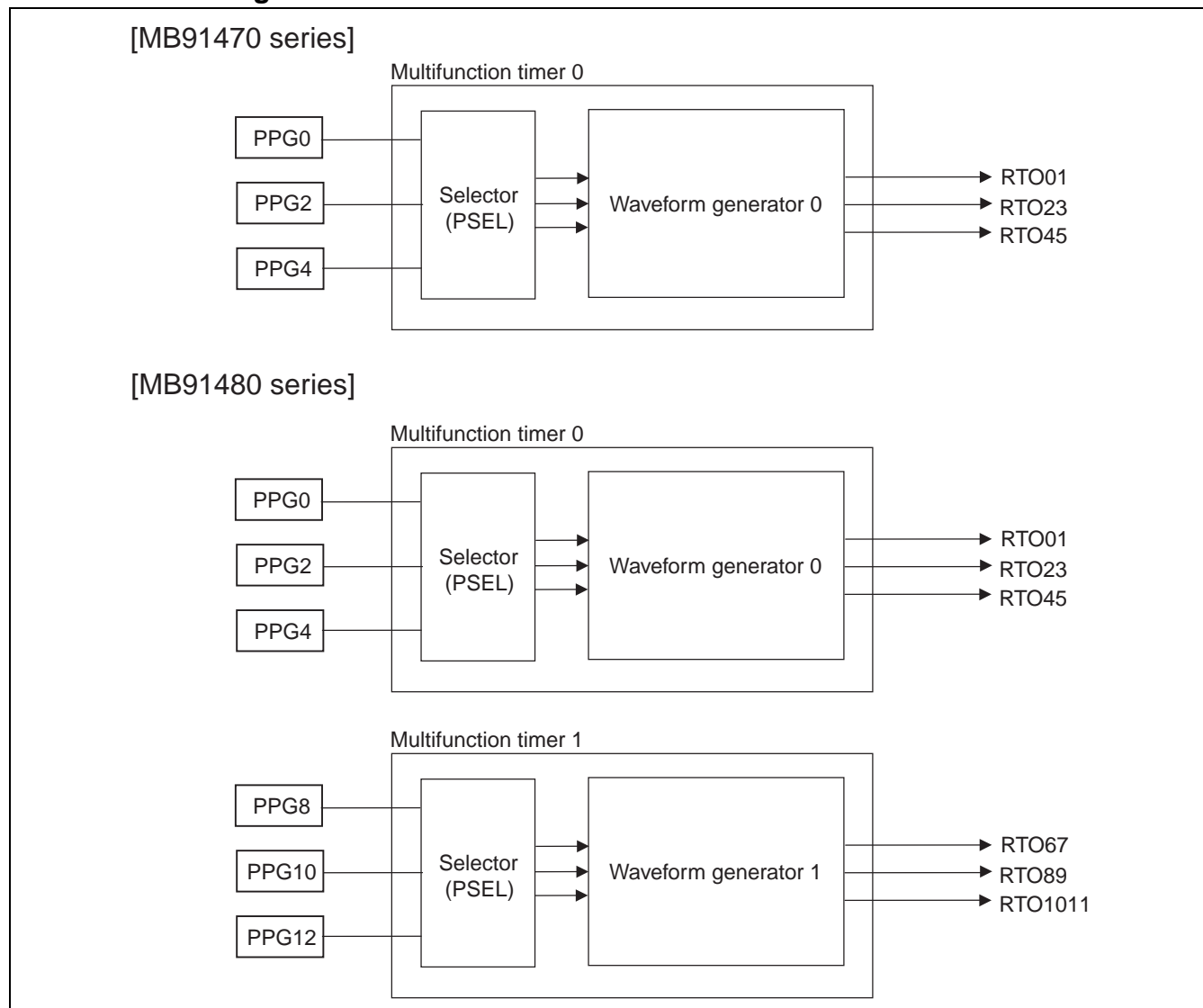


■ **Block Diagram of the 8-bit PPG ch.3, ch.7, ch.11 and ch.15**

Figure 11.2-3 Block Diagram of the 8-bit PPG (ch.3, ch.7, ch.11, ch.15)

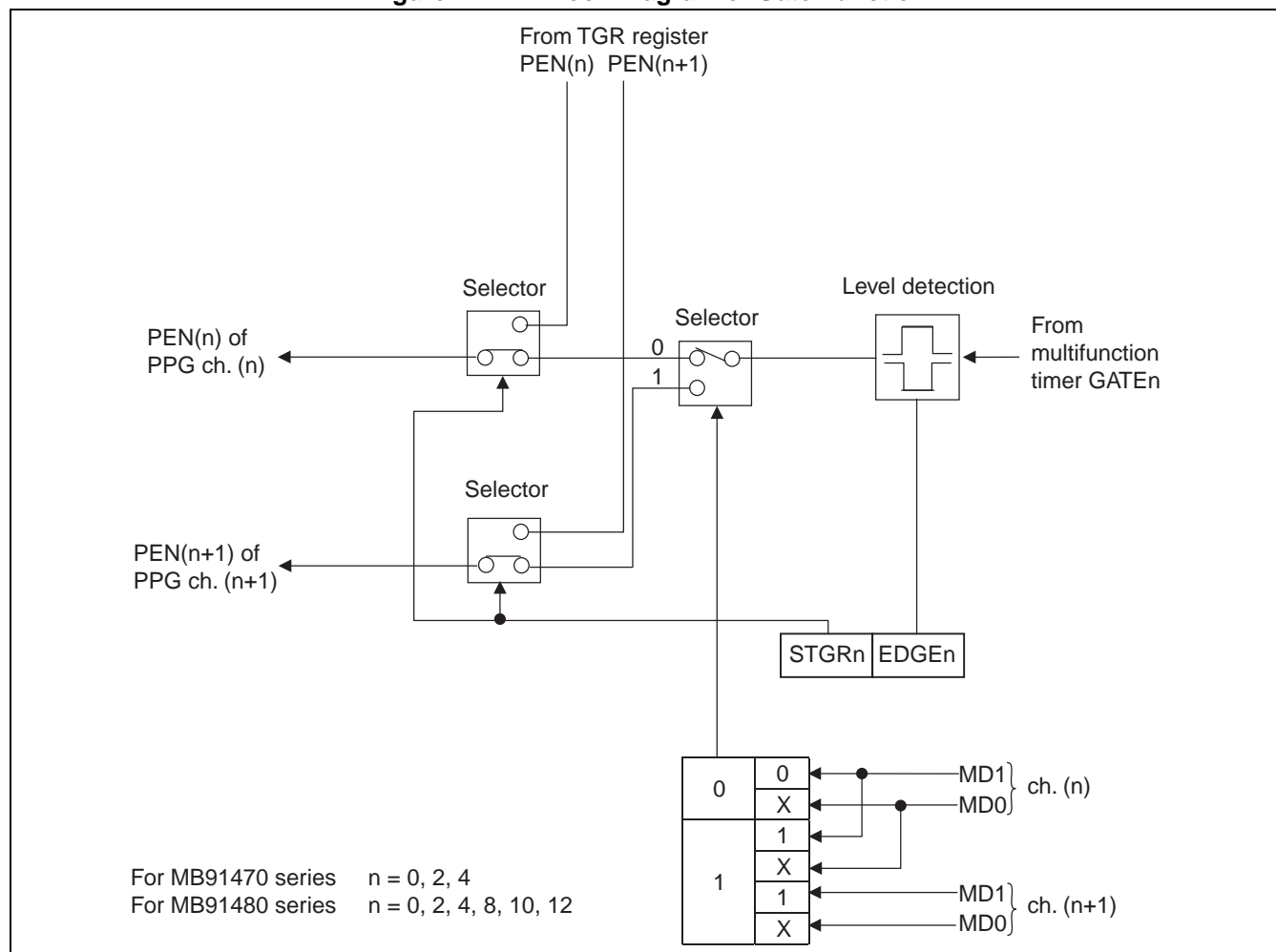


■ Connection Diagram between PPG and Multi-function Timer



■ **Block Diagram of Gate Function**

Figure 11.2-4 Block Diagram of Gate Function



MB91470/480 Series

11.3 Registers of PPG

This section lists the registers of the PPG.

■ PPG Registers

PPG trigger register (TRG)

*TRGH

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000130 _H	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TRGL

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00	00000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

*PEN15 to PEN08 are not available for MB91470 series.

Output inversion register (REVC)

*REVCH

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
REV15	REV14	REV13	REV12	REV11	REV10	REV09	REV08	00000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

REVCL

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
REV07	REV06	REV05	REV04	REV03	REV02	REV01	REV00	00000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

*REV15 to REV08 are not available for MB91470 series.

GATE function control register (GATECn)

GATECn

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ch.0: 000133 _H ch.4: 000137 _H ch.8: 00013B _H ch.12: 00013F _H	-	-	STGR(n+2)	EDGE(n+2)	-	-	STGR(n)	EDGE(n)	GATEC0, GATEC8: --00--00 _B GATEC4, GATEC12: -----00 _B
	-	-	R/W	R/W	-	-	R/W	R/W	

n = 0, 4

MB91470 series

n = 0, 4, 8, 12

MB91480 series

PPG operation mode control register (PPGC0 to PPGC15)

PPGCn

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Initial value
ch.0: 000108 _H to ch.15: 00012F _H	PIEn	PUFn	INTMn	PCS1	PCS0	MD1*	MD0*	TTRGn	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

n=0 to 7 PPG0 to 7 (MB91470 series)

n=0 to 15 PPG0 to 15 (MB91480 series)

*: MD1 and MD0 exist only in even-numbered channel, but they do not exist in odd-numbered channel.

The initial value of odd-numbered channel is undefined. Writing to them is meaningless.

n = 8 to 15 are not available in MB91470 series

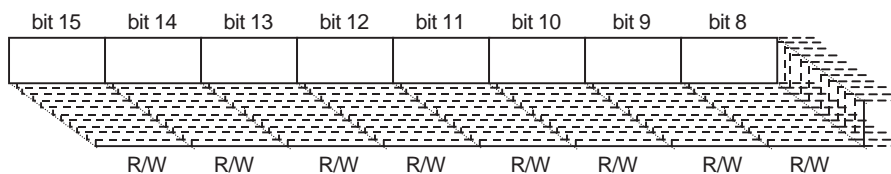
R/W: Readable/writable

● Reload registers: 8-bit PPG mode

Reload register H (PRLH0 to PRLH15)

PPLHn

Address
ch.0:
000100_H to
ch.15:
00012A_H



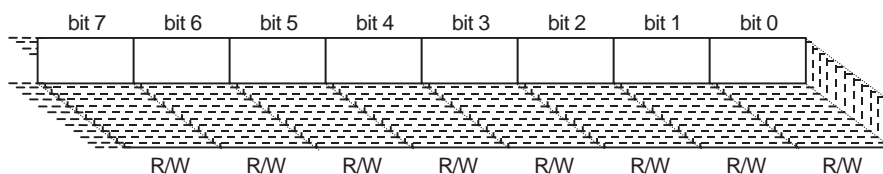
Initial value
XXXXXXXX_B

n=0 to 7 PPG0 to 7 (MB91470 series)
n=0 to 15 PPG0 to 15 (MB91480 series)

Reload register L (PRL0 to PRL15)

PRLLn

Address
ch.0:
000101_H to
ch.15:
00012B_H

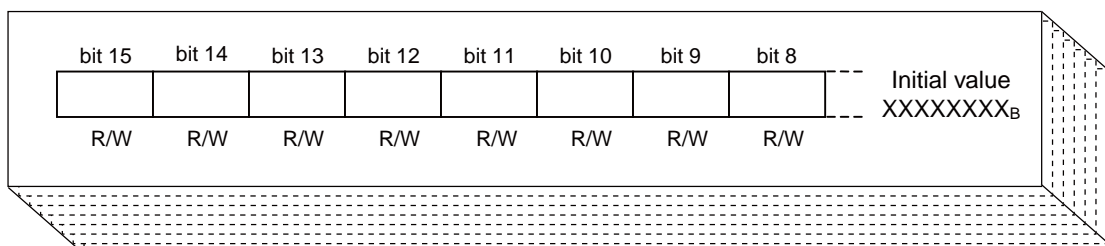


Initial value
XXXXXXXX_B

n=0 to 7 ch.0 to ch.7 (MB91470 series)
n=0 to 15 ch.0 to ch.15 (MB91480 series)
R/W: Readable/writable

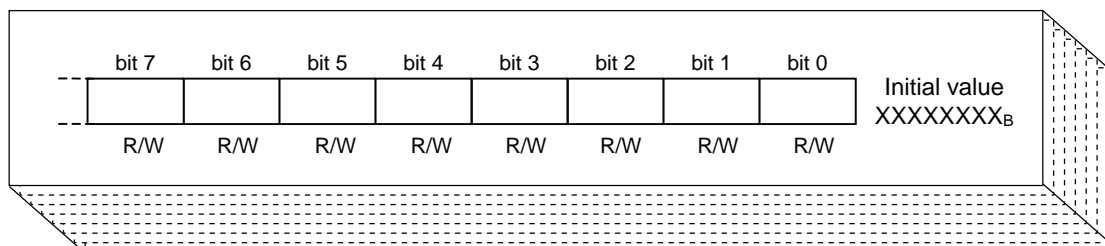
■ Reload Registers: 16-bit PPG Mode

Reload register H (PRLH0, PRLH2, PRLH4, PRLH6, PRLH8, PRLH10, PRLH12, PRLH14)



n=0,2,4,6 PPG0/2/4/6 (MB91470 series)
n=0,2,4,6,8,10,12,14 PPG0/2/4/6/8/10/12/14 (MB91480 series)

Reload register L (PRL0, PRL2, PRL4, PRL6, PRL8, PRL10, PRL12, PRL14)



n=0,2,4,6 PPG0/2/4/6 (MB91470 series)
n=0,2,4,6,8,10,12,14 PPG0/2/4/6/8/10/12/14 (MB91480 series)

The address of PRLLn in 16-bit PPG mode is different from 8-bit PPG mode.

R/W: Readable/writable

MB91470/480 Series

11.3.1 PPG Operation Mode Control Registers (PPGC0 to PPGC15)

The PPG operation mode control registers can make interrupt, operation mode, prescaler, and other settings.

■ PPG Operation Mode Control Registers (PPGC0 to PPGC15)

PPG operation mode control registers (PPGC0 to PPGC15)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ch.0: 000108 _H to ch.15: 00012F _H	PIE	PUF	INTM	PCS1	PCS0	MD1	MD0	TTRG	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: PPGC8 to PPGC15 are not available for MB91470 series.

R/W: Readable/writable

[bit7] PIE (Ppg Interrupt Enable): PPG interrupt enable bit

This bit enables or disables PPG interrupts as follows:

0	Disables PPG interrupts.
1	Enables PPG interrupts.

- An interrupt request occurs when the PUF bit is set to "1" with this bit containing "1".
- No interrupt request occurs with this bit containing "0".
- The bit is initialized to "0" by a reset.
- The bit is readable and writable.

[bit6] PUF (Ppg Underflow Flag): PPG counter underflow bit

This bit indicates the detection status of a PPG counter underflow as follows:

0	Indicates that no PPG counter underflow has been detected.
1	Indicates that the PPG counter underflow has been detected.

- This bit is set to "1" on occurrence of an underflow when the count value for each channel changes from "00_H" to "FF_H" in either 8-bit PPG 2-channel mode or 8-bit prescaler + 8-bit PPG mode.
- In 16-bit PPG 1-channel mode, the bit is set to "1" on occurrence of an underflow when the count value of ch(n+1)/ch(n) (n = 0/2/4/6/8/10/12/14) changes from "0000_H" to "FFFF_H".
- Writing "0" to the bit set it to "0".
- Writing "1" to this bit is meaningless.
- The bit returns "1" when read of a read modify write (RMW) instruction.
- The bit is initialized to "0" by a reset.
- The bit is readable and writable.

[bit5] INTM (Interrupt Mode): Interrupt mode bit

This bit allows the PUFn bit to detect only the underflow from PRLBH.

0	The PUF bit is set to "1" when the underflow occurs.
1	The PUFn bit is set to "1" only when the underflow from PRLBHn occurs.

- The bit is initialized to "0" by a reset.
- The bit is readable and writable.
- Setting this bit to "1" allows an interrupt to occur when one cycle of the PPG waveform is output.
- Do not rewrite this bit with interrupts enabled.

[bit4, bit3] PCS1, PCS0 (Ppg Count Select): Count clock select bits

These bits select the operating clock for the down counter as follows:

PCS1	PCS0	Operation mode
0	0	Peripheral clock (CLKP) (25-ns peripheral clock at 40 MHz)
0	1	Peripheral clock (CLKP) /4 (100-ns peripheral clock at 40 MHz)
1	0	Peripheral clock (CLKP) /16 (400-ns peripheral clock at 40 MHz)
1	1	Peripheral clock (CLKP) /64 (1.6-μs peripheral clock at 40 MHz)

- These bits are initialized to "00_B" by a reset.
- The bits are readable and writable.

[bit2, bit1] MD1, MD0 (ppg count MoDe): Operation mode select bits

These bits select the operation mode of the PPG timer as follows:

MD1	MD0	Operation mode
0	0	8-bit PPG 2-channel independent mode
0	1	8-bit prescaler + 8-bit PPG mode
1	0	16-bit PPG mode
1	1	16-bit prescaler + 16-bit PPG mode

- These bits are initialized to "00_B" by a reset.
- The bits are readable and writable.
- The bits exist only for even-numbered channels.

[bit0] TTRG (Timing TRGer): Timing trigger select bit

This bit allows the PPG to get started only with the trigger signal from the timing generator.

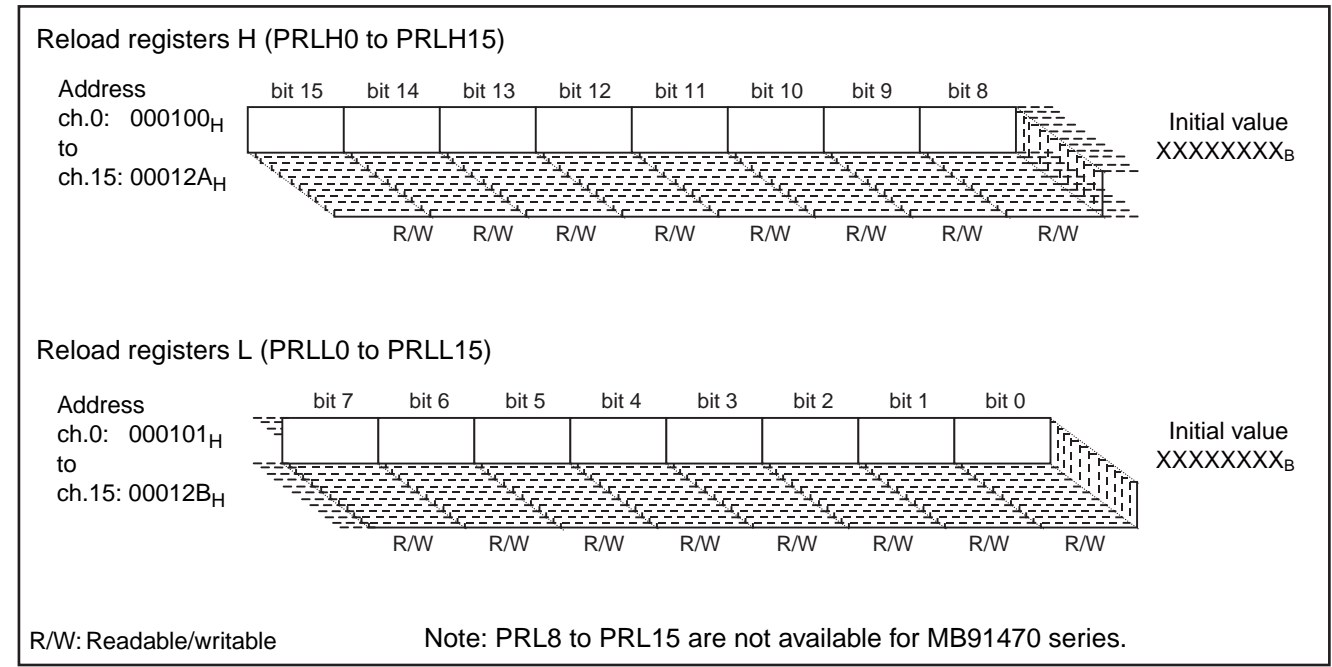
0	Starts the PPG in response to the TRG register or multi-function timer.
1	Starts the PPG only in response to the timing generator.

- The bit is initialized to "0" by a reset.
- The bit is readable and writable.

11.3.2 Reload Registers (PRLH0 to PRLH15, PRL0 to PRL15)

Reload registers can hold the reload values for the down counter.

■ Reload Registers (PRLH0 to PRLH15, PRL0 to PRL15)



Register Name	Function
PRL	Holds the "L"-side reload value.
PRLH	Holds the "H"-side reload value.

Note:

When the PPG is used either in 8-bit prescaler + 8-bit PPG mode or in 16-bit prescaler + 16-bit PPG mode, the PPG waveform may vary from cycle to cycle if the prescaler-side PRL and PRLH registers are set to different values. They should therefore be set to the same value.

MB91470/480 Series

11.3.3 PPG Trigger Register (TRG)

The PPG trigger register can enable the operation of each PPG.

■ PPG Trigger Register (TRG)

PPG trigger register (TRG)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000130 _H	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: PEN08 to PEN15 are not available for MB91470 series (PPG8).

R/W: Readable/writable

[bit15 to bit0] PEN15 to PEN00 (Ppg ENable): PPG operation enable bits

These bits are used to select the PPG operation start and the operation mode:

PEN15 to PEN00	Operation Status
0	Stops the PPG from operating (while holding the output at "L" level).
1	Enables the PPG to operate.

- The bits are initialized to "0" by a reset.
- The bits are readable and writable.

11.3.4 Output Inversion Register (REVC)

The output inversion register can enable the inverted output of each PPG output value.

■ Output Inversion Register (REVC)

Output inversion register (REVC)									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
Address	REV15	REV14	REV13	REV12	REV11	REV10	REV09	REV08	Initial value
000134 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00000000 _B
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	REV07	REV06	REV05	REV04	REV03	REV02	REV01	REV00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Note: REV08 to REV15 are not available for MB91470 series (PPG8).									
R/W: Readable/writable									

[bit15 to bit0] REV15 to REV00: Output inversion bits

These bits are used to invert the PPG output values including the initial level.

REV15 to REV00	Output Level
0	Normal
1	Inverted

- The bits are initialized to "0" by a reset.
- The bits are readable and writable.
- As the register simply inverts the PPG output, it inverts the initial level as well. It also exchanges the "L" and "H" relationships between reload registers with each other.

MB91470/480 Series

11.3.5 GATE Function Control Registers (GATEC0/GATEC4/GATEC8/GATEC12)

The GATE function control registers can enable the PPG to start/stop in response to the signal from the multi-function timer.

■ GATE Function Control Registers (GATEC0/GATEC4/GATEC8/GATEC12)

GATE function control register (GATECn)									Initial value
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ch.0: 000133 _H	-	-	STGR(n+2)	EDGE(n+2)	-	-	STGR(n)	EDGE(n)	GATEC0, GATEC8: --00--00 _B GATEC4, GATEC12: -----00 _B
ch.4: 000137 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.8: 00013B _H									
ch.12: 00013F _H									

n = 0, 4 MB91470 series
n = 0, 4, 8, 12 MB91480 series

R/W: Readable/writable

[bit5, bit1] STGR: Gate function select bits

These bits are used to select whether to use the trigger signal from the multi-function timer or the TRG register to start the PPG as follows:

STGR	Operation mode
0	Starts the PPG according to the TRG register.
1	Starts the PPG in response to the trigger signal from the multi-function timer.

- The bits are initialized to "0" by a reset.
- The bits are readable and writable.

[bit4, bit0] EDGE: Trigger edge select bits

These bits are used to select the trigger edge from the multi-function timer as follows:

EDGE	Operation Mode
0	Start at the rising edge -> Stop at the falling edge ^{*1}
1	Start at the falling edge -> Stop at the rising edge ^{*2}

- The bits are initialized to "0" by a reset.
- The bits are readable and writable.

*1: The PPG remains on with the signal at the "H" level.

*2: The PPG remains on with the signal at the "L" level.

11.4 Operation Explanation of PPG

MB91480 series has 16 channels for 8-bit PPG and can operate four modes in total: independent mode, 8-bit prescaler + 8-bit PPG mode, 16-bit PPG 1 channel mode, and 16-bit prescaler + 16-bit PPG mode.



There are 8 channels of the 8-bit PPG in the MB91470 series.

■ Operation Explanation

Each of 8-bit length PPG units has two 8-bit-length reload registers for the "L" and "H" sides (PRL, PRLH). The "L"-side and "H"-side values written to this register are alternately reloaded to the 8-bit down-counter (PCNT) which counts down on each cycle of the count clock. The value of the pin output (PPG) is toggled each time a counter borrow occurs to trigger another reload. With this operation, the pin output (PPG) becomes a pulse output, which has "L"/"H" width corresponding to the value of reload register.

The operation starts/restarts when the bit of the register is written.

The relationship between the reload operation and the pulse output is shown below.

Reload operation	Pin output change
PRLH → PCNT	PPG [0 → 1] 
PRL → PCNT	PPG [1 → 0] 

When bit7 (PIEn) of the PPGCn register is "1", an interrupt request is output when the counter goes from "00_H" to "FF_H" causing a borrow (or a borrow from "0000_H" to "FFFF_H" in 16-bit PPG mode).

● Operation Modes

There are four operation modes: independent mode, 8-bit prescaler + 8-bit PPG mode, 16-bit PPG 1 channel mode, and 16-bit prescaler + 16-bit PPG mode.

- In the independent mode, a channel can operate as 8-bit PPG independently. The PPG output of ch.(n) is connected to PPG(n) pin. (n = 0 to 9)
- The 8-bit prescaler + 8-bit PPG mode makes 1 channel operate as an 8-bit prescaler, counts its borrow output, and then allows the 8-bit PPG waveform in any cycle to be output. For example, the prescaler output of ch.1 is connected to the PPG1 pin; the PPG output of ch.0 is connected to the PPG0 pin.
- In the 16-bit PPG 1 channel mode, two channels are combined, and the combined channel operates as 16-bit PPG. For example, if ch.0 and ch.1 are combined, 16-bit PPG outputs are connected to both PPG0 pin and PPG1 pin.

● PPG Output Operation

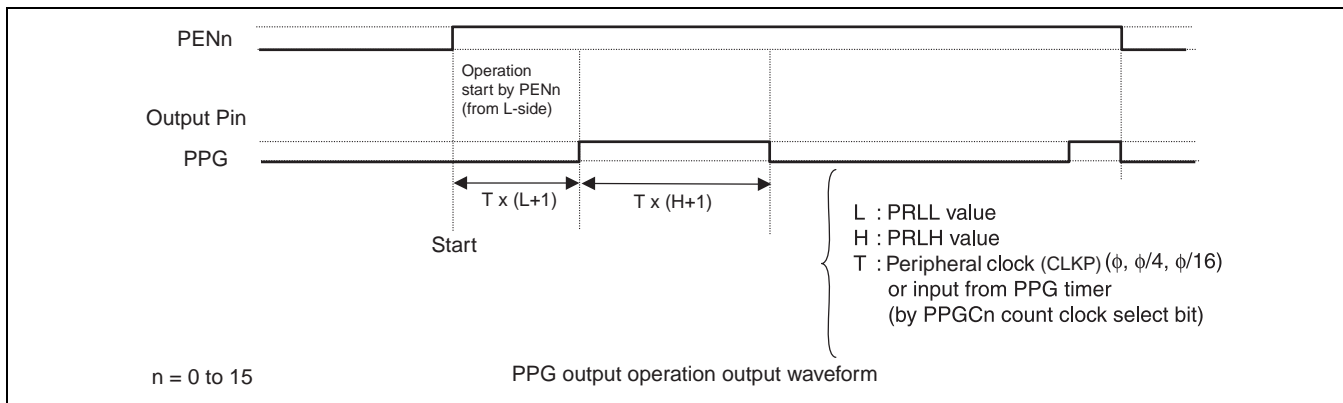
The PPG in this block is activated and starts counting when the bit corresponding to each channel in the TRG register (PPG trigger register) is set to "1". After operation starts, the count operation is stopped when each channel bit of TRG register is set to 0. After having stopped, the pulse output holds "L" level.

Do not set the PPG channel as the operating state, with the prescaler channel as the stopped state, in the 8-bit prescaler + 8-bit PPG mode and the 16-bit prescaler + 16-bit PPG mode.

In 16-bit PPG mode, use the PENn bits for each channel in the TRG register to simultaneously start and stop operation. (n = 0 to 15)

PPG output operation is explained below.

In PPG operation, the pulse wave with any frequency/duty ratio (the ratio between "H" level period and "L" level period in pulse wave) is output continuously. If the pulse wave output is started, PPG will not stop it before operation stop is set.



● Relationship between Reload Value and Pulse Width

The pulse width to be output is the value that multiplies the cycle of the count clock by the value in the reload register plus 1. Note that the pulse width will be one cycle of the count clock when the reload register value is set to "00_H" at operating the 8-bit PPG and when the reload register value is set to "0000_H" at operating the 16-bit PPG. Note that the pulse width will be 256 cycles of the count clock when the reload register value is set to "FF_H" at operating the 8-bit PPG and the pulse width will be 65536 cycles of the count clock when the reload register value is set to "FFFF_H" at operating the 16-bit PPG.

The equations for calculating the pulse width are shown below:

$$\begin{aligned} \text{Pl} &= T \times (L + 1) \\ \text{Ph} &= T \times (H + 1) \end{aligned} \quad \left\{ \begin{array}{l} \text{L : PRL value} \\ \text{H : PRLH value} \\ \text{T : Period of input clock} \\ \text{Ph : "H" pulse width} \\ \text{Pl : "L" pulse width} \end{array} \right.$$

● Count Clock Selection

The count clock to be used are the same input for the peripheral clock (CLKP), and can be selected from one of the following four types of count clock inputs.

The count clock operates as shown below.

PPGC0 to PPGC15 registers		Count clock operation
PCS1	PCS0	
0	0	Count clock is counted for peripheral clock (CLKP)
0	1	Count clock is counted for 4 cycles of peripheral clock (CLKP)
1	0	Count clock is counted for 16 cycles of peripheral clock (CLKP)
1	1	Count clock is counted for 64 cycles of peripheral clock (CLKP)

Note that, in 8-bit prescaler + 8-bit PPG mode and 16-bit prescaler + 16-bit PPG mode, the period of the initial count may vary if the PPG is started when the prescaler is running and the PPG is halted.

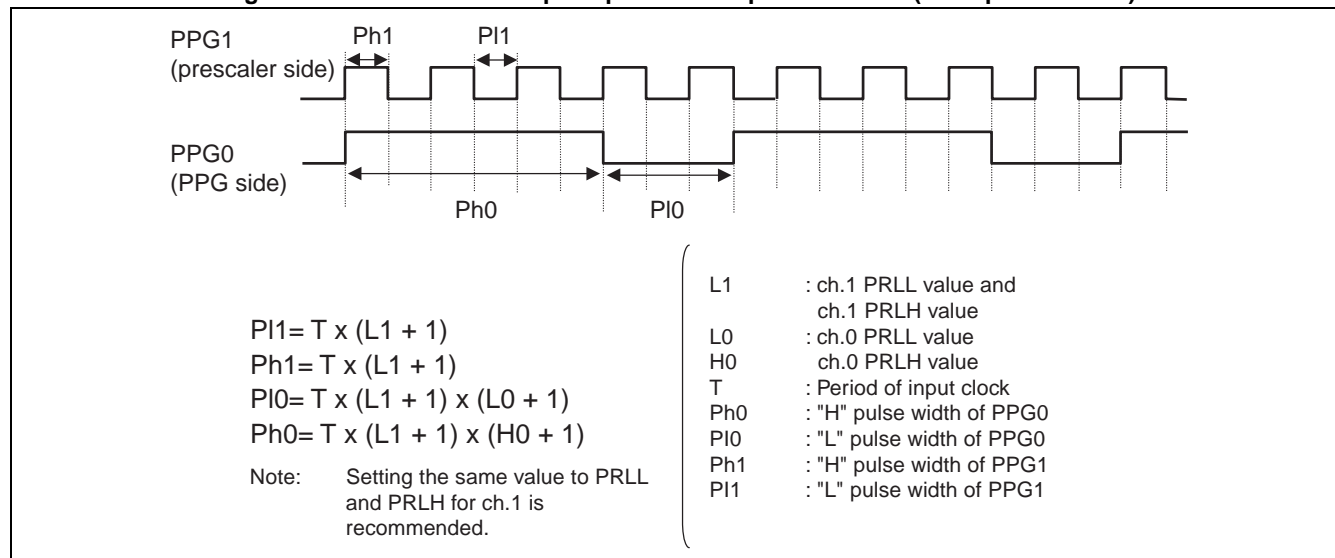
● Control of Pulse Pin Output

The pulse output generated by operating this module can be output from external pins (PPG0 to PPG15).

As both PPG(m) and PPG(m + 1) output the same waveform in 16-bit PPG mode, the same output can be obtained whichever of these is enabled as an external output pin. (m = 0, 2, 4, 6, 8, 10, 12, 14)

In the 8-bit prescaler + 8-bit PPG mode and the 16-bit prescaler + 16-bit PPG mode, the 8-bit prescaler toggle waveform is output on the prescaler side, and the 8-bit PPG waveform is output on the PPG side. The following shows an example of the output waveform in this mode.

Figure 11.4-1 8 + 8 PPG Output Operation Output Waveform (Example ch.1/ch.0)



● Interrupt

The interrupt on this module becomes active when a reload value is counted out and a borrow occurs.

However, the interrupt only goes to active if the INTMn bit is "1" when an underflow (borrow) occurs for PRLBHn. The interrupt occurs when H width pulse ends.

In the 8-bit PPG mode and the 8-bit prescaler +8-bit PPG mode, an interrupt request is performed by the relevant counter borrow. However, in 16-bit PPG mode and 16-bit prescaler +16-bit PPG mode, PUF(m) and PUF(m+1) are concurrently set by the borrow of the 16-bit counter. For this reason, it is recommended that either PIE (m) or PIE (m + 1) is enabled in order to unify the interrupt sources. Similarly, it is recommended that you write to PUF(m) and PUF(m + 1) simultaneously when clearing the interrupt source. (m = 0, 2, 4, 6, 8, 10, 12, 14)

● GATE Function

By using the multi-function timer signal, PPG can be: started-stopped.

- In the 8-bit PPG mode and the 8-bit prescaler + 8-bit PPG mode, this function can activate the PPG ch.(n).
- In the 16-bit PPG mode and the 16-bit prescaler + 16-bit PPG mode, this function can activate the PPG ch. (n), ch. (n+1).

The activation for each mode is determined by the MD register for each PPG.

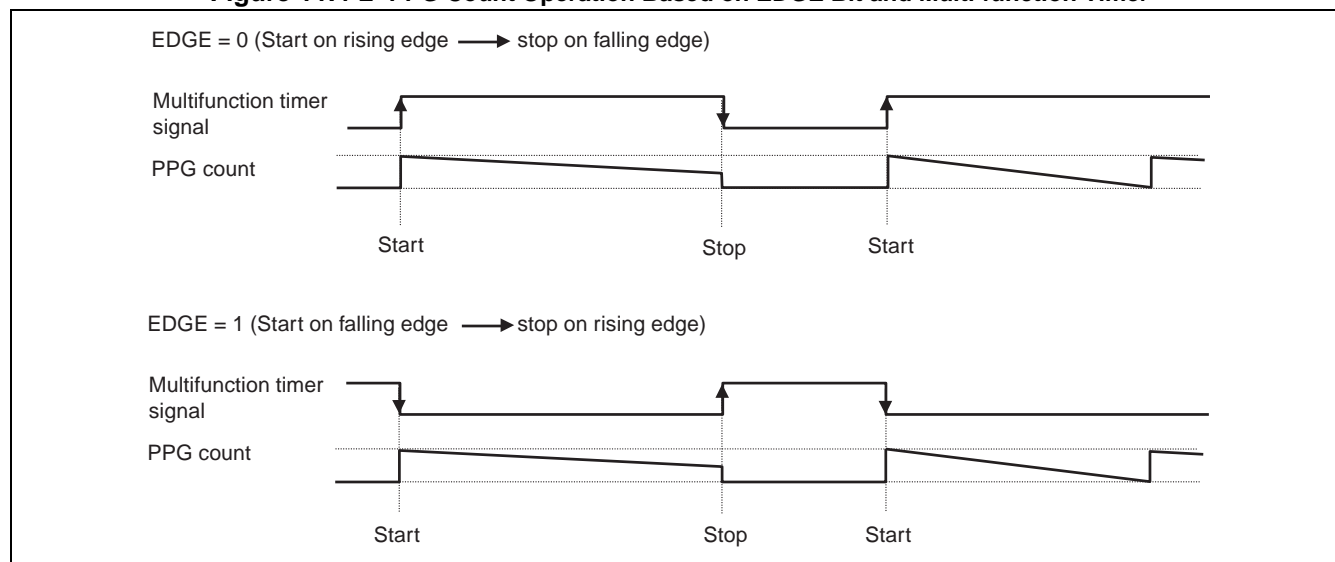
- When PPG ch. (n): MD1, MD0 = 0, X, PPG ch. (n) is started (8-bit PPG)
- When PPG ch. (n): MD1, MD0 = 1, X, PPG ch. (n), ch. (n+1) are started (16-bit PPG)

EDGE bit and the multi-function timer signal can control the period when PPG activation is valid.

MB91470 series: n = 0/2/4 (multi-function timer 0)

MB91480 series: n = 0/2/4 (multi-function timer 0), n = 8/10/12 (multi-function timer 1)

Figure 11.4-2 PPG Count Operation Based on EDGE Bit and Multi-function Timer



● Initial values for hardware

Each hardware is initialized by a reset as shown below.

< Register > PPGC(n) → 00000000_B
 TRG → 00000000_00000000_B
 REVC → 00000000_00000000_B
 GATEC0 → XX00XX00_B
 GATEC4 → XXXXXX00_B
 GATEC8 → XX00XX00_B
 GATEC12 → XXXXXX00_B

 < Pulse output > PPG(n) → "L"
 < Interruption request > IRQ(n) → "L" (n = 0 to 15)

Any hardware other than those above is not initialized.

● PPG combinations

ch.0: PPGC		ch.2: PPGC		ch.0	ch.1	ch.2	ch.3
MD1	MD0	MD1	MD0				
0	0	0	0	8-bit PPG	8-bit PPG	8-bit PPG	8-bit PPG
0	0	0	1	8-bit PPG	8-bit PPG	8-bit PPG ←	8-bit prescaler
0	0	1	0	8-bit PPG	8-bit PPG	16-bit PPG	
0	0	1	1	Setting disabled			
0	1	0	0	8-bit PPG ←	8-bit prescaler	8-bit PPG	8-bit PPG
0	1	0	1	8-bit PPG ←	8-bit prescaler	8-bit PPG ←	8-bit prescaler
0	1	1	0	8-bit PPG ←	8-bit prescaler	16-bit PPG	
0	1	1	1	Setting disabled			
1	0	0	0	16-bit PPG		8-bit PPG	8-bit PPG
1	0	0	1	16-bit PPG		8-bit PPG	8-bit prescaler
1	0	1	0	16-bit PPG		16-bit PPG	
1	0	1	1	Setting disabled			
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1	16-bit PPG ←		16-bit prescaler	

The operations for ch.4 to ch.7, ch.8 to ch.11* and ch.12 to ch.15* can be combined in the same way as for ch. (0, 1, 2, 3).

Replace as shown below.

$$\left\{ \begin{array}{l} \text{ch.0} = \text{ch.4} / * \text{ch.8} / * \text{ch.12} \\ \text{ch.1} = \text{ch.5} / * \text{ch.9} / * \text{ch.13} \\ \text{ch.2} = \text{ch.6} / * \text{ch.10} / * \text{ch.14} \\ \text{ch.3} = \text{ch.7} / * \text{ch.11} / * \text{ch.15} \end{array} \right.$$

*: ch.8 to ch.15 do not exist in the MB91470 series.

CHAPTER 12

MULTI-FUNCTION TIMER

This chapter explains the overview of the multi-function timer, the configuration and functions of registers, and operation of the multi-function timer.

- 12.1 Overview of the Multi-function Timer
- 12.2 Block Diagram of the Multi-function Timer
- 12.3 Pins of the Multi-function Timer
- 12.4 Multi-function Timer Register
- 12.5 Multi-function Timer Interrupt
- 12.6 Operation of the Multi-function Timer
- 12.7 Notes on Using the Multi-function Timer
- 12.8 Example Program for Multi-function Timer

12.1 Overview of the Multi-function Timer

Multi-function timer consists of three 16-bit free-run timer, six 16-bit output compares, four 16-bit input captures, one waveform generator, and three A/D activation compares. If this waveform generator is used with the PPG timer, 6 different waveforms can be output from the 16-bit free-run timer, an input pulse width and an external clock cycle can be measured. In MB91470 series, one multi-function timer and 8-channel PPG are installed. In MB91480 series and two multi-function timers are installed.

■ Structure of Multi-function Timer

● 16-bit free-run timer (× 3)

- The 16-bit free-run timer consists of 16-bit up/down counters, control registers, 16-bit compare clear registers (with buffer registers), and prescalers.
- Nine different counter operating clocks are available (ϕ , $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/128$, and $\phi/256$). (ϕ : Peripheral clock (CLKP))
- A compare clear interrupt is generated when a compare clear register compares and matches 16-bit free-run timer. A zero-detection interrupt is generated while the 16-bit free-run timer detects a count value "0".
- A compare clear register has selectable buffer registers. (Data written to this buffer register is transferred to a compare clear register.) When the 16-bit free-run timer is stopped and data is written to the buffer, the transfer is performed immediately. When the timer value "0" is detected during the 16-bit free-run timer operation, data is transferred from the buffer.
- When a compare match with a reset, a software clear, or a compare clear register occurs in the up count mode, the counter value is reset to "0000_H".
- This counter output value can be used as a multi-function timer output compare and an input capture clock count.
- When a zero-detection or a compare match occurs, A/D can be activated.
- The connections between the free-run timer and the resource can be set by the free-run timer selector or the resource input selector. However, the connections may be fixed depending on the series.

● 16-bit output compare (× 6)

- The 16-bit output compare consists of six 16-bit compare registers (with selectable buffer registers), compare output latches, and compare control registers. An interrupt is generated and the output level is inverted when a match occurs between the value of the selected 16-bit free-run timer and the compare register.
- Six compare registers can be operated independently. An output pin and an interrupt flag are assigned to each compare register.
- Two compare registers can be paired to control an output pin. The output pin can be reversed with using two compare registers together.
- The initial value of each output pin can be set.
- An interrupt can be generated when the output compare register matches the 16-bit free-run timer.
- Any channel of free-run timer for each compare unit can be set.

● 16-bit input capture (× 4)

- The input capture consists of four independent external input pins, and capture registers and capture control registers associated to these pins. Detection of an edge on the input signal from the external pin causes the value of the selected 16-bit free-run timer to be stored to the capture register and an interrupt to be generated.
- The trigger edge for the external input signal can be selected from the three types: Rising edge, falling edge, and both edges. Also there are registers that indicate whether the trigger edge is a rising edge or a falling edge.
- Four input capture can be used independently.
- An interrupt can be generated when a valid edge of an external input signal is detected.
- The input free-run timers of input capture unit can be configured in all series.

● 8/16-bit PPG timer (× 8: MB91470 series, × 16: MB91480 series)

- In MB91470 series, the PPG ch.0/ch.2/ch.4 are used for the output waveform to the waveform generator.
- In MB91480 series, the PPG ch.0/ch.2/ch.4 can be used for the output waveform to the waveform generator 0. The PPG ch.8/ch.10/ch.12 can be used for the output waveform to the waveform generator 1.
- See "CHAPTER 11 PPG" for details of the PPG timer.

● **Waveform generator**

- The waveform generator consists of three 16-bit dead timer registers, three timer control registers, and one 16-bit waveform control register.
- The waveform generator can generate real-time output, 16-bit PPG waveform output, non-overlap 3-phase waveform output (for inverter control), and DC chopper waveform output.
- The non-overlap waveform output can be generated on the basis of the dead time of the 16-bit dead timer (dead time timer function).
- The non-overlap waveform output can be generated when the real-time output is activated in 2 channel mode (dead time timer function).
- When a real-time output compare match is detected, GATE signal is generated to start or stop the PPG timer operation (GATE function).
- When a real-time output compare match is detected, the 16-bit dead timer becomes active. With generating the GATE signal for controlling the PPG operation, the input PPG timer can be started or stopped easily (GATE function).
- DTTI pins can be used to control stopping forcibly.
- DTTI registers can be used to control stopping forcibly.

● **A/D activation compare (× 3)**

- The A/D can be activated when a match occurs between the value of the 16-bit free-run timer and the compare register. On the waveform generator 0, the 16-bit free-run timer ch.0/ch.1/ch.2 is selected as the free-run timer input. On the waveform generator 1, the 16-bit free-run timer ch.3/ch.4/ch.5 is selected as the free-run timer input.
- The A/D can be activated when the free-run timer value is corresponding to the compare register at up-counting 16-bit free-run timer.
- The A/D can be activated when the free-run timer value is corresponding to the compare register at down-counting 16-bit free-run timer.
- The A/D can be activated when the free-run timer value is corresponding to the compare register at up/down-counting 16-bit free-run timer.
- A separate value can be set to the two compare registers respectively. The A/D can be activated when the free-run timer value is corresponding to the compare register 0 at up-counting 16-bit free-run timer. And the A/D can be activated when the free-run timer value is corresponding to the compare register 1 at down-counting 16-bit free-run timer.

■ Difference between Multi-function Timers 0 and 1

- 16-bit free-run timer
 - Multi-function timer 0 has three channel free-run timer 0 to 2.
 - Multi-function timer 1 has three channel free-run timer 3 to 5.
- 16-bit output compare
 - In multi-function timer 0, there are six channel output compare 0 to 5, and either of free-run timer 0 to 2 can be selected respectively as an output compare 0 to 5 input.
 - In multi-function timer 1, there are six channel output compare 6 to 11, and multi-function timer 0 is installed, either of free-run timer 0 to 5 can respectively be selected as an input for output compare 0 to 11.
- 16-bit input capture
 - In multi-function timer 0, there are four channel input capture 0 to 3, and either of free-run timer 0 to 2 can be selected respectively as an input for input capture 0 to 3.
 - In multi-function timer 1, there are four channel input capture 4 to 7, and multi-function timer 0 is installed, either of free-run timer 0 to 5 can respectively be selected as an input for input capture 0 to 11.
- Waveform generator
 - In multi-function timer 0, there is one unit waveform generator 0 with real-time output 0 to 5, and PPG0/2/4 is selected as an output waveform and it is possible to use it.
 - In multi-function timer 1, there is one unit waveform generator 1 with real-time output 6 to 11, and PPG8/10/12 is selected as an output waveform and it is possible to use it.
- A/D activating compare
 - In multi-function timer 0, there are three channel trigger output 0 to 2 for the A/D activating, and either of free-run timer 0 to 2 can be selected respectively as an A/D activating compare 0 to 2 input.
 - In multi-function timer 1, there are three channel trigger output 3 to 5 for the A/D activating, and either of free-run timer 3 to 5 can be selected respectively as an A/D activating compare 3 to 5 input.

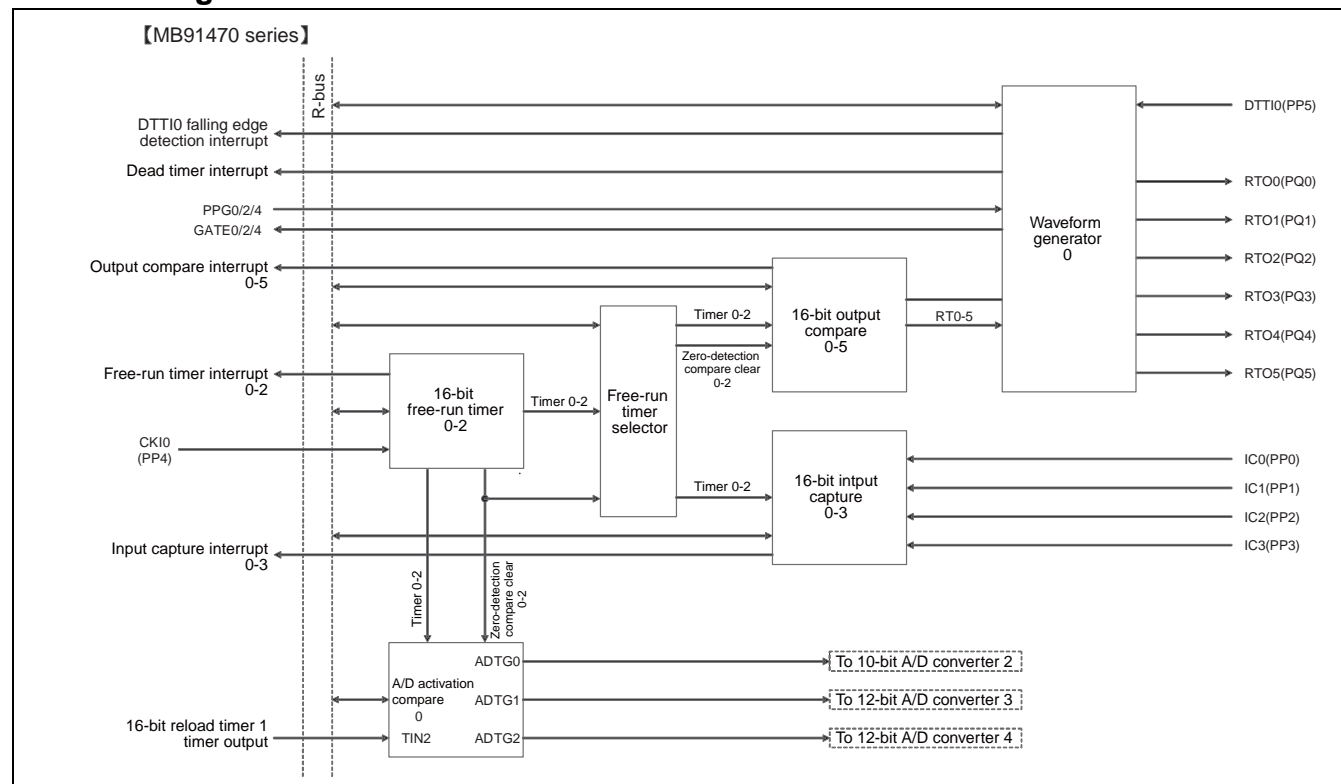
■ Difference Point of Multi-function Timer Construction in Each Series

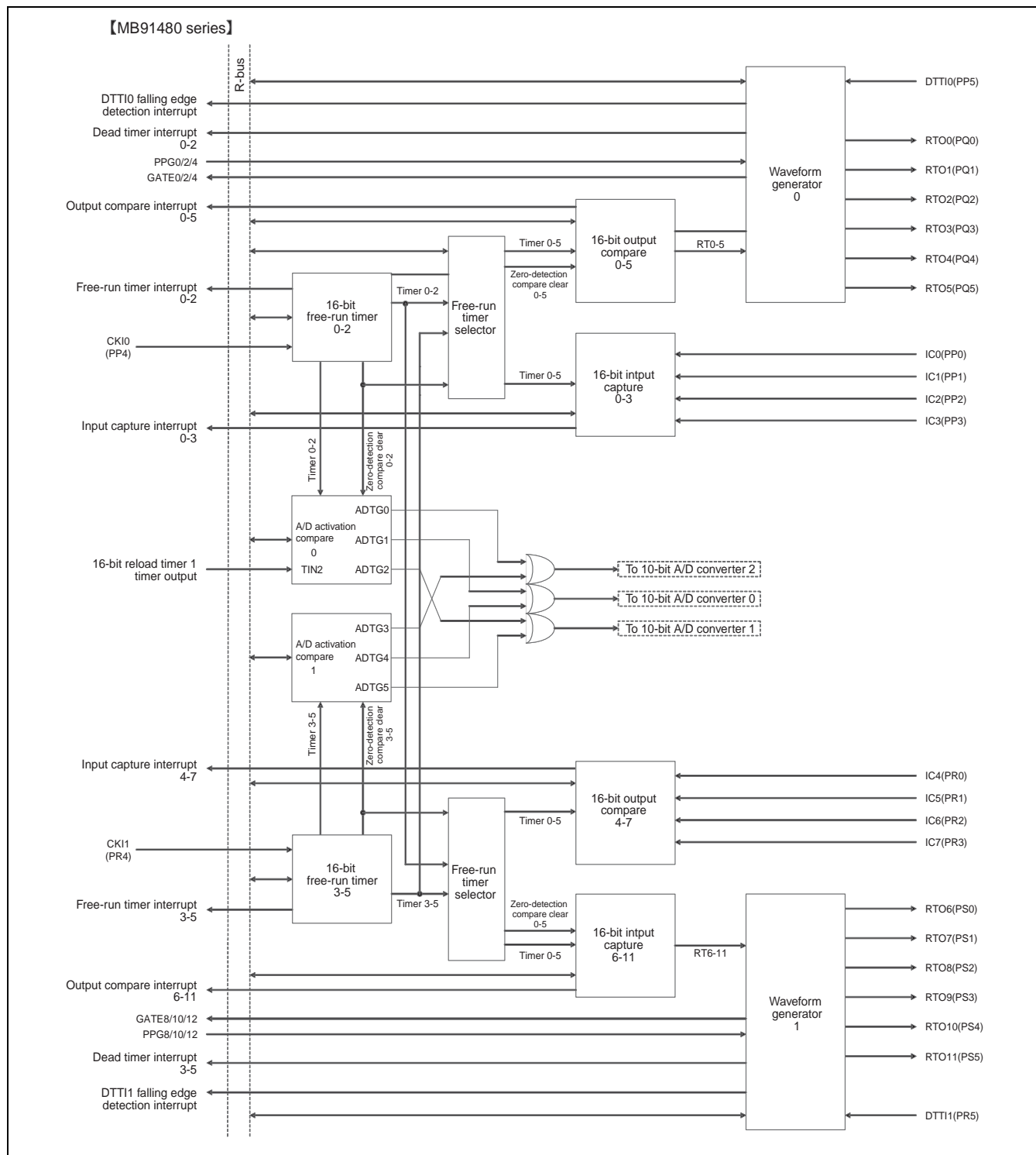
- MB91470 series
 - Only multi-function timer 0 is installed.
 - Only PPG ch.0/ch.2/ch.4 can be used as the output waveform to the waveform generator 0.
- MB91480 series
 - Both multi-function timers 0 and 1 are installed.
 - Either of PPG ch.0/ch.2/ch.4 is selected as an output waveform to the waveform generator 0.
 - Either of PPG ch.8/ch.10/ch.12 is selected as an output waveform to the waveform generator 1.

12.2 Block Diagram of the Multi-function Timer

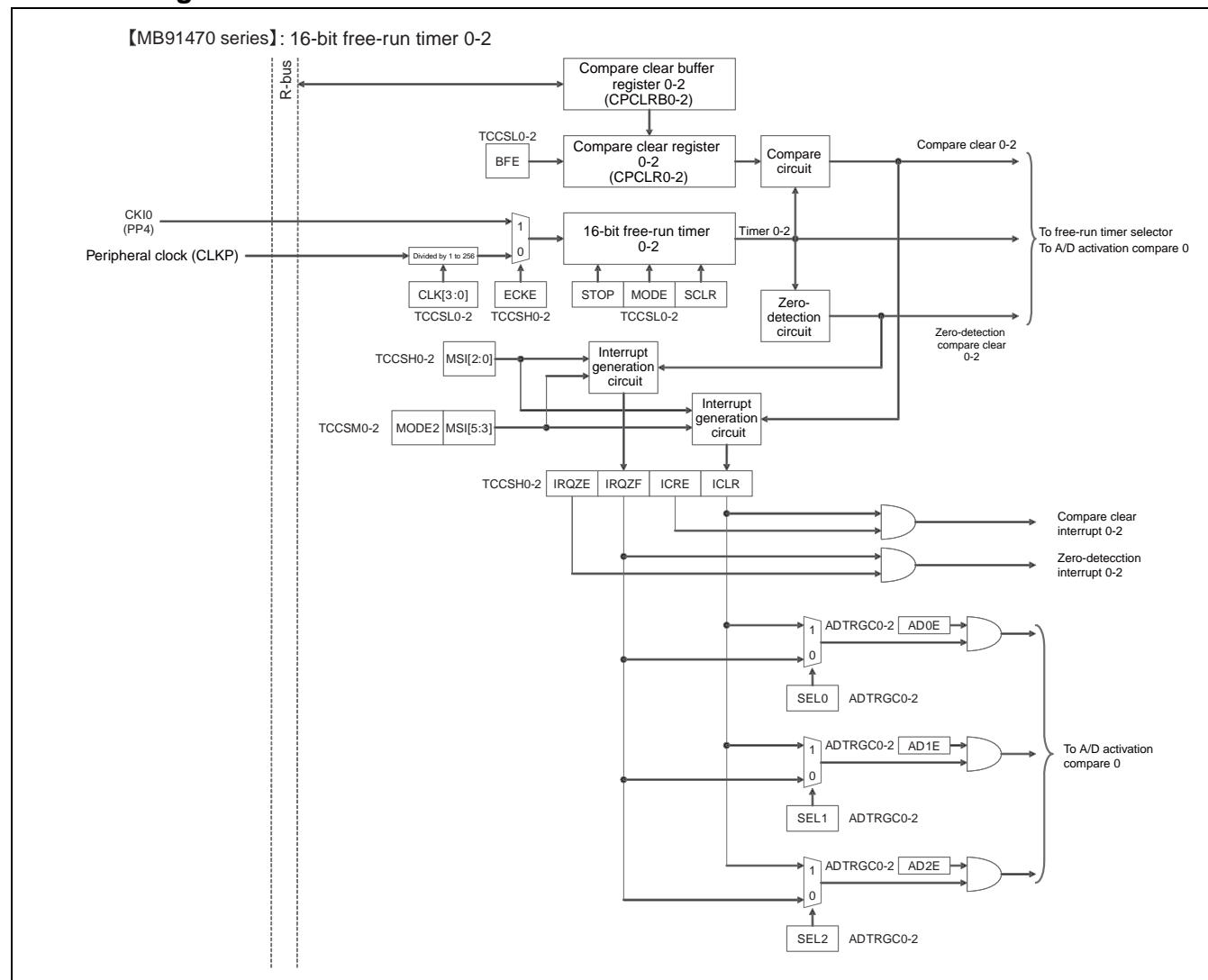
This section shows the block diagram of the multi-function timer.

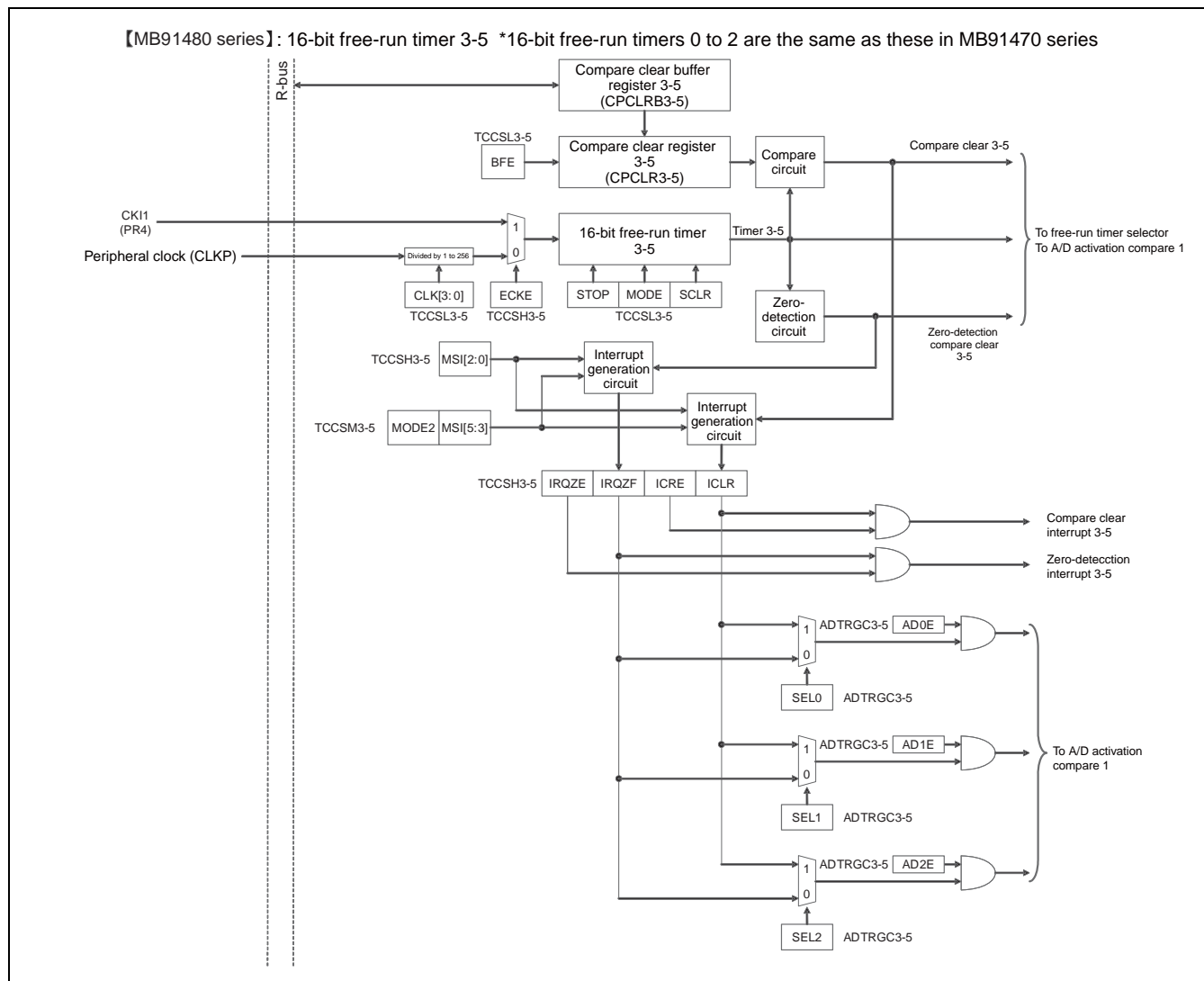
■ Block Diagram of the Multi-function Timer



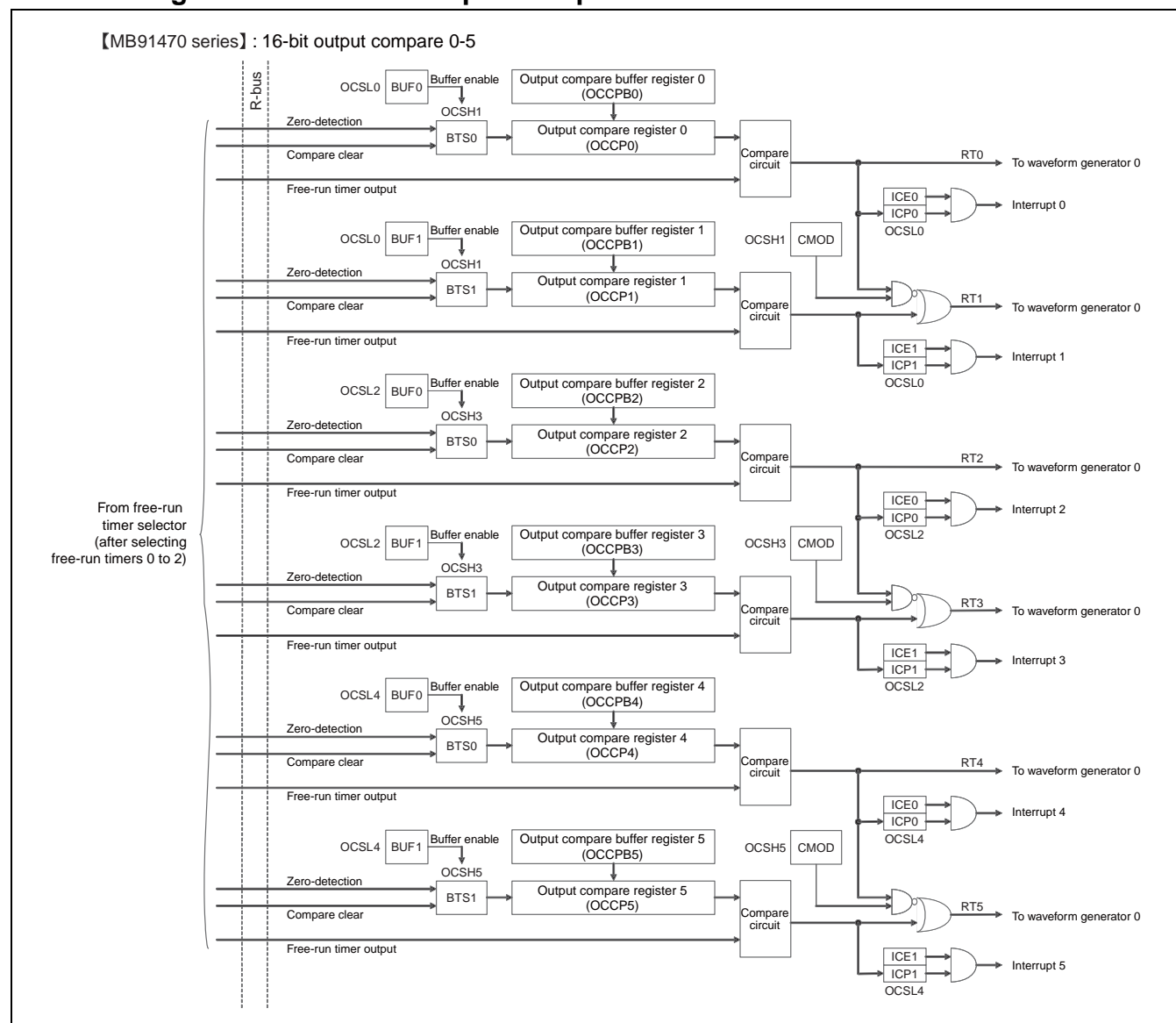


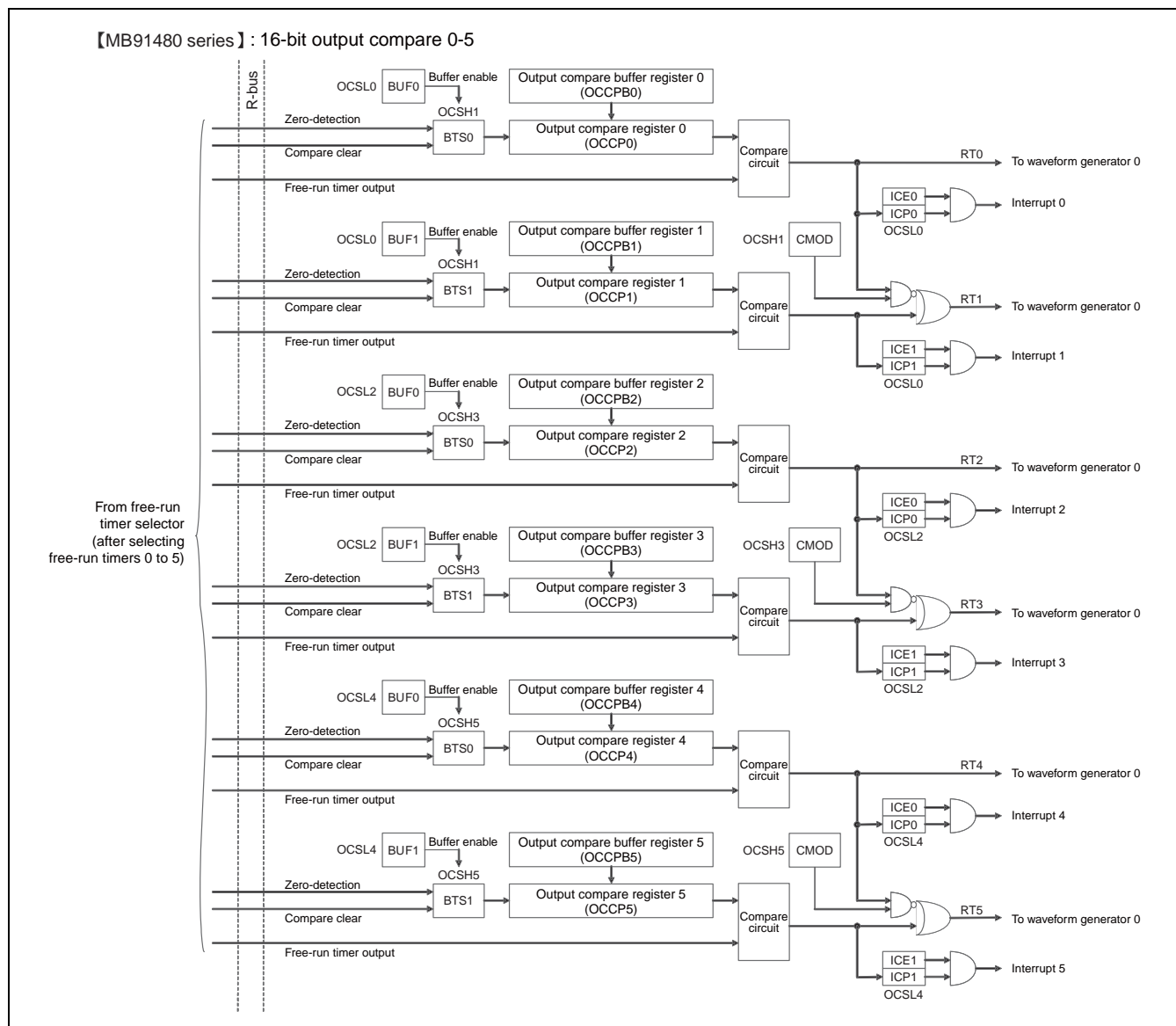
■ Block Diagram of 16-bit Free-run Timer

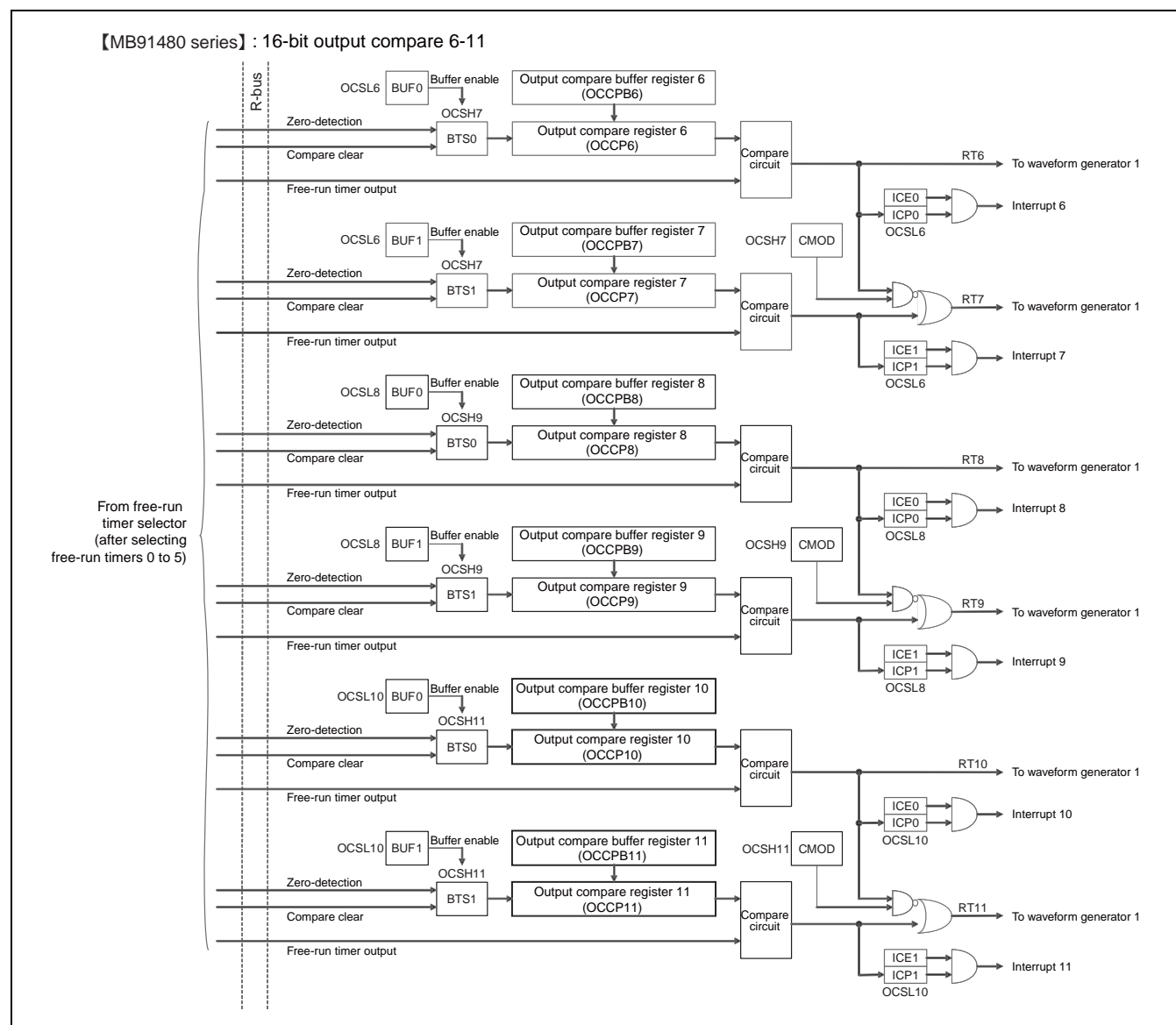




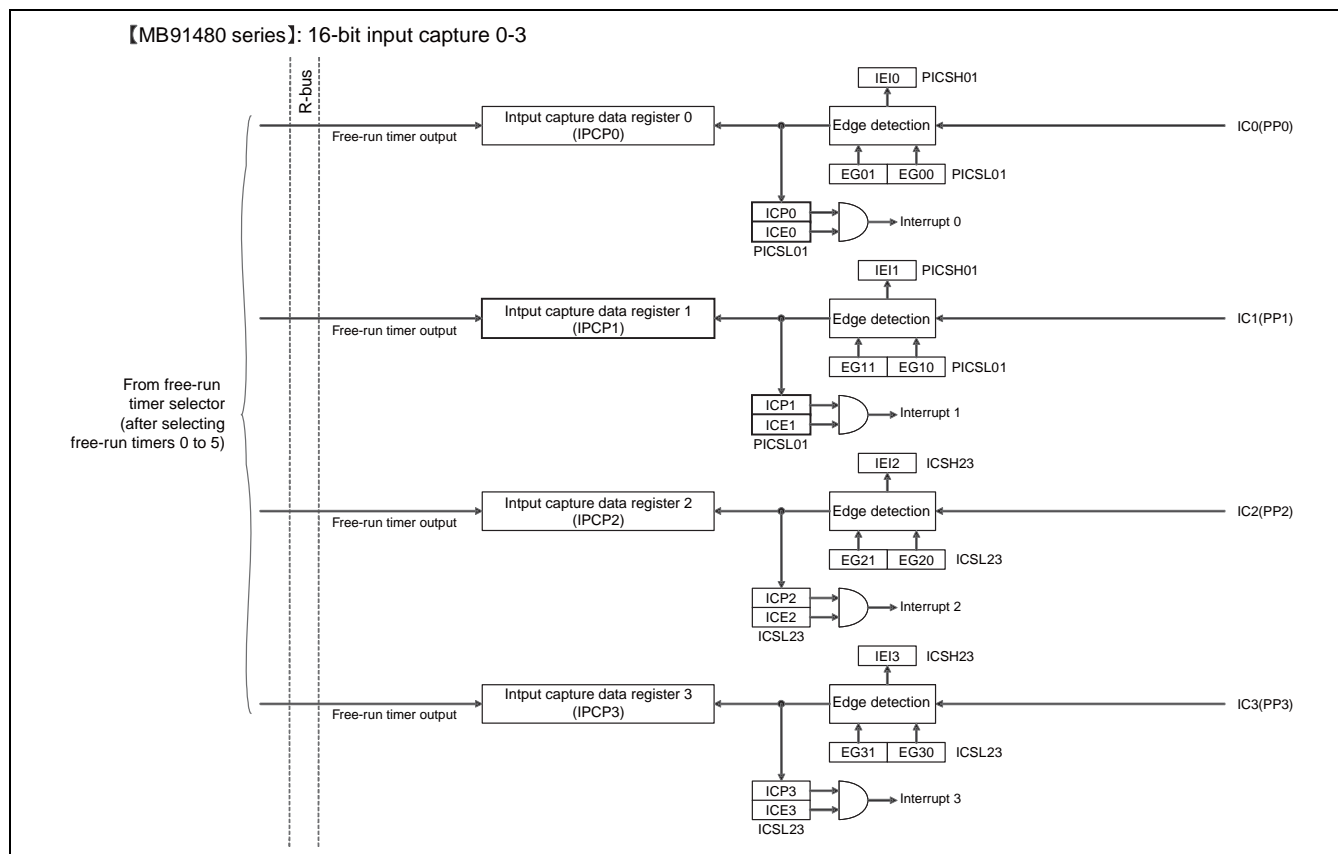
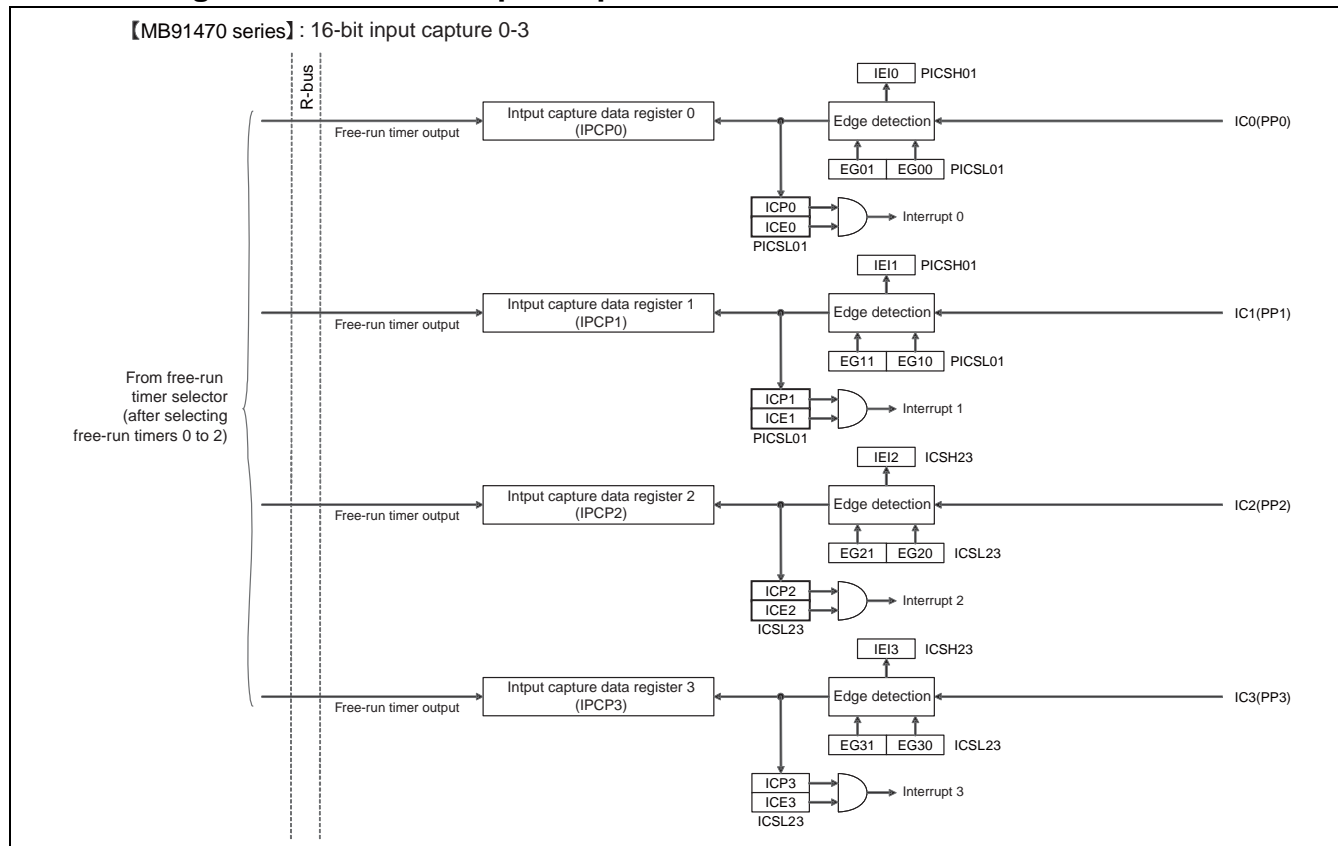
■ Block Diagram of the 16-bit Output Compare

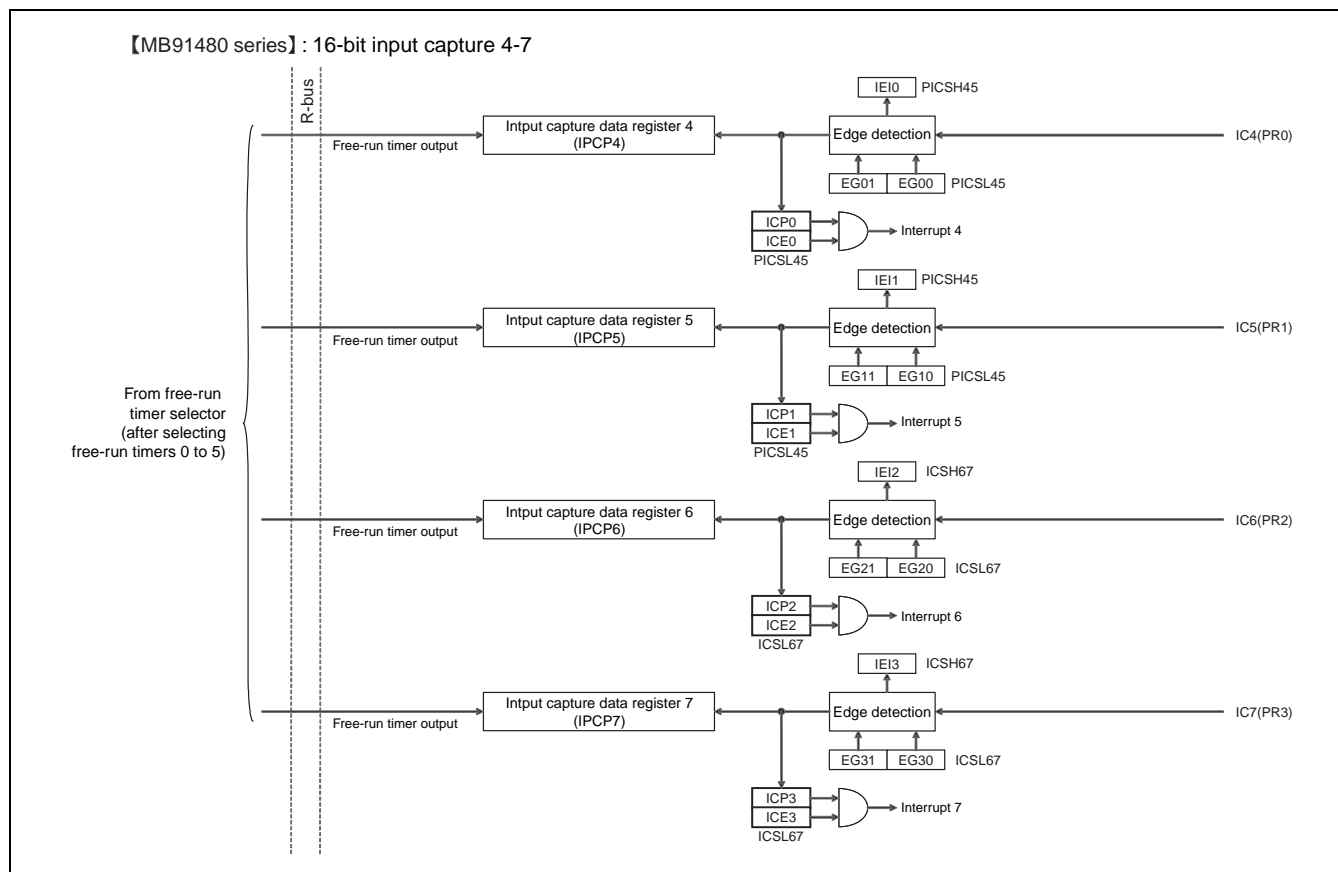




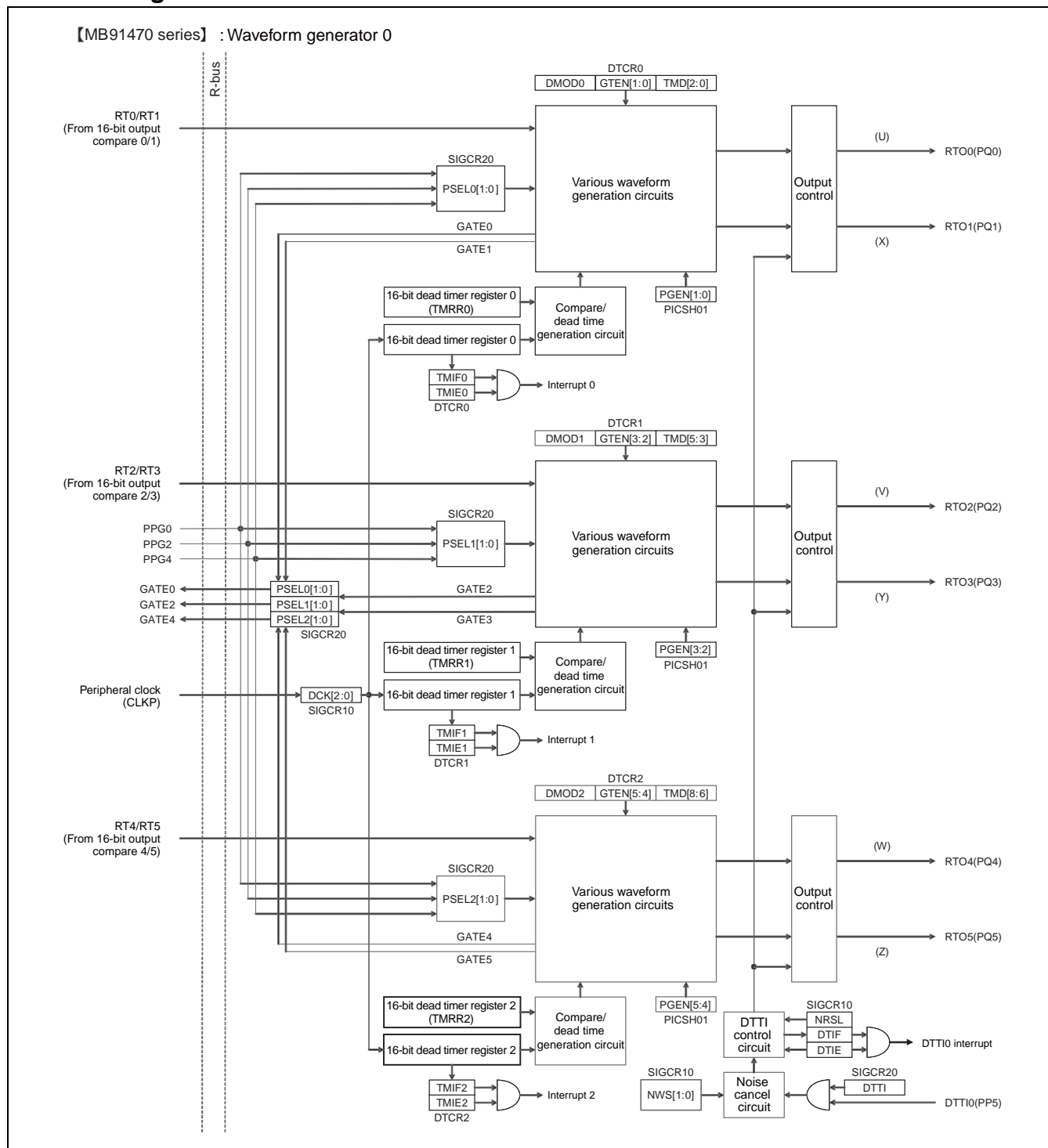


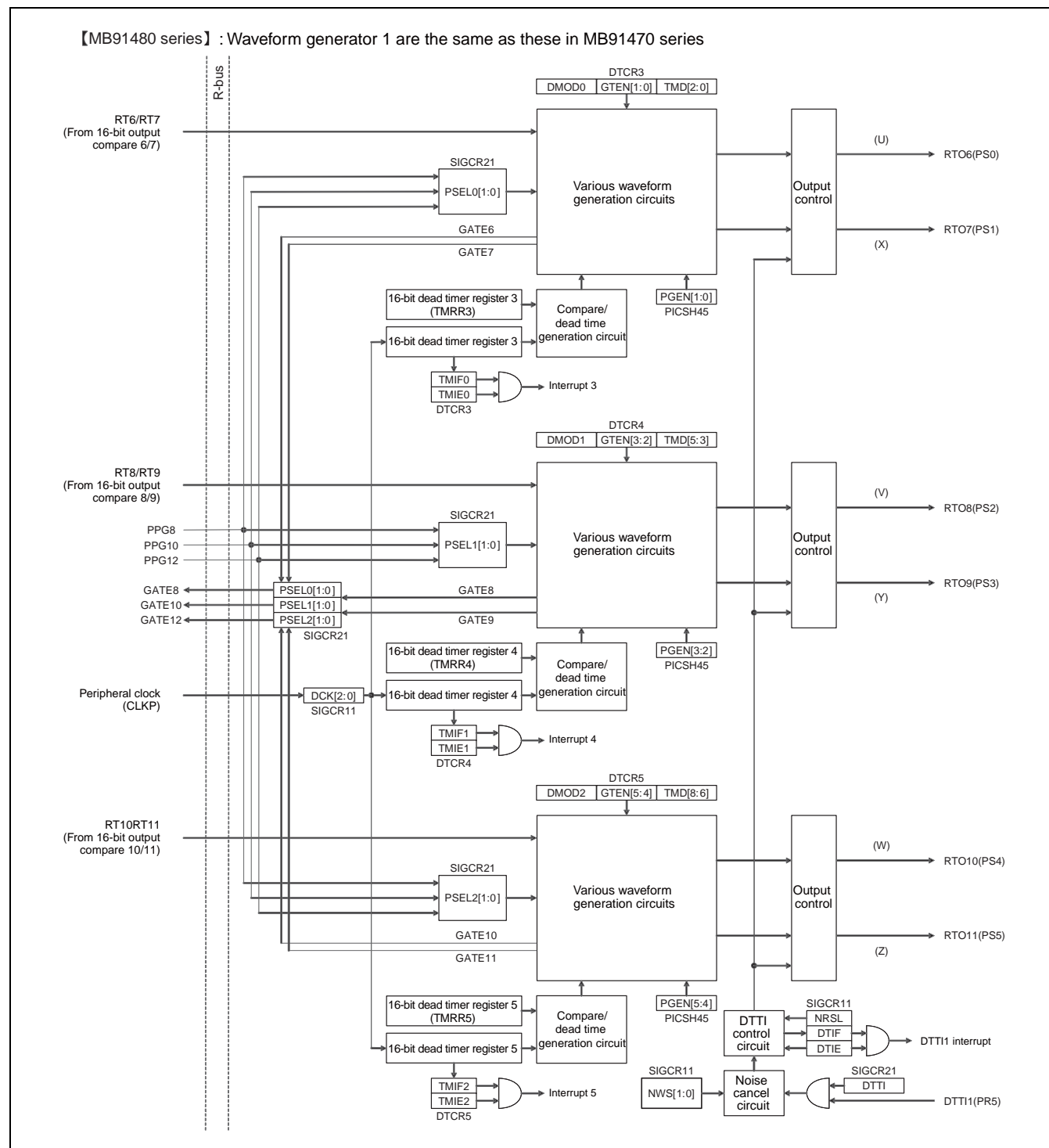
■ Block Diagram of the 16-bit Input Capture



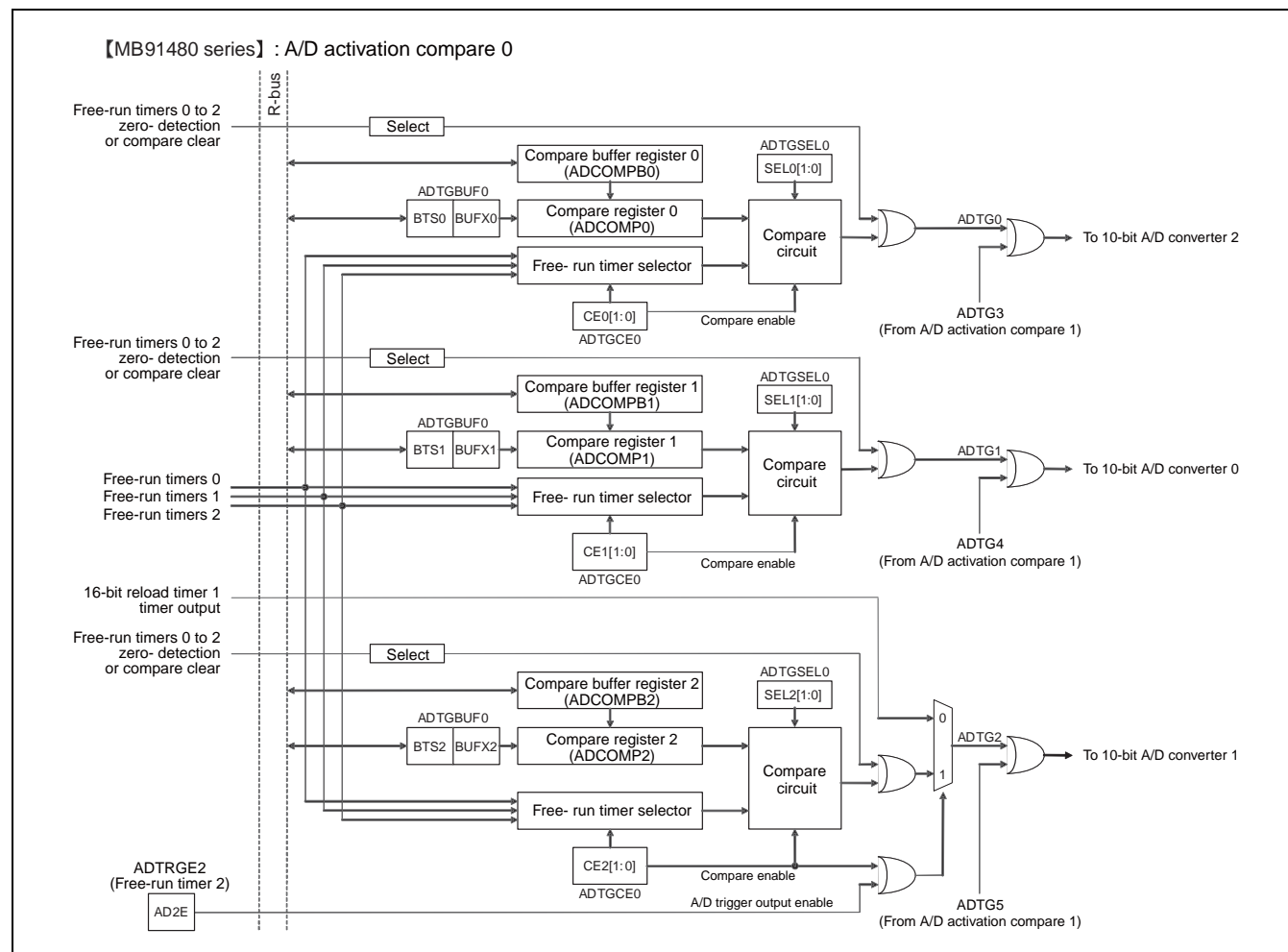


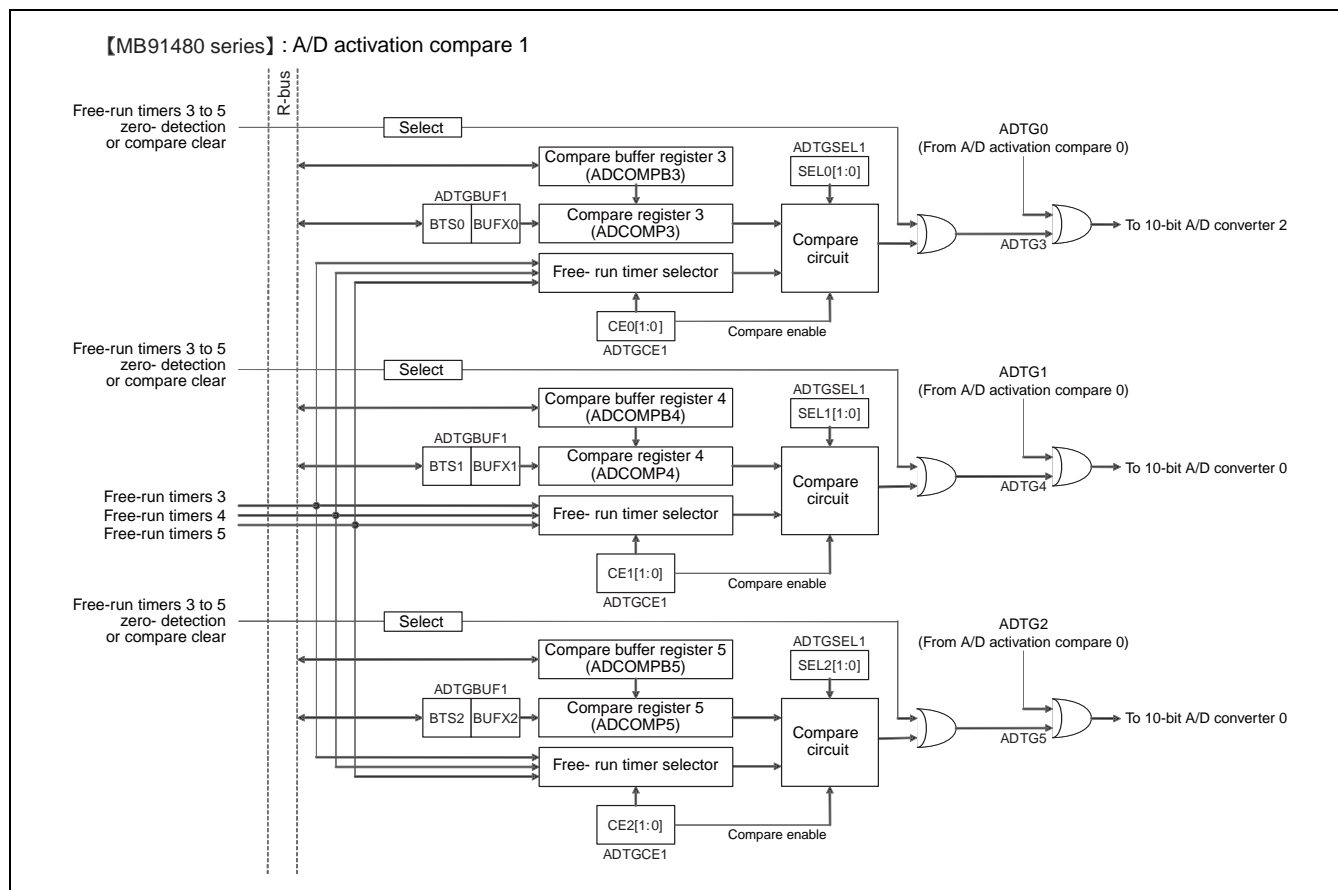
■ Block Diagram of the Waveform Generator



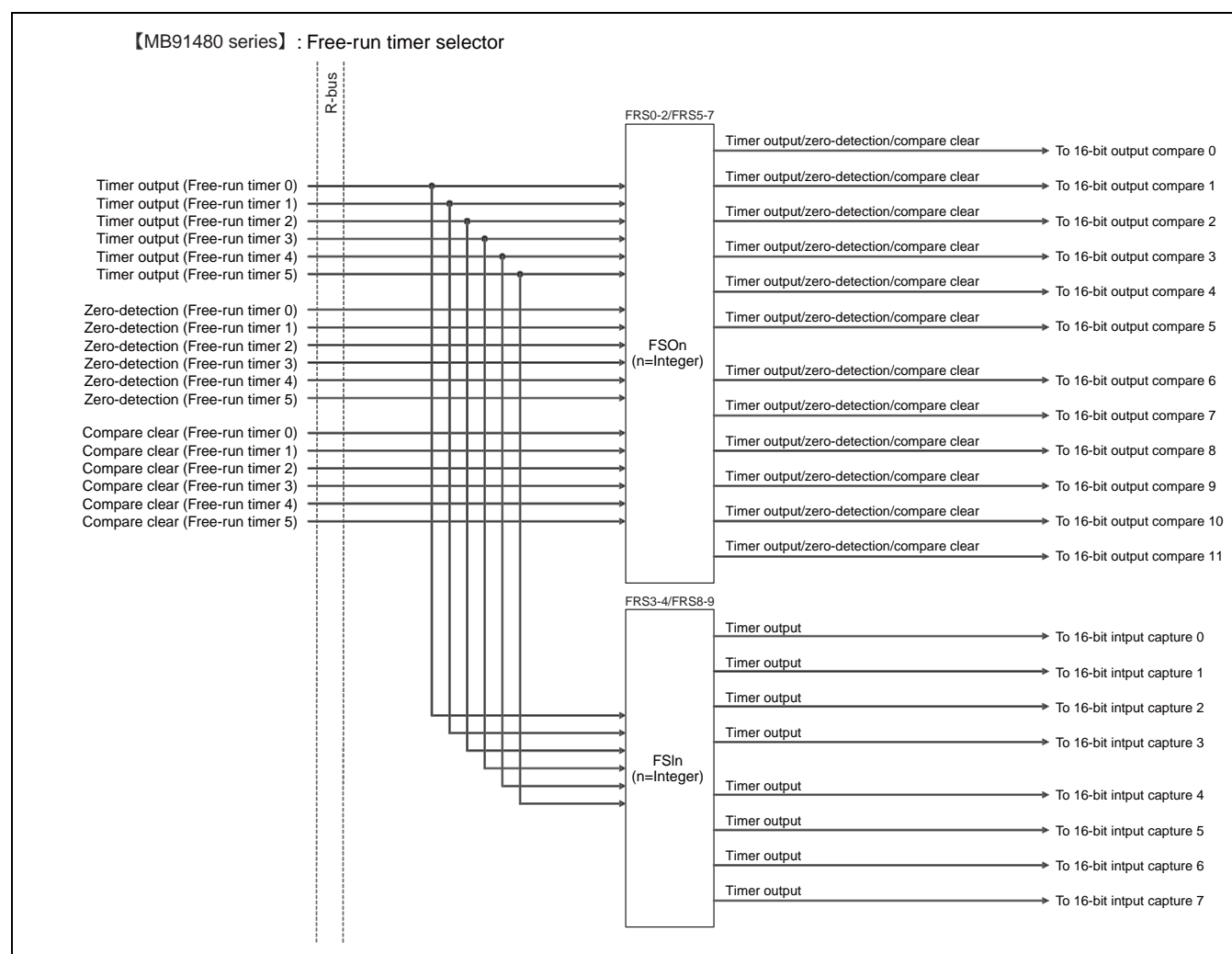
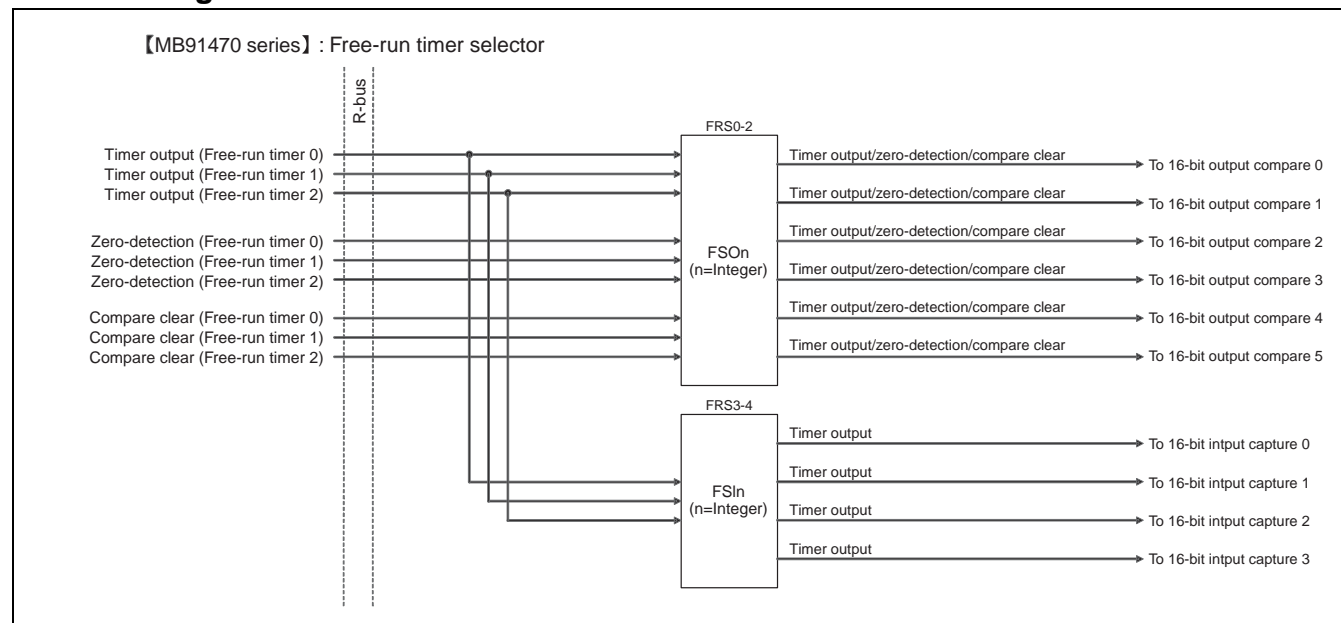


[illegible]





■ Block Diagram of the Free-run Timer Selector



MB91470/480 Series

12.3 Pins of the Multi-function Timer

This section describes the pins of the multi-function timer.

■ Pins of the Multi-function Timer

Table 12.3-1 Pins of the Multi-function Timer 0

Pin name	Pin function	I/O Type	Pull-up option	Standby control	Pin setting
PP5/DTTI0	Port P, I/O, DTTI	CMOS output, CMOS hysteresis input	Selectable	Yes	Set a pin as input port (DDRP: bit5 = 0)
PP4/CKI0	Port P, I/O, External clock				Set a pin as input port (DDRP: bit4 = 0)
PP0/IC0	Port P, I/O, Input capture 0				Set a pin as input port (DDRP: bit0 = 0)
PP1/IC1	Port P, I/O, Input capture 1				Set a pin as input port (DDRP: bit1 = 0)
PP2/IC2	Port P, I/O, Input capture 2				Set a pin as input port (DDRP: bit2 = 0)
PP3/IC3	Port P, I/O, Input capture 3				Set a pin as input port (DDRP: bit3 = 0)
PQ0/RTO0 (U)	Port Q, I/O, RTO0				Set RTO0 output (DDRQ: bit0 = 1)
PQ1/RTO1 (X)	Port Q, I/O, RTO1				Set RTO1 output (DDRQ: bit1 = 1)
PQ2/RTO2 (V)	Port Q, I/O, RTO2				Set RTO2 output (DDRQ: bit2 = 1)
PQ3/RTO3 (Y)	Port Q, I/O, RTO3				Set RTO3 output (DDRQ: bit3 = 1)
PQ4/RTO4 (W)	Port Q, I/O, RTO4				Set RTO4 output (DDRQ: bit4 = 1)
PQ5/RTO5 (Z)	Port Q, I/O, RTO5				Set RTO5 output (DDRQ: bit5 = 1)

DDRx: Port direction register

Table 12.3-2 Pins of the Multi-function Timer 1

Pin name	Pin function	I/O Type	Pull-up option	Standby control	Pin setting
PR5/DTTI1	Port R, I/O, DTTI	CMOS output, CMOS hysteresis input	Selectable	Yes	Set a pin as input port (DDRR: bit5 = 0)
PR4/CKI1	Port R, I/O, External clock				Set a pin as input port (DDRR: bit4 = 0)
PR0/IC4	Port R, I/O, Input capture 4				Set a pin as input port (DDRR: bit0 = 0)
PR1/IC5	Port R, I/O, Input capture 5				Set a pin as input port (DDRR: bit1 = 0)
PR2/IC6	Port R, I/O, Input capture 6				Set a pin as input port (DDRR: bit2 = 0)
PR3/IC7	Port R, I/O, Input capture 7				Set a pin as input port (DDRR: bit3 = 0)
PS0/RTO6 (U)	Port S, I/O, RTO6				Set RTO6 output (DDRQ: bit0 = 1)
PS1/RTO7 (X)	Port S, I/O, RTO7				Set RTO7 output (DDRQ: bit1 = 1)
PS2/RTO8 (V)	Port S, I/O, RTO8				Set RTO8 output (DDRQ: bit2 = 1)
PS3/RTO9 (Y)	Port S, I/O, RTO9				Set RTO9 output (DDRQ: bit3 = 1)
PS4/RTO10 (W)	Port S, I/O, RTO10				Set RTO10 output (DDRQ: bit4 = 1)
PS5/RTO11 (Z)	Port S, I/O, RTO11				Set RTO11 output (DDRQ: bit5 = 1)

DDRx: Port direction register

12.4 Multi-function Timer Register

This section describes the multi-function timer 0/1 registers.

* marks mean the multi-function timer 1.

16-bit Free-run Timer Register

Compare clear buffer register, Compare clear register (Upper)

CPCLRBHn/CPCLRHn

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	Initial value 11111111 _B
CPCLRBH Write	→ W	→ W	→ W	→ W	→ W	→ W	→ W	→ W	
CPCLRH Read	→ R	→ R	→ R	→ R	→ R	→ R	→ R	→ R	

Address:

ch.0: 0000B4H ch.3*: 0001B4H
ch.1: 0000BCH ch.4*: 0001BCH
ch.2: 0000C4H ch.5*: 0001C4H

Compare clear buffer register, Compare clear register (Lower)

CPCLRBLn/CPCLRLn

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	Initial value 11111111 _B
CPCLRBL Write	→ W	→ W	→ W	→ W	→ W	→ W	→ W	→ W	
CPCLRL Read	→ R	→ R	→ R	→ R	→ R	→ R	→ R	→ R	

Timer data register (Upper)

TCDTHn

Address:

ch.0: 0000B6H
ch.1: 0000BEH
ch.2: 0000C6H
ch.3*: 0001B6H
ch.4*: 0001BEH
ch.5*: 0001C6H

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	T15	T14	T13	T12	T11	T10	T09	T08	Initial value 00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer data register (Lower)

TCDTLn

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	T07	T06	T05	T04	T03	T02	T01	T00	Initial value 00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

n = 0/1/2:FRT0/1/2, n = 3/4/5:FRT3/4/5

*: In MB91470 series, only ch.0 to ch.2 are available.
In MB91480 series, ch.0 to ch.5 are available.

(Continued)

(Continued)

Timer state control register (Upper)

TCCSHn

Address:

ch.0: 0000B8H

ch.1: 0000C0H

ch.2: 0000C8H

ch.3*: 0001B8H

ch.4*: 0001C0H

ch.5*: 0001C8H

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE	Initial value 00000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer state control register (Lower)

TCCSLn

Address:

ch.0: 0000B9H

ch.1: 0000C1H

ch.2: 0000C9H

ch.3*: 0001B9H

ch.4*: 0001C1H

ch.5*: 0001C9H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0	Initial value 01000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer state control register M

TCCSMn

Address:

ch.0: 0000BAH

ch.1: 0000C2H

ch.2: 0000CAH

ch.3*: 0001BAH

ch.4*: 0001C2H

ch.5*: 0001CAH

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
-	-	-	-	MODE2	MSI5	MSI4	MSI3	Initial value ----0000 _B
-	-	-	-	R/W	R/W	R/W	R/W	

A/D trigger control register

ADTRGCn

Address:

ch.0: 0000BBH

ch.1: 0000C3H

ch.2: 0000CBH

ch.3*: 0001BBH

ch.4*: 0001C3H

ch.5*: 0001CBH

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
-	SEL2	SEL1	SEL0	-	AD2E	AD1E	AD0E	Initial value -000-000 _B
-	R/W	R/W	R/W	-	R/W	R/W	R/W	

R/W: Readable/writable

n = 0/1/2:FRT0/1/2, n = 3/4/5:FRT3/4/5

*: In MB91470 series, only ch.0 to ch.2 are available.
In MB91480 series, ch.0 to ch.5 are available.

Free-run Timer Selection Registers

Free-run timer selection register (Upper) for output compare

FRS1,FRS6*

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value	
FRS1: 0000CE _H	-	FSO14	FSO13	FSO12	-	FSO10	FSO9	FSO8	FRS1	FRS6
FRS6*: 0001CE _H	-	R/W	R/W	R/W	-	R/W	R/W	R/W	-000-000 _B	-011-011 _B

Free-run timer selection register (Lower) for output compare

FRS0,FRS2,

FRS5*,FRS7*

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
FRS0: 0000CF _H	-	FSO6	FSO5	FSO4	-	FSO2	FSO1	FSO0	FRS0/2	FRS5/7
FRS2: 0000CD _H	-	R/W	R/W	R/W	-	R/W	R/W	R/W	-000-000 _B	-011-011 _B
FRS5*: 0001CF _H										
FRS7*: 0001CD _H										

Free-run timer selection register (Upper) for input capture

FRS4,FRS9**

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value	
FRS4: 0000D2 _H	-	FSI14	FSI13	FSI12	-	FSI10	FSI9	FSI8	FRS4	FRS9
FRS9*: 0001D2 _H	-	R/W	R/W	R/W	-	R/W	R/W	R/W	-000-000 _B	-011-011 _B

Free-run timer selection register (Lower) for input capture

FRS3,FRS8*

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value	
FRS3: 0000D3 _H	-	FSI6	FSI5	FSI4	-	FSI2	FSI1	FSI0	FRS3	FRS8
FRS8*: 0001D3 _H	-	R/W	R/W	R/W	-	R/W	R/W	R/W	-000-000 _B	-011-011 _B

R/W: Readable/writable

*: Not available in MB91470 series.

■ 16-bit Output Compare Register

Output compare buffer register, Output compare register (Upper)

OCCPBH0 to OCCPBH5/ OCCPH0 to OCCPH5	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCCPBH6* to OCCPBH11*/ OCCPH6* to OCCPH11*	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	Initial value 00000000 _B

OCCPBH Write → W
OCCPH Read → R

Address

ch.0: 0000A0H ch.6: 0001A0H
ch.1: 0000A2H ch.7: 0001A2H
ch.2: 0000A4H ch.8: 0001A4H
ch.3: 0000A6H ch.9: 0001A6H
ch.4: 0000A8H ch.10: 0001A8H
ch.5: 0000AAH ch.11: 0001AAH

Output compare buffer register, Output compare register (Lower)

OCCPBL0 to OCCPBL5/ OCCPL0 to OCCPL5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCCPBL6* to OCCPBL11*/ OCCPL6* to OCCPL11*	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	Initial value 00000000 _B

OCCPBL Write → W
OCCPL Read → R

Compare control register 1,3,5,7*,9*,11* (Upper)

OCSH1,OCSH3,OCSH5	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCSH7*,OCSH9*,OCSH11*	-	BTS1	BTS0	CMOD	-	-	OTD1	OTD0	Initial value -110--00 _B
	-	R/W	R/W	R/W	-	-	R/W	R/W	

Address

ch.1: 0000ACH ch.7: 0001ACH
ch.3: 0000AEH ch.9: 0001AEH
ch.5: 0000B0H ch.11: 0001B0H

Compare control register 0,2,4,6*,8*,10* (Lower)

OCSL0,OCSL2,OCSL4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCSL6*,OCSL8*,OCSL10*	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0	Initial value 00001100 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address

ch.0: 0000ADH ch.6: 0001ADH
ch.2: 0000AFH ch.8: 0001AFH
ch.4: 0000B1H ch.10: 0001B1H

Compare mode control register

OCMOD0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCMOD1*	-	-	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	Initial value --000000 _B
	-	-	R/W	R/W	R/W	R/W	R/W	R/W	

Address

ch.0: 0000B2H
*ch.1: 0001B2H

R/W: Readable/writable

R: Read only

W: Write only

*: Not available in MB91470 series.

■ 16-bit Input Capture Register

Input capture data register (Upper)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
IPCPH0 to IPCPH3									
IPCPH4* to IPCPH7*	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	Initial value XXXXXXXX _B
Address	R	R	R	R	R	R	R	R	
ch.0: 0000D4H									
*ch.4: 0001D4H									
ch.1: 0000D6H									
*ch.5: 0001D6H									
ch.2: 0000D8H									
*ch.6: 0001D8H									
ch.3: 0000DAH									
*ch.7: 0001DAH									

Input capture data register (Lower)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
IPCPL0 to IPCPL3									
IPCPL4* to IPCPL7*	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	Initial value XXXXXXXX _B
	R	R	R	R	R	R	R	R	

Input capture state control register (ch. 2, ch. 3*) (Upper)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ICSH23									
ICSH67	-	-	-	-	-	-	IEI3	IEI2	Initial value -----00 _B
Address							R	R	
ch.0: 0000DEH									
ch.1: 0001DEH									

Input capture state control register (ch. 2, ch. 3, ch. 6*, ch. 7*) (Lower)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICSL23									
ICSL67	ICP3	ICP2	ICE3	ICE2	EG31	EG30	EG21	EG20	Initial value 00000000 _B
Address	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.0: 0000DFH									
ch.1: 0001DFH									

PPG output control / Input capture state control register (ch. 0, ch. 1, ch. 4*, ch. 5*) (Upper)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
PICSH01									
PICSH45	PGEN5	PGEN4	PGEN3	PGEN2	PGEN1	PGEN0	IEI1	IEI0	Initial value 00000000 _B
Address	W	W	W	W	W	W	R	R	
ch.0: 0000DCH									
ch.1: 0001DCH									

Input capture state control register (ch. 0, ch. 1, ch. 4*, ch. 5*) (Lower)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
PICSL01									
PICSL45	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	Initial value 00000000 _B
Address	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.0: 0000DDH									
ch.1: 0001DDH									

*: Not available in MB91470 series.

R/W: Readable/writable

R: Read only

W: Write only

■ Waveform Generator Register

16-bit dead timer register (Upper)

TMRRH0, TMRRH1, TMRRH2, TMRRH3*, TMRRH4*, TMRRH5*

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
Waveform generator0:	TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	Initial value XXXXXXXX _B
ch.0: 0000E0H									
ch.1: 0000E2H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 0000E4H									
Waveform generator1*:									
ch.3: 0001E0H									
ch.4: 0001E2H									
ch.5: 0001E4H									

16-bit dead timer register (Lower)

TMRRL0, TMRRL1, TMRRL2, TMRRL3*, TMRRL4*, TMRRL5*

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	Initial value XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit dead timer control register 0, 3*

DTCR0

DTCR3

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
Waveform generator0 : 0000E8H	DMOD0	GTEN1	GTEN0	TMIF0	TMIE0	TMD2	TMD1	TMD0	Initial value 00000000 _B
Waveform generator1* : 0001E8H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit dead timer control register 1, 4*

DTCR1

DTCR4

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
Waveform generator0 : 0000E9H	DMOD1	GTEN3	GTEN2	TMIF1	TMIE1	TMD5	TMD4	TMD3	Initial value 00000000 _B
Waveform generator1* : 0001E9H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16-bit dead timer control register 2, 5*

DTCR2

DTCR5

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
Waveform generator0 : 0000EAH	DMOD2	GTEN5	GTEN4	TMIF2	TMIE2	TMD8	TMD7	TMD6	Initial value 00000000 _B
Waveform generator1* : 0001EAH	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Waveform control register 10, 11*

SIGCR10

SIGCR11

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
Waveform generator0 : 0000EDH	DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0	Initial value 00000000 _B
Waveform generator1* : 0001EDH	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Waveform control register 20, 21

SIGCR20

SIGCR21

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
Waveform generator0 : 0000EFH	PSEL2[1]	PSEL2[0]	PSEL1[1]	PSEL1[0]	PSEL0[1]	PSEL0[0]	-	DTTI	Initial value 000000-1 _B
Waveform generator1* : 0001EFH	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	

R/W: Readable/writable

*: The waveform generator 1 is not available in MB91470 series.

■ A/D Activation Compare Register

Compare register 0, 1, 2, 3*, 4*, 5* (Upper)

ADCOMPB0/ADCOMP0
ADCOMPB1/ADCOMP1
ADCOMPB2/ADCOMP2
ADCOMPB3/ADCOMP3
ADCOMPB4/ADCOMP4
ADCOMPB5/ADCOMP5

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	Initial value 00000000 _B

ADCOMPB0 to ADCOMP5
Read/write
ADCOMP0 to ADCOMP5
Read/write

→ R	R	R	R	R	R	R	R	
→ W	W	W	W	W	W	W	W	

Address

ch.0: 0000F0H	ch.3: 0001F0H	ch.0: 0000F2H	ch.3: 0001F2H
ch.1: 0000F4H	ch.4: 0001F4H	ch.1: 0000F6H	ch.4: 0001F6H
ch.2: 0000F8H	ch.5: 0001F8H	ch.2: 0000FAH	ch.5: 0001FAH

Address (ADCOMPDP)

Compare register 0, 1, 2, 3*, 4*, 5* (Lower)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	Initial value 00000000 _B

ADCOMPB0 to ADCOMP5
Read/write
ADCOMP0 to ADCOMP5
Read/write

→ R	R	R	R	R	R	R	R	
→ W	W	W	W	W	W	W	W	

Compare enable register

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
-	-	CE2[1]	CE2[0]	CE1[1]	CE1[0]	CE0[1]	CE0[0]	Initial value --000000 _B

ADTGCE0
ADTGCE1*

Address - - R/W R/W R/W R/W R/W R/W

A/D activation compare 0: 0000FFH
A/D activation compare 1: 0001FFH

Count direction selection for comparison register

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
-	-	SEL2[1]	SEL2[0]	SEL1[1]	SEL1[0]	SEL0[1]	SEL0[0]	Initial value --000000 _B

ADTGSEL0
ADTGSEL1*

Address - - R/W R/W R/W R/W R/W R/W

A/D activation compare 0: 0000FEH
A/D activation compare 1: 0001FEH

Buffer control register

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
-	BTS2	BTS1	BTS0	-	BUFX2	BUFX1	BUFX0	Initial value -000-111 _B

ADTGBUF0
ADTGBUF1*

Address - R/W R/W R/W - R/W R/W R/W

A/D activation compare 0: 0000FDH
A/D activation compare 1: 0001FDH

R/W: Readable/writable
R: Read only
W: Write only

* Not available in MB91470 series.

12.4.1 Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5) / Compare Clear Register (CPCLRH0 to CPCLRH5, CPCLRL0 to CPCLRL5)

The compare clear buffer register (CPCLRBH, CPCLRBL) is a 16-bit buffer register which exists in the compare clear register (CPCLRH, CPCLRL). Both the register (CPCLRBH, CPCLRBL) and the register (CPCLRH, CPCLRL) exist in the same address.

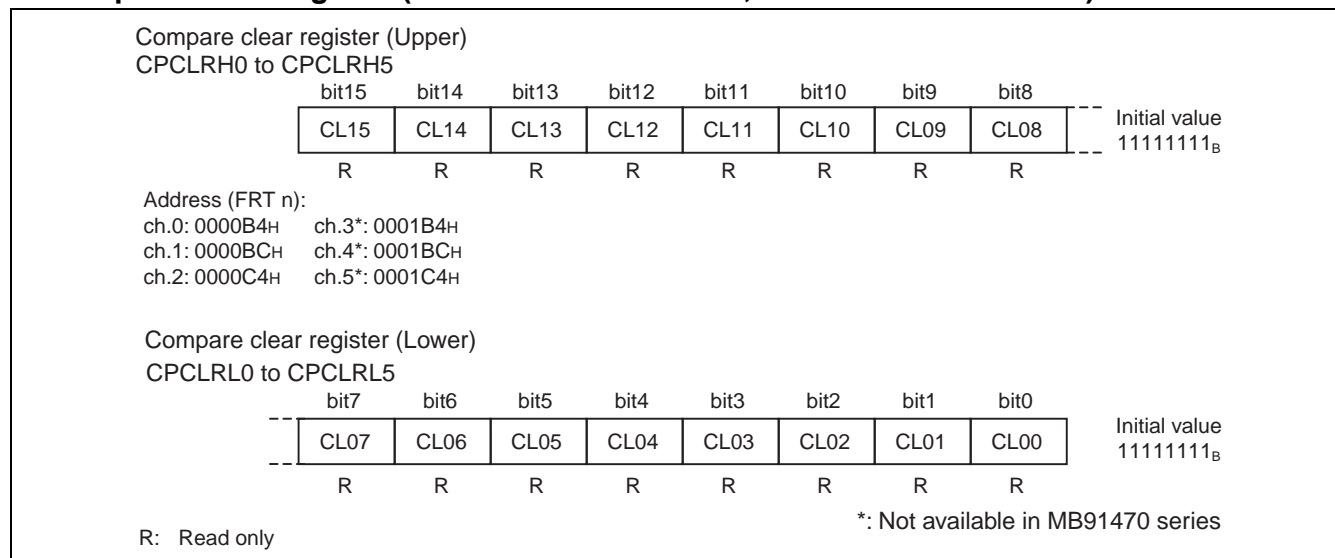
■ Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5)

Compare clear buffer register (Upper)									
CPCLRBH0 to CPCLRBH5	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	Initial value 11111111 _B
Address (ch.n):	W	W	W	W	W	W	W	W	
ch.0: 0000B4H	ch.3*: 0001B4H								
ch.1: 0000BCH	ch.4*: 0001BCH								
ch.2: 0000C4H	ch.5*: 0001C4H								
Compare clear buffer register (Lower)									
CPCLRBL0 to CPCLRBL5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	Initial value 11111111 _B
	W	W	W	W	W	W	W	W	
W: Write only					*: Not available in MB91470 series				

The compare clear buffer register is a buffer register which exists in the same address of the compare clear register (CPCLRH, CPCLRL). When the buffer function is disabled (the timer state control register lower (TCCSL), BFE: bit7 = 0), or the free-run timer is stopped, the value of the compare clear buffer register is transferred to the compare clear register immediately. If the buffer function is enabled, the value is transferred to the compare clear register when the count value 0 of the 16-bit free-run timer is detected.

To access this register, use a half-word or word access instruction. Do not access this register with the read modify write (RMW) instructions.

■ Compare Clear Register (CPCLRH0 to CPCLRH5, CPCLRL0 to CPCLRL5)



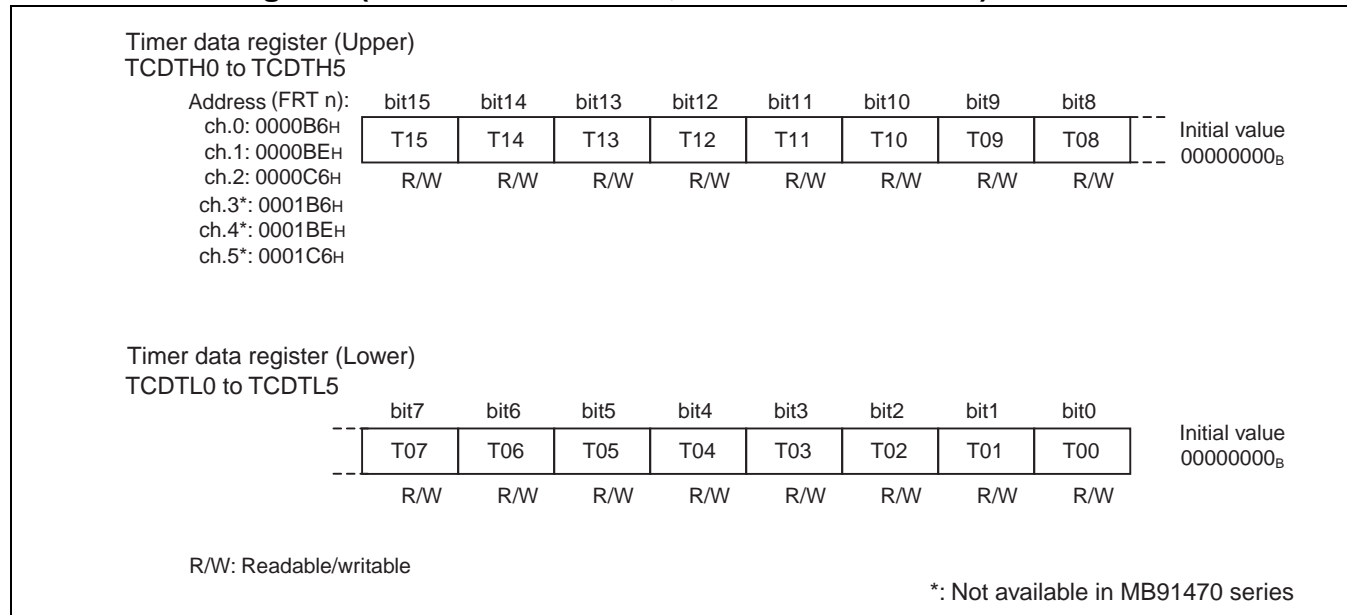
The compare clear register is used to compare with the count value of the 16-bit free-run timer. In up-count mode, the 16-bit free-run timer is reset to "0000_H" when the 16-bit free-run timer count value matches the value in this register. In up/down count mode, the 16-bit free-run timer changes its mode from the up counting to the down counting when this register matches the count value of the 16-bit free-run timer; or changes from the down counting to up counting when a zero-detection occurs.

To access this register, use a half-word or word access instruction. Do not access this register with the read modify write (RMW) instructions.

12.4.2 Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)

The timer data register (TCDTH, TCDTL) is used to read the count value of the 16-bit free-run timer. Also the count value of the 16-bit free-run timer can be set to this register.

■ Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)



The timer data register is used to read the count value of the 16-bit free-run timer. The count value is cleared to "0000_H" immediately when a reset occurs. Writing the value to this register can be set the timer value. Please write a value while the timer is stopped (the timer state control register lower (TCCSL), STOP: bit6 = 1). To access the timer data register, use a half-word or word access instruction.

16-bit free-run timer is initialized immediately when the following factors occur.

- Reset
- The clear bit (SCLR: bit 4) in the timer state control register lower (TCCSL)=1 while the 16-bit free-run timer is operating (STOP: bit 6 in the timer state control register lower (TCCSL)=0)

(Note) The 16-bit free-run timer is not initialized as the clear bit (SCLR: bit 4) in the timer state control register lower (TCCSL)=1 while the 16-bit free-run timer is stopped (STOP: bit6 in the timer state control register lower (TCCSL)=1)

- Match of the compare clear register and the timer count value in the up count mode (the timer state control register lower (TCCSL), MODE: bit5 = 0)

MB91470/480 Series

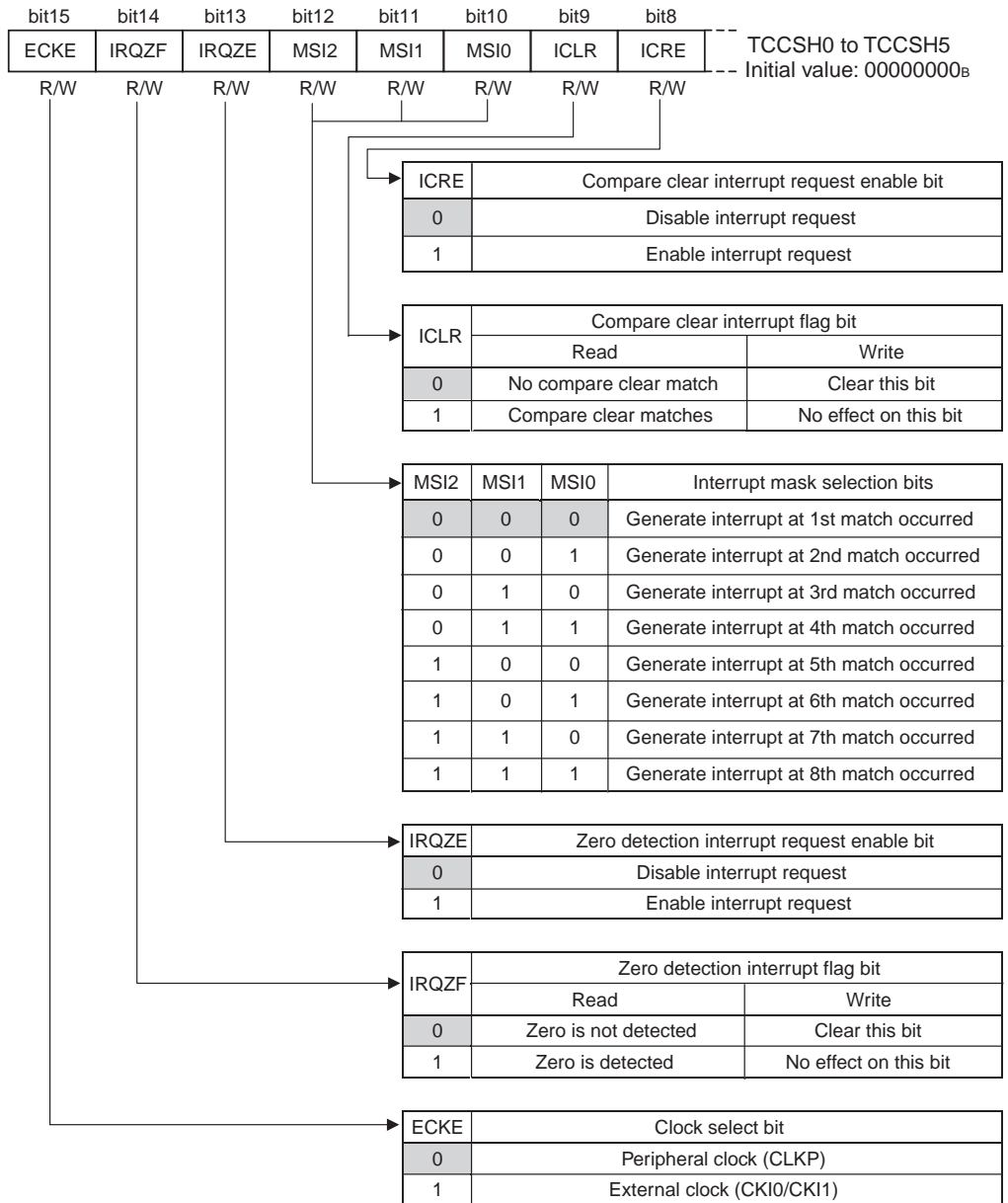
12.4.3 Timer State Control Register (TCCSH0 to TCCSH5, TCCSL0 to TCCSL5, TCCSM0 to TCCSM5)

The timer state control register (TCCSH, TCCSL and TCCSM) is a 16-bit and 8-bit registers which are used to control the operation of the 16-bit free-run timer.

■ Timer State Control Register Upper (TCCSH0 to TCCSH5)

Timer state control register (Upper)

Address:
ch.0: 0000B8H
ch.1: 0000C0H
ch.2: 0000C8H
ch.3*: 0001B8H
ch.4*: 0001C0H
ch.5*: 0001C8H



R/W: Readable/Writable

: Initial value

*: They are not available in MB91470 series.

Table 12.4-1 Timer State Control Register Upper (TCCSH) (1 / 2)

Bit name		Function
bit15	ECKE: Clock select bit	<ul style="list-style-type: none"> This bit is used to select the peripheral clock (CLKP) or the external clock (CKI0/CKI1) as a count clock of the 16-bit free-run timer. Setting this bit to "0" selects the peripheral clock (CLKP). To select a count clock frequency, you also need to select a clock frequency select bits of the TCCSL register (CLK3 to CLK0: bit3 to bit0). Setting this bit to "1" selects the external clock (CKI0/CKI1). The external clock (CKI0/CKI1) is input from CKI pins. Therefore, write "0" to the bit4 of the port direction register (DDRP, DDRR) to enable the external clock input. <p>Note: The count clock is changed immediately when this bit is set. Therefore, this bit must be changed when the output compare and the input capture are stopped.</p>
bit14	IRQZF: Zero-detection interrupt flag bit	<ul style="list-style-type: none"> When the count value of the 16-bit free-run timer is 0000_H, this bit is set to "1". When this bit is set to "0": Clears the bit. Setting this bit to "1" has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read. <p>Note: This bit is not set by a software clear (writing "1" to bit4 (SCLR) of the timer state control register lower (TCCSL)) when the 16-bit free-run timer is operating (bit6 (STOP) of the timer state control register lower (TCCSL)=0). In up/down count mode (bit5 (MODE) in the timer state control register lower (TCCSL) = 1), this bit is set to "1" when an interrupt set in the interrupt mask selection bits (bit12 to bit10 (MSI2 to MSI0) in the timer state control register upper (TCCSH) \neq 000_B) occurs. When no interrupt occurs, this bit is not set to "1". In the up count mode (MODE: bit5 = 0), this bit is set every time the zero-detection occurs regardless of the value of the MSI2 to MSI0: bit12 to bit10.</p>
bit13	IRQZE: Zero detection interrupt request enable bit	When this bit and the interrupt flag bit (IRQZF: bit14) are set to "1", an interrupt request to the CPU can be generated.

Table 12.4-1 Timer State Control Register Upper (TCCSH) (2 / 2)

Bit name		Function
bit12 to bit10	MSI2 to MSI0: Interrupt mask selection bits	<p>When MODE2:bit11 of timer state control register M=0</p> <ul style="list-style-type: none"> These bits are used to set the mask counting of the compare clear interrupt at up-count mode (MODE:bit5 of the timer state control register lower (TCCSL)=0). These bits are used to set the mask counting of 0 detection interrupt at up/down-count mode (MODE:bit5 of the timer state control register lower (TCCSL)=1). When 0 is set to this bit, the interrupt factor is not masked. <p>When MODE2:bit11 of the timer state control register M=1</p> <ul style="list-style-type: none"> These bits are used to set the mask counting of "0" detection interrupt at up/down-count mode (MODE:bit5 of the timer state control register lower (TCCSL)=1). The setting of up-count mode (MODE:bit5 of the timer state control register lower (TCCSL)=0) is prohibited. <p>Note: Reading returns the value of the mask counter. For read-modify-write instruction, reading returns the value of the mask register. Write data at writing to the mask register. When free-run timer is running (STOP:bit6 of the timer state control register lower (TCCSL)=0), writing value to the mask register will be reloaded to the counter after the mask counter reached to zero. When free-run timer is stopped (STOP:bit6 of the timer control register lower (TCCSL)=1), writing value to the mask register will reload to the counter immediately.</p>
bit9	ICLR: Compare clear interrupt flag bit	<ul style="list-style-type: none"> This bit is set to "1" when the value of the compare clear matches the value of the 16-bit free-run timer. When this bit is set to "0": Clears the bit. Setting this bit to "1" has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read. <p>Note: In up-count mode (bit5 (MODE) in the timer state control register lower (TCCSL) = 0), this bit is set to "1" when an interrupt set in the interrupt mask selection bits occurs. When no interrupt occurs, this bit is not set to "1". In the up/down count mode (MODE:bit5 of the timer state control register lower (TCCSL)=1), this bit is set every time a compare clear occurs regardless of the value of the MSI2 to MSI0.</p>
bit8	ICRE: Compare clear interrupt request enable bit	<p>An interrupt request to the CPU can be generated when this bit and the compare clear interrupt flag bit (ICLR: bit9) are set to "1".</p>

■ Timer State Control Register Lower (TCCSL0 to TCCSL5)

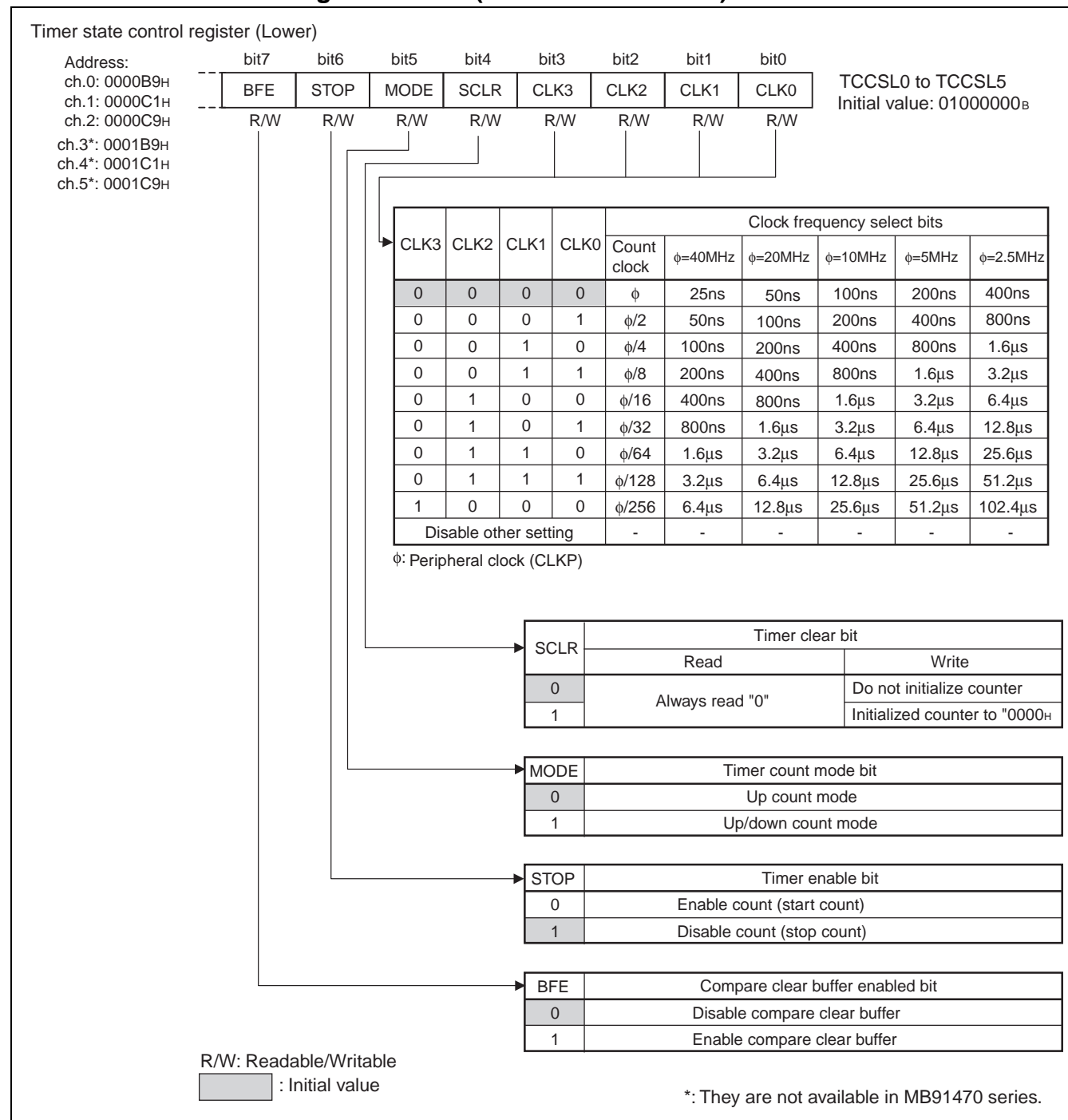


Table 12.4-2 Timer State Control Register Lower (TCCSL)

Bit name		Function
bit7	BFE: Compare clear buffer enable bit	<ul style="list-style-type: none"> This bit is used to enable a compare clear buffer register (CPCLRBH, CPCLRBL). Setting this bit to "0" disables the compare clear buffer register (CPCLRBH, CPCLRBL). Accordingly, you can write directly to the compare clear registers (CPCLRH and CPCLRL). Setting this bit to "1" enables the compare clear buffer register (CPCLRBH, CPCLRBL). The data written to and stored in the compare clear buffer register (CPCLRBH, CPCLRBL) is transferred to the compare clear register when a count value of "0" is detected in the 16-bit free-run timer.
bit6	STOP: Timer enable bit	<ul style="list-style-type: none"> This bit is used to start/stop the 16-bit free-run timer counting. Setting this bit to "0" starts the 16-bit free-run timer counting. Setting this bit to "1" stops the 16-bit free-run timer counting. Even if SCLR: bit4 of timer state control register lower (TCCSL) is "1" when the free-run timer is stopped (this bit=1), the free-run timer cannot be initialized.
bit5	MODE: Timer count mode bit	<ul style="list-style-type: none"> This bit is used to select a count mode of the 16-bit free-run timer. Selecting this bit to "0" selects the up-count mode. The timer continues to perform incremental counting until the count value matches a compare clear register and is reset "0000_H". Then, the timer restarts to perform incremental counting. Setting this bit to "1" selects the up/down count mode. The timer continues to perform incremental counting until the count value matches a compare clear register. Then, the mode changes to the down count. After that, the mode changes to the up count again when the count value reaches to "0000_H". This bit can be written even if the timer is operating or stopped. When the timer is running, the value written to this bit is stored in a buffer and the count mode changes based on the buffer value the next time the timer value goes to "0000_H".
bit4	SCLR: Timer clear bit	<ul style="list-style-type: none"> This bit is used to initialize the 16-bit free-run timer to "0000_H". Initialize the 16-bit free-run timer: When this bit is set to "1" while the 16-bit free-run timer is running (STOP: bit6 of the timer state control register lower (TCCSL)=0), the timer is initialized to "0000_H" at the next count clock. When the bit is set to "1" while the timer is stopped (STOP: bit6 of the timer state control register lower (TCCSL)=1), the timer is not initialized. The read value is always "0". <p>Note: Writing "1" to this bit does not generate a zero-detection interrupt. No timer clear is performed if you set "1" and then write "0" before the next count clock.</p>
bit3 to bit0	CLK3 to CLK0: Clock frequency select bits	<ul style="list-style-type: none"> These bits are used to select a count clock frequency of the 16-bit free-run timer. The count clock is changed immediately after the bits are set.

■ **Timer State Control Register M (TCCSM0 to TCCSM5)**

Timer state control register M

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	TCCSM0 to TCCSM5	
-	-	-	-	MODE2	MSI5	MSI4	MSI3	Address:	
-	-	-	-	R/W	R/W	R/W	R/W	ch.0: 0000BAH ch.3*2: 0001BAH	
								ch.1: 0000C2H ch.4*2: 0001C2H	
								ch.2: 0000CAH ch.5*2: 0001CAH	
								Initial value: ----0000B	

MSI5	MSI4	MSI3	Compare clear interrupt mask selection bits
0	0	0	Generate interrupt at 1st match occurred
0	0	1	Generate interrupt at 2nd match occurred
0	1	0	Generate interrupt at 3rd match occurred
0	1	1	Generate interrupt at 4th match occurred
1	0	0	Generate interrupt at 5th match occurred
1	0	1	Generate interrupt at 6th match occurred
1	1	0	Generate interrupt at 7th match occurred
1	1	1	Generate interrupt at 8th match occurred

MODE2	MODE*1	Interrupt mask mode bit 2
0	0	Setting value of MSI5 to MSI3 are invalid
0	1	Setting value of MSI5 to MSI3 are invalid
1	0	Prohibited Setting (Operation does not guarantee)
1	1	Setting value of MSI5 to MSI3 are valid

R/W : Readable/Writable
 : Initial value
 - : Undefined bit

*1: bit5 in the timer state control register lower (TCCSL)
 *2: They are not available in MB91470 series

Table 12.4-3 Timer State Control Register M (TCCSM)

Bit name		Function
bit15 to bit12	Undefined bits	<ul style="list-style-type: none"> The read value of these bits are undefined. Write to these bits takes no effect.
bit11	MODE2: Interrupt mask mode bit 2	<ul style="list-style-type: none"> This bit is used to mask the zero-detection interrupt and compare clear interrupt separately when the 16-bit free-run timer is in up/down count mode (MODE: bit5 of the timer state control register lower (TCCSL)=1). If the bit is set to "1" when MODE: bit5 of the timer state control register lower (TCCSL) is 1, the value set is MSI5 to MSI3: bit10 to bit8 of this register is valid and the number of the specified compare clear interrupt is masked. The number of masking of zero-detection interrupt is the value set in MSI2 to MSI0: bit12 to bit10 of the timer state control register upper (TCCSH). <p>Note: If the bit is set to "0" when MODE: bit5 of the timer state control register lower (TCCSL) is 1, the operation is not guaranteed.</p>
bit10 to bit8	MSI5 to MSI3: Compare clear interrupt mask selection bits	<ul style="list-style-type: none"> This bit is only valid when MODE2: bit11 of this register and MODE: bit5 of the timer state control register lower (TCCSL) are 1. It is used to set the number of masking of the compare clear interrupt. The number of masking of the zero-detection interrupt is set by MSI2 to MSI0: bit12 to bit10 of the timer state control register upper (TCCSH). Setting the bits to "000_B" do not mask the compare clear interrupt cause. <p>Note: Reading returns the value of the mask counter. For read-modify-write instruction, reading returns the value of the mask register. Write data at writing to the mask register. When free-run timer is running (STOP:bit6 of the timer state control register lower (TCCSL)=0), writing value to the mask register will be reloaded to the counter after the mask counter reached to "0". When free-run timer is stopped (STOP:bit6 of the timer state control register lower (TCCSL)=1), writing value to the mask register will reload to the counter immediately.</p>

12.4.4 A/D Trigger Control Register (ADTRGC0 to ADTRGC5)

Controls A/D trigger signal output when a free-run timer compare match or zero-detection occurs.

■ A/D Trigger Control Register (ADTRGC0 to ADTRGC5)

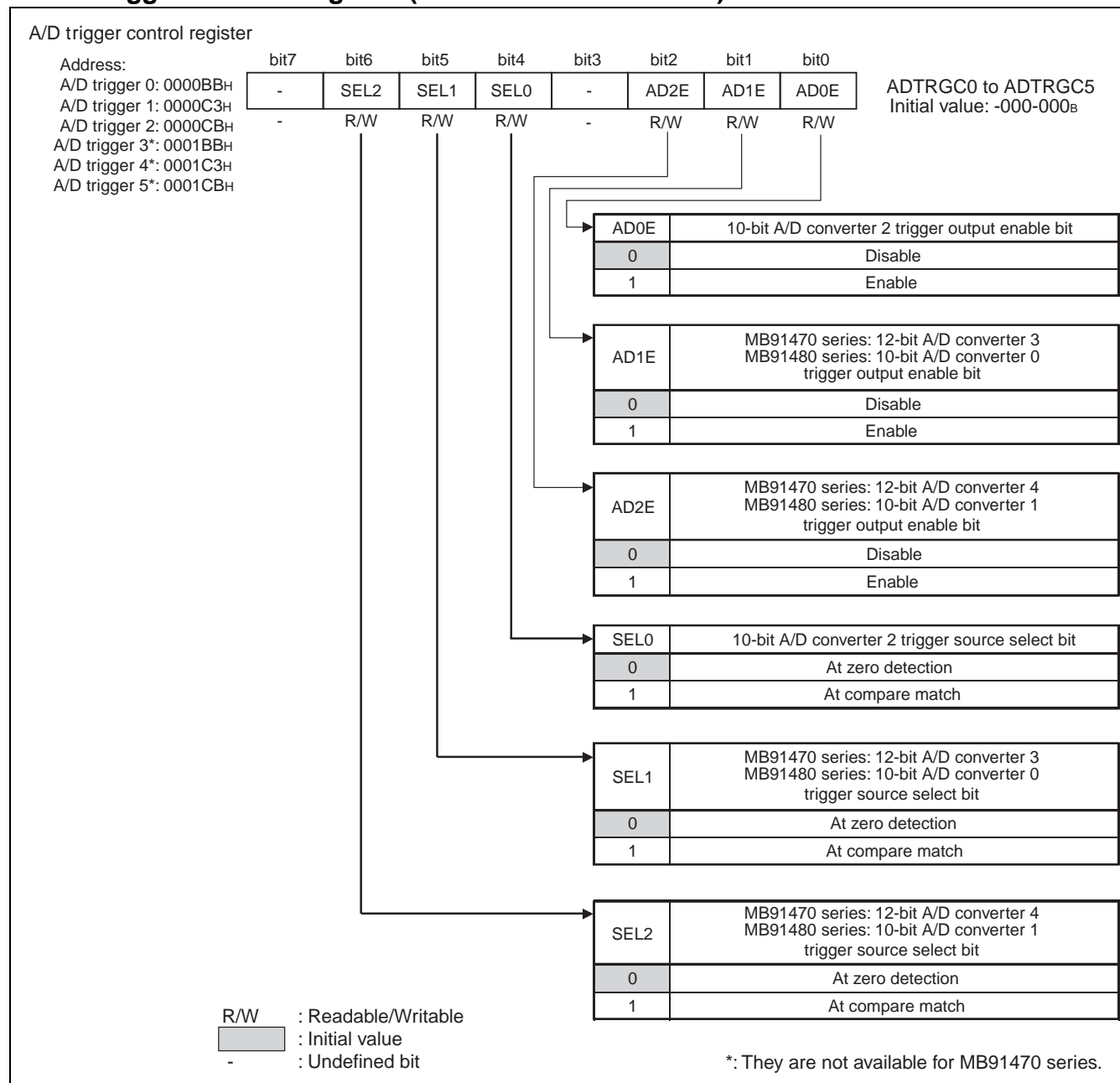


Table 12.4-4 A/D Trigger Control Register (ADTRGC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit6	SEL2: MB91470 series: 12-bit A/D converter 4 MB91480 series: 10-bit A/D converter 1 trigger source select bit	This bit is the bit that selects whether the MB91470 series: 12-bit A/D converter 4, MB91480 series: 10-bit A/D converter 1 trigger is output at a zero detection of the free-run timer or a compare match.
bit5	SEL1: MB91470 series: 12-bit A/D converter 3 MB91480 series: 10-bit A/D converter 0 trigger source select bit	This bit is the bit that selects whether the MB91470 series: 12-bit A/D converter 3, MB91480 series: 10-bit A/D converter 0 trigger is output at a zero detection of the free-run timer or a compare match.
bit4	SEL0: 10-bit A/D converter 2 trigger source select bit	This bit is the bit that selects whether the 10-bit A/D converter 2 trigger is output at a zero detection of the free-run timer or a compare match.
bit3	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit2	AD2E: MB91470 series: 12-bit A/D converter 4 MB91480 series: 10-bit A/D converter 1 trigger output enable bit	<ul style="list-style-type: none"> When this bit is set to "0", the MB91470 series: 12-bit A/D converter 4, MB91480 series: 10-bit A/D converter 1 trigger signal is not output. When this bit is set to "1", the MB91470 series: 12-bit A/D converter 4, MB91480 series: 10-bit A/D converter 1 trigger signal can be output.
bit1	AD1E: MB91470 series: 12-bit A/D converter 3 MB91480 series: 10-bit A/D converter 0 trigger output enable bit	<ul style="list-style-type: none"> When this bit is set to "0", the MB91470 series: 12-bit A/D converter 3, MB91480 series: 10-bit A/D converter 0 trigger signal is not output. When this bit is set to "1", the MB91470 series: 12-bit A/D converter 3, MB91480 series: 10-bit A/D converter 0 trigger signal can be output.
bit0	AD0E: 10-bit A/D converter 2 trigger output enable bit	<ul style="list-style-type: none"> When this bit is set to "0", the 10-bit A/D converter 2 trigger signal is not output. When this bit is set to "1", the 10-bit A/D converter 2 trigger signal can be output.

12.4.5 Free-run Timer Selection Register (FRS0 to FRS9)

The free-run timer selection register is used to select one of the free-run timers with 6 channels for input capture and output compare.

Free-run Timer Selection Register (Upper) for Output Compare (FRS1, FRS6)

Free-run timer selection register (Upper) for output compare

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
-	FSO14	FSO13	FSO12	-	FSO10	FSO9	FSO8	FRS1/FRS6*1
-	R/W	R/W	R/W	-	R/W	R/W	R/W	Address:
								FRS1: 0000CE _H FRS6*1: 0001CE _H
								Initial value:
								FRS1 : -000-000 _B
								FRS6*1: -011-011 _B

FSO10	FSO9	FSO8	Free-run timer selection bits for output compare	
0	0	0	FRT0	→ OC2 / OC8
0	0	1	FRT1	→ OC2 / OC8
0	1	0	FRT2	→ OC2 / OC8
0	1	1	FRT3	→ OC2 / OC8*2
1	0	0	FRT4	→ OC2 / OC8*2
1	0	1	FRT5	→ OC2 / OC8*2
Other settings			Prohibited (Operation not guarantee)	

FSO14	FSO13	FSO12	Free-run timer selection bits for output compare	
0	0	0	FRT0	→ OC3 / OC9
0	0	1	FRT1	→ OC3 / OC9
0	1	0	FRT2	→ OC3 / OC9
0	1	1	FRT3	→ OC3 / OC9*2
1	0	0	FRT4	→ OC3 / OC9*2
1	0	1	FRT5	→ OC3 / OC9*2
Other settings			Prohibited (Operation not guarantee)	

*1: FRS6 is not available in MB91470 series.

*2: FRT3 to FRT5 do not exist in MB91470 series.

R/W : Readable/Writable
 : Initial value
 - : Undefined bit

Table 12.4-5 Free-run Timer Selection Register (Upper) for Output Compare (FRS1, FRS6)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit14 to bit12	FSO14 to FSO12: free-run timer selection bits for output compare	These bits are used to select one of the free-run timers for output compare 3/9. Note: Be sure to check that the free-run timer is stopped before using these bits.
bit11	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit10 to bit8	FSO10 to FSO8: free-run timer selection bits for output compare	These bits are used to select one of the free-run timers for output compare 2/8. Note: Be sure to check that the free-run timer is stopped before using these bits.

Free-run Timer Selection Register (Lower) for Output Compare (FRS0, FRS2, FRS5, FRS7)

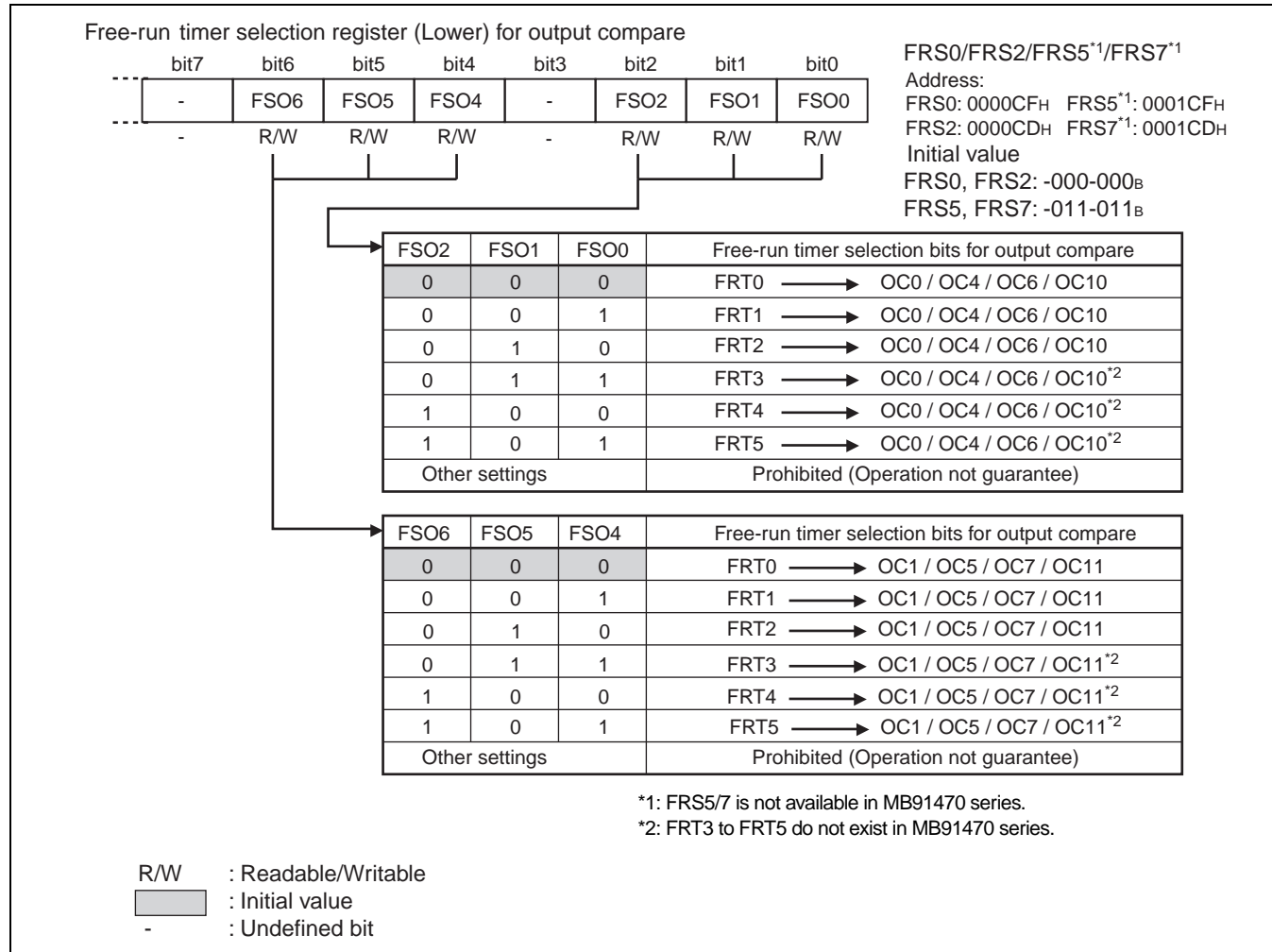


Table 12.4-6 Free-run Timer Selection Register (Lower) for Output Compare (FRS0, FRS2, FRS5, FRS7)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit6 to bit4	FSO6 to FSO4: free-run timer selection bits for output compare	These bits are used to select one of the free-run timers for output compare 1/5/7/11. Note: Be sure to check that the free-run timer is stopped before using these bits.
bit3	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has have no effect on operation.
bit2 to bit0	FSO2 to FSO0: free-run timer selection bits for output compare	These bits are used to select one of the free-run timers for output compare 0/4/6/10. Note: Be sure to check that the free-run timer is stopped before using these bits.

■ Free-run Timer Selection Register (Upper) for Input Capture (FRS4, FRS9)

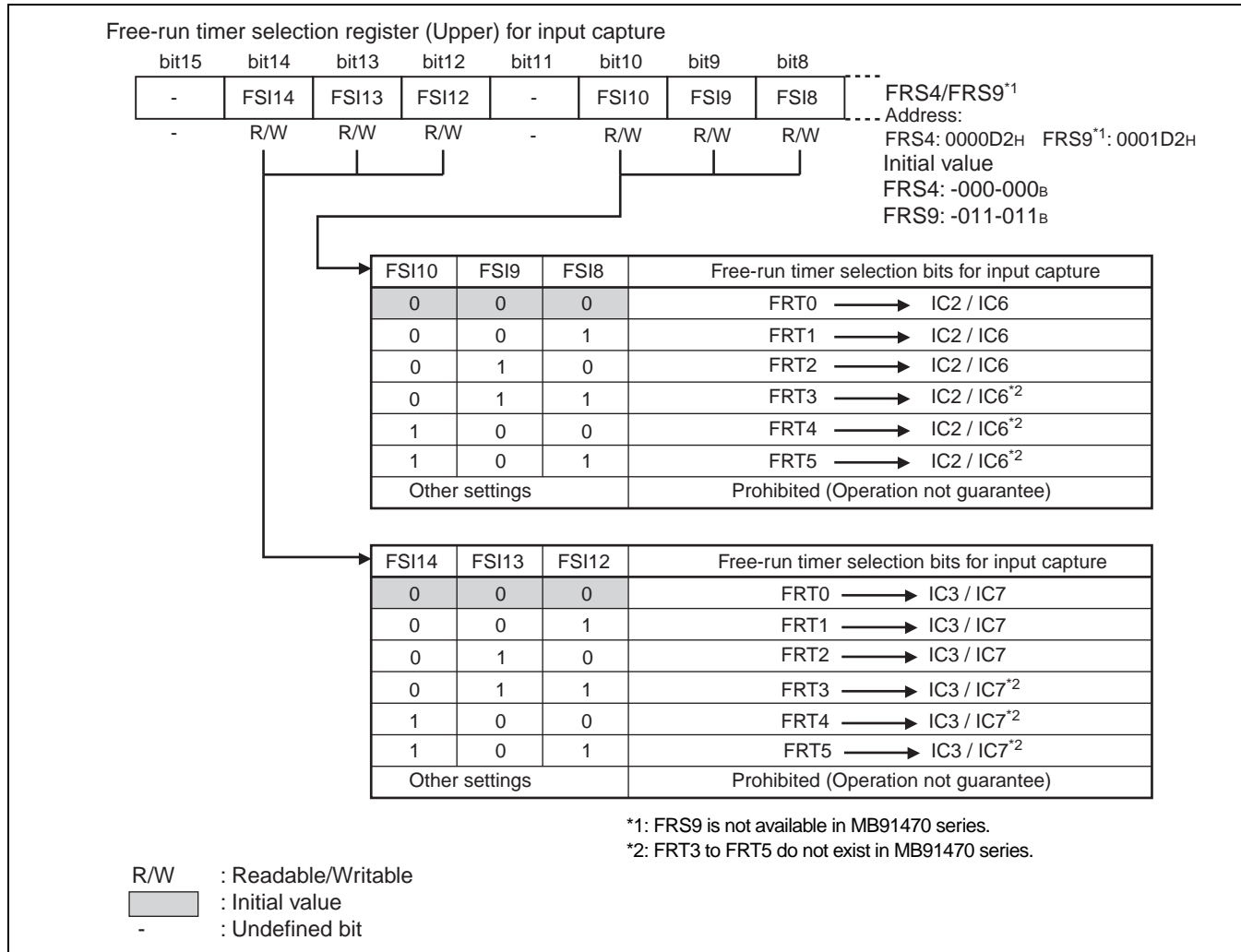


Table 12.4-7 Free-run Timer Selection Register (Upper) for Input Capture (FRS4, FRS9)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit14 to bit12	FSI14 to FSI12: free-run timer selection bits for input capture	These bits are used to select one of the free-run timers for input capture3/7. Note: Be sure to check that the free-run timer is stopped before using these bits.
bit11	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has have no effect on operation.
bit10 to bit8	FSI10 to FSI8: free-run timer selection bits for input capture	These bits are used to select one of the free-run timers for input capture2/6. Note: Be sure to check that the free-run timer is stopped before using these bits.

Free-run Timer Selection Register (Lower) for Input Capture(FRS3, FRS8)

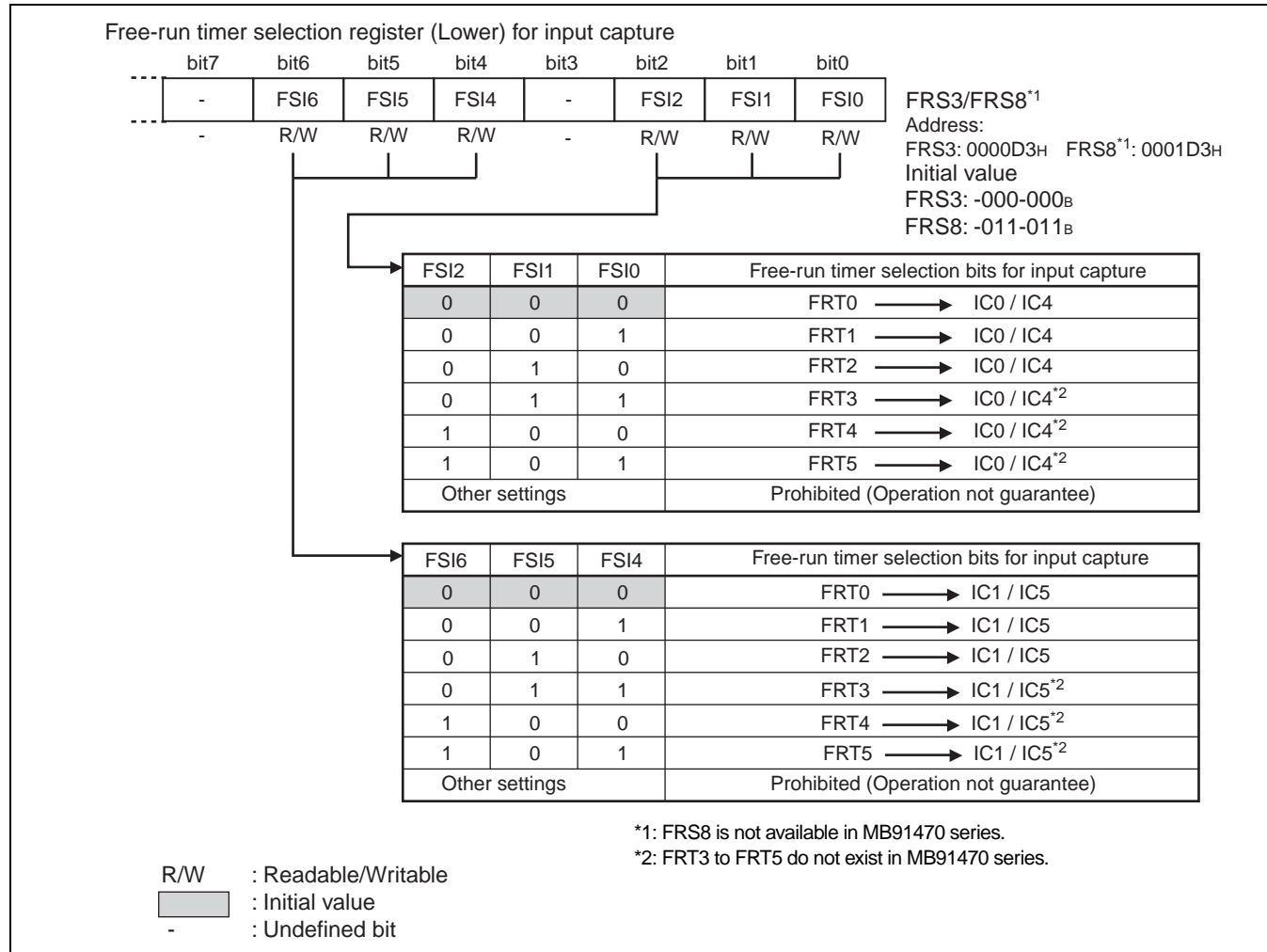


Table 12.4-8 Free-run Timer Selection Register (Lower) for Input Capture (FRS3, FRS8)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit6 to bit4	FSI6 to FSI4: free-run timer selection bits for input capture	These bits are used to select one of the free-run timers for input capture1/5. Note: Be sure to check that the free-run timer is stopped before using these bits.
bit3	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit2 to bit0	FSI2 to FSI0: free-run timer selection bits for input capture	These bits are used to select one of the free-run timers for input capture0/4. Note: Be sure to check that the free-run timer is stopped before using these bits.

MB91470/480 Series

12.4.6 Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11) /Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11)

The output compare buffer register (OCCPBH, OCCPBL) is a 16-bit buffer register for the output compare register (OCCPH, OCCPL).

Both the register (OCCPBH, OCCPBL) and the register (OCCPH, OCCPL) exist in the same address.

■ Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11)

Output compare buffer register (Upper)										
OCCPBH0 to OCCPBH5		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value 00000000 _B
OCCPBH6* to OCCPBH11*		OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	
Address		W	W	W	W	W	W	W	W	
ch.0: 0000A0H	ch.6: 0001A0H									
ch.1: 0000A2H	ch.7: 0001A2H									
ch.2: 0000A4H	ch.8: 0001A4H									
ch.3: 0000A6H	ch.9: 0001A6H									
ch.4: 0000A8H	ch.10: 00001A8H									
ch.5: 0000AAH	ch.11: 00001AAH									

Output compare buffer register (Lower)										
OCCPBL0 to OCCPBL5		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value 00000000 _B
OCCPBL6* to OCCPBL11*		OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	
		W	W	W	W	W	W	W	W	

W: Write only

*: Not available in MB91470 series.

W: Write only

*: Not available in MB91470 series.

The output compare buffer register is a buffer register for the output compare register (OCCPH, OCCPL). When the buffer function is disabled (the lower of the compare control register (OCSL0, OCSL2, OCSL4), BUF1, BUF0: bit3, bit2 = 11_B), or the free-run timer is stopped, the value of the output compare buffer register is transferred to the output compare register immediately. When the buffer function is enabled (the lower of the compare control register (OCSL0, OCSL2, OCSL4), BUF1, BUF0: bit3, bit2=00_B), the value is transferred when the compare clear match or zero-detection occurs according to the transfer selection bits (BTS1, BTS0: bit14, bit13) in the upper of the compare control register (OCSH1, OCSH3, OCSH5).

To access this register, use a half-word or word access instruction.

The free-run timer in the above explanation refers to the operation of the free-run timer selected for output compare. Do not access this register with the read modify write (RMW) instructions.

■ Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11)

Output compare register (Upper)		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value 00000000 _B
OCCPH0 to OCCPH5	OCCPH6* to OCCPH11*	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	
Address		R	R	R	R	R	R	R	R	
ch.0: 0000A0H	ch.6: 0001A0H									
ch.1: 0000A2H	ch.7: 0001A2H									
ch.2: 0000A4H	ch.8: 0001A4H									
ch.3: 0000A6H	ch.9: 0001A6H									
ch.4: 0000A8H	ch.10: 0001A8H									
ch.5: 0000AAH	ch.11: 0001AAH									
Output compare register (Lower)		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value 00000000 _B
OCCPL0 to OCCPL5	OCCPL6* to OCCPL11*	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	
		R	R	R	R	R	R	R	R	

R: Read only

*: Not available in MB91470 series.

The output compare register is a 16-bit register used to compare with the count value of the 16-bit free-run timer. Set the value of the output compare buffer register (OCCPBH, OCCPBL) before the timer operation is enabled.

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 (the lower of the compare control register (OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/OCSL10), IOP1, IOP0: bit7, bit6) is set. When the output level is set (the upper of the compare control register (OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/OCSH11), OTD1, OTD0: bit9, bit8), an output level waveform generator (RTO0 to RTO5) corresponding to the output compare register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11) can be reversed.

The compare signal is not generated when the value of this register matches the peak value when the 16-bit free-run timer is in up/down mode.

● Up/down mode

- In CMOD = 0

When this register is set to "FFFF_H", the RT output goes to "0" regardless of the 16-bit free-run timer value and inversion mode. The output goes to "1" when "0000_H" is set.

- In CMOD = 1

When this register is set to "FFFF_H", the RT output goes to "1" regardless of the 16-bit free-run timer value and inversion mode. The output goes to "0" when "0000_H" is set.

To access this register, use a half-word or word access instruction. Do not access this register with the read modify write (RMW) instructions.

The free-run timer in the above explanation refers to the operation of the free-run timer selected for output compare.

MB91470/480 Series**12.4.7 Compare Control Register (OCSH0 to OCSH11, OCSL0 to OCSL11)**

The compare control register is used to control the output level, output enable, output level reverse mode, compare operation enable, compare match interrupt enable, and compare match interrupt flag of RT0/RT6 to RT5/RT11.

■ **Compare Control Register, Upper Byte (OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/OCSH11)**

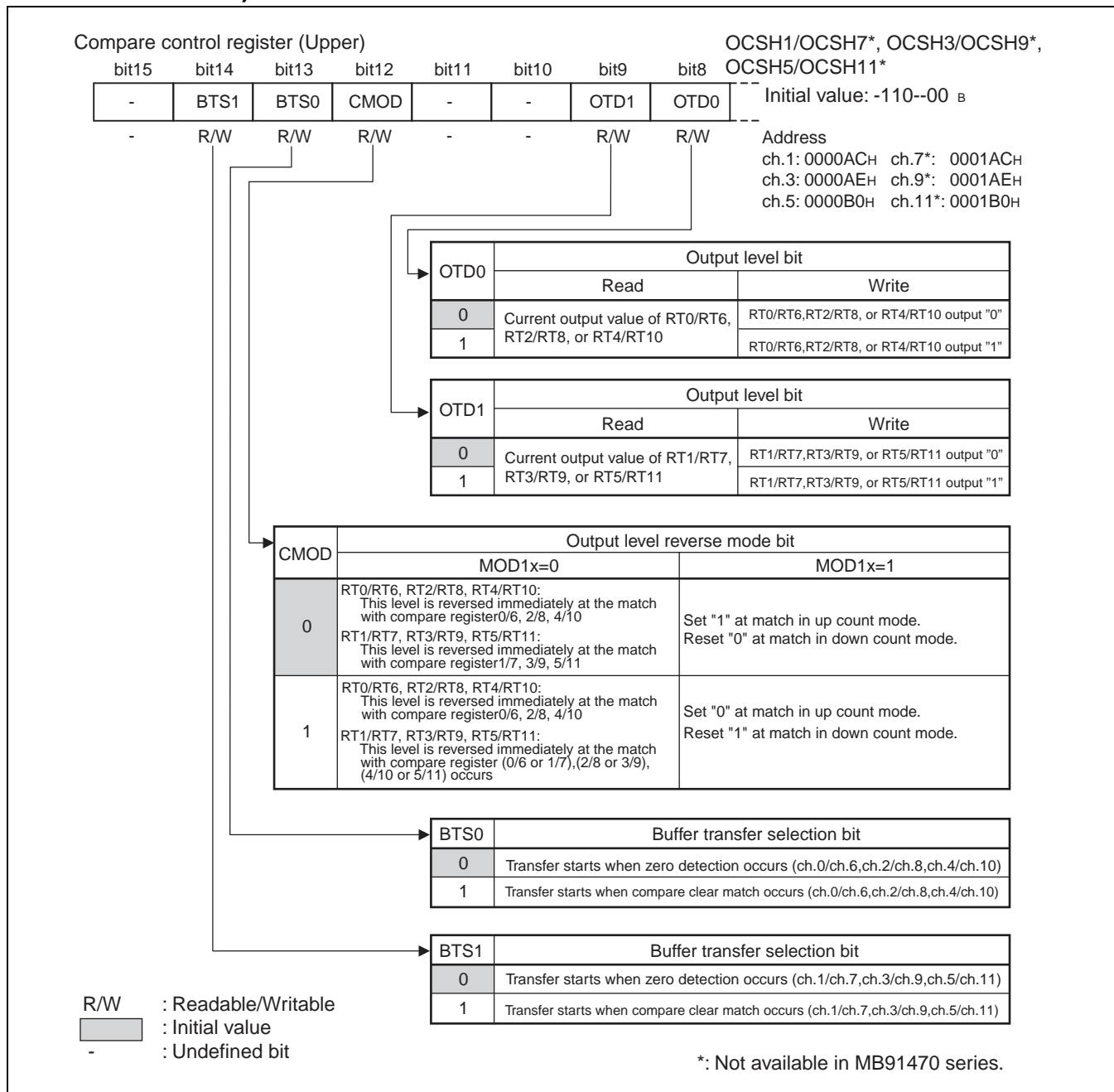


Table 12.4-9 Compare Control Register, Upper Byte (OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/OCSH11) (1 / 2)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit14	BTS1: Buffer transfer selection bit	<ul style="list-style-type: none"> This bit is used to select the time when the data is transferred from the output compare buffer registers (OCCPBH1/OCCPBH7, OCCPBH3/OCCPBH9, OCCPBH5/OCCPBH11, OCCPBL1/OCCPBL7, OCCPBL3/OCCPBL9, OCCPBL5/OCCPBL11) to the output compare registers (OCCPH1/OCCPH7, OCCPH3/OCCPH9, OCCPH5/OCCPH11, OCCPL1/OCCPL7, OCCPL3/OCCPL9, OCCPL5/OCCPL11). Setting this bit to "0" starts the data transfer when the count value "0" of the 16-bit free-run timer is detected. Setting this bit to "1" starts the data transfer when a compare clear match of the 16-bit free-run timer occurs.
bit13	BTS0: Buffer transfer selection bit	<ul style="list-style-type: none"> This bit is used to select the time when the data is transferred from the output compare buffer registers (OCCPBH0/OCCPBH6, OCCPBH2/OCCPBH8, OCCPBH4/OCCPBH10, OCCPBL0/OCCPBL6, OCCPBL2/OCCPBL8, OCCPBL4/OCCPBL10) to the output compare registers (OCCPH0/OCCPH6, OCCPH2/OCCPH8, OCCPH4/OCCPH10, OCCPL0/OCCPL6, OCCPL2/OCCPL8, OCCPL4/OCCPL10). Setting this bit to "0" starts the data transfer when the count value "0" of the 16-bit free-run timer is detected. Setting this bit to "1" starts the data transfer when a compare clear match of the 16-bit free-run timer occurs.

Table 12.4-9 Compare Control Register, Upper Byte (OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/OCSH11) (2 / 2)

Bit name		Function
bit12	CMOD: Output level reverse mode bit	<ul style="list-style-type: none"> This bit is used to change the pin output level inversion mode immediately after a match occurs. When this bit is set to "0": <p>The compare mode control register (OCMOD): MOD1x = 0</p> <ul style="list-style-type: none"> RT0/RT6, RT2/RT8, RT4/RT10: The level is reversed immediately when the compare registers 0/6, 2/8, 4/10 match the 16-bit free-run timer. RT1/RT7, RT3/RT9, RT5/RT11: The level is reversed immediately when the compare registers 1/7, 3/9, 5/11 match the 16-bit free-run timer. <p>The compare mode control register (OCMOD): MOD1x = 1</p> <ul style="list-style-type: none"> Set to "1" when the match occurs in the up-count mode. Reset to "0" when the match occurs in the down-count mode. When this bit is set to "1": <p>The compare mode control register (OCMOD): MOD1x = 0</p> <ul style="list-style-type: none"> RT0/RT6, RT2/RT8, RT4/RT10: The level is reversed immediately when the compare registers 0/6, 2/8, 4/10 match the 16-bit free-run timer. RT1/RT7, RT3/RT9, RT5/RT11: The level is reversed immediately when the compare registers (0/6 or 1/7) (2/8 or 3/9) (4/10 or 5/11) match the 16-bit free-run timer. When the value of the compare register 0/6, 2/8, 4/10 and 1/7, 3/9, 5/11 is the same, the operation is the same operation as one compare register is used. <p>The compare mode control register (OCMOD): MOD1x = 1</p> <ul style="list-style-type: none"> Reset to "0" when the match occurs in the up-count mode. Set to "1" when the match occurs in the down-count mode.
bit11, bit10	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect on operation.
bit9	OTD1: Output level bit	<ul style="list-style-type: none"> This bit is used to change a pin output level of the output compare 1/7, 3/9, 5/11 (RT1/RT7, RT3/RT9, RT5/RT11). The initial value of the compare pin output is "0". Always halt compare operation before writing to the value of this bit. The read value of this bit indicates the output compare value of RT1/RT7, RT3/RT9, RT5/RT11. This bit can be written when CST1: bit1 of the compare control register lower (OCSL) is 0.
bit8	OTD0: Output level bit	<ul style="list-style-type: none"> This bit is used to change a pin output level of the output compare 0/6, 2/8, 4/10 (RT0/RT6, RT2/RT8, RT4/RT10). The initial value of the compare pin output is "0". Always halt compare operation before writing to the value of this bit. The read value of this bit indicates the output compare value of RT0/RT6, RT2/RT8, RT4/RT10. This bit can be written when CST0: bit0 of the compare control register lower (OCSL) is 0.

■ **Compare Control Register, Lower Byte**
(OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/OCSL10)

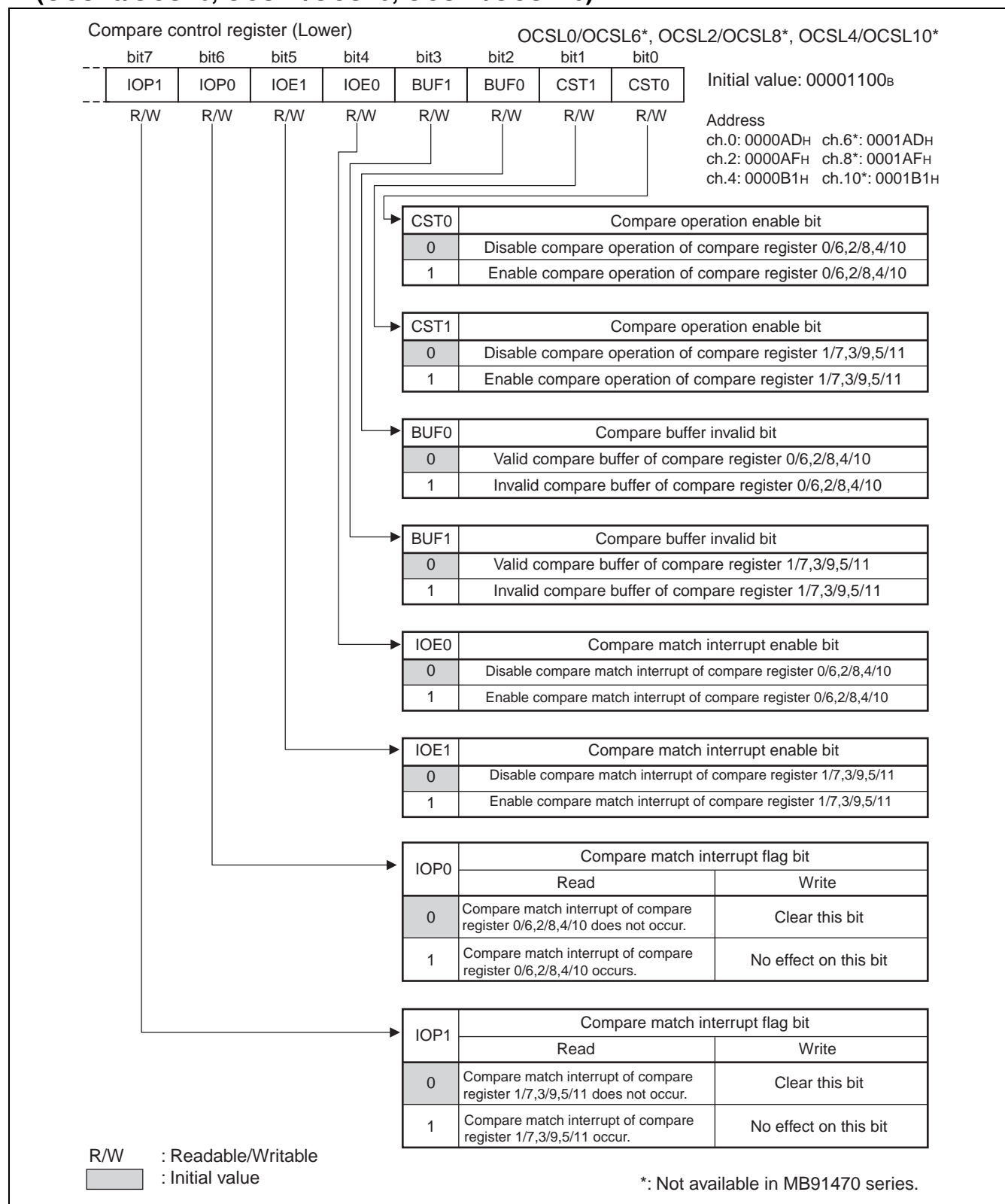


Table 12.4-10 Compare Control Register, Lower Byte (OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/OCSL10)

Bit name		Function
bit7	IOP1: Compare match interrupt flag bit	<ul style="list-style-type: none"> This bit is an interrupt flag which indicates that the compare register 1/7, 3/9, 5/11 matches the value of the 16-bit free-run timer. This bit is set to "1" when the value of the compare register matches the value of the 16-bit free-run timer. When this bit is set while the compare match interrupt enable bit (IOE1: bit5) is enabled, the output compare interrupt occurs. When this bit is set to "0": Clears the bit. Setting this bit to "1" has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit6	IOP0: Compare match interrupt flag bit	<ul style="list-style-type: none"> This bit is an interrupt flag which indicates that the compare register 0/6, 2/8, 4/10 matches the value of the 16-bit free-run timer. This bit is set to "1" when the value of the compare register matches the value of the 16-bit free-run timer. When this bit is set while the compare match interrupt enable bit (IOE0: bit4) is enabled, the output compare interrupt occurs. When this bit is set to "0": Clears the bit. Setting this bit to "1" has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit5	IOE1: Compare match interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable the output compare interrupt of the compare register 1/7, 3/9, 5/11. An output compare interrupt is generated if this bit is "1" when the compare match interrupt flag bit (IOP1:bit7) is set.
bit4	IOE0: Compare match interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable the output compare interrupt of the compare register 0/6, 2/8, 4/10. An output compare interrupt is generated if this bit is "1" when the compare match interrupt flag bit (IOP0:bit6) is set.
bit3	BUF1: Compare buffer disable bit	<ul style="list-style-type: none"> This bit is used to disable the buffer function of the output compare register 1/7, 3/9, 5/11. Setting this bit to "0" enables the buffer function. Setting this bit to "1" disables the buffer function.
bit2	BUF0: Compare buffer disable bit	<ul style="list-style-type: none"> This bit is used to disable the buffer function of the output compare register 0/6, 2/8, 4/10. Setting this bit to "0" enables the buffer function. Setting this bit to "1" disables the buffer function.
bit1	CST1: Compare operation enable bit	<ul style="list-style-type: none"> This bit is used to enable the compare operation between the 16-bit free-run timer and compare register 1/7, 3/9, 5/11. Before enabling compare operation, always write values to compare registers 1/7, 3/9, and 5/11, and to the timer data registers (TCDTH and TCDTL).
bit0	CST0: Compare operation enable bit	<ul style="list-style-type: none"> This bit is used to enable the compare operation between the 16-bit free-run timer and compare register 0/6, 2/8, 4/10. Before enabling compare operation, always write values to compare registers 0/6, 2/8, and 4/10, and to the timer data registers (TCDTH and TCDTL).

12.4.8 Compare Mode Control Register (OCMOD0/OCMOD1)

The compare mode control register controls the mode for inverting the output level when a compare match occurs and whether to set or reset.

■ Compare Mode Control Register (OCMOD0/OCMOD1)

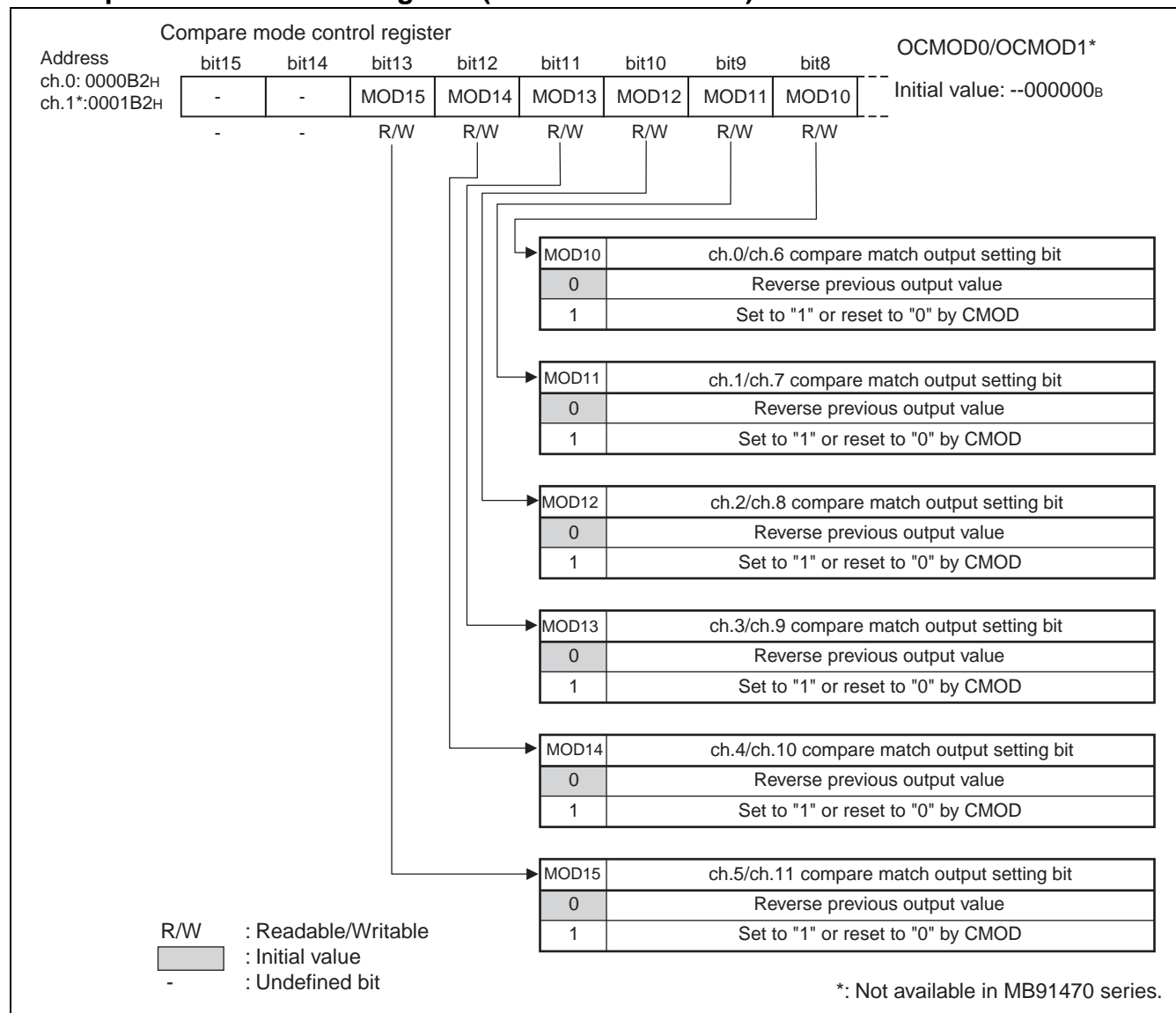


Table 12.4-11 Compare Mode Control Register (OCMOD0/OCMOD1)

Bit name		Function
bit15, bit14	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect the operation.
bit13	MOD15: ch.5/ch.11 compare match mode setting bit	<ul style="list-style-type: none"> These bits specify the operation when the compare match of the output compare output occurs. The initial value is "0". When the bits are set to "0", reverse the output value temporarily when the match occurs. When the bits are set to "1", set the output value to "1" or reset the output value to "0" when the match occurs. CMOD bit of the compare control register (OCSH) sets the set/reset switch. Before data is written, be sure to stop the compare operation. CMOD is set for ch.0/ch.6 and ch.1/ch.7, ch.2/ch.8 and ch.3/ch.9, ch.4/ch.10 and ch.5/ch.11. <ul style="list-style-type: none"> Reset/set is not available to ch.0/ch.6 and ch.1/ch.7 separately. Reset/set is not available to ch.2/ch.8 and ch.3/ch.9 separately. Reset/set is not available to ch.4/ch.10 and ch.5/ch.11 separately.
bit12	MOD14: ch.4/ch.10 compare match mode setting bit	
bit11	MOD13: ch.3/ch.9 compare match mode setting bit	
bit10	MOD12: ch.2/ch.8 compare match mode setting bit	
bit9	MOD11: ch.1/ch.7 compare match mode setting bit	
bit8	MOD10: ch.0/ch.6 compare match mode setting bit	

12.4.9 Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)

The input capture data register is used to store the count value of the free-run timer on detection of a valid edge of the input waveform.

■ Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)

Input capture data register (Upper)									
IPCPH0 to IPCPH3	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value XXXXXXXX _B
IPCPH4* to IPCPH7*	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	
Address	R	R	R	R	R	R	R	R	
ch.0: 0000D4H	ch.4*: 0001D4H								
ch.1: 0000D6H	ch.5*: 0001D6H								
ch.2: 0000D8H	ch.6*: 0001D8H								
ch.3: 0000DAH	ch.7*: 0001DAH								
Input capture data register (Lower)									
IPCPL0 to IPCPL3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value XXXXXXXX _B
IPCPL4* to IPCPL7*	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	
	R	R	R	R	R	R	R	R	
R: Read only									
* Not available in MB91470 series.									

This register is used to store the value of the free-run timer each time the specified edge is detected on the waveform input to the corresponding external pin. (Always use half-word or word access instructions to access this register. Writing data to this register is not permitted.)

The free-run timer in the above explanation refers to the operation of the free-run timer selected for input capture.

MB91470/480 Series

12.4.10 Input Capture State Control/PPG Output Control Register (ICSH23/ICSH67, ICSL23/ICSL67, PICSH01/PICSH45, PICSL01/PICSL45)

The input capture state control/PPG output control register (ICSH23/ICSH67, ICSL23/ICSL67, PICSH01/PICSH45, PICSL01/PICSL45) is used to control the edge selection, the interrupt request enable, the interrupt request flag, and the PPG output. This register is also used to indicate a valid edge which was detected on the input capture 2/6 and 3/7.

■ Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Upper Byte (ICSH23/ICSH67)

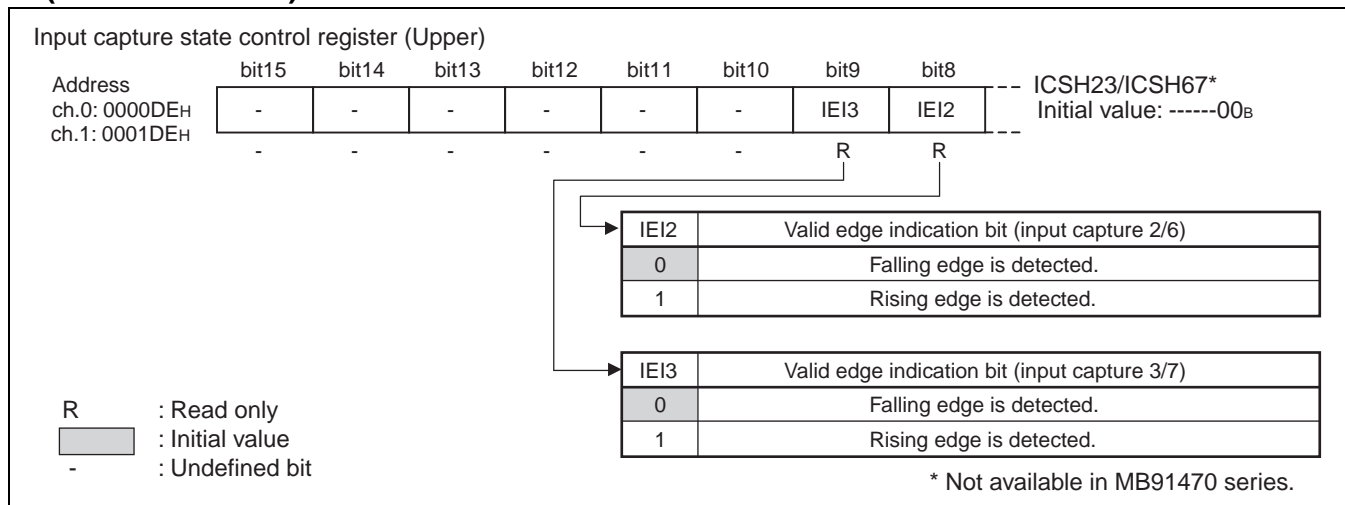


Table 12.4-12 Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Upper Byte (ICSH23/ICSH67)

Bit name		Function
bit15 to bit10	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect on operation.
bit9	IEI3: Valid edge indication bit (Input capture 3/7)	<ul style="list-style-type: none"> This bit specifies the valid edge indication bit for capture register 3/7 and indicates whether a rising edge or falling edge was detected. "0" is written to this bit when a falling edge is detected. "1" is written to this bit when a rising edge is detected. This bit is a read only bit. <p>Note: When the lower of the input capture state control register (ICSL23/ICSL67), EG31, EG30: bit3, bit2 = 00_B, the read value has no meaning.</p>
bit8	IEI2: Valid edge indication bit (Input capture 2/6)	<ul style="list-style-type: none"> This bit specifies the valid edge indication bit for capture register 2/6 and indicates whether a rising edge or falling edge was detected. "0" is written to this bit when a falling edge is detected. "1" is written to this bit when a rising edge is detected. This bit is a read only bit. <p>Note: When the lower of the input capture state control register (ICSL23/ICSL67), EG21, EG20: bit1, bit0 = 00_B, the read value has no meaning.</p>

■ Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Lower Byte (ICSL23/ICSL67)

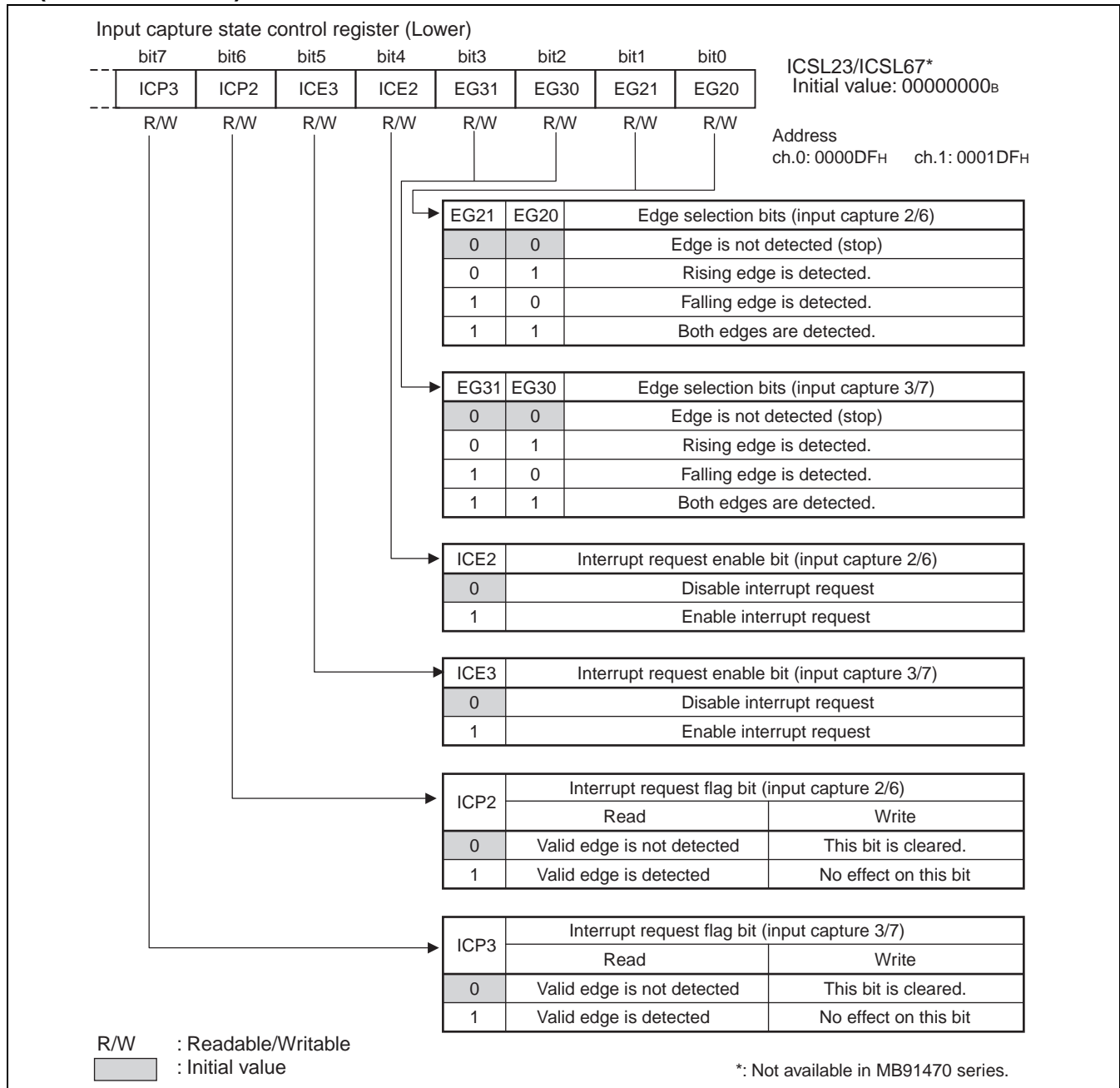


Table 12.4-13 Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Lower Byte (ICSL23/ICSL67)

Bit name		Function
bit7	ICP3: Interrupt request flag bit (Input capture 3/7)	<ul style="list-style-type: none"> • This bit is used as an interrupt request flag of the input capture 3/7. • This bit is set to "1" immediately when a valid edge of an external input pin is detected. • When a valid edge is detected while the interrupt request enable bit (ICE3: bit5) is set, the interrupt can be generated immediately. • When this bit is set to "0": Clears the bit. • Setting this bit to "1" has no effect on this bit. • When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit6	ICP2: Interrupt request flag bit (Input capture 2/6)	<ul style="list-style-type: none"> • This bit is used as an interrupt request flag of the input capture 2/6. • This bit is set to "1" immediately when a valid edge of an external input pin is detected. • When a valid edge is detected while the interrupt request enable bit (ICE2: bit4) is set, the interrupt can be generated immediately. • When this bit is set to "0": Clears the bit. • Setting this bit to "1" has no effect on this bit. • When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit5	ICE3: Interrupt request enable bit (Input capture 3/7)	<ul style="list-style-type: none"> • This bit is used to enable an input capture interrupt request of the input capture 3/7. • When the interrupt request flag bit (ICP3: bit7) is set while this bit is set to "1", the input capture 3/7 interrupt is generated.
bit4	ICE2: Interrupt request enable bit (Input capture 2/6)	<ul style="list-style-type: none"> • This bit is used to enable an input capture interrupt request of the input capture 2/6. • When the interrupt request flag bit (ICP2: bit6) is set while this bit is set to "1", the input capture 2/6 interrupt is generated.
bit3, bit2	EG31, EG30: Edge selection bits (Input capture 3/7)	<ul style="list-style-type: none"> • These bits are used to specify the active edge polarity for the external input to input capture 3/7. • These bits are used also to enable an operation of the input capture 3/7.
bit1, bit0	EG21, EG20: Edge selection bits (Input capture 2/6)	<ul style="list-style-type: none"> • These bits are used to specify the active edge polarity for the external input to input capture 2/6. • These bits are used also to enable an operation of the input capture 2/6.

■ PPG Output Control Register Upper Byte (PICS01/PICS45)

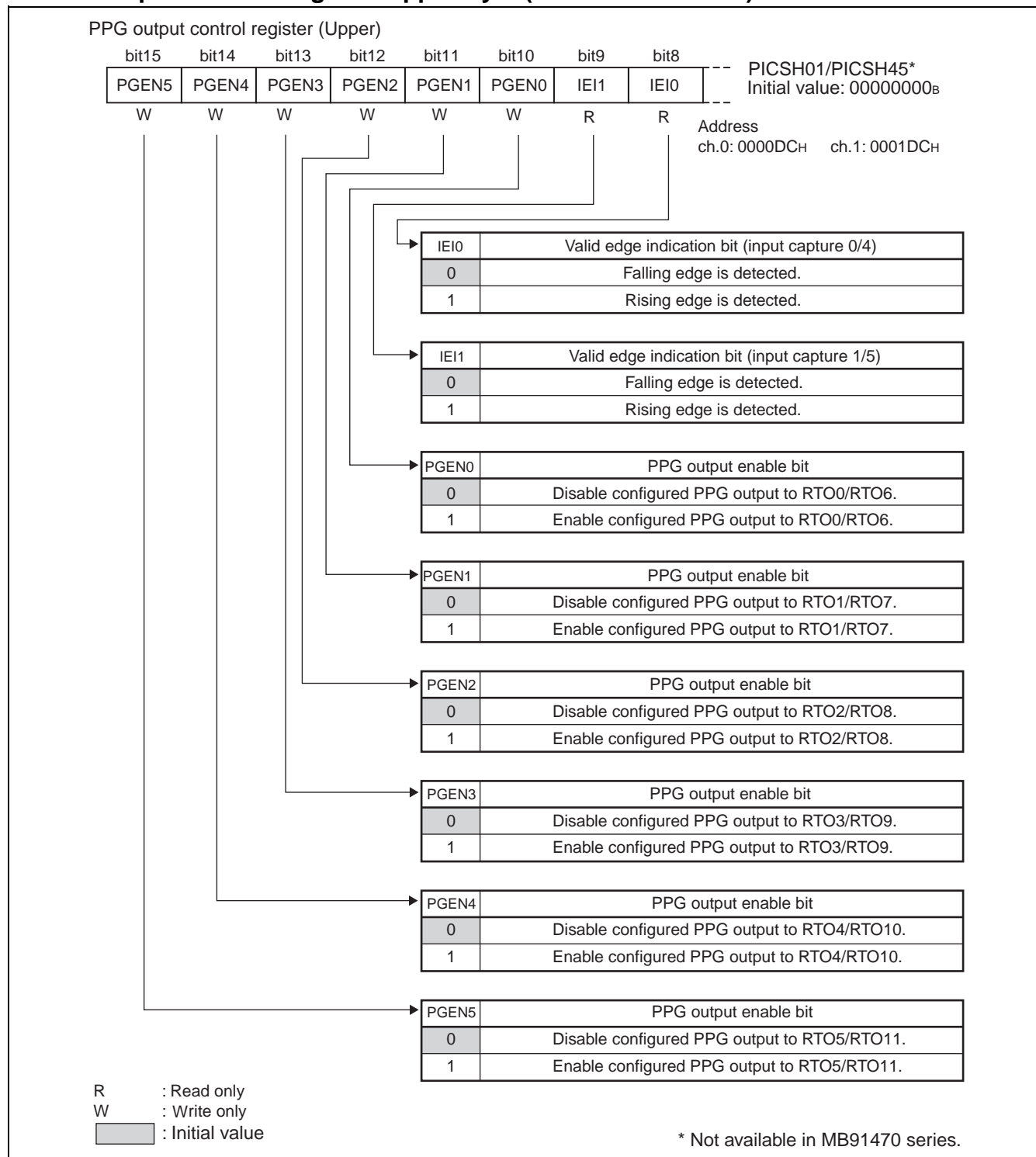


Table 12.4-14 PPG Output Control Register Upper Byte (PICSH01/PICSH45)

Bit name		Function
bit15 to bit10	PGEN5 to PGEN0: PPG output enable bits	<ul style="list-style-type: none"> These bits are used to select the PPG output to RTO0/RTO6 to RTO5/RTO11. These bits are write-only.
bit9	IEI1: Valid edge indication bit (Input capture 1/5)	<ul style="list-style-type: none"> This bit specifies the valid edge indication bit for capture register 1/5 and indicates whether a rising edge or falling edge was detected. "0" is written to this bit when a falling edge is detected. "1" is written to this bit when a rising edge is detected. This bit is a read only bit. <p>Note: When the lower of the input capture state control register (PICSL01/PICSL45), EG11, EG10: bit3, bit2 = 00_B, the read value has no meaning.</p>
bit8	IEI0: Valid edge indication bit (Input capture 0/4)	<ul style="list-style-type: none"> This bit specifies the valid edge indication bit for capture register 0/4 and indicates whether a rising edge or falling edge was detected. "0" is written to this bit when a falling edge is detected. "1" is written to this bit when a rising edge is detected. This bit is a read only bit. <p>Note: When the lower of the input capture state control register (PICSL01/PICSL45), EG01, EG00: bit1, bit0 = 00_B, the read value has no meaning.</p>

MB91470/480 Series

■ Input Capture State Control Register (ch.0/ch.4,ch.1/ch.5) Lower Byte (PICSL01/ PICSL45)

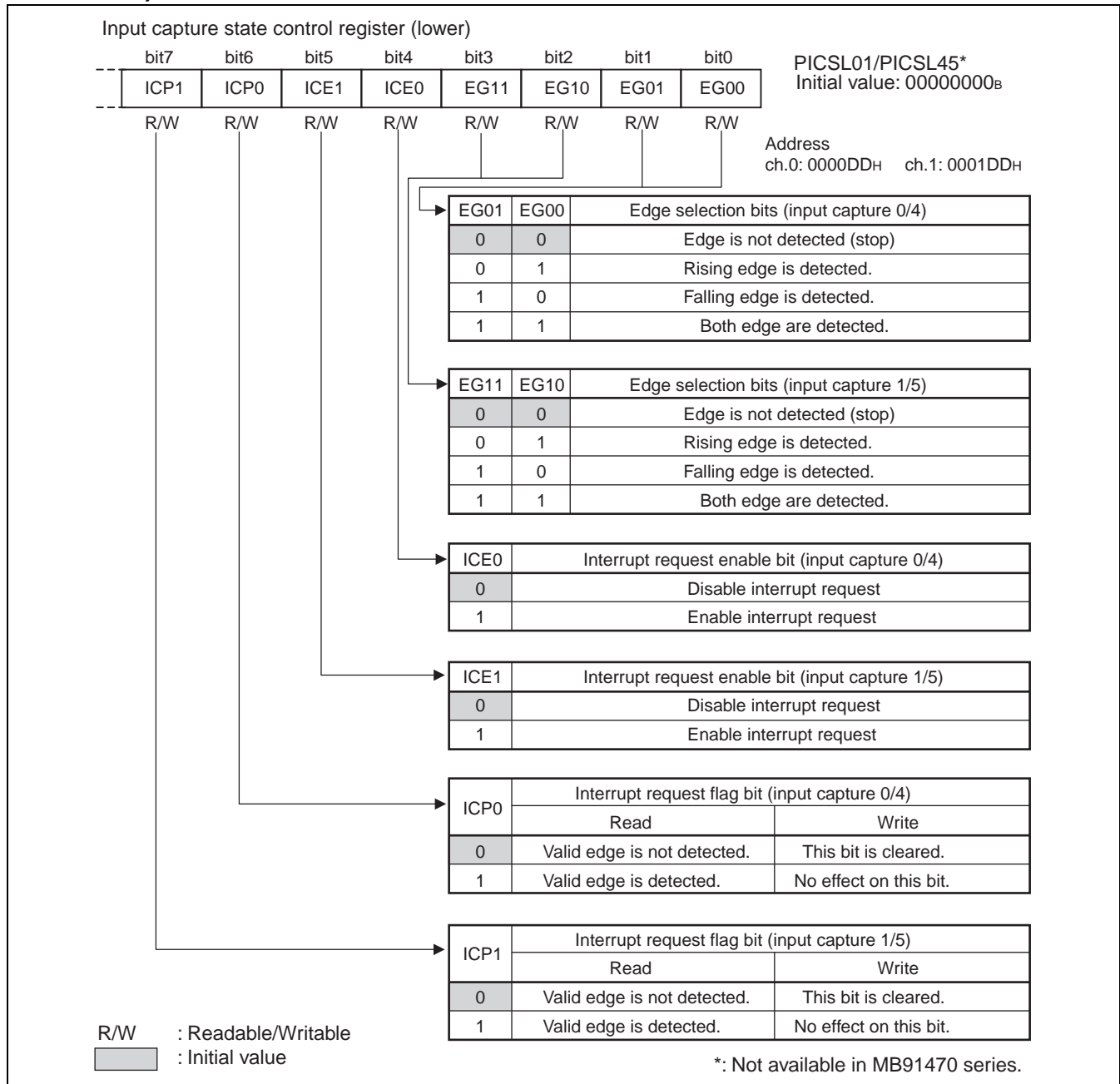


Table 12.4-15 Input Capture State Control Register (ch.0/ch.1,ch.4/ch.5) Lower Byte (PICSL01/PICSL45)

Bit name		Function
bit7	ICP1: Interrupt request flag bit (Input capture 1/5)	<ul style="list-style-type: none"> • This bit is used as an interrupt request flag of the input capture 1/5. • This bit is set to "1" immediately when a valid edge of an external input pin is detected. • When a valid edge is detected while the interrupt request enable bit (ICE1: bit5) is set, the interrupt is generated immediately. • When this bit is set to "0": Clears the bit. • Setting this bit to "1" has no effect on this bit. • When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit6	ICP0: Interrupt request flag bit (Input capture 0/4)	<ul style="list-style-type: none"> • This bit is used as an interrupt request flag of the input capture 0/4. • This bit is set to "1" immediately when a valid edge of an external input pin is detected. • When a valid edge is detected while the interrupt request enable bit (ICE0: bit4) is set, the interrupt is generated immediately. • When this bit is set to "0": Clears the bit. • Setting this bit to "1" has no effect on this bit. • When this bit is read to a read modify write (RMW) instruction, "1" is always read.
bit5	ICE1: Interrupt request enable bit (Input capture 1/5)	<ul style="list-style-type: none"> • This bit is used to enable an input capture interrupt request of the input capture 1/5. • When the interrupt request flag bit (ICP1: bit7) is set while this bit is set to "1", the input capture 1/5 interrupt is generated.
bit4	ICE0: Interrupt request enable bit (Input capture 0/4)	<ul style="list-style-type: none"> • This bit is used to enable an input capture interrupt request of the input capture 0/4. • When the interrupt request flag bit (ICP0: bit6) is set while this bit is set to "1", the input capture 0/4 interrupt is generated.
bit3, bit2	EG11,EG10: Edge selection bits (Input capture 1/5)	<ul style="list-style-type: none"> • These bits are used to specify the active edge polarity for the external input to input capture 1/5. • These bits are used also to enable an operation of the input capture 1/5.
bit1, bit0	EG01,EG00: Edge selection bits (Input capture 0/4)	<ul style="list-style-type: none"> • These bits are used to specify the active edge polarity for the external input to input capture 0/4. • These bits are used also to enable an operation of the input capture 0/4.

MB91470/480 Series

12.4.11 16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)

The 16-bit dead timer register stores the compare value of the 16-bit dead timer.

■ 16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)

16-bit dead timer register (Upper)										
TMRRH0 to TMRRH2		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value XXXXXXXX _B
TMRRH3* to TMRRH5*		TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	
Address										
WG0	WG1*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.0: 0000E0H	ch.3: 0001E0H									
ch.1: 0000E2H	ch.4: 0001E2H									
ch.2: 0000E4H	ch.5: 0001E4H									

16-bit dead timer register (Lower)										
TMRRL0 to TMRRL2		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value XXXXXXXX _B
TMRRL3* to TMRRL5*		TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/Writable

*: Not available in MB91470 series.

These registers are used to store the compare value for the 16-bit dead timer.

These register values are reloaded when the 16-bit dead timer starts the operation.

If the values are rewritten to these registers during the timer operation, these new values are enabled in the next timer start/operation.

To access these registers, use a half-word or word access instruction.

In the dead time timer mode, these registers are used to set the non-overlap time.

$$\text{Non-overlap time} = (\text{Setting value}) \times \text{Selected clock}$$

Note:

"0000_H" cannot be set.

In the timer mode, these registers are used to set the GATE time of the PPG timer operation.

$$\text{GATE time} = (\text{Setting value}) \times \text{Selected clock}$$

Note:

"0000_H" cannot be set.

12.4.12 16-bit Dead Timer Control Register (DTCR0 to DTCR5)

The 16-bit dead timer control register (DTCR0/DTCR3 to DTCR2/DTCR5) is used to control the operation mode of the waveform generator, the interrupt request enable, the interrupt request flag, the GATE signal enable, and the output level polarity.

16-bit Dead Timer Control Register, Upper Byte (DTCR0/DTCR3)

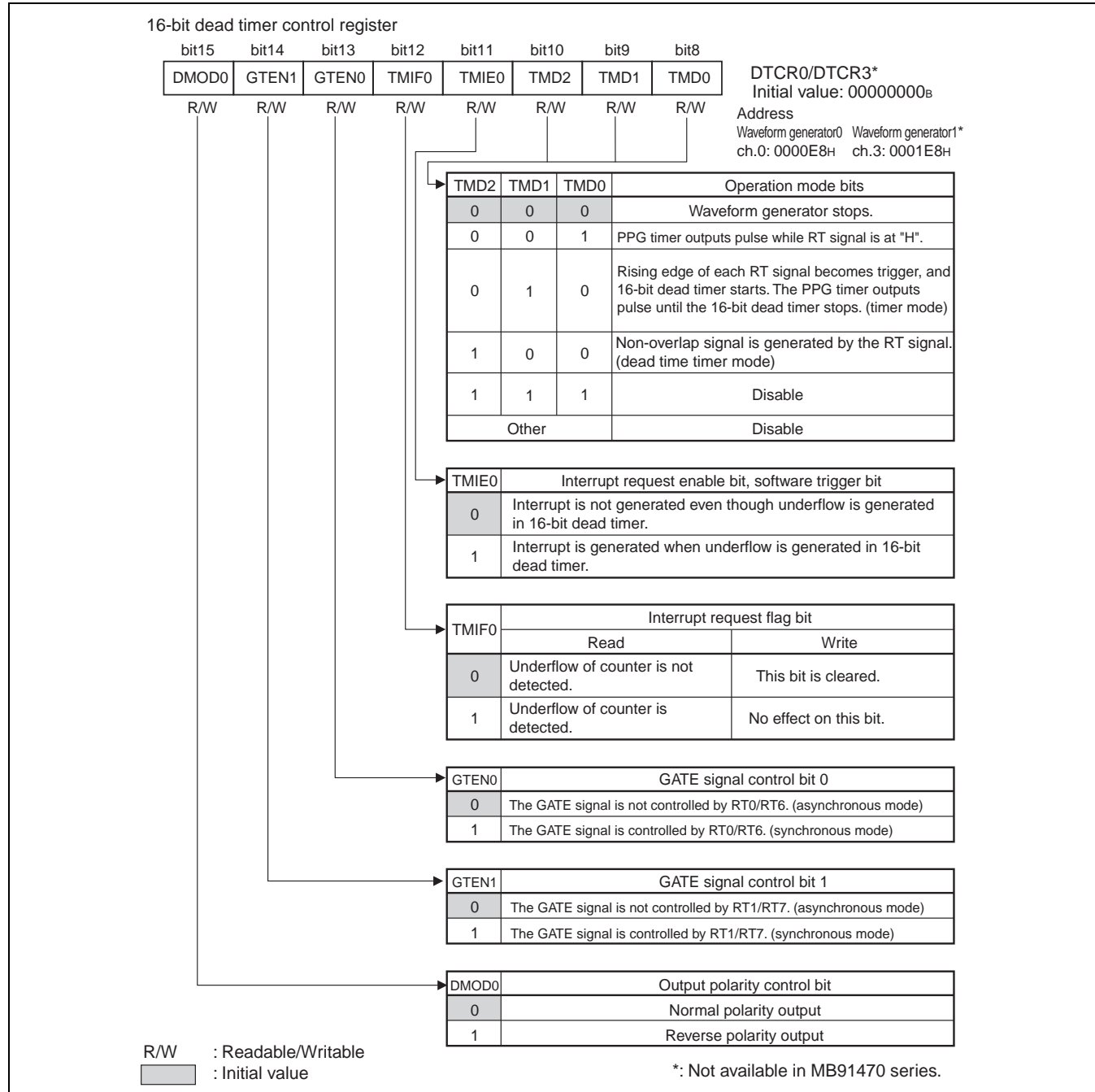


Table 12.4-16 16-bit Dead Timer Control Register, Upper Byte (DTCR0/DTCR3)

Bit name		Function
bit15	DMOD0: Output polarity control bit	<ul style="list-style-type: none"> This bit is used to set the U/V/W output in the dead time timer mode. Setting this bit reverses the U/V/W output polarity. <p>Note: When the dead time timer mode is not selected, (TMD2: bit10 = 0), this bit has no meaning.</p>
bit14	GTEN1: GATE signal control bit1	This bit is used to control the GATE signal of the PPG timer with RT1/RT7.
bit13	GTEN0: GATE signal control bit0	This bit is used to control the GATE signal of the PPG timer with RT0/RT6.
bit12	TMIF0: Interrupt request flag bit	<ul style="list-style-type: none"> This bit is used as an interrupt request flag of the 16-bit dead timer. This bit is set to "1" when an underflow occurs on the 16-bit dead timer. Writing "0" to this bit clears the bit. Writing "1" to this bit has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read. <p>Note: This bit works only when the register values (TMD2 to TMD0: bit10 to bit8) are 000_B or 001_B, and becomes always "0" when they are other values. If a software clear (writing "0") and hardware set (underflow on 16-bit dead timer 0) occur simultaneously, the software operation has precedence and the bit is cleared.</p>
bit11	TMIE0: Interrupt request enable bit, software trigger bit	<ul style="list-style-type: none"> This bit is used as a software trigger bit and an interrupt enable bit of the 16-bit dead timer. TMD2 to TMD0: bit10 to bit8 = 000_B or 001_B: This bit is used as a software trigger of the 16-bit dead timer. When this bit changes from "0" to "1", it becomes a trigger of the 16-bit dead timer, reloads the value, and starts the down count. When this bit is "1" and the interrupt request flag bit (TMIF0: bit12) is "1", an interrupt request is sent to the CPU. <p>Note: To trigger the 16-bit dead timer again, you must write "0" to this bit before writing "1".</p>
bit10 to bit8	TMD2 to TMD0: Operation mode bits	<ul style="list-style-type: none"> These bits are used to select the operation mode of the waveform generator. TMD2 to TMD0: bit10 to bit8 = 000_B: The RT0/RT6 and RT1/RT7 signals of the output compare are respectively output from the RTO0/RTO6 and RTO1/RTO7. The 16-bit dead timer is also used as a reload timer. TMD2 to TMD0: bit10 to bit8 = 001_B: The RT0/RT6 and RT1/RT7 signals of the output compare are respectively output from RTO0/RTO6 and RTO1/RTO7 when the PPG output is disabled (the upper of the PPG output control/input capture state control register (PICSH01/PICSH45), PGEN0: bit10 = 0, PGEN1: bit11 = 0). The 16-bit dead timer is also used as a reload timer. <p>Note: Always select two-channel mode for RT1 (set bit12 (CMOD) in the upper compare control register (OSCH1/OSCH7)=1) to use the waveform generator in dead time timer mode.</p>

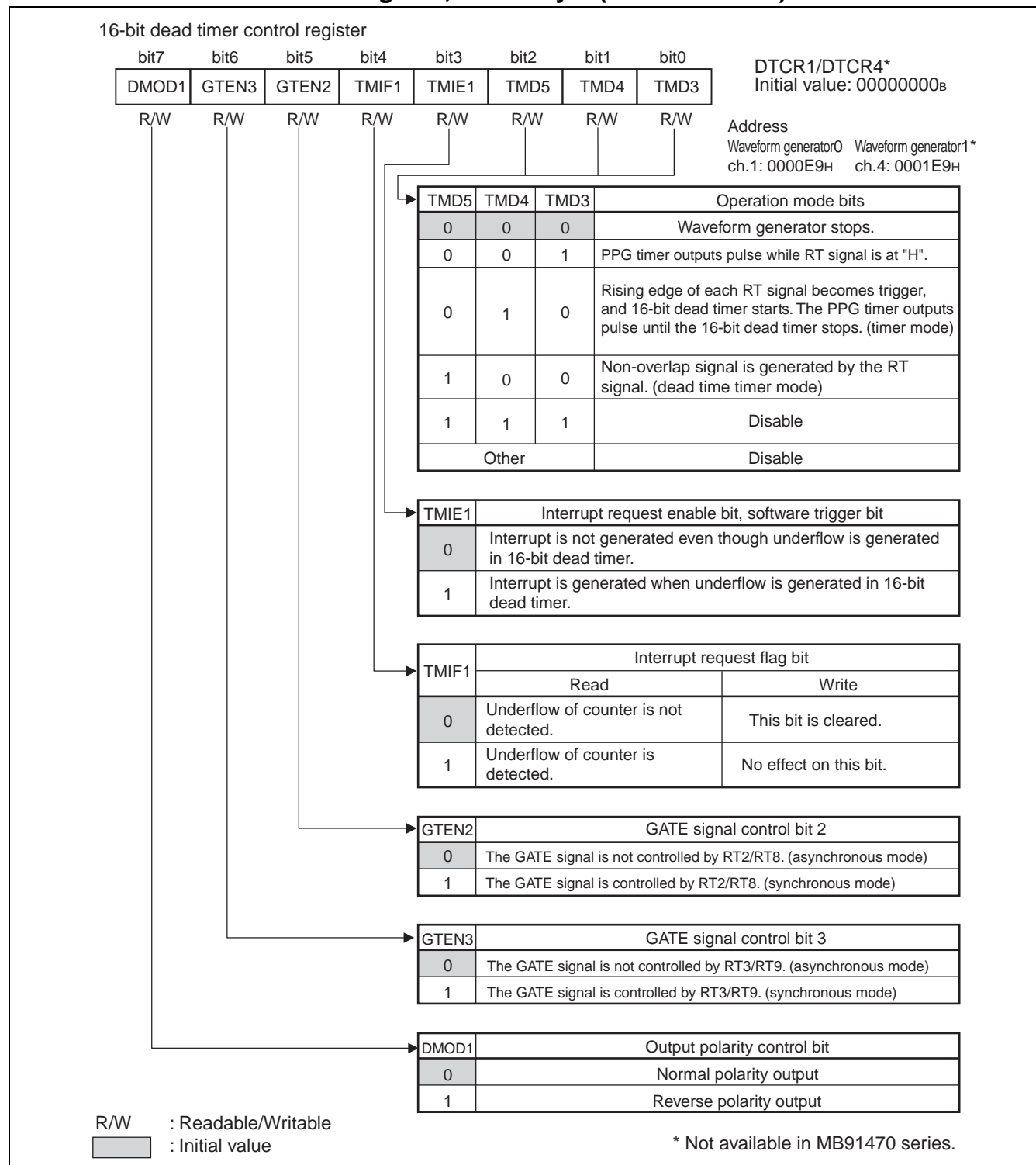
16-bit Dead Timer Control Register, Lower Byte (DTCR1/DTCR4)

Table 12.4-17 16-bit Dead Timer Control Register, Lower Byte (DTCR1/DTCR4)

Bit name		Function
bit7	DMOD1: Output polarity control bit	<ul style="list-style-type: none"> This bit is used to set the U/V/W output in the dead time timer mode. Setting this bit reverses the U/V/W output polarity. <p>Note: When the dead time timer mode is not selected, (TMD5: bit2 = 0), this bit has no meaning.</p>
bit6	GTEN3: GATE signal control bit3	This bit is used to control the GATE signal of the PPG timer with RT3/RT9.
bit5	GTEN2: GATE signal control bit2	This bit is used to control the GATE signal of the PPG timer with RT2/RT8.
bit4	TMIF1: Interrupt request flag bit	<ul style="list-style-type: none"> This bit is used as an interrupt request flag of the 16-bit dead timer. This bit is set to "1" when an underflow occurs on the 16-bit dead timer. Writing "0" to this bit clears the bit. Writing "1" to this bit has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read. <p>Note: This bit works only when the register values (TMD5 to TMD3: bit2 to bit0) are "000_B" or "001_B", and becomes always 0 when they are other values. If a software clear (writing "0") and hardware set (underflow on 16-bit dead timer 1) occur simultaneously, the software operation has precedence and the bit is cleared.</p>
bit3	TMIE1: Interrupt request enable bit, software trigger bit	<ul style="list-style-type: none"> This bit is used as a software trigger bit and an interrupt enable bit of the 16-bit dead timer. TMD5 to TMD3: bit2 to bit0 = 000_B or 001_B: This bit is used as a software trigger of the 16-bit dead timer. When this bit changes from "0" to "1", it becomes a trigger of the 16-bit dead timer, reloads the value, and starts the down count. When this bit is "1" and the interrupt request flag bit (TMIF1: bit4) is "1", an interrupt request is sent to the CPU. <p>Note: To trigger the 16-bit dead timer again, you must write "0" to this bit before writing "1".</p>
bit2 to bit0	TMD5 to TMD3: Operation mode bits	<ul style="list-style-type: none"> These bits are used to select the operation mode of the waveform generator. TMD5 to TMD3: bit2 to bit0 = 000_B: The RT2/8 and RT3/9 signals of the output compare are respectively output from the RTO2/RTO8 and RTO3/RTO9. The 16-bit dead timer is also used as a reload timer. TMD5 to TMD3: bit2 to bit0 = 001_B: The RT2/RT8 and RT3/RT9 signals of the output compare are respectively output from the RTO2/RTO8 and RTO3/RTO9 when the PPG0 output is disabled (the upper of the PPG output control/input capture state control register (PICSH01/PICSH45), PGEN2: bit12=0, PGEN3: bit13=0). The 16-bit dead timer is also used as a reload timer. <p>Note: Always select two-channel mode for RT3/RT9 (set bit12 (CMOD) in the upper compare control register (OCSH3/OCSH9)=1) to use the waveform generator in dead time timer mode.</p>

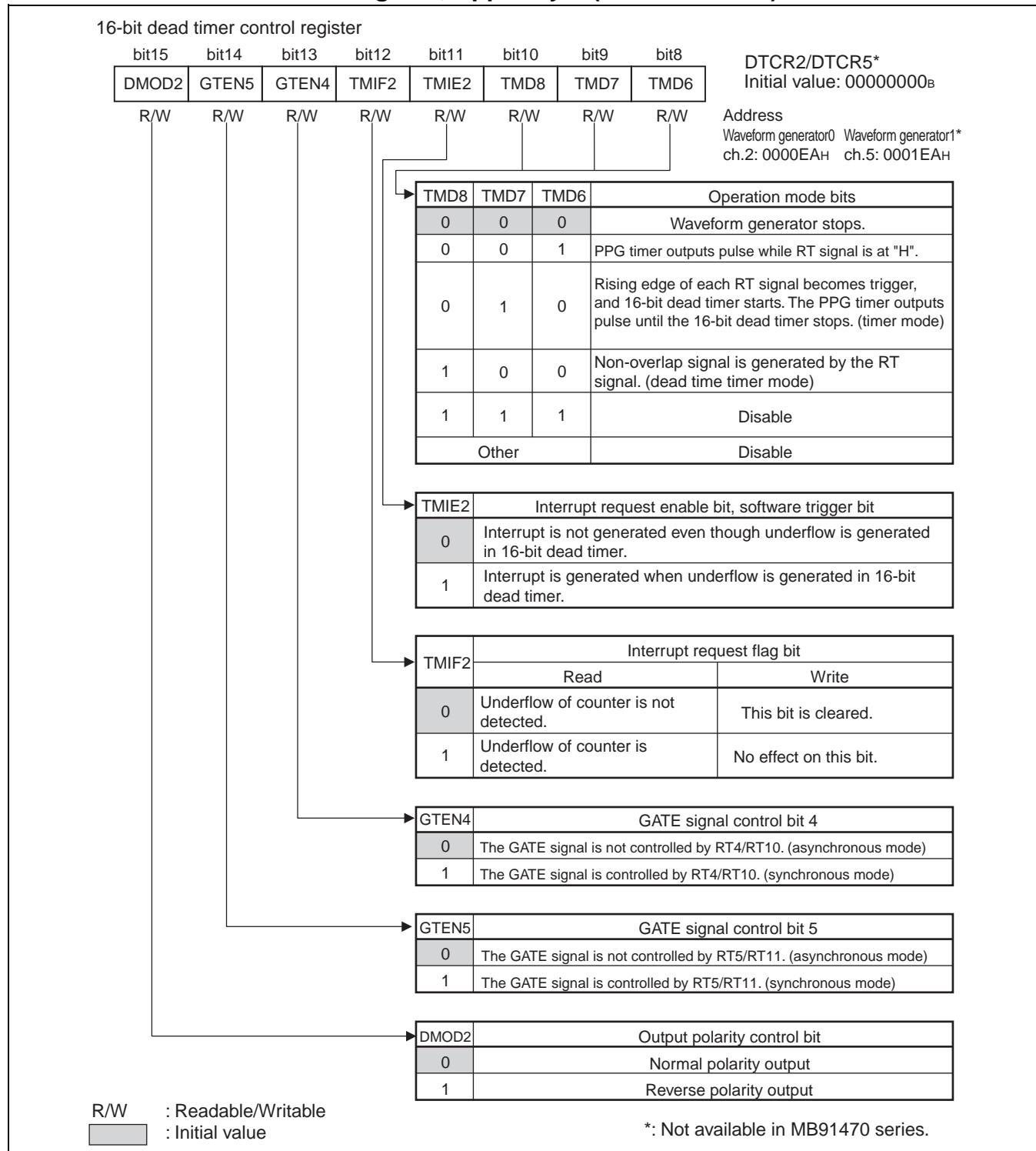
■ 16-bit Dead Timer Control Register, Upper Byte (DTCR2/DTCR5)

Table 12.4-18 16-bit Dead Timer Control Register, Upper Byte (DTCR2/DTCR5)

Bit name		Function
bit15	DMOD2: Output polarity control bit	<ul style="list-style-type: none"> This bit is used to set the U/V/W output in the dead time timer mode. Setting this bit reverses the U/V/W output polarity. <p>Note: When the dead time timer mode is not selected, (TMD8: bit10 = 0), this bit has no meaning.</p>
bit14	GTEN5: GATE signal control bit5	This bit is used to control the GATE signal of the PPG timer with RT5/RT11.
bit13	GTEN4: GATE signal control bit4	This bit is used to control the GATE signal of the PPG timer with RT4/RT10.
bit12	TMIF2: Interrupt request flag bit	<ul style="list-style-type: none"> This bit is used as an interrupt request flag of the 16-bit dead timer. This bit is set to "1" when an underflow occurs on the 16-bit dead timer. Writing "0" to this bit clears the bit. Writing "1" to this bit has no effect on this bit. When this bit is read to a read modify write (RMW) instruction, "1" is always read. <p>Note: This bit works only when the register values (TMD8 to TMD6: bit10 to bit8) are "000_B" or "001_B", and becomes always "0" when they are other values. If a software clear (writing "0") and hardware set (underflow on 16-bit dead timer 2) occur simultaneously, the software operation has precedence and the bit is cleared.</p>
bit11	TMIE2: Interrupt request enable bit, software trigger bit	<ul style="list-style-type: none"> This bit is used as a software trigger bit and an interrupt enable bit of the 16-bit dead timer. TMD8 to TMD6: bit10 to bit8 = 000_B or 001_B: This bit is used as a software trigger of the 16-bit dead timer. When this bit changes from "0" to "1", it becomes a trigger of the 16-bit dead timer, reloads the value, and starts the down count. When this bit is "1" and the interrupt request flag bit (TMIF2: bit12) is "1", an interrupt request is sent to the CPU. <p>Note: To trigger the 16-bit dead timer again, you must write "0" to this bit before writing "1".</p>
bit10 to bit8	TMD8 to TMD6: Operation mode bits	<ul style="list-style-type: none"> These bits are used to select the operation mode of the waveform generator. TMD8 to TMD6: bit10 to bit8 = 000_B: The RT4/RT10 and RT5/RT11 signals of the output compare are respectively output from the RTO4/RTO10 and RTO5/RTO11. The 16-bit dead timer is also used as a reload timer. TMD8 to TMD6: bit10 to bit8 = 001_B: The RT4/RT10 and RT5/RT11 signals of the output compare are respectively output from the RTO4/RTO10 and RTO5/RTO11 when the PPG output is disabled (the upper of the PPG output control/input capture state control register (PICSH01/PICSH45), PGEN4: bit14 = 0, PGEN5: bit15 = 0). The 16-bit dead timer is also used as a reload timer. <p>Note: Always select two-channel mode for RT5/RT11 (set bit12 (CMOD) in the upper compare control register (OCSH5)=1) to use the waveform generator in dead time timer mode.</p>

12.4.13 Waveform Control Register (SIGCR1/SIGCR2)

The waveform control register is used to control the operating clock frequency, enable the noise cancellation feature, enable DTTI input, and control DTTI interrupts.

■ Waveform Control Register 1 (SIGCR1)

Waveform control register 10/11

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SIGCR1*
DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0	Initial value: 00000000 _B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Address Waveform generator0 Waveform generator1* ch.10: 0000ED _H ch.11: 0001ED _H
			NWS1		NWS0	DTTI0/DTTI1 noise width selection bits		
			0		0	Cancel noise of 4 peripheral clock (CLKP) cycles		
			0		1	Cancel noise of 8 peripheral clock (CLKP) cycles		
			1		0	Cancel noise of 16 peripheral clock (CLKP) cycles		
			1		1	Cancel noise of 32 peripheral clock (CLKP) cycles		
			DCK2		DCK1	DCK0	Operation clock selection bits	
			0		0	0	φ (50 ns, φ=20 MHz)	
			0		0	1	φ/2 (100 ns, φ=20 MHz)	
			0		1	0	φ/4 (200 ns, φ=20 MHz)	
			0		1	1	φ/8 (400 ns, φ=20 MHz)	
			1		0	0	φ /16 (800 ns, φ=20 MHz)	
			1		0	1	φ /32 (1.6 μs, φ=20 MHz)	
			1		1	0	φ /64 (3.2 μs, φ=20 MHz)	
			1		1	1	Disable	
			φ: Peripheral clock (CLKP)					
			NRSL		Noise cancel function valid bit			
			0		Noise cancel circuit of DTTI0/DTTI1 input is invalid.			
			1		Noise cancel circuit of DTTI0/DTTI1 input is valid.			
			DTIF		DTTI0/DTTI1 interrupt flag bit			
					Read		Write	
			0		No interrupt request		This bit is cleared.	
			1		Interrupt request		No effect on this bit.	
			DTIE		DTTI0/DTTI1 input valid bit			
			0		Invalid DTTI0/DTTI1 input			
			1		Valid DTTI0/DTTI1 input			

R/W

: Readable/Writable

R/W : Readable/Writable
 : Initial value

*: Not available in MB91470 series.

Table 12.4-19 Waveform Control Register1 (SIGCR1)

Bit name		Function
bit7	DTIE: DTTI0/DTTI1 input enabled bit	This bit is used to enable the output level control DTTI signal of RTO pins 0/6 to 5/11.
bit6	DTIF: DTTI0/DTTI1 interrupt flag bit	<ul style="list-style-type: none"> • This bit is the DTTI0/DTTI1 interrupt flag. • When DTTI0/DTTI1 input is enabled (DTIE: bit7 = 1) and DTTI0/DTTI1 "L" level is detected, this bit is set, and an interrupt request is generated. • When this bit is set to "0": Clears the bit. • Setting this bit to "1" has no effect on this bit. • "1" is always read during read modify write (RMW) instruction. <p>Note: When the noise cancellation feature is enabled (NRSL: bit5 = 1), this bit is set to "1" when a noise pulse is generated. If a software clear (write 0) and hardware set (DTTI0/DTTI1 "L" level detected) occur simultaneously, the software clear is given precedence over the hardware set, and this bit is cleared.</p>
bit5	NRSL: Noise cancellation feature enabled bit	<ul style="list-style-type: none"> • This bit is used to enable the noise cancellation feature. • The noise canceling circuit accepts the DTTI0/DTTI1 input signal if it remains at the "L" level until an overflow occurs on the counter. The counter is an n-bit counter operated by Low level input. n is set by NWS1 bit and NWS0 bit; Based on the settings of bit1 and bit0, the value of n is 2, 3, 4, or 5. <p>Note: Approximately 2ⁿ peripheral clock (CLKP) are required to cancel the noise pulse width. When a noise cancellation circuit is selected, input is disabled when in a mode that stops the peripheral clock (CLKP) (e.g. stop mode).</p>
bit4 to bit2	DCK2 to DCK0: Operation clock selection bits	These bits are used to select the operating clock for the 16-bit dead timer.
bit1, bit0	NWS1, NWS0: DTTI0/DTTI1 noise width selection bits	These bits are used to select the width of noise pulses to reject on the DTTI0/DTTI1 pin.

■ **Waveform Control Register 2 (SIGCR2)**

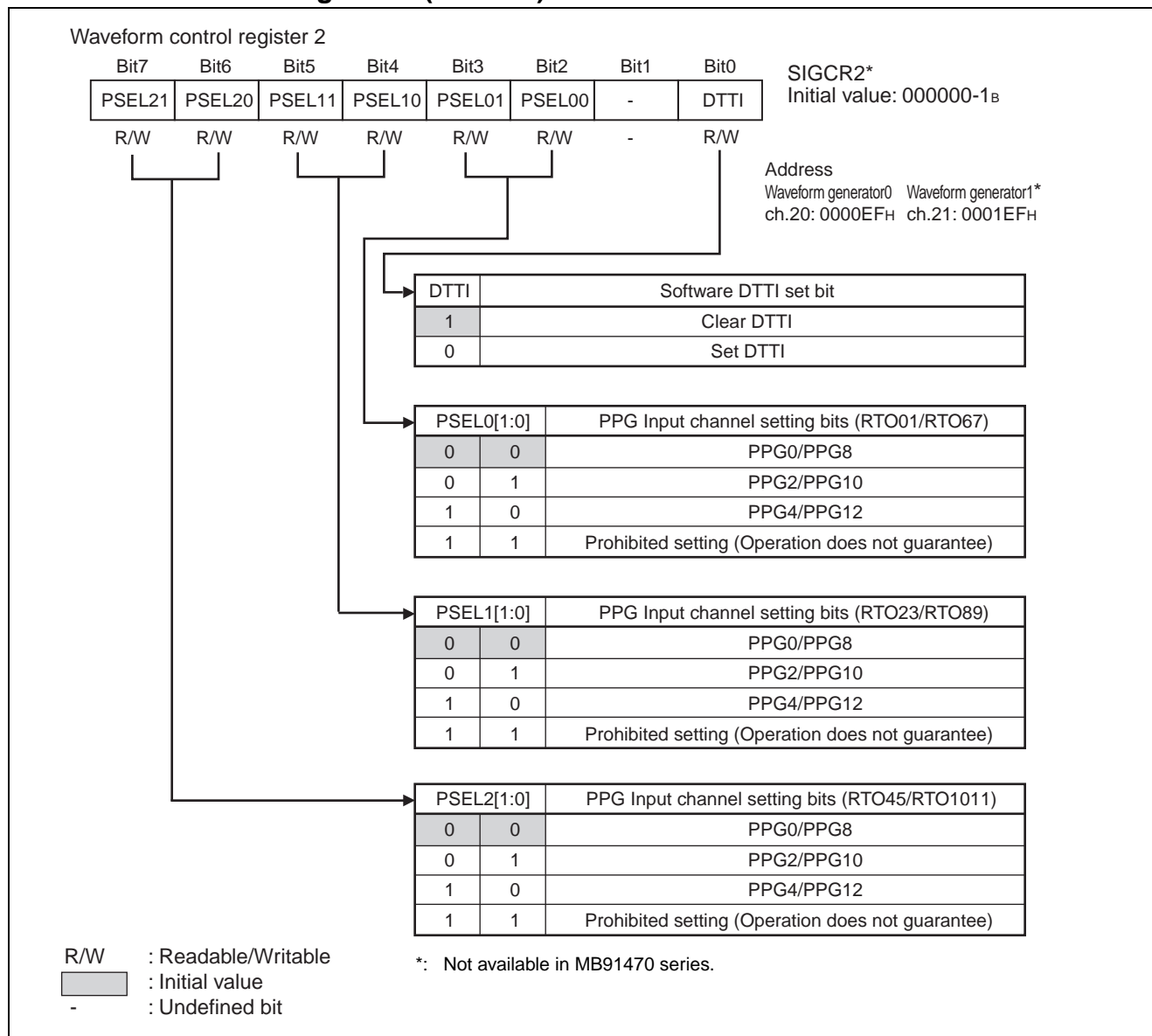


Table 12.4-20 Waveform Control Register 20/21 (SIGCR2)

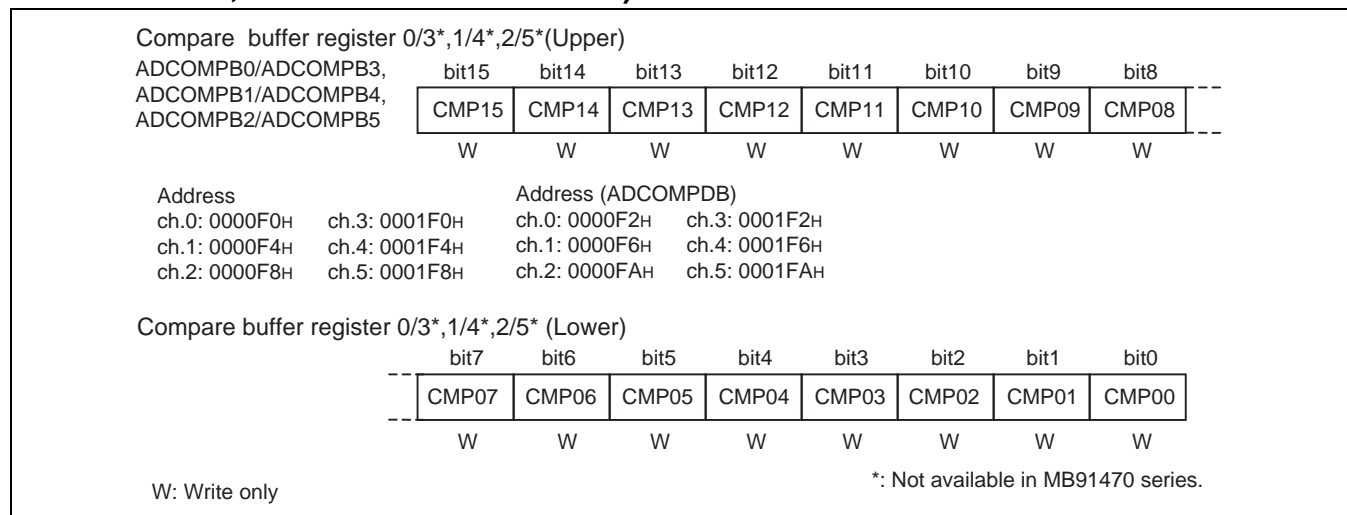
Bit name		Function
bit7, bit6	PSEL2[1:0]: PPG input channel setting bits (RTO45/ RTO1011)	<ul style="list-style-type: none"> These register bits are to set the input PPG for output RTO45/RTO1011. PSEL2[1:0]=11_B is prohibited setting.
bit5, bit4	PSEL1[1:0]: PPG input channel setting bits (RTO23/ RTO89)	<ul style="list-style-type: none"> These register bits are to set the input PPG for output RTO23/RTO89. PSEL1[1:0]=11_B is prohibited setting.
bit3, bit2	PSEL0[1:0]: PPG input channel setting bits (RTO01/ RTO67)	<ul style="list-style-type: none"> These register bits are to set the input PPG for output ROT01/RTO67 *. PSEL0[1:0]=11_B is prohibited setting.
bit1	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit0	DTTI: Software DTTI bit	<ul style="list-style-type: none"> Write "0" to set DTTI0/DTTI1. Write "1" to this bit to clear it. <p>Note: As this uses the external input DTTI0/DTTI1 and OR, however, DTTI0/DTTI1 depends on the external input level.</p>

*: SIGCR20 controls RTO01/RTO23/RTO45 of the waveform generator 0.
SIGCR21 controls RTO67/RTO89/RTO1011 of the waveform generator 1.

12.4.14 A/D Activation Compare Register (ADCOMPB0 to ADCOMPB5, ADCOMP0 to ADCOMP5, ADTGCE0/ADTGCE1, ADTGSEL0/ADTGSEL1, ADTGBUF0/ADTGBUF1)

Compare registers 0/3, 1/4, 2/5 activate A/D converters when their values match that of the free-run timer. The compare register buffer is used to write compare values. The control register can select whether the A/D activation request is generated when the compare match occurs.

■ Compare Buffer Register 0/3, 1/4, 2/5 (ADCOMPB0/ADCOMPB3, ADCOMPB1/ADCOMPB4, ADCOMPB2/ADCOMPB5)



The compare buffer register is a buffer register for A/D activation compare register (ADCOMP).

When the buffer function is disabled (buffer control register (ADTGBUF), BUF2, BUF1, BUF0:bit2, bit1, bit0 = 111_B), or the free-run timer is stopped, the value of the compare buffer register is transferred to the compare register immediately. When the buffer function is enabled (buffer control register (ADTGBUF), BUF2, BUF1, BUF0:bit2, bit1, bit0 = 000_B), the value is transferred to the compare register when a compare match or a zero detection occurs.

When count direction selection register (ADTGSEL), SEL1, SEL0 = 11_B, ADCOMPDB0 to ADCOMPDB5 are operating as a buffer registers for ADCOMP0 to ADCOMP5.

To write this register, use a halfword or word access instruction.

MB91470/480 Series

■ Compare Register 0/3, 1/4, 2/5 (ADCOMP0/ADCOMP3, ADCOMP1/ADCOMP4, ADCOMP2/ADCOMP5)

Compare register 0/3*, 1/4*, 2/5* (Upper)										
ADCOMP0/ADCOMP3, ADCOMP1/ADCOMP4, ADCOMP2/ADCOMP5		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value 00000000 _B
		CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	
		R	R	R	R	R	R	R	R	
Address		Address (ADCOMPD)								
ch.0: 0000F0H	ch.3: 0001F0H	ch.0: 0000F2H	ch.3: 0001F2H							
ch.1: 0000F4H	ch.4: 0001F4H	ch.1: 0000F6H	ch.4: 0001F6H							
ch.2: 0000F8H	ch.5: 0001F8H	ch.2: 0000FAH	ch.5: 0001FAH							
Compare register 0/3*, 1/4*, 2/5* (Lower)										
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value 00000000 _B
		CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	
		R	R	R	R	R	R	R	R	
R: Read only										
*: Not available in MB91470 series.										

The compare register is used to write data for comparison with the 16-bit free-run timer count value. It is possible to activate A/D when the free-run timer and compare values match.

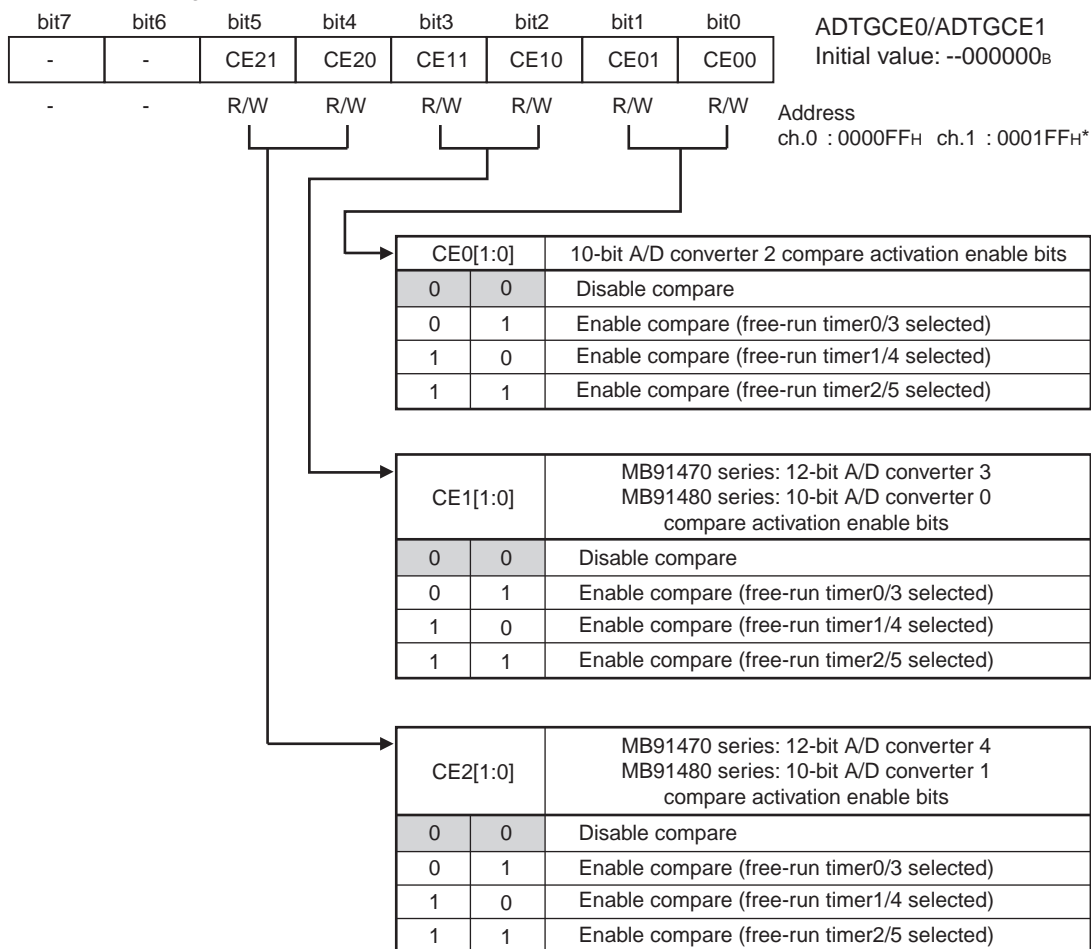
The value written to the compare register is used for a comparison immediately.

When count direction selection register (ADTGSEL), SEL1, SEL0 = 11_B, ADCOMP0 to ADCOMP5 and ADCOMPD0 to ADCOMPD5 are doing compare match operation. The former is only during up counting, the later only during down counting of the free-run timer.

Always use word or half-word access to read the compare register. Do not access this register with the read modify write (RMW) instructions.

■ **Compare Enable Register (ADTGCE0/ADTGCE1)**

Compare enable register



R/W : Readable/Writable
 : Initial value
 - : Undefined bit

Note: In MB91470 series, only free-run timer0 to free-run timer2 are available.
 In MB91480 series, there are two units of the multifunction timer.
 A/D trigger 0 can select the input free-run timers from free-run timer0 to free-run timer2.
 A/D trigger 1 can select the input free-run timers from free-run timer3 to free-run timer5.
 *: Not available in MB91470 series

Table 12.4-21 Compare Enable Register (ADTGCE0/ADTGCE1)

Bit name		Function
bit7, bit6	Undefined bits.	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect on operation.
bit5, bit4	CE21, CE20: MB91470 series: 12-bit A/D converter 4, MB91480 series: 10-bit A/D converter 1 compare activation enable bits	<ul style="list-style-type: none"> Write "00_B" to these bits to disable compare operation. Write other than "00_B" to these bits to output a activation request for MB91470 series: 12-bit A/D converter 4, MB91480 series: 10-bit A/D converter 1 when there is a free-run timer and a compare value match.
bit3, bit2	CE11, CE10: MB91470 series: 12-bit A/D converter 3, MB91480 series: 10-bit A/D converter 0 compare activation enable bits	<ul style="list-style-type: none"> Write "00_B" to these bits to disable compare operation. Write other than "00_B" to these bits to output a activation request for MB91470 series: 12-bit A/D converter 3, MB91480 series: 10-bit A/D converter 0 when there is a free-run-timer and a compare value match.
bit1, bit0	CE01, CE00: 10-bit A/D converter 2 compare activation enable bits	<ul style="list-style-type: none"> Write "00_B" to these bits to disable compare operation. Write other than "00_B" to these bits to output a activation request for 10-bit A/D converter 2 when there is a free-run timer and a compare value match.

Note:

If these registers are set, be sure to check that the free-run timer is stopped.

■ Count Direction Selection (for Comparison) Register (ADTGSEL0/ADTGSEL1)

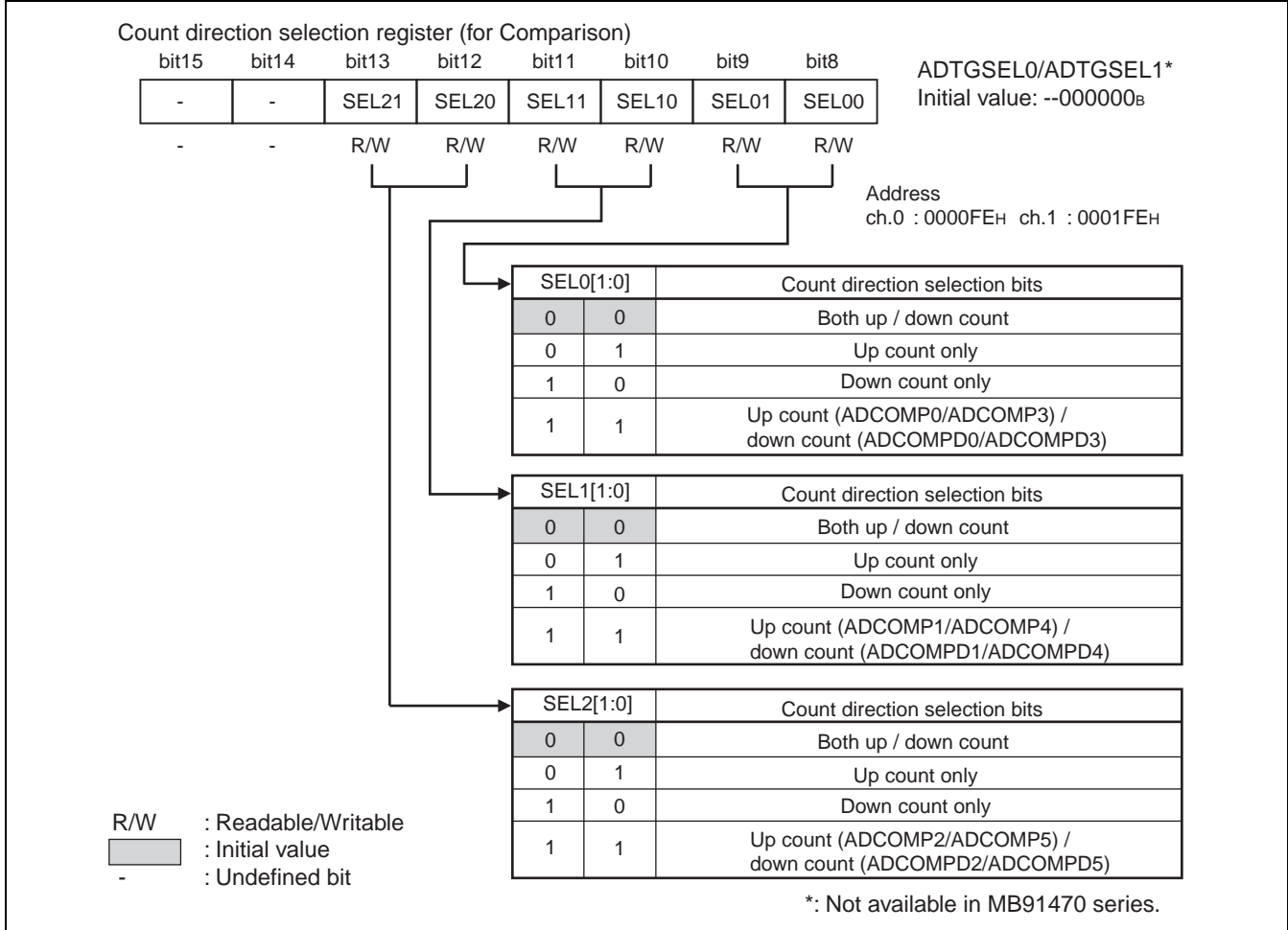
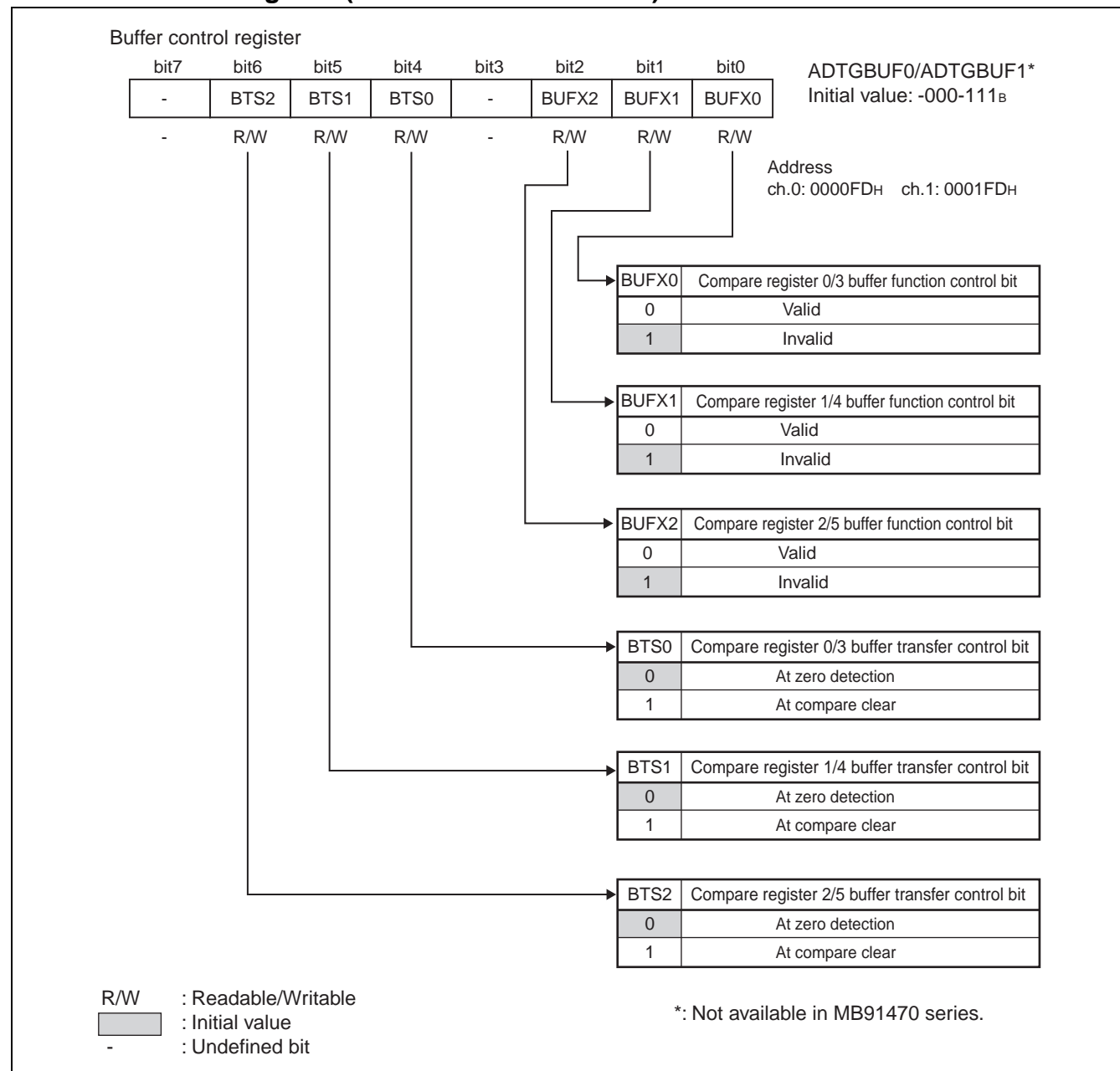


Table 12.4-22 Count Direction Selection (for Comparison) Register (ADTGSEL0/ADTGSEL1)

Bit name		Function
bit15, bit14	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect on operation.
bit13, bit12	SEL2: Count direction selection bits	<ul style="list-style-type: none"> Write "00_B" to activate the comparison in up/down count mode of free-run timer. Write "01_B" to activate the comparison in up count mode of free-run timer. Write "10_B" to activate the comparison in down count mode of free-run timer. When setting "11_B", ADCOMP2/ADCOMP5 execute compare match only while free-run timer is up count, and ADCOMPD2/ADCOMPD5 execute compare match only while free-run timer is down count.
bit11, bit10	SEL1: Count direction selection bits	<ul style="list-style-type: none"> Write "00_B" to activate the comparison in up/down count mode of free-run timer. Write "01_B" to activate the comparison in up count mode of free-run timer. Write "10_B" to activate the comparison in down count mode of free-run timer. When setting "11_B", ADCOMP1/ADCOMP4 execute compare match only while free-run timer is up count, and ADCOMPD1/ADCOMPD4 execute compare match only while free-run timer is down count.
bit9, bit8	SEL0: Count direction selection bits	<ul style="list-style-type: none"> Write "00_B" to activate the comparison in up/down count mode of free-run timer. Write "01_B" to activate the comparison in up count mode of free-run timer. Write "10_B" to activate the comparison in down count mode of free-run timer. When setting "11_B", ADCOMP0/ADCOMP3 execute compare match only while free-run timer is up count, and ADCOMPD0/ADCOMPD3 execute compare match only while free-run timer is down count.

■ **Buffer Control Register (ADTGBUF0/ADTGBUF1)**



Note:

Be sure to stop the free-run timer before rewriting the BTS bit.

Table 12.4-23 Buffer Control Register (ADTGBUF0/ADTGBUF1)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit6 to bit4	BTS0/BTS3 to BTS2/BTS5: compare register buffer transfer control bits	<ul style="list-style-type: none"> Write "0" to this bit to transfer the compare value to the buffer, at zero detection of free-run timer. Write "1" to this bit to transfer the compare value to the buffer at compare match of free-run timer.
bit3	Undefined bit	<ul style="list-style-type: none"> The read value is indetermine. Writing to this bit has no effect on operation.
bit2 to bit0	BUFX0/BUFX3 to BUFX2/BUFX5: compare register buffer function control bits	<ul style="list-style-type: none"> Write "1" to this bit to disable compare buffering. Write "0" to this bit to enable compare buffering.

12.5 Multi-function Timer Interrupt

The multi-function timer can generate 16-bit free-run timer interrupts, 16-bit output compare interrupts, 16-bit input capture interrupts, and waveform generator interrupts.

■ 16-bit Free-run Timer Interrupt

See Table 12.5-1 for 16-bit free-run timer interrupt control bits and interrupt causes.

Table 12.5-1 16-bit Free-run Timer Interrupt Control Bits and Interrupt Causes

	16-bit free-run timer	
	Compare clear	Zero-detect
Interrupt request flag bit	Timer state control register upper (TCCSH) ICLR: bit9	Timer state control register upper (TCCSH) IRQZF: bit14
Interrupt request enable bit	Timer state control register upper (TCCSH) ICRE: bit8	Timer state control register upper (TCCSH) IRQZE: bit13
Interrupt cause	The 16-bit free-run timer value and compare-clear register (CPCLRH/CPCLRL) match.	16-bit free-run timer goes to "0"

When the 16-bit free-run timer value and compare-clear register (CPCLRH/CPCLRL) match, the timer state control register upper (TCCSH) ICLR: bit9 is set to 1. When interrupt requests are enabled in this state (TCCSH: ICRE bit8=1), the interrupt request is output to the interrupt controller.

When the timer value is "0000_H", the timer state control register upper (TCCSH) IRQZF: bit14 is set to "1". When interrupt requests are enabled in this state (TCCSH IRQZE:bit13=1), the interrupt request is output to the interrupt controller.

Note:

An example of the multi-function timer 0 is shown. For setting of the multi-function timer 1, see Sections "12.4 Multi-function Timer Register" and "12.7 Notes on Using the Multi-function Timer".

■ 16-bit Output Compare Interrupt

See Table 12.5-2 for 16-bit output compare interrupt control bits and interrupt causes.

Table 12.5-2 16-bit Output Compare 0 to 5/6 to 11 Interrupt Control Bits and Interrupt Causes

	16-bit output compare 0/6, 1/7	16-bit output compare 2/8, 3/9	16-bit output compare 4/10, 5/11
Interrupt request flag bit	Compare control register Low (OCSL0/OCSL6) IOP1, IOP0 (bit7, bit6)	Compare control register Low (OCSL2/OCSL8) IOP1, IOP0 (bit7, bit6)	Compare control register Low (OCSL4/OCSL10) IOP1, IOP0 (bit7, bit6)
Interrupt request enable bit	Compare control register Low (OCSL0/OCSL6) IOE1, IOE0 (bit5, bit4)	Compare control register Low (OCSL2/OCSL8) IOE1, IOE0 (bit5, bit4)	Compare control register Low (OCSL4/OCSL10) IOE1, IOE0 (bit5, bit4)
Interrupt cause	The 16-bit free-run timer value and output compare register (OCCPH0/OCCPH6, OCCPH1/OCCPH7, OCCPL0/OCCPL6, OCCPL1/OCCPL7) match.	The 16-bit free-run timer value and output compare register (OCCPH2/OCCPH8, OCCPH3/OCCPH9, OCCPL2/OCCPL8, OCCPL3/OCCPL9) match.	The 16-bit free-run timer value and output compare register (OCCPH4/OCCPH10, OCCPH5/OCCPH11, OCCPL4/OCCPL10, OCCPL5/OCCPL11) match.

When the 16-bit free-run timer value and output compare register (OCCPH0/OCCPH6 to OCCPH5/OCCPH11, OCCPL0/OCCPL6 to OCCPL5/OCCPL11) match, the compare control register low-order (OCSL0/OCSL6, OCSL2/OCSL8, and OCSL4/OCSL10) IOP 1 and IOP 0: bit7 and bit6 are set to 1. When interrupt requests are enabled in this state (OCSL0/OCSL6, OCSL2/OCSL8, and OCSL4/OCSL10 registers IOE1 and IOE0: bit5/bit4 = 11_B), the interrupt request is output to the interrupt controller.

■ 16-bit Input Capture Interrupt

See Table 12.5-3 for 16-bit input capture interrupt control bits and interrupt causes.

Table 12.5-3 16-bit Input Capture 0 to 3/4 to 7 Interrupt Control Bits and Interrupt Causes

	16-bit input capture 0/4, 1/5	16-bit input capture 2/6, 3/7
Interrupt request flag bit	Input capture status control register Low (PICSL01/PICSL45) ICP1/ICP5, ICP0/ICP4 (bit7, bit6)	Input capture status control register Low (ICSL23/ICSL67) ICP3/ICP7, ICP2/ICP6 (bit7, bit6)
Interrupt request enable bit	Input capture status control register Low (PICSL01/PICSL45) ICE1/ICE5, ICE0/ICE4 (bit5, bit4)	Input capture status control register Low (ICSL23/ICSL67) ICP3/ICP7, ICP2/ICP6 (bit5, bit4)
Interrupt cause	Valid edges are detected by IC pins 0/4 and 1/5.	Valid edges are detected by IC pins 2/6 and 3/7.

With 16-bit input capture, when a valid edge is detected by IC pins 0 to 3/4 to 7, the input capture-status control registers (PICSL01/PICSL45 and ICSL23/ICSL67) ICP3/ICP7, ICP2/ICP6, ICP1/ICP5, and ICP0/ICP4: bit7 and bit6 are both set to 11_B. When interrupt requests are enabled in this state (PICSL01/PICSL45 and ICSL23/ICSL67 registers ICE3/ICE7, ICE2/ICE6, ICE1/ICE5, ICE0/ICE4: bit5 and bit4 are both 11_B), interrupt requests are output to the interrupt controller.

■ Waveform Generator Interrupts

See Table 12.5-4 for waveform generator interrupt control bits and interrupt causes.

Table 12.5-4 Waveform Generator Interrupt Control Bits and Interrupt Causes

	Waveform generator	
	16-bit dead timer 0 to 2/3 to 5	DTTI0/DTTI1
Interrupt request flag bit	16-bit dead timer control register High, Low (DTCR0/DTCR3 to DTCR2/DTCR5) TMIF0/TMIF3 to TMIF2/TMIF5 (High is bit12, Low is bit4)	Waveform control register1/2 (SIGCR1/SIGCR2) DTIF (bit6)
Interrupt request enable bit	16-bit dead timer control register High, Low (DTCR0/DTCR3 to DTCR2/DTCR5) TMIE0/TMIE3 to TMIE2/TMIE5 (High is bit11, Low is bit3)	-
Interrupt cause	Underflow in 16-bit dead timer 0 to 2/3 to 5	Low level detected in DTTI.

The waveform generator sets TMIF0 to TMIF2 (upper bit12 and lower bit4) in the 16-bit dead timer control register (DTCR0/DTCR3 to DTCR2/DTCR5) to "1" when an underflow occurs on the 16-bit dead timer and the TMD8 to TMD0 bits in the DTCR0/DTCR3 to DTCR2/DTCR5 registers (upper bit10 to bit8, lower bit2 to bit0) are set to "000_B" or "001_B". When interrupt requests are enabled in this state (DTCR0/DTCR3 to and DTCR2/DTCR5 registers TMIE0 - TMIE2 (upper bit is 11, lower bit is 3) = 1), the interrupt request is output to the interrupt controller.

12.6 Operation of the Multi-function Timer

The operation of the multi-function timer is described below.

■ Operation of the Multi-function Timer

- 16-bit free-run timer

When the 16-bit free-run timer enables count operation, the counter begins counting up from the value set in the timer data register (TCDTH/TCDTL). The count value is used as the standard time of the 16-bit output compare and 16-bit input capture.

- Free-run timer selector

The free-run timer input can be selected for the 16-bit output compare, 16-bit input capture, A/D activation compare. The output compare/input capture can be selected by the free-run timer selector and the A/D activation compare by the compare enable register (ADTGCE).

- 16-bit output compare

16-bit output compare is used to compare the value set in the output compare register with the 16-bit free-run timer value. If a match is detected, the interrupt flag is set, and the output level is reversed.

- 16-bit input capture

16-bit input capture is used to detect specified valid edges.

When a valid edge is detected, the interrupt flag is set, and the value of the 16-bit free-run timer is retrieved, and stored in the input capture data register.

- Waveform generator

The waveform generator generates a variety of waveforms (including dead times) using real-time output (RTO0 to RTO5/RTO6 to RTO11), the input 16-bit PPG timer, and 16-bit dead timer.

- A/D activation compare

An A/D activation is generated when the selected 16-bit free-run timer reaches the specified value.

The channel of the free-run timer can be selected by the register setting.

12.6.1 Operation of 16-bit Free-run Timer

Three 16-bit free-run timer units are provided and these start up-counting from the value set in the timer data register (TCDTH/TCDTL) after a reset completes. The count value is used as the standard time of the 16-bit output compare and 16-bit input capture.

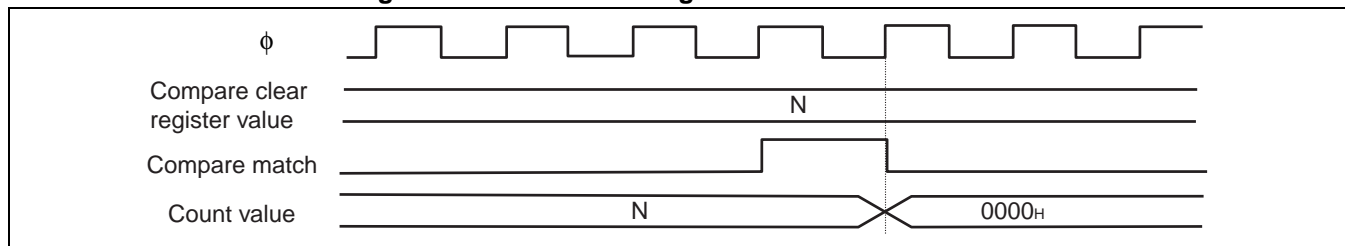
■ Timer Clear

The count value of the 16-bit free-run timer is cleared when one of the following holds:

- A match with the compare-clear register is detected via up-count mode (TCCSL registers MODE: bit5 = 0)
- "1" is written to bit4 (SCLR) of the TCCSL register during operation
- "0000_H" is written to the TCDTH/TCDTL register when operation is halted
- A reset occurs

After a reset, the counter is immediately cleared. The counter is cleared, synchronized with the count timing, when cleared by software or when a match with the compare clear register occurs

Figure 12.6-1 Clear Timing of 16-bit Free-run Timer



Note:

The count value of the 16-bit free-run timer is not cleared even if "1" is written to bit4 (SCLR) of the TCCSL register while the timer is stopped.

■ Timer Mode

The following modes can be selected for the 16-bit free-run timer.

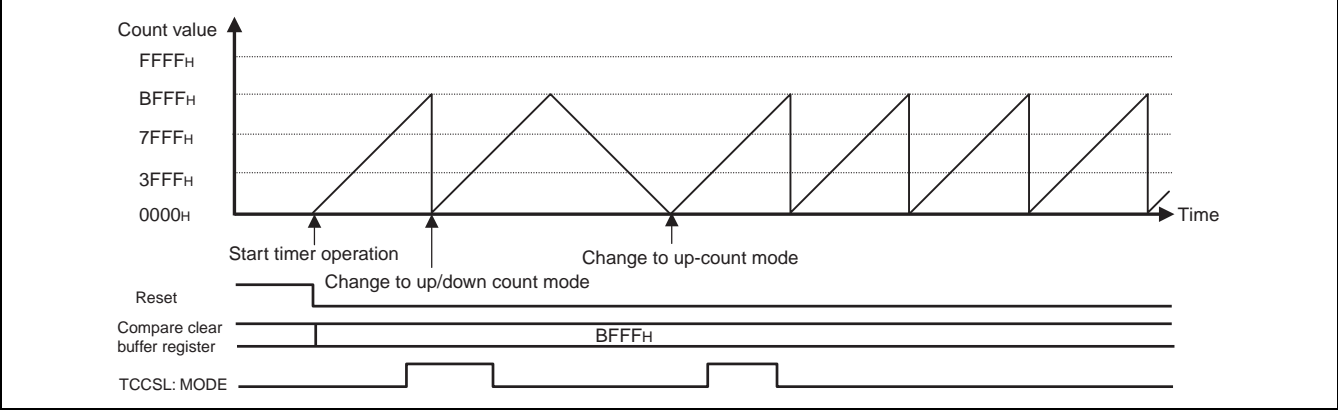
- Up-count mode (TCCSL registers MODE: bit5 = 0)
- Up/down count mode (TCCSL registers MODE: bit5=1)

In the up-count mode, the counter starts counting from the preset timer data register (TCDTH/TCDTL) and continues counting up until the count value matches the value in the compare clear register (CPCLR_H/CPCLR_L). Then, the counter is cleared to "0000_H" and the counter restarts counting up.

In the up/down count mode, the counter starts counting from the preset timer data register (TCDTH/TCDTL) and continues counting up until the count value matches the value in the compare clear register (CPCLR_H/CPCLR_L). Then, counting changes from the up-count mode to the down-count mode, the counter value performs counting down until it reaches to "0000_H", and the counter restarts counting up.

You can write to the mode bit (bit5 (MODE) in the TCCSL register) at any time regardless of whether the timer is running or halted. The value written to the bit when the timer is running is stored in a buffer and the actual count mode does not change until the timer value reaches "0000_H".

Figure 12.6-2 Change Timer Mode during Timer Operation



■ Compare Clear Buffer

The compare-clear register (CPCLR_H/CPCLR_L) has a buffer feature that can be enabled or disabled. When the buffer function is enabled (bit7 (BFE) = 1 in the TCCSL register), data written to the compare clear buffer register (CPCLR_{BH}/CPCLR_{BL}) is transferred to the CPCLR_H/CPCLR_L register when zero is detected on the 16-bit free-run timer. When the buffer function is disabled (bit7 (BFE) in the TCCSL register = 0) and data is written directly to the CPCLR_H/CPCLR_L register.

Figure 12.6-3 Operation in Up-count Mode when Compare Clear Buffer is Disabled
(TCCSL Register's BFE:bit7 = 0)

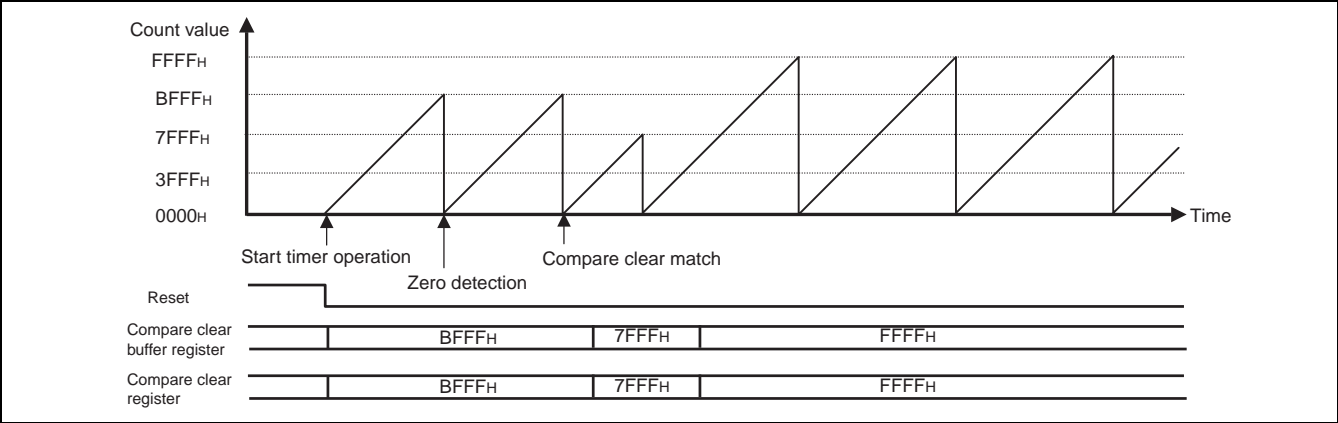


Figure 12.6-4 Operation in Up-count Mode when Compare Clear Buffer is Enabled
(TCCSL Register's BFE:bit7=1)

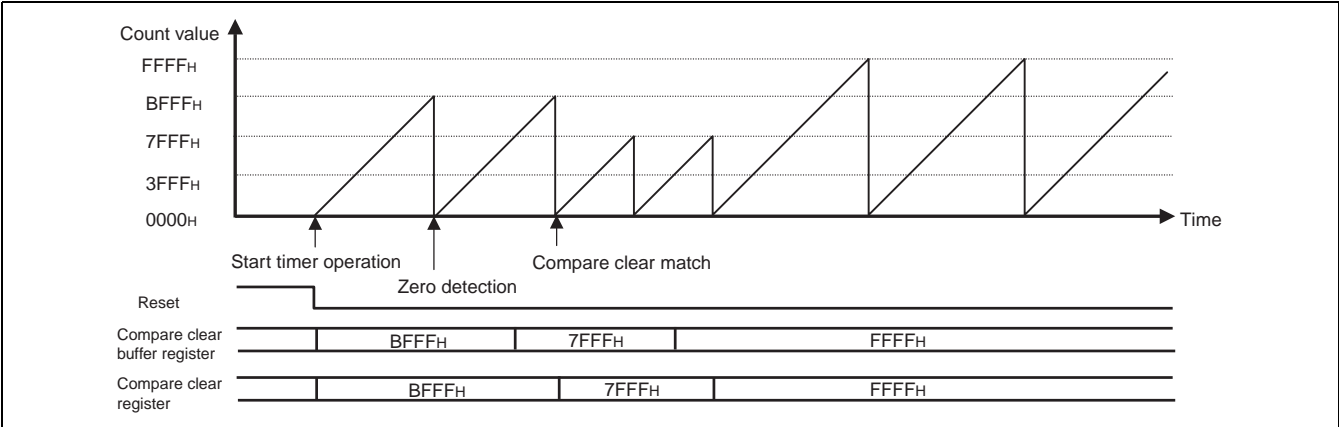
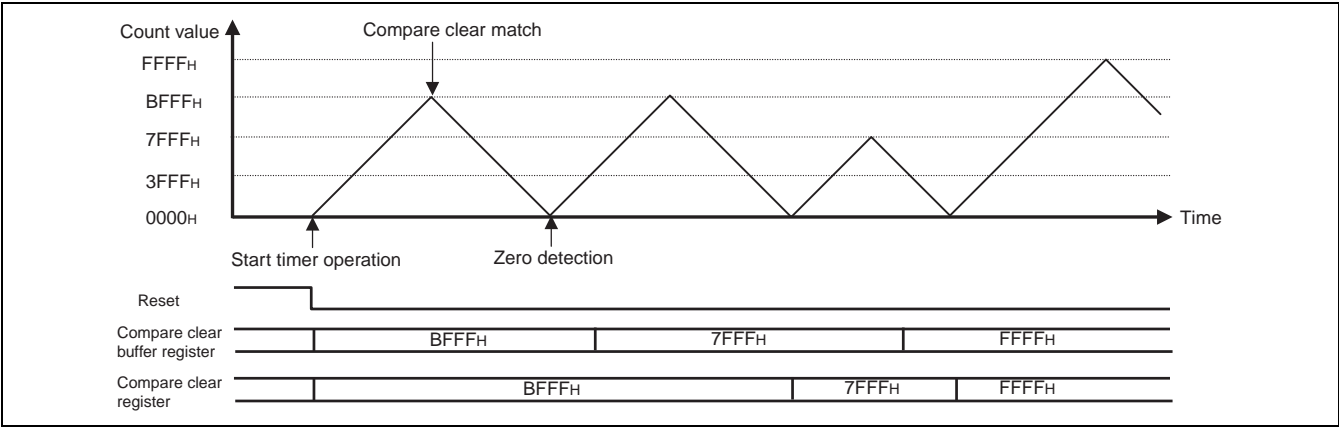


Figure 12.6-5 Operation in Up/Down Count Mode when Compare Clear Buffer is Enabled
(TCCSL Register's BFE:bit 7 = 1)



■ Timer Interrupt

The 16-bit free-run timer can generate the following two interrupts.

- Compare clear interrupt
- Zero-detect interrupt

Compare-clear interrupts are generated when the timer value matches the value of the compare-clear register.

Zero-detect interrupts are generated when the timer value reaches "0000_H".

Note:

A software clear (setting bit4 (SCLR) in the TCCSL register = 1) does not generate a zero-detect interrupt.

Figure 12.6-6 Interrupt Generated in Up-count Mode (TCCSL Register MODE:bit5 = 0)

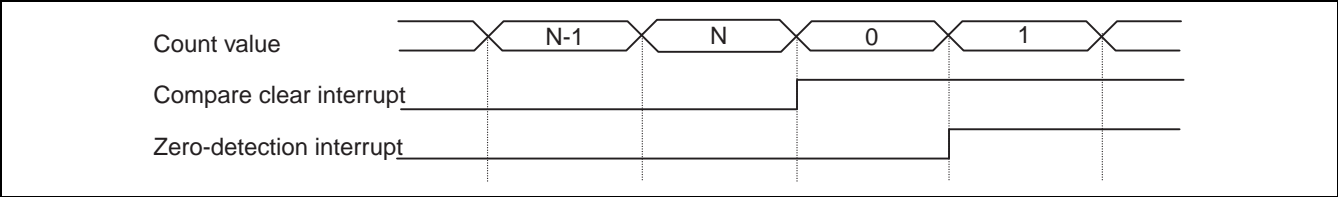
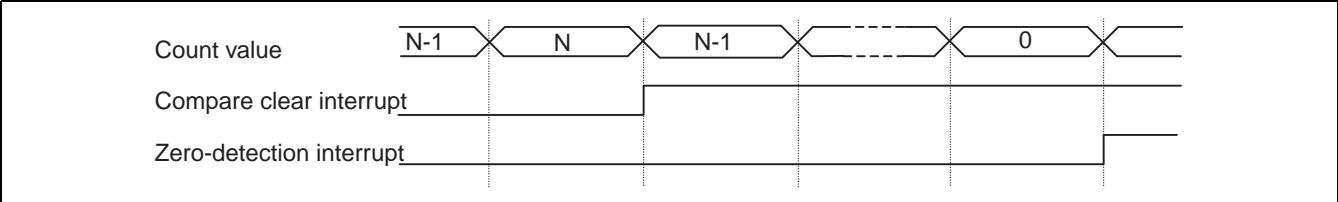


Figure 12.6-7 Interrupt Generated in Up/Down-count Mode (TCCSL Register MODE:bit5 = 1)



■ Interrupt Mask Function

It is possible to mask either one of zero detection interrupt or compare match interrupt or both.

The following describes how to mask either one of interrupt.

- It is possible to mask interrupt requests by setting the TCCSH registers MSI2 - MSI0 : bit12-bit10. MSI2 bit - MSI0 bit are a 3-bit reload down register that reload when the count value reaches "000_B". The count value can also be loaded by writing directly to the MSI2 to MSI0 bits. The mask count is the value set in MSI2 bit - MSI0 bit. When MSI2 bit - MSI0 bit are "000_B", interrupt request are not masked.
- The interrupt request depends on the count mode (TCCSL registers MODE : bit5). In up-count mode, it is only possible to mask compare-clear interrupts, and zero-detect interrupts are generated each time "0" is detected. In up/down count mode, it is only possible to mask zero-detect interrupts.

The following explains how to mask both interrupt requests.

- Both interrupt can be masked when the free-run timer is only in up/down count mode and in MODE2 in the TCCSM register =1 and MODE in the TCCSL register = 1.
- MSI2 to MSI0 bits in the TCCSH register is used for zero detect interrupt mask and MSI5 to MSI3 bits in the TCCSM register is used for compare clear interrupt mask.

Note:

A software clear (setting bit4 (SCLR) in the TCCSL register = 1) does not generate a 0-detect interrupt.

Figure 12.6-8 Compare Clear Interrupt Masked in Up-count Mode

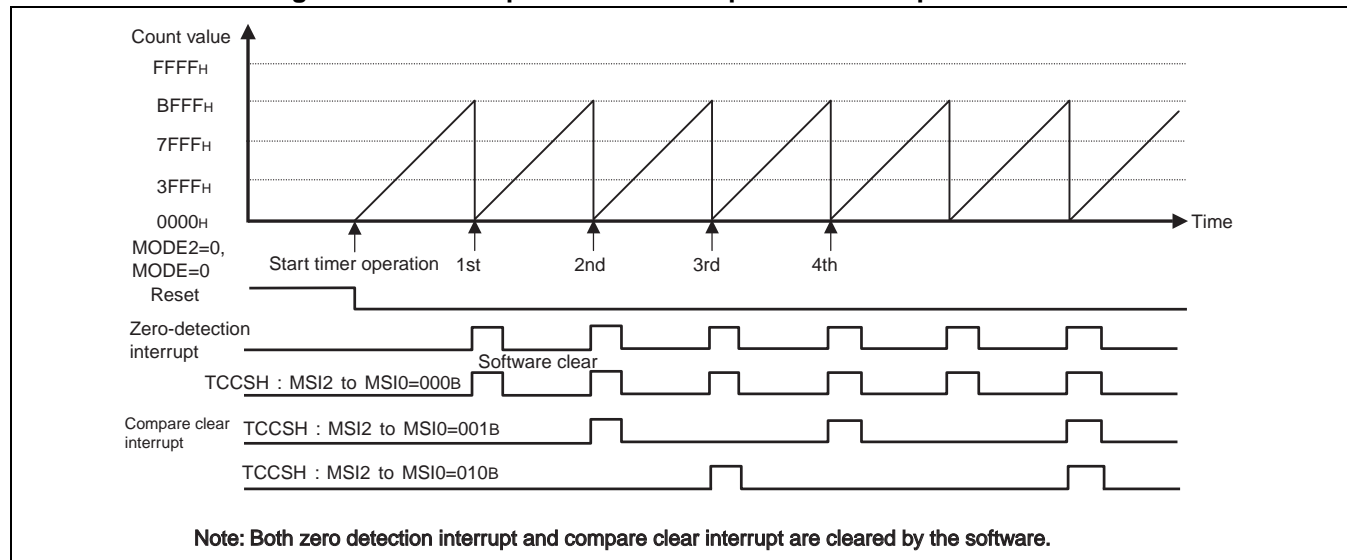


Figure 12.6-9 Zero Detection Interrupt Masked in Up/Down Count Mode

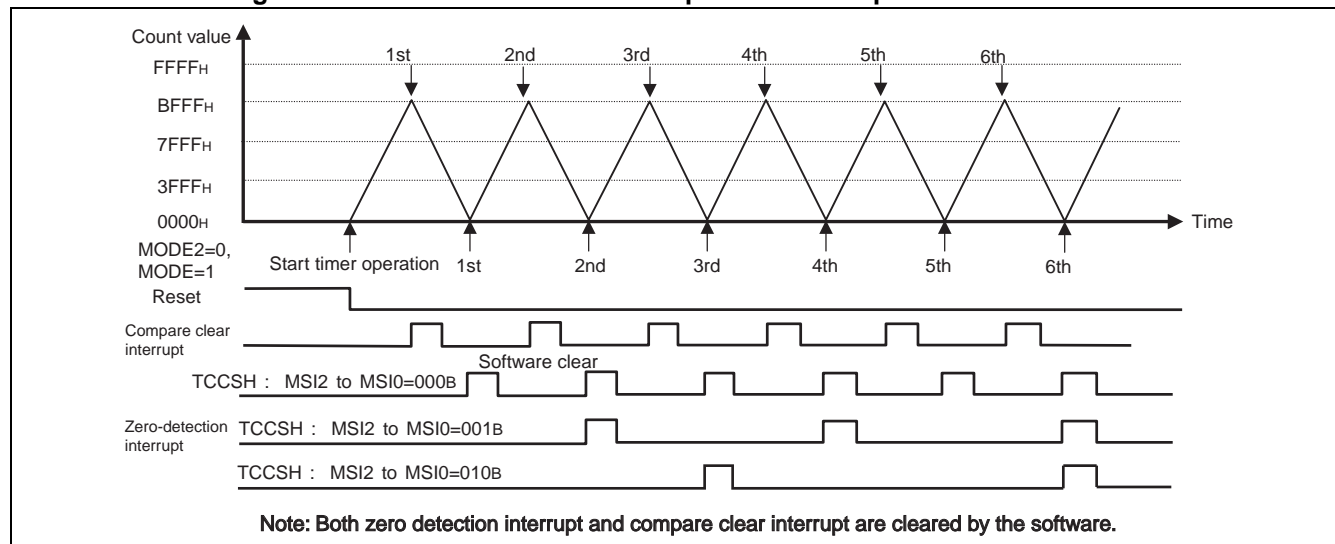
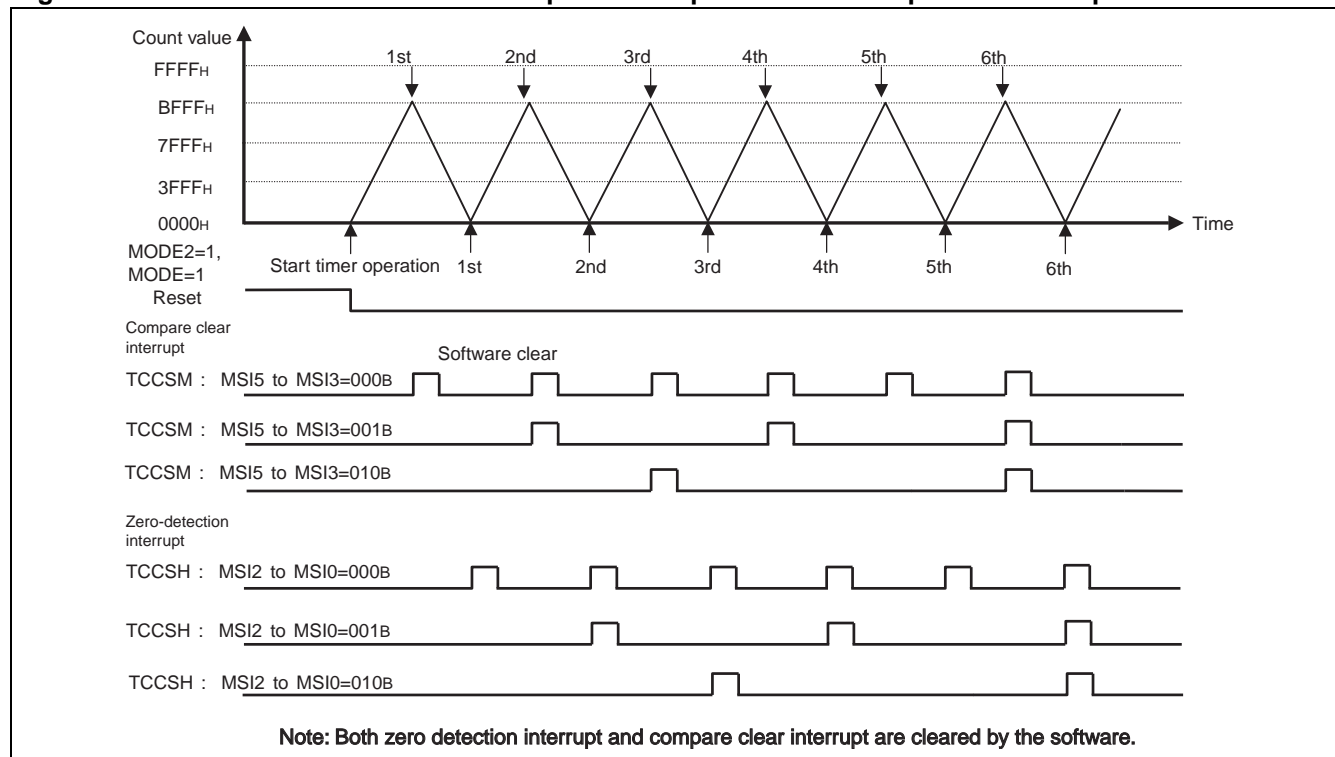


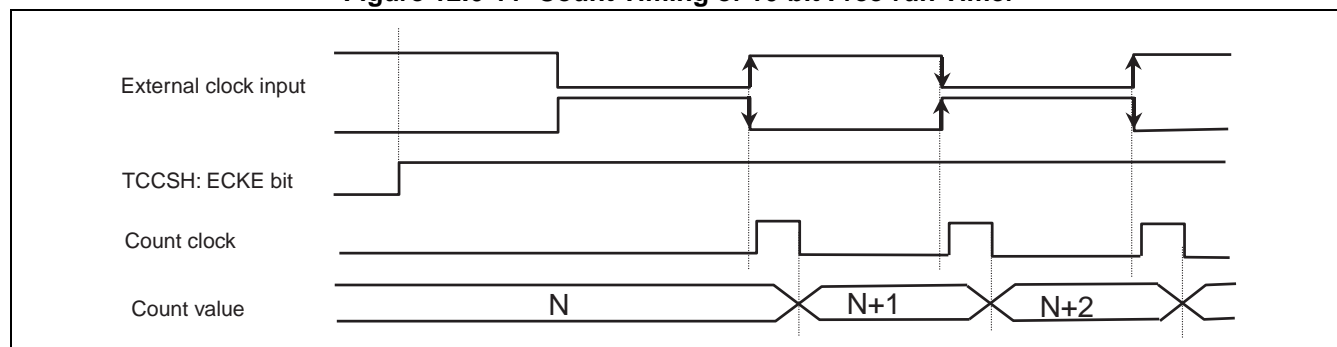
Figure 12.6-10 Both Zero Detection interrupt and Compare Clear Interrupt Masked in Up/Down Count Mode



■ Selected External Count Clock

The 16-bit free-run timer is incremented based on the input clock (peripheral clock (CLKP) or external clock). When the external clock is selected in external clock mode (bit15 (ECKE) in the TCCSH register = 1), the 16-bit free-run timer starts counting up on rising edges if the initial value of the external input is "1". Subsequently, it counts up on both edges. If the initial value of external input is "0", it starts counting up on a falling edge. Subsequently, it counts up on both edges.

Figure 12.6-11 Count Timing of 16-bit Free-run Timer



Note:

The external clock input is counted on both edges.

■ A/D Activation by Free-run Timer

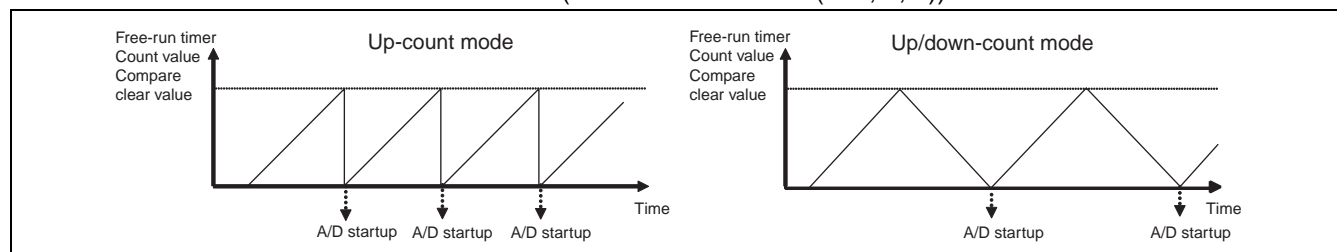
It is possible to activate A/D upon a compare match or zero detection of the 16-bit free-run timer. The activation trigger can be selected by means of the A/D trigger cause selection bits (SEL0 to SEL2: bit4 to bit6) of the A/D trigger control register (ADTRGC).

It is possible to halt A/D activation signals, even upon compare match or zero-detection, via the A/D trigger output enable/disable bits (AD0E to AD2E: bit0 to bit2) of the A/D trigger control register (ADTRGC).

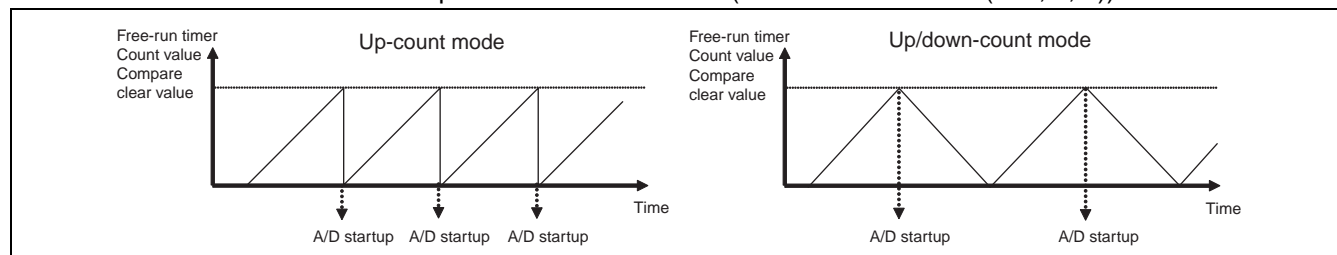
Note:

If A/D activation signal output is enabled from disabled state, and in the meantime, an activation trigger compare match or 0-detection is arrived, the A/D activation signal will be asserted immediately after the A/D activation signal output was enabled.

● Activate A/D when zero is detected (ADTRGC: SELn = 0 (n=0, 1, 2))



● Activate A/D when a compare clear match occurs (ADTRGC: SELn = 1 (n=0, 1, 2))



12.6.2 Operation of Free-run Timer Selector

Free-run timer selector is used to select the input free-run timers to output compare and input capture unit.

In MB91480 series, there are two multi-function timers (MFT0, MFT1). They contains 6 free-run timers, 12 output compare, 8 input capture.

■ Operation of Free-run Timer Selector in MB91470 Series

In MB91470 series, there is one multi-function timer. Multi-function timer contains 3 free-run timers, 6 output compare, 4 input capture.

The mapping information is shown in Table 12.6-1 and Table 12.6-2.

Table 12.6-1 Registers for Mapping Free-run Timers

Resource	Register
OCU0	FRS0[2:0]
OCU1	FRS0[6:4]
OCU2	FRS1[10:8]
OCU3	FRS1[14:12]
OCU4	FRS2[2:0]
OCU5	FRS2[6:4]
ICU0	FRS3[2:0]
ICU1	FRS3[6:4]
ICU2	FRS4[10:8]
ICU3	FRS4[14:12]

Table 12.6-2 Set Value List

Set value	Free-run timer
000 _B	free-run timer0 (Default status of multifunction timer0)
001 _B	free-run timer1
010 _B	free-run timer2
Others	Prohibited (Operation not guarantee)

■ Operation of Free-run Timer Selector in MB91480 Series

In MB91480 series, there are two multi-function timers. They contains 6 free-run timers, 12 output compare, 8 input capture.

The mapping information is shown in Table 12.6-3 and Table 12.6-4.

Table 12.6-3 Registers for Mapping Free-run Timers

Resource	Register	Description
OCU0	FRS0[2:0]	Multi function timer0
OCU1	FRS0[6:4]	
OCU2	FRS1[10:8]	
OCU3	FRS1[14:12]	
OCU4	FRS2[2:0]	
OCU5	FRS2[6:4]	
ICU0	FRS3[2:0]	
ICU1	FRS3[6:4]	
ICU2	FRS4[10:8]	
ICU3	FRS4[14:12]	
OCU6	FRS5[2:0]	Multi function timer1
OCU7	FRS5[6:4]	
OCU8	FRS6[10:8]	
OCU9	FRS6[14:12]	
OCU10	FRS7[2:0]	
OCU11	FRS7[6:4]	
ICU4	FRS8[2:0]	
ICU5	FRS8[6:4]	
ICU6	FRS9[10:8]	
ICU7	FRS9[14:12]	

Table 12.6-4 Set Value List

Set value	Free-run timer
000 _B	free-run timer0 (Default status of multifunction timer0)
001 _B	free-run timer1
010 _B	free-run timer2
011 _B	free-run timer3 (Default status of multifunction timer1)
100 _B	free-run timer4
101 _B	free-run timer5
Others	Prohibited (Operation not guarantee)

Note:

Be sure to stop the free-run timer before configuration of the free-run timer selection register.

12.6.3 Operation of 16-bit Output Compare

Output compare is used to compare the value set in the compare clear register with the 16-bit free-run timer value. If a match is detected, the interrupt flag is set, and the output level is reversed.

If the free-run timer is in up/down count mode, match signals are ignored when the count peak and compare register value match.

■ 16-bit Output Compare Operation (Inversion Mode, MOD1x = 0)

- Compare operation can be performed on each channel (compare control register higher-orders (OCSH1/OCSH7, OCSH3/OCSH9, and OCSH5/OCSH11) CMOD: bit12 = 0)

Figure 12.6-12 Example of Output Waveform when the Initial Output is "0" and Compare Register 0 and Compare Register 1 are Used Separately (Free-run Timer in Up-count Mode)

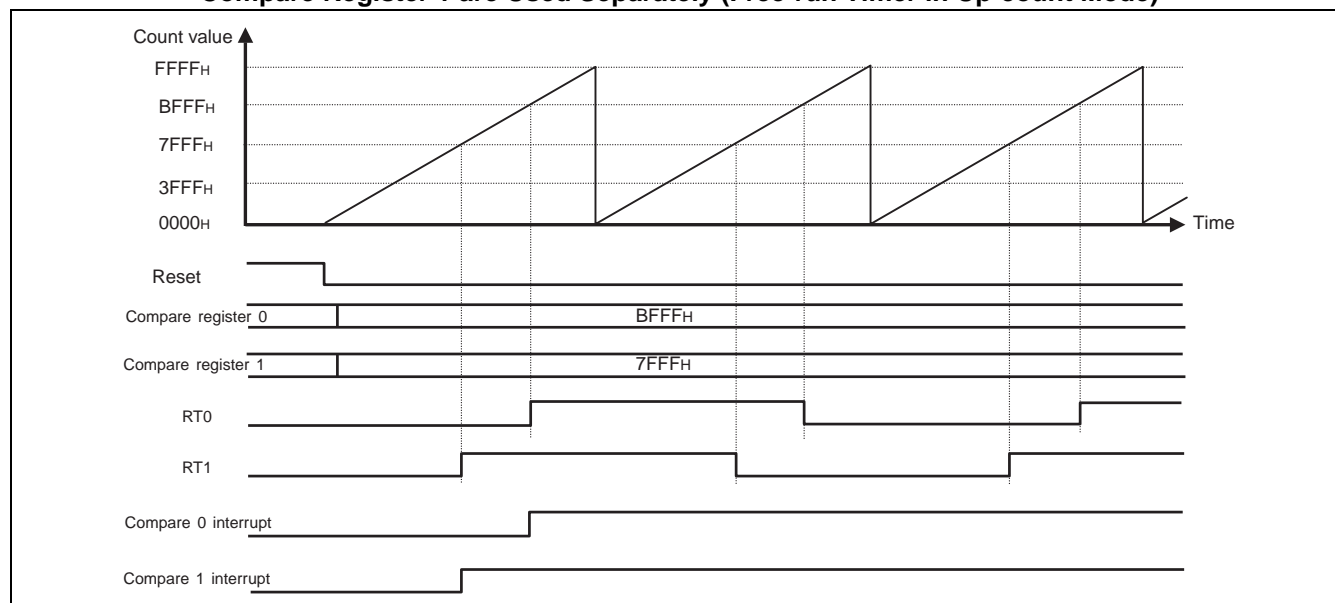
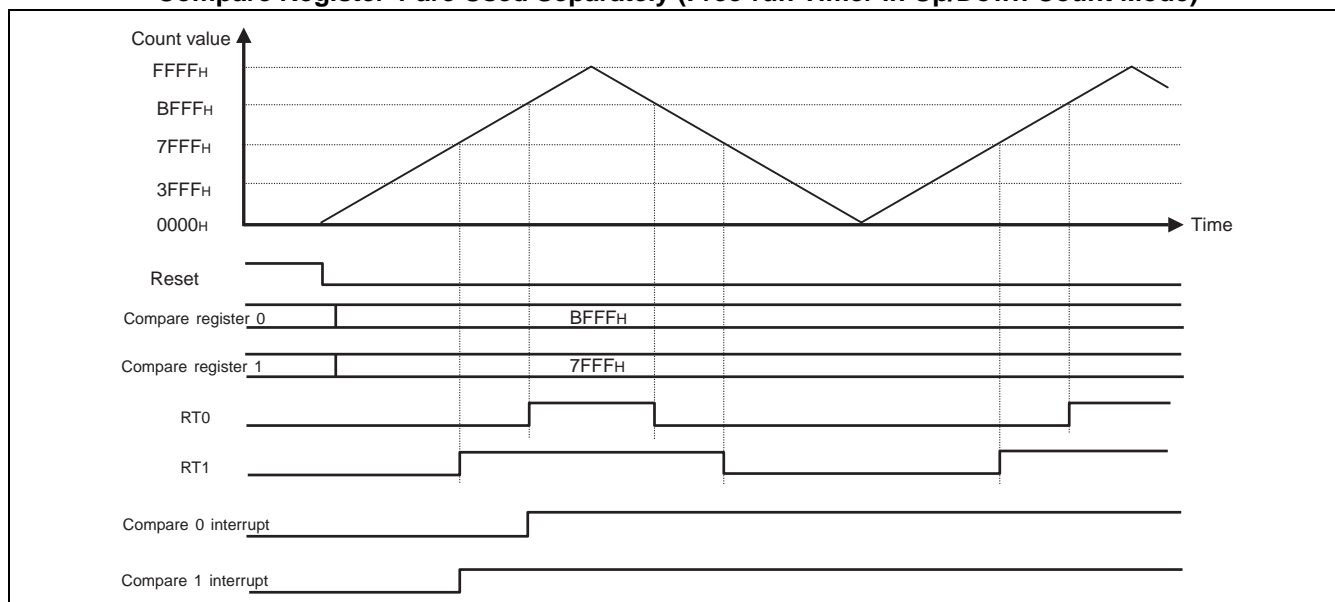


Figure 12.6-13 Example of Output Waveform when the Initial Output is "0" and Compare Register 0 and Compare Register 1 are Used Separately (Free-run Timer in Up/Down Count Mode)



- The output level can be changed using a single compare register (compare control register higher-orders (OCSH1/OCSH7, OCSH3/OCSH9, and OCSH5/OCSH11) CMOD: bit12 = 1)

Figure 12.6-14 Example of Output Waveform when the Initial Output is "0" and Compare Register 0 and Compare Register 1 are Used Together (Free-run Timer in Up-count Mode)

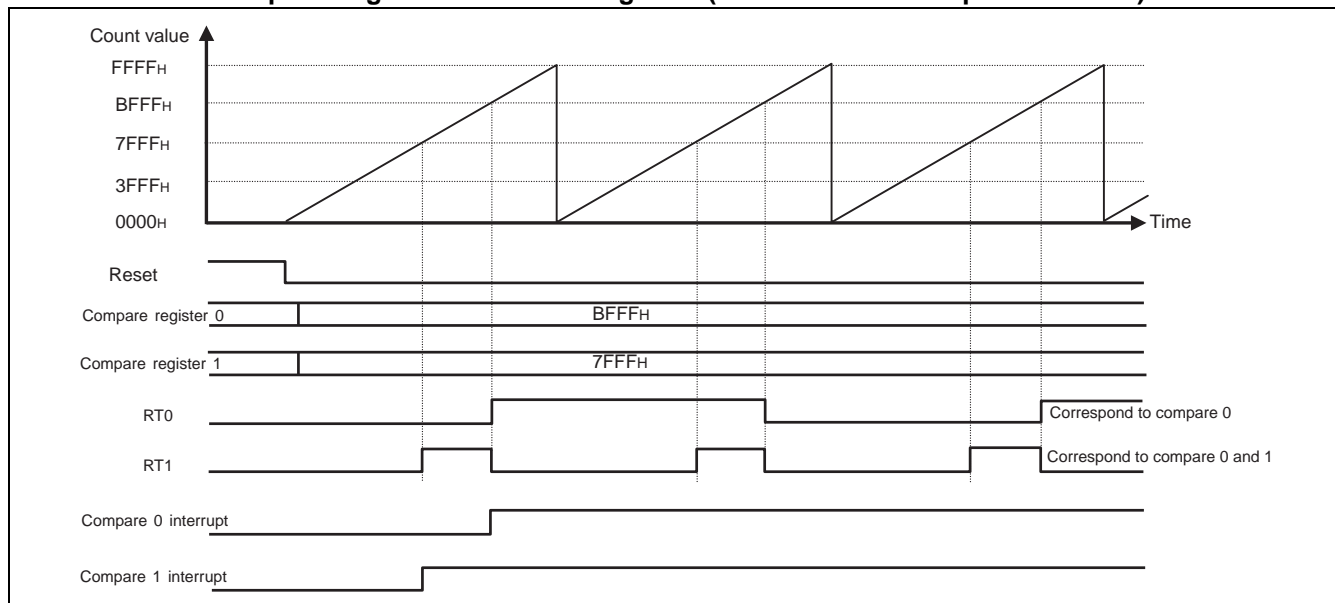
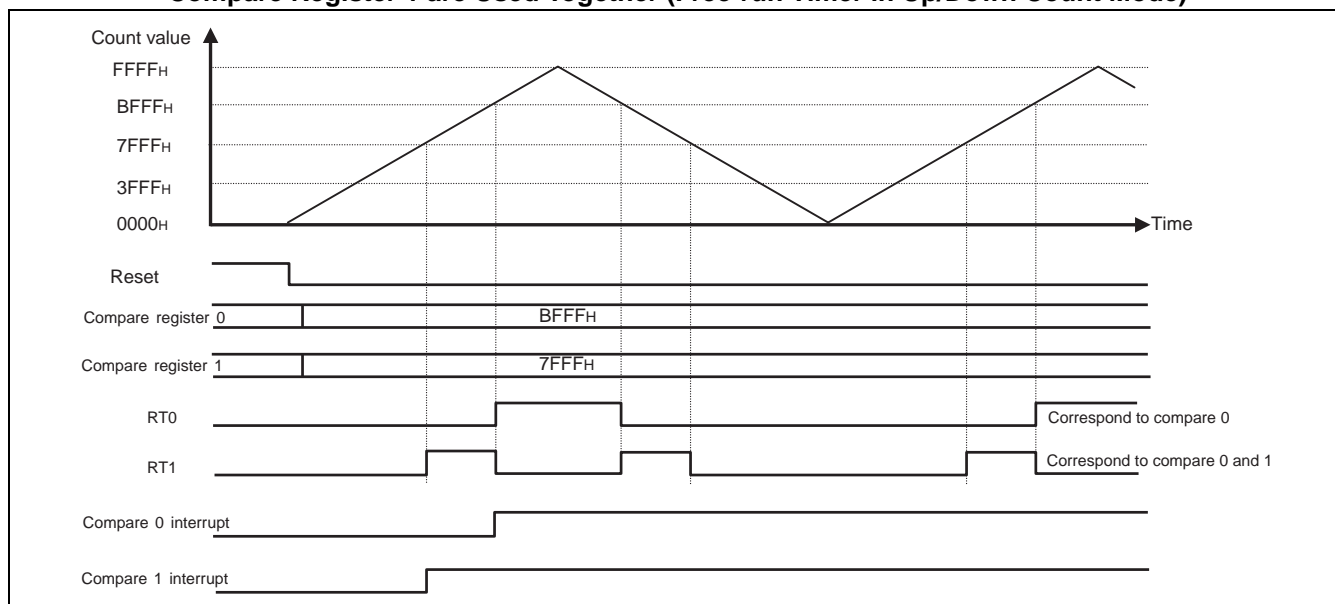
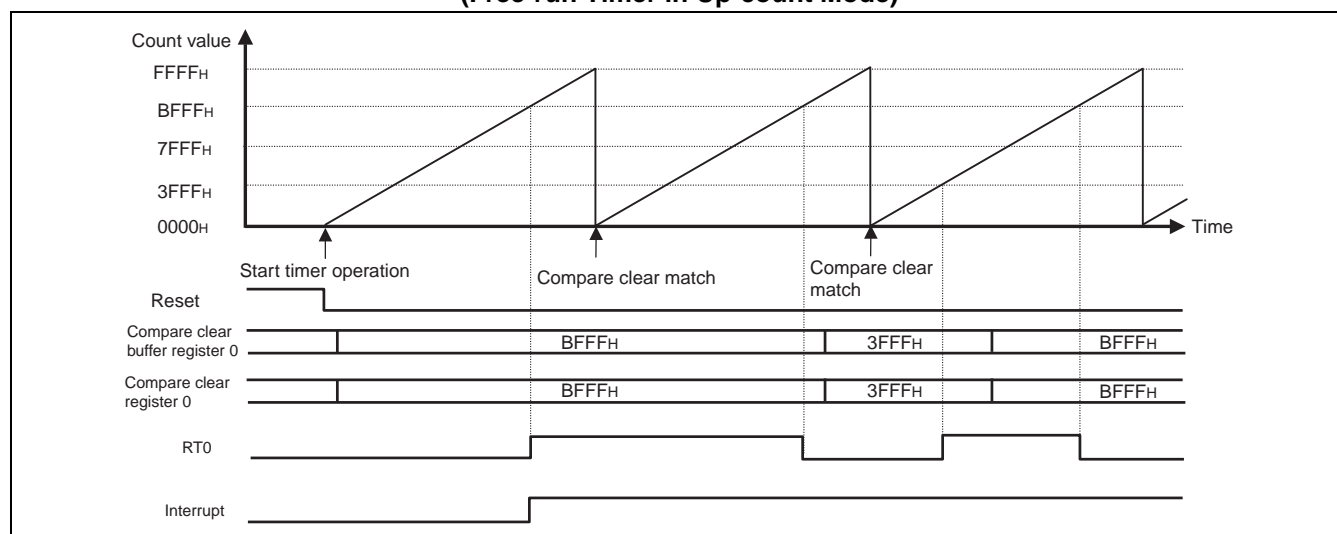


Figure 12.6-15 Example of Output Waveform when the Initial Output is "0" and Compare Register 0 and Compare Register 1 are Used Together (Free-run Timer in Up/Down Count Mode)



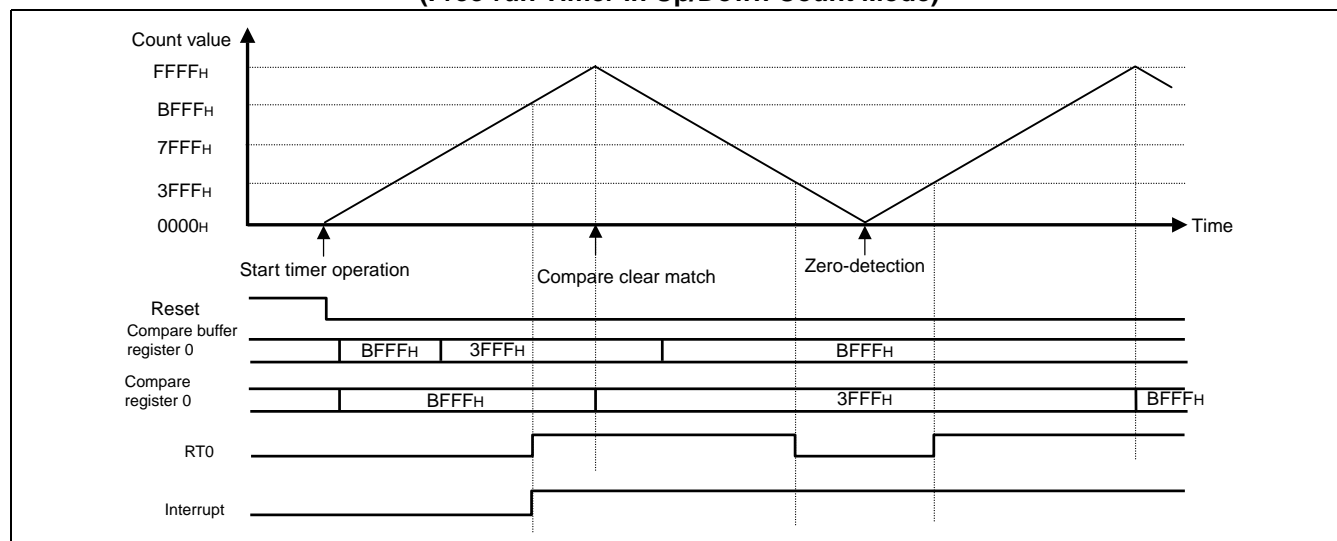
- Output level when compare buffer is disabled

**Figure 12.6-16 Example of Output Waveform when the Compare Buffer is Disabled
(Free-run Timer in Up-count Mode)**

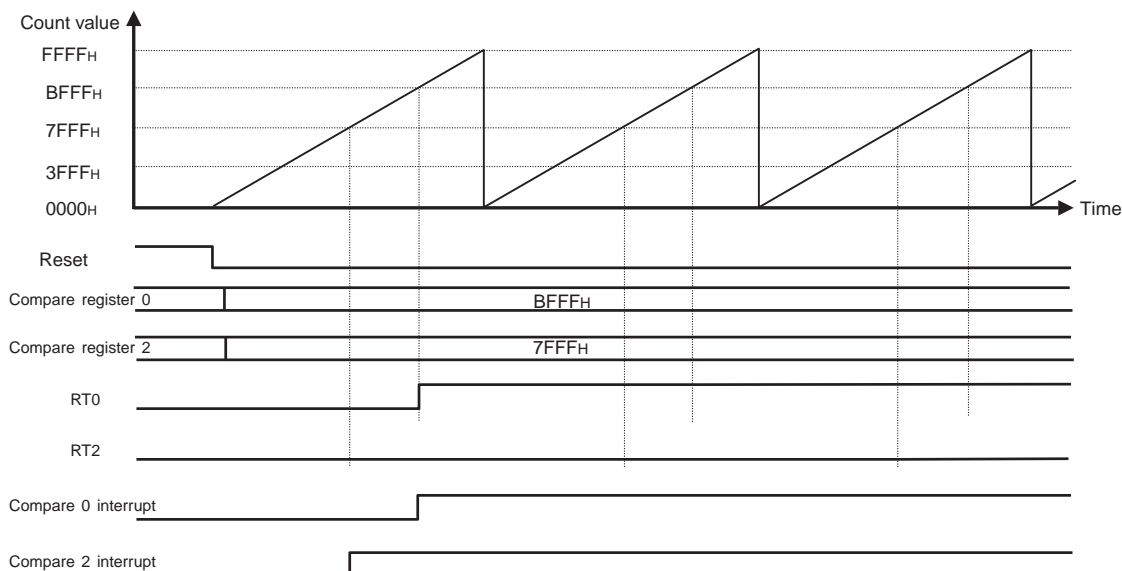


- Output level when compare buffer is selected, and a compare clear match occurs:

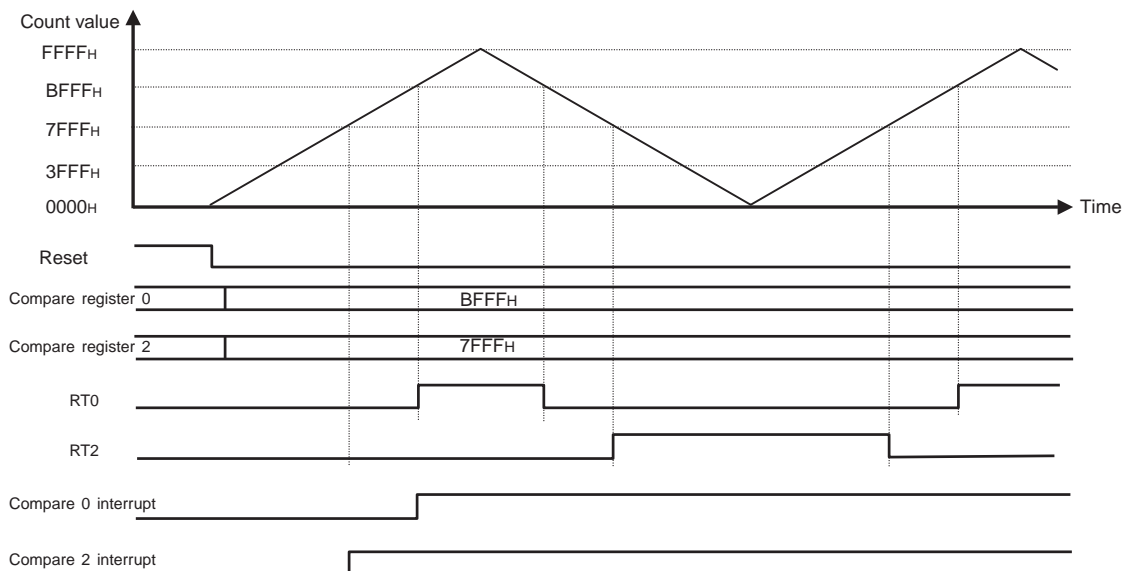
**Figure 12.6-17 Example of Output Waveform when the Compare Buffer is Valid
(Free-run Timer in Up/Down Count Mode)**



■ 16-bit Output Compare Operation (Set/Reset Mode, MOD1x = 1)



Up count of ch.0 is set, down count of ch.0 is reset
Up count of ch.2 is reset, down count of ch.2 is set
Note: Keeps "1" if ch.0 compare clear match occurs. ch.2 is always "0".



Up count of ch.0 is set, down count of ch.0 is reset
Up count of ch.2 is reset, down count of ch.2 is set

■ 16-bit Output Compare Timing

When a match occurs between the free-run timer and compare register, the output compare generates the compare match signal to reverse the output and generates an interrupt. When a compare match occurs, output is reversed in synchronization with the count timing of the counter.

Figure 12.6-18 Compare Register Interrupt Timing

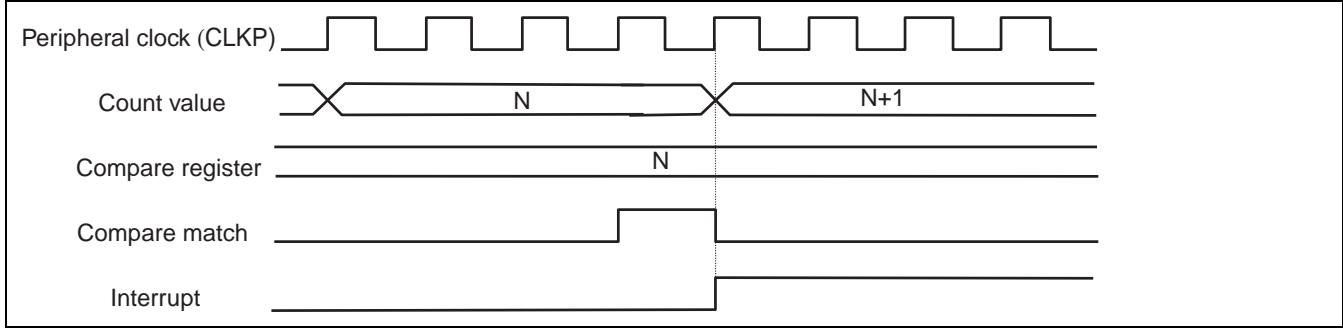
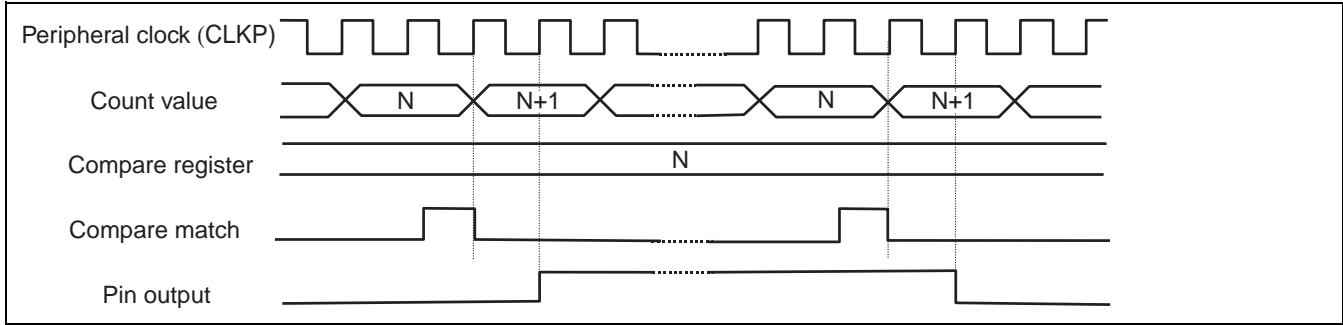


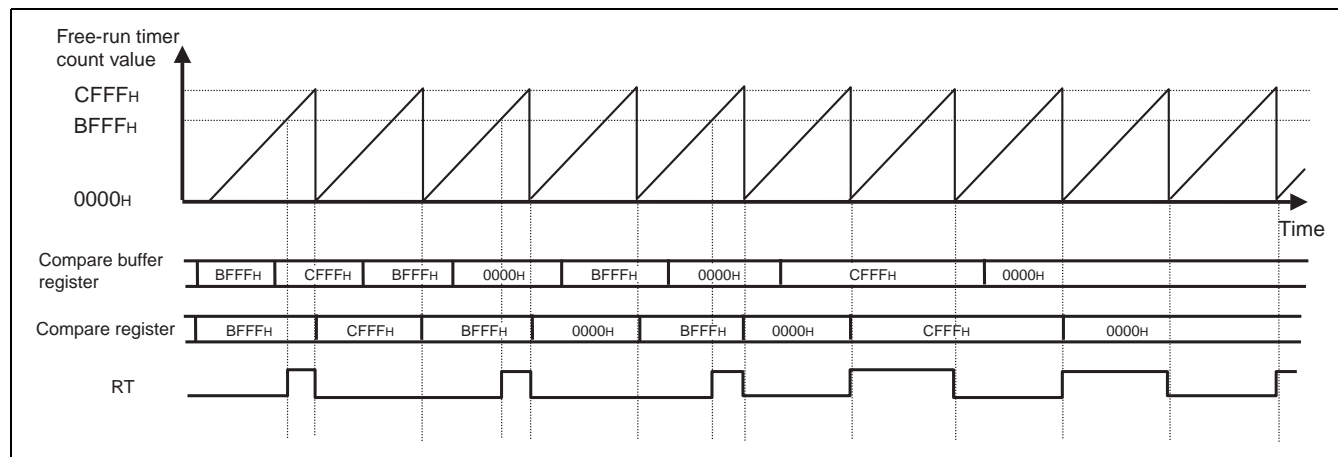
Figure 12.6-19 Pin Output Change Timing



■ Operation of 16-bit Output Compare and Free-run Timer

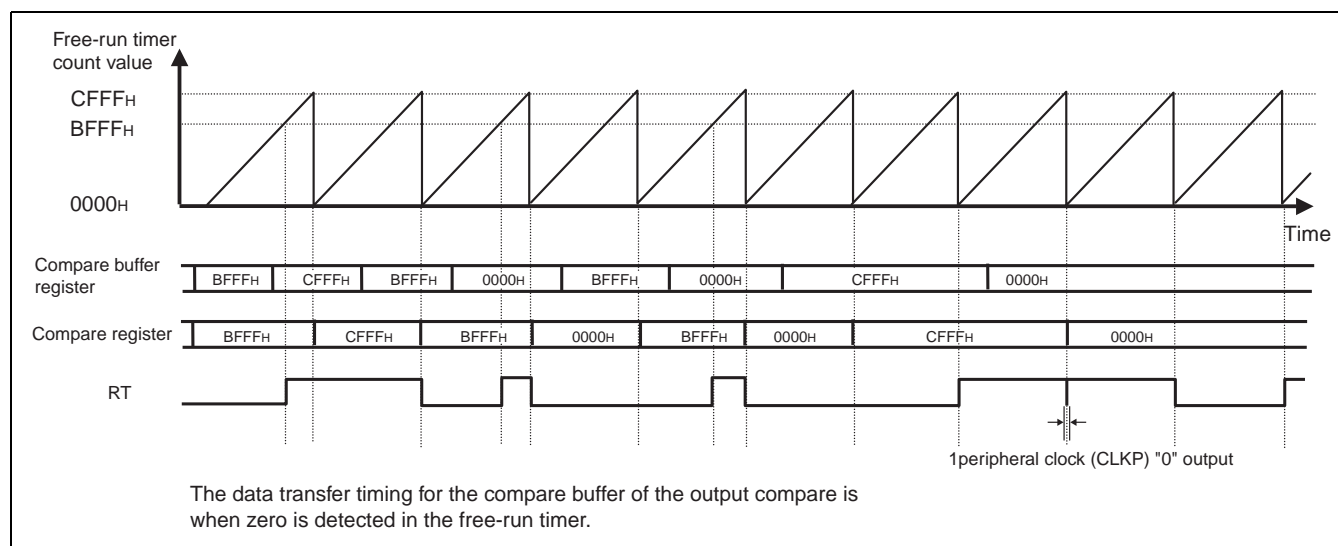
● When the free-run timer up-counts

The data transfer timing for the compare buffer of the output compare is when a compare clear match occurs with the free-run timer.



● When the free-run timer up-counts

The data transfer timing for the compare buffer of the output compare is when zero is detected in the free-run timer.

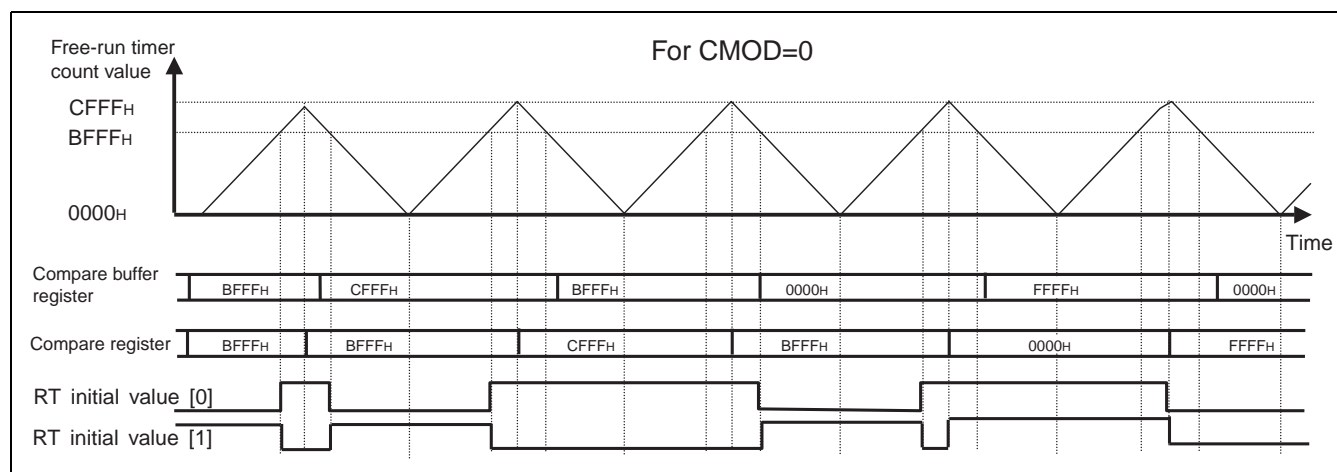


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when a compare clear match occurs with the free-run timer.
- When the output compare output mode is set to reverse the output when a match occurs

Notes:

- RT is set to "1" when the compare register is set to "0000_H" regardless of the free-run timer count value (reset to "0" when CMOD = 1).
- RT is reset to "0" when the compare register is set to "FFFF_H", regardless of the free-run timer count value (set to "1" when CMOD = 1).
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

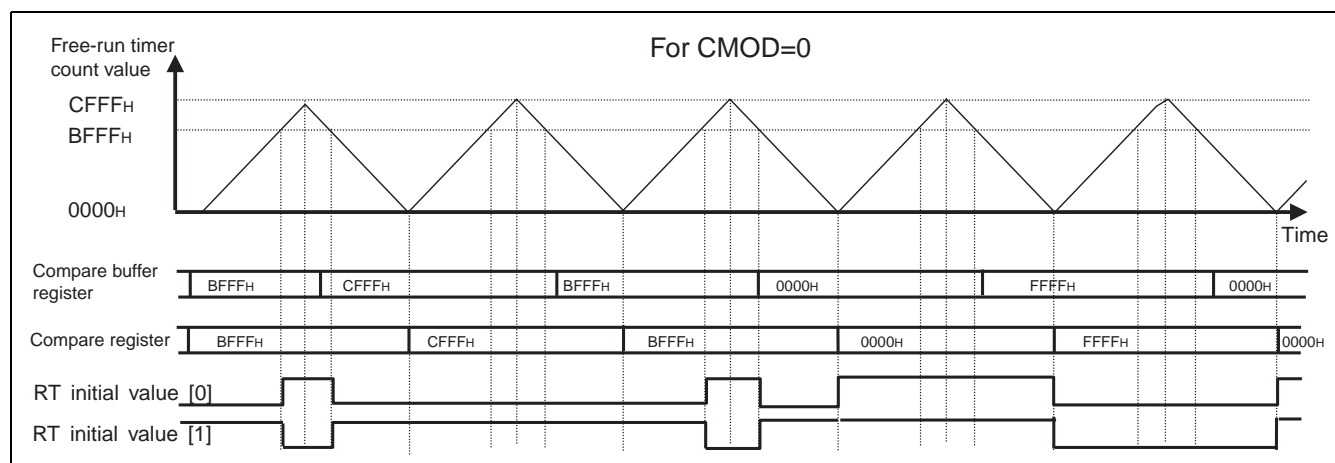


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when zero is detected in the free-run timer.
- When the output compare output mode is set to reverse the output when a match occurs

Notes:

- RT is set to "1" when the compare register is set to "0000_H" regardless of the free-run timer count value (reset to "0" when CMOD = 1).
- RT is reset to "0" when the compare register is set to "FFFF_H", regardless of the free-run timer count value (set to "1" when CMOD = 1).
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

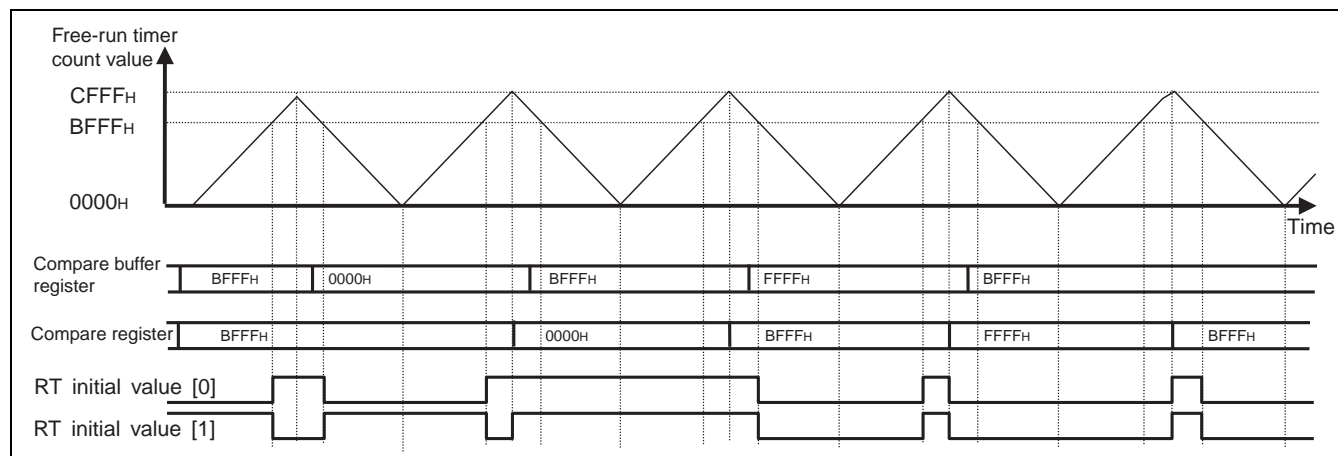
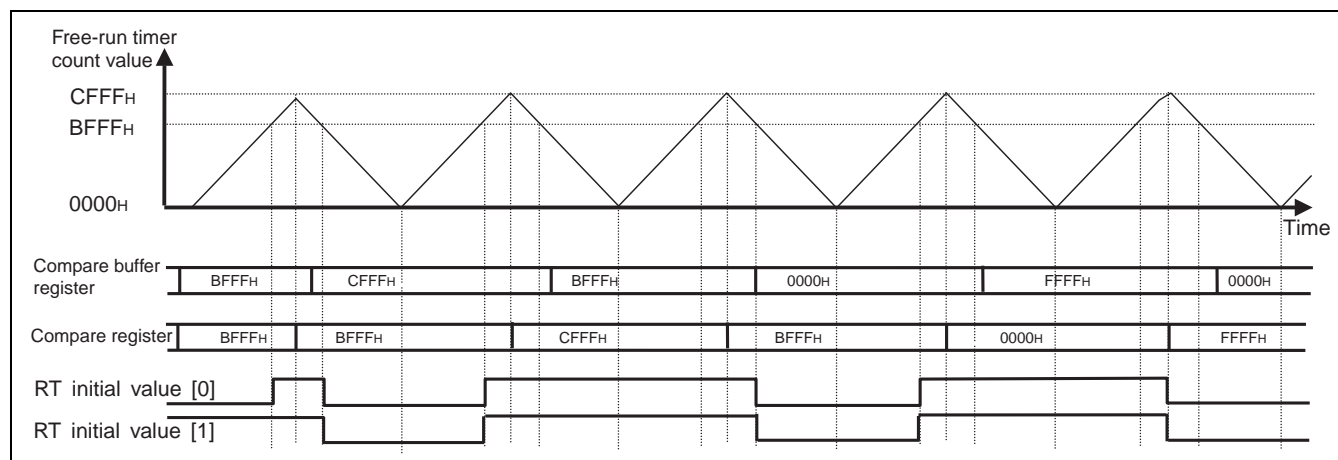


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when a compare clear match occurs with the free-run timer.
- The output compare output is set to "1" for a match due to an up-count and reset to "0" for a match due to a down-count (CMOD=0).

Notes:

- RT is set to "1" when the compare register is set to "0000_H" regardless of the free-run timer count value. When the compare register value is changed from "0000_H" to an arbitrary value of "0001_H" to "FFFE_H", RT is "1".
- RT is reset to "0" when the compare register is set to "FFFF_H", regardless of the free-run timer count value. When the compare register value is changed from "FFFF_H" to an arbitrary value of "0001_H" to "FFFE_H", RT becomes "1".
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

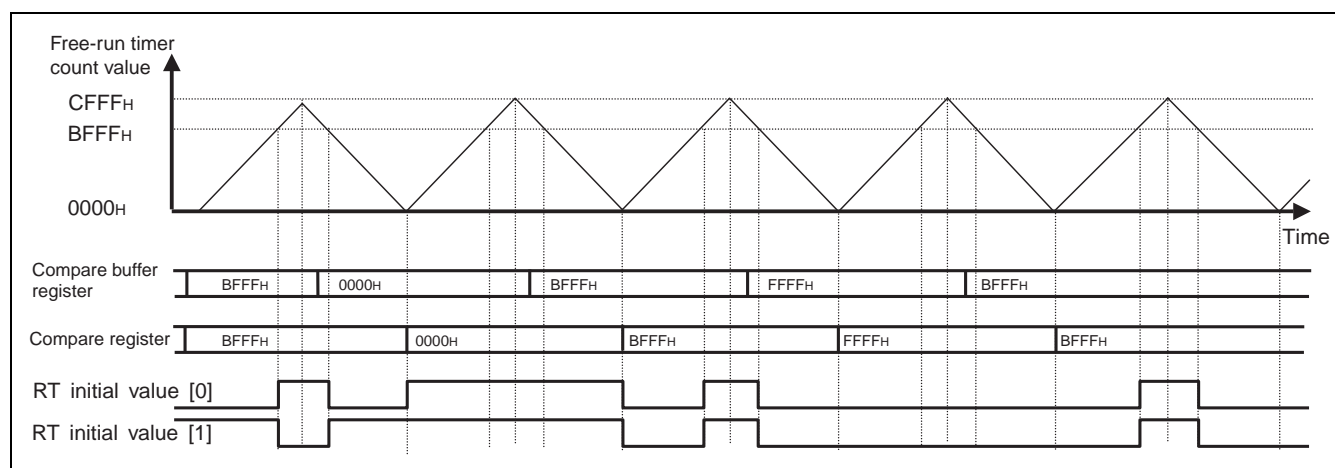
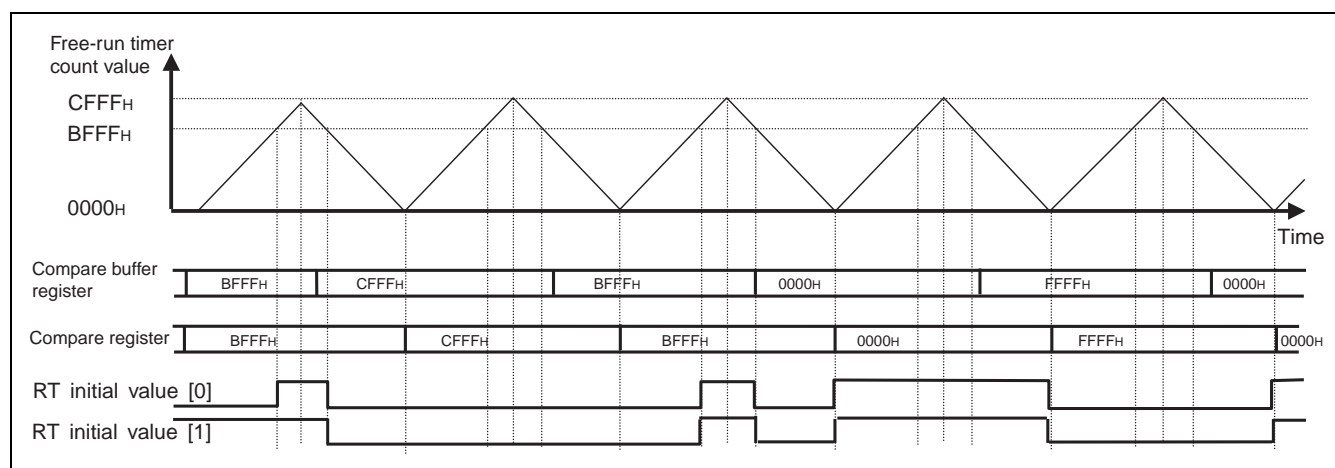


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when zero is detected in the free-run timer.
- The output compare output is set to "1" for a match due to an up-count and reset to "0" for a match due to a down-count (CMOD=0).

Notes:

- RT is set to "1" when the compare register is set to "0000_H" regardless of the free-run timer count value. When the compare register value is changed from "0000_H" to an arbitrary value of "0001_H" to "FFFE_H", RT becomes "0".
- RT is reset to "0" when the compare register is set to "FFFF_H", regardless of the free-run timer count value. When the compare register value is changed from "FFFF_H" to an arbitrary value of "0001_H" to "FFFE_H", RT is "0".
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

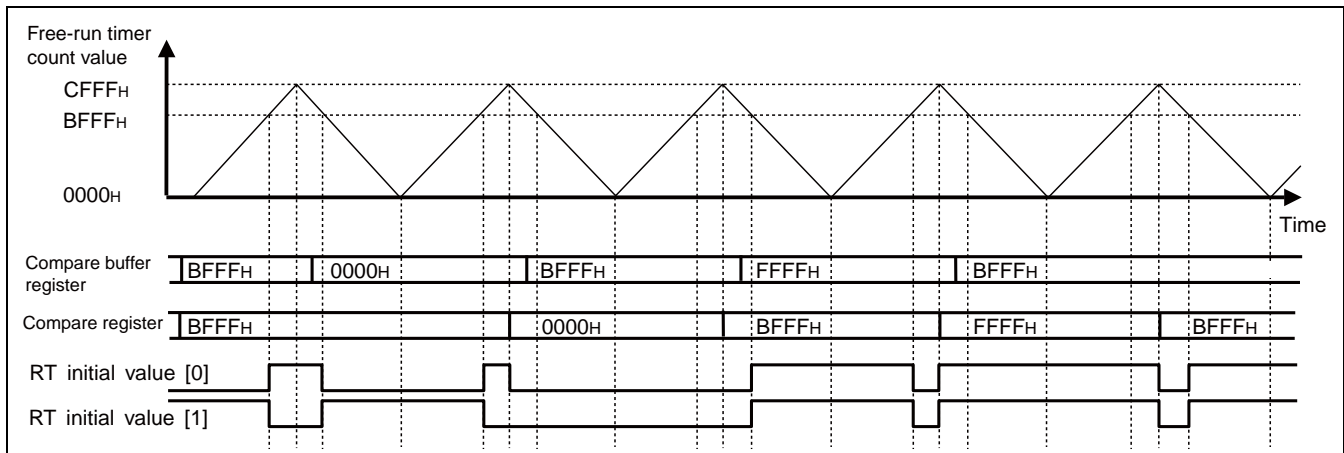
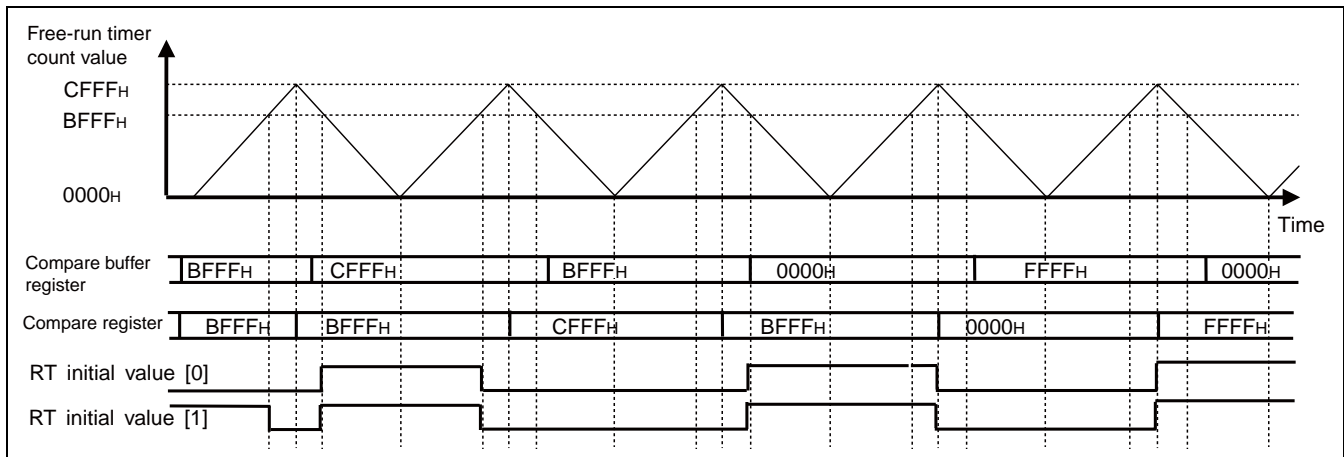


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when a compare clear match occurs with the free-run timer.
- Output compare output when up-count match is reset to 0, and down-count match is set to "1" (CMOD=1):

Notes:

- RT is reset to "0" when the compare register is set to "0000_H", regardless of the free-run timer count value. When the compare register value is changed from "0000_H" to an arbitrary value of "0001_H" to "FFFE_H", RT is "0".
- RT is set to "1" when the compare register is set to "FFFF_H" regardless of the free-run timer count value. When the compare register value is changed from "FFFF_H" to an arbitrary value of "0001_H" to "FFFE_H", RT becomes "0".
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

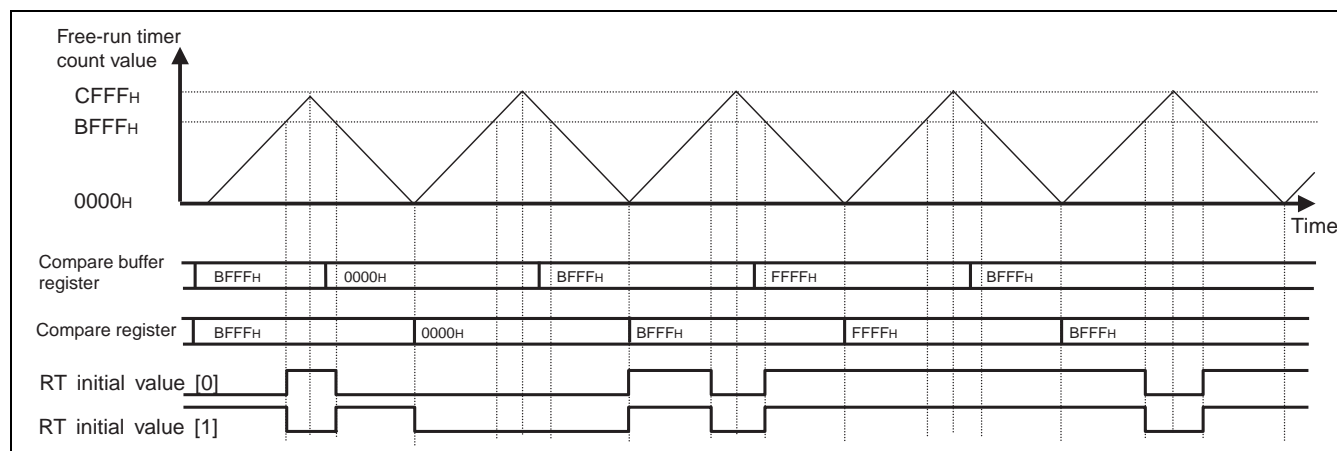
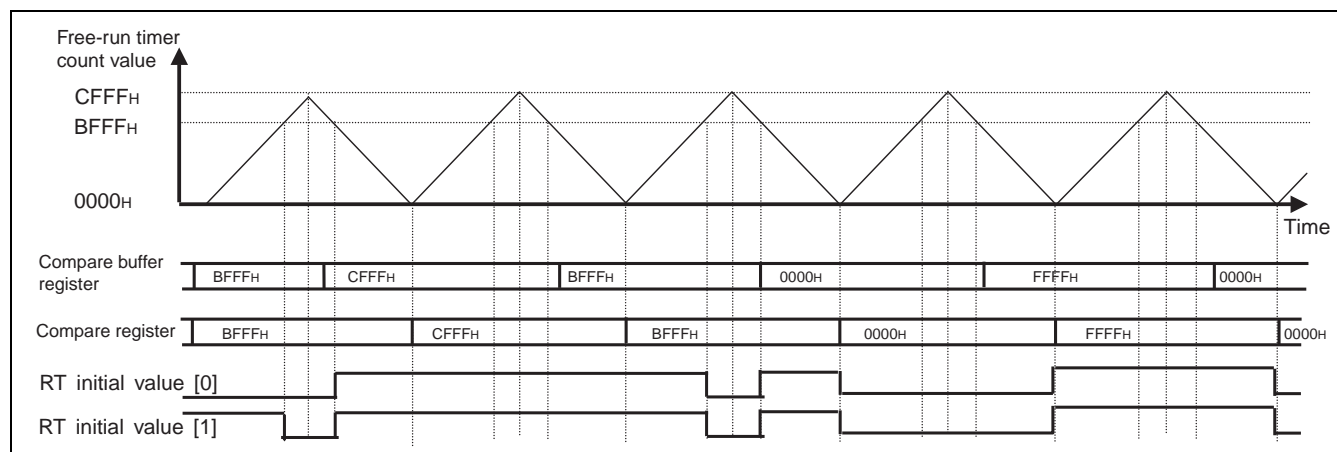


● When the free-run timer up/down counts

- The data transfer timing for the compare buffer of the output compare is when zero is detected in the free-run timer.
- Output compare output when up-count match is reset to "0", and down-count match is set to "1" (CMOD=1):

Notes:

- RT is reset to "0" when the compare register is set to "0000_H", regardless of the free-run timer count value. When the compare register value is changed from "0000_H" to an arbitrary value of "0001_H" to "FFFE_H", RT becomes "1".
- RT is set to "1" when the compare register is set to "FFFF_H" regardless of the free-run timer count value. When the compare register value is changed from "FFFF_H" to an arbitrary value of "0001_H" to "FFFE_H", RT is "1".
- No comparison is performed when the value of the compare clear register in the free-run timer is the same as the compare register in the output compare. In this case, RT is reset to "0" when both the compare clear register and compare register are set to "FFFF_H", regardless of the free-run timer count value.

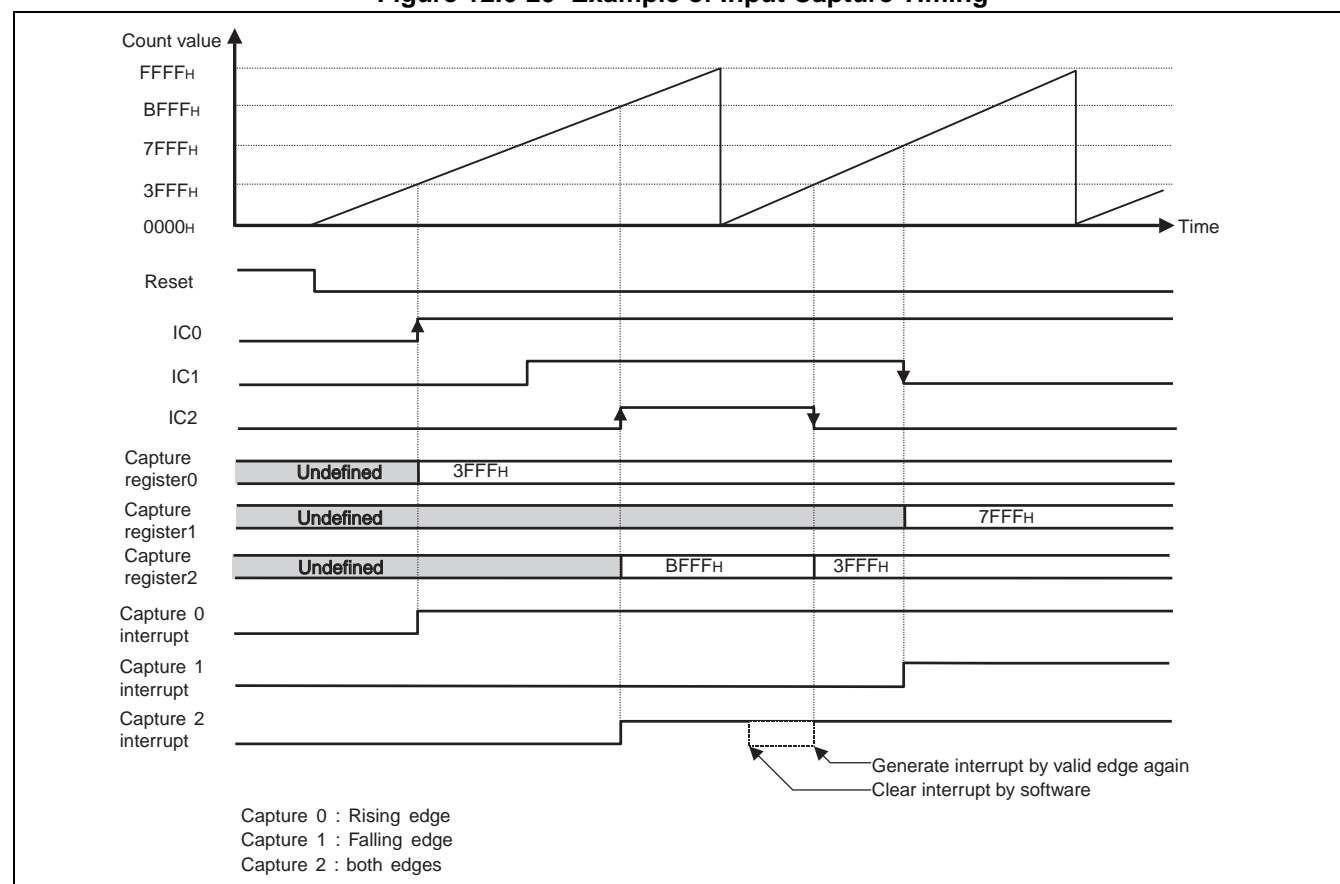


12.6.4 16-bit Input Capture Operation

Input capture is used to detect specified valid edges. When a valid edge is detected, the interrupt flag is set, and the value of the 16-bit free-run timer is loaded into the capture register.

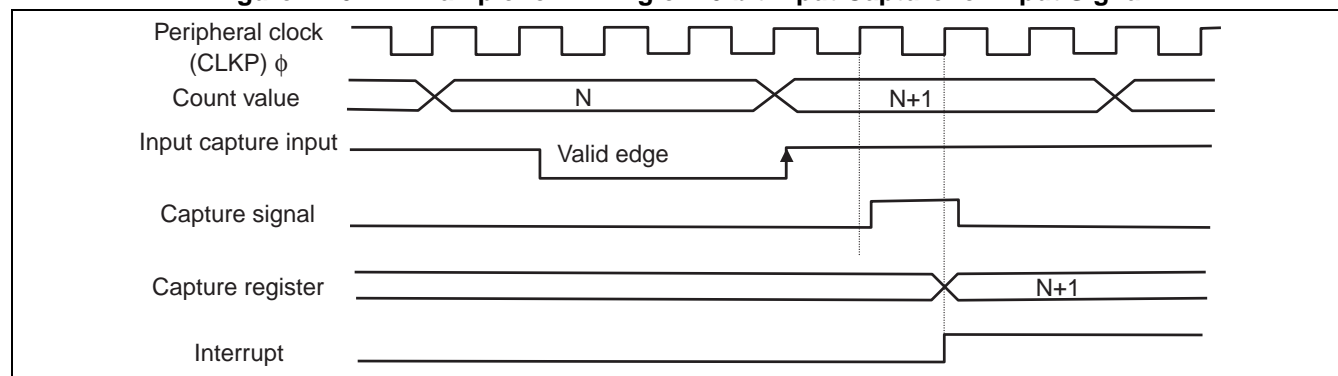
■ 16-bit Input Capture Operation

Figure 12.6-20 Example of Input Capture Timing



■ Input Timing for 16-bit Input Capture

Figure 12.6-21 Example for Timing of 16-bit Input Capture for Input Signal



12.6.5 Waveform Generator Operation

The waveform generator can generate a variety of waveforms (including dead times) using real-time output (RTO0 to RTO5), the 16-bit PPG timer 0/2/4, and 16-bit dead timers 0, 1, and 2.

■ Output Status of RTO0 to RTO5 and GATE

Table 12.6-5 RTO0 to RTO5/GATE Output Status and Bit Settings (1 / 2)

TMD2	TMD1	TMD0	GTENx	PGENx	RTOx	GATE
0	0	0	X	X	Real-time output RTx (16-bit output compare output)	Always "0"
0	0	1	X	0	Real-time output RTx (16-bit output compare output)	(RTx and GTENx) *3
0	0	1	0	1	RTx outputs a PPG0/PPG2/PPG4 pulse for duration H. *1	Always "0"
0	0	1	1	1	RTx outputs the PPG0/PPG2/PPG4 pulse activated by the GATE signal for duration H.	(RT0 RT1 RT2 RT3 RT4 RT5)
0	1	0	X	0	16-bit dead timer 0 starts on a rising edge on RT0 or RT1 and "H" is output until the 16-bit dead timer 0 underflows.	"H" is output during timer operation *4
			X		16-bit dead timer 1 starts on a rising edge on RT2 or RT3 and "H" is output until the 16-bit dead timer 1 underflows.	
			X		16-bit dead timer 2 starts on a rising edge on RT4 or RT5 and "H" is output until the 16-bit dead timer 2 underflows.	
0	1	0	0	1	16-bit dead timer 0 starts on a rising edge on RT0 or RT1 and the PPG0/PPG2/PPG4 pulse is output until the 16-bit dead timer 0 underflows. *1	Always "0"
			0		16-bit dead timer 1 starts on a rising edge on RT2 or RT3 and the PPG0/PPG2/PPG4 pulse is output until the 16-bit dead timer 1 underflows. *1	
			0		16-bit dead timer 2 starts on a rising edge on RT4 or RT5 and the PPG0/PPG2/PPG4 pulse is output until the 16-bit dead timer 2 underflows. *1	

Table 12.6-5 RTO0 to RTO5/GATE Output Status and Bit Settings (2 / 2)

TMD2	TMD1	TMD0	GTENx	PGENx	RTOx	GATE
0	1	0	1	1	16-bit dead timer 0 starts on a rising edge on RT0 or RT1 and the PPG0/PPG2/PPG4 pulse activated by the GATE signal is output until the 16-bit dead timer 0 underflows.	"H" is output during timer operation ^{*4}
			1		16-bit dead timer 1 starts on a rising edge on RT2 or RT3 and the PPG0/PPG2/PPG4 pulse activated by the GATE signal is output until the 16-bit dead timer 1 underflows.	
			1		16-bit dead timer 2 starts on a rising edge on RT4 or RT5 and the PPG0/PPG2/PPG4 pulse activated by the GATE signal is output until the 16-bit dead timer 2 underflows.	
1	0	0	X	X	RT1 generates a non-overlapping signal. ^{*2}	Always "0"
			X		RT3 generates a non-overlapping signal. ^{*2}	
			X		RT5 generates a non-overlapping signal. ^{*2}	
1	1	1	0	X	Setting disabled	-
1	1	1	1	X	Setting disabled	-
The others					Always "0"	Always "0"

*1: PPG needs to select the channel used from PPG0/PPG2/PPG4, and to be started beforehand.

*2: In order to generate a non-overlapping signal, first select 2-channel mode (compare control registers higher-order (OCSH1, OCSH3, and OCSH5) CMOD: bit12 = 1 for RT1, RT3, and RT5.

*3: The GATE signal is generated from the RTx whose GTENx bit is set to "1".

*4: The GATE signal is generated while the timer activated by the RTx whose GTENx bit is set to "1" is operating. If more than one GATEx bit is set to "1", the GATE signal is the OR of the signals of each of the operating timers.

Note:

RTO0 and RTO1 are controlled by the 16-bit dead timer control register higher-order (DTCR0) TMD2 to TMD0: bit10 to bit8, RTO2 and RTO3 are controlled by the lower-order of the DTCR1 register TMD5 to TMD3: bit2 to bit0, and RTO4 and RT5 are controlled by the higher-order of the DTCR2 register TMD8 to TMD6: bit10 to bit8.

■ PPG Output Control

PPG output to RTO0 pin to RTO5 pin can be enabled by means of the PPG output control/input capture-status control registers higher-order (PICSH01) PGEN5 to PGEN0: bit15 to bit10.

■ PPG Output by Gate Trigger

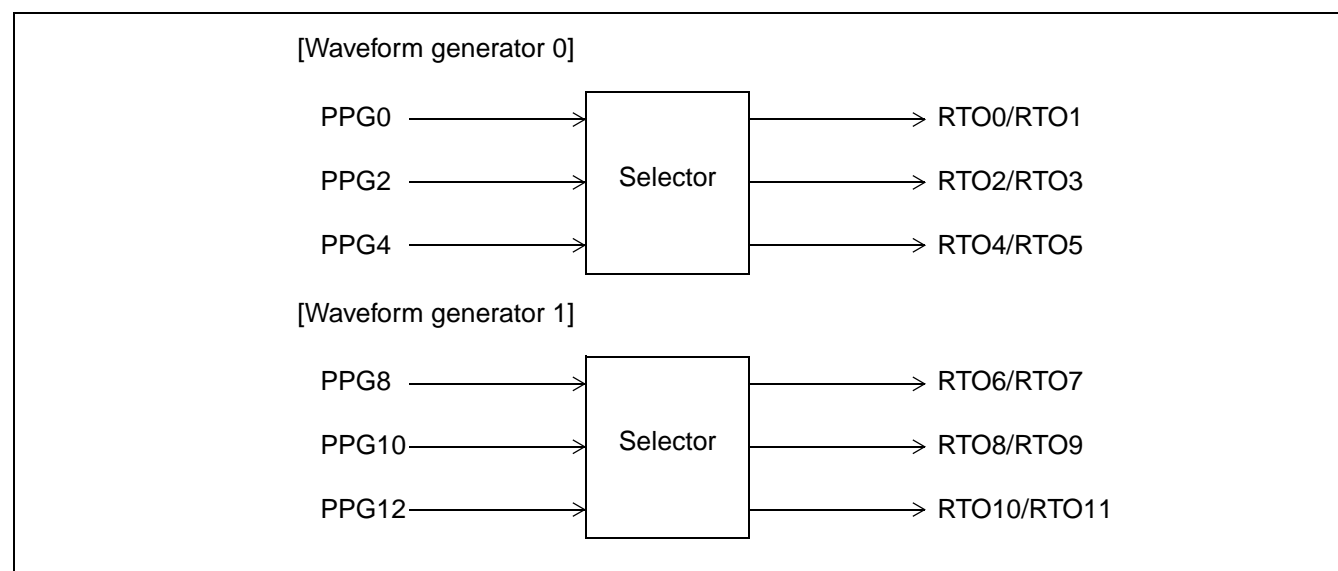
The waveform generator can generate a GATE signal via real-time output RTO0 to RTO5, and the 16-bit dead timer 0, 1, and 2 can operate the PPG count as a trigger. Two real-time outputs (RTO0/RTO2/RTO4 and RTO1/RTO3/RTO5) are controlled by one 16-bit dead timer 0, 1, and 2, generating six separate gate signals. Six gate signals are used logical sum and the GATE signal is generated, causing trigger of the PPG count. Also, if PGEN 0 to PGEN 5 signals are used, it is possible to output 6 different waveforms to RTO0 pin to RTO5 pin using PPG alone.

Note:

This section explains waveform generator 0 as an example. The PPG channel that can be selected is shown in the following figures for waveform generator 1.

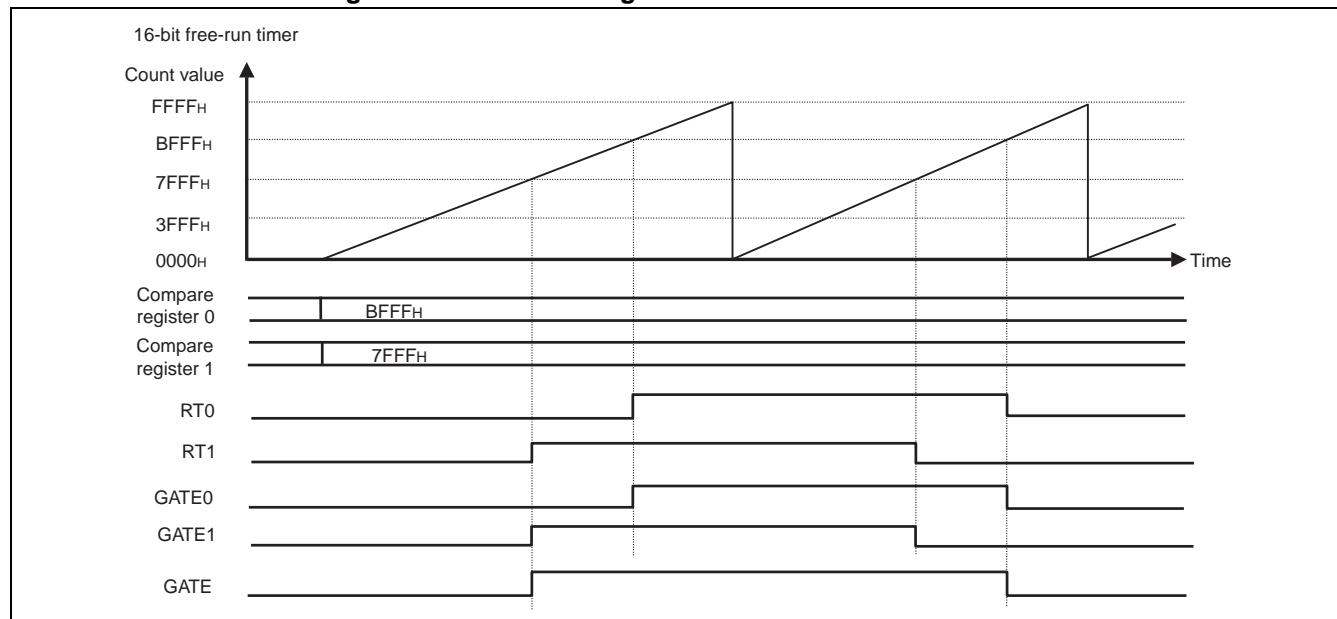
In the MB91470 series, waveform generator 0 only exists.

In the MB91480 series, waveform generator 0/1 exists.



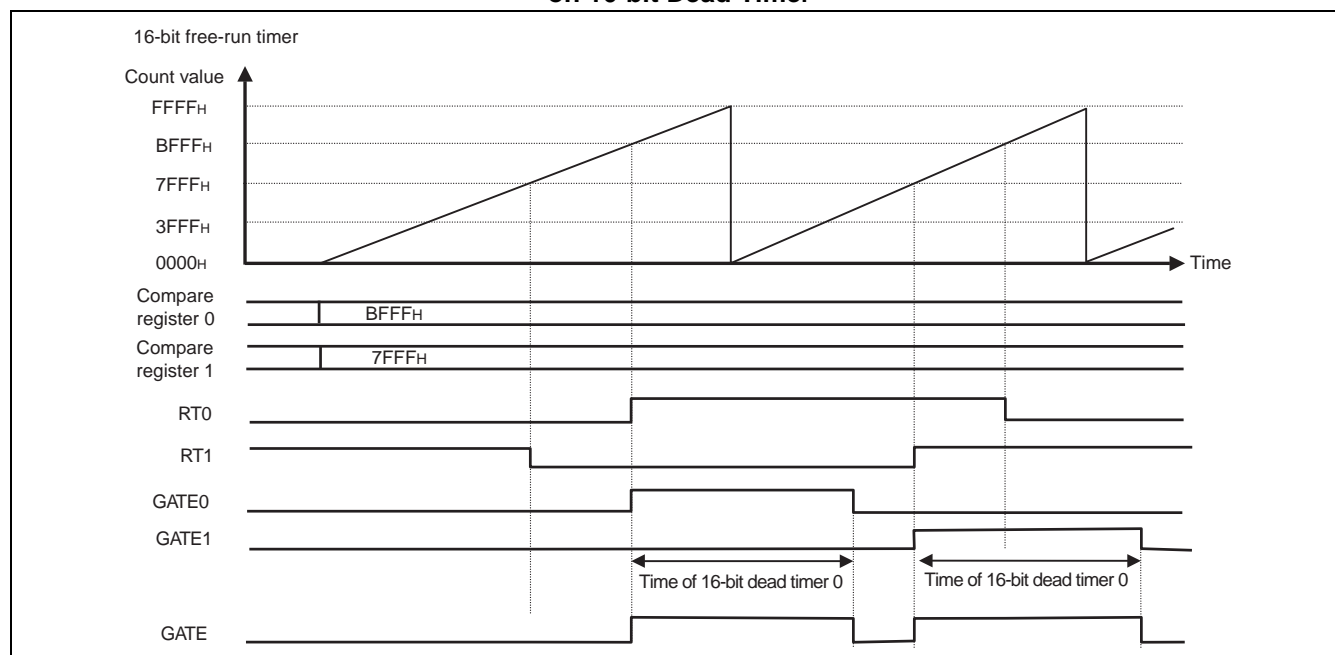
- GATE signal generation when GATEx is active and each RTx is at the "H" level (TMD8 to TMD0 (upper bit10 to bit8, lower bit2 to bit0) in the 16-bit dead timer control registers (DTCR0, DTCR1, DTCR2) are "001_B" or "111_B")

Figure 12.6-22 GATE Signal Generation when RTx is "H"



- GATE signal generation from rising edge on RTx until underflow on 16-bit dead timer 0, 1, 2 when GTENx is active (TMD8 to TMD0 in the DTCR0, DTCR1, DTCR2 registers = 010_B)

Figure 12.6-23 GATE Signal Generation from Rising Edge on RTx until Underflow Occurs on 16-bit Dead Timer



Note:

Each 16-bit dead timer is used for two RTs. In other words, 16-bit dead timer 0 is used for RT0 and RT1; 16-bit dead timer 1 is used for RT2 and RT3; and 16-bit dead timer 2 is used for RT4 and RT5. Consequently, you must not try to use a RT to activate a timer that is already operating. Attempting to do this will extend the GATE signal and therefore result in misoperation.

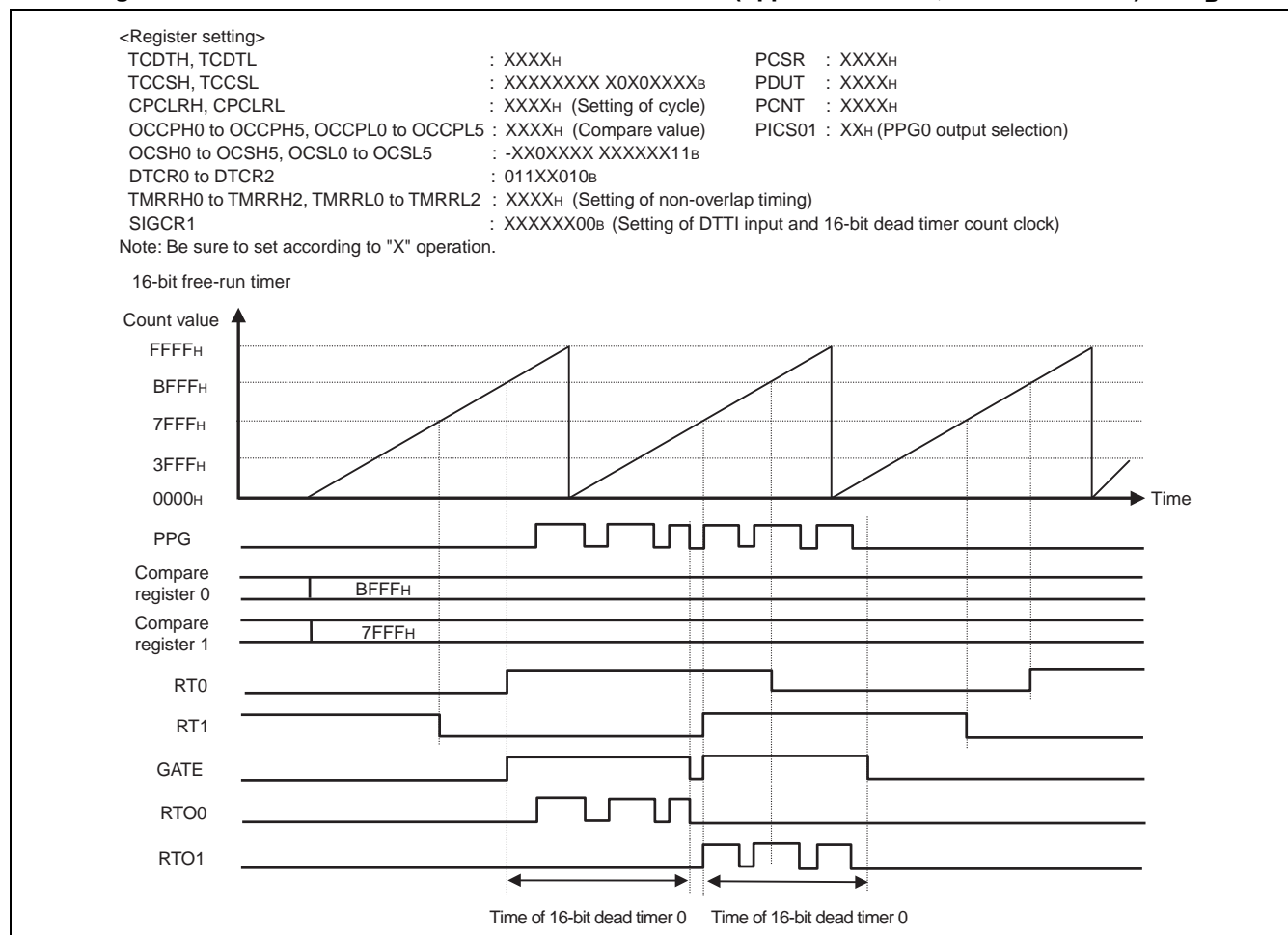
12.6.5.1 Operation of Timer Mode

When an RT0 to RT5 pins rising edge is detected, the value is reloaded into the 16-bit dead timer, and the 16-bit dead timer starts counting down. PPG timer continues to output to the RTO0 to RTO5 pins until an underflow occurs on the 16-bit dead timer.

■ Operation of Timer Mode

- Generation of PPG output pulse from a rising edge on RT until an underflow on the 16-bit dead timer (TMD8 to TMD0 (upper bit10 to bit8, lower bit2 to bit0) in the DTCR0, DTCR1, DTCR2 registers are "010_B")

Figure 12.6-24 Waveform Generated when TMD2 to TMD0 (Upper bit10 to bit8, Lower bit2 to bit0) = 010_B



Note:

Each 16-bit dead timer is used for two RTs. In other words, 16-bit dead timer 0 is used for RT0 and RT1; 16-bit dead timer 1 is used for RT2 and RT3; and 16-bit dead timer 2 is used for RT4 and RT5. Consequently, you must not try to use a RT to activate a PPG that is already operating. Attempting to do this will extend the GATE signal and therefore result in misoperation.

12.6.5.2 Operation during Dead Time Timer Mode

The dead-time generator inputs real-time output (RT1, RT3, and RT5) and outputs a non-overlapping signal (reverse signal) to the external pins (RTO0 to RTO5).

■ Operation during Dead Time Timer Mode

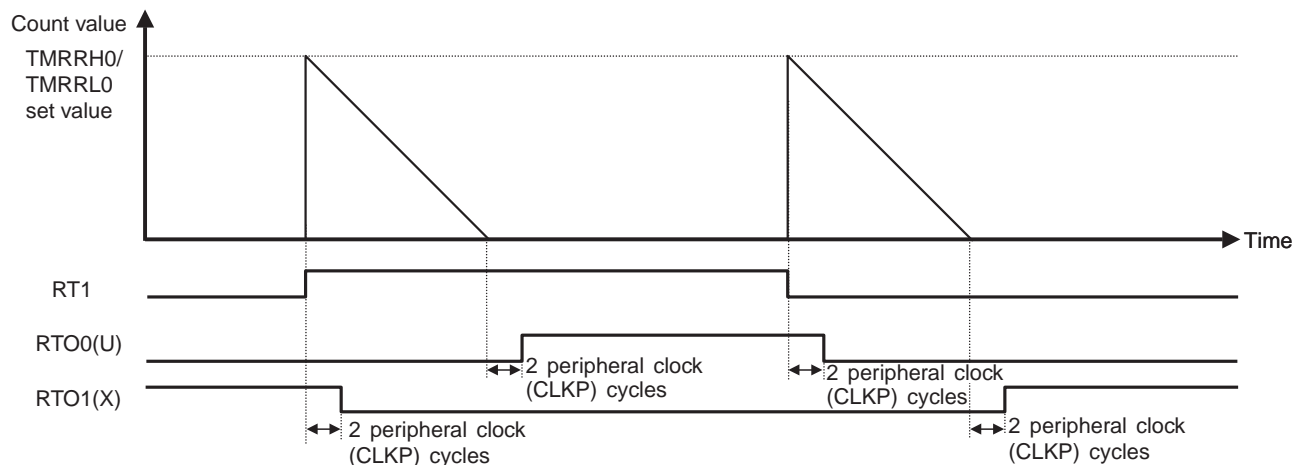
- This non-overlapping signal is generated via normal-polarity RT1, RT3, and RT5 (16-bit dead timer control registers (DTCR0, DTCR1, and DTCR2) TMD8 to TMD0 (higher-order bits are 10 to 8; lower-order bits are 2 to 0) =100_B)

When the DTCR 0, DTCR 1, and DTCR 2 registers DMOD2 to DMOD0 select the non-overlapping signal with a value of 0 (normal polarity), a delay equivalent to the non-overlap time set in the 16-bit dead timer registers (TMRRH0 to TMRRH2, TMRRL0 to TMRRL2) is applied. This delay is generated on the rising edge and falling edge of the RT1, RT3, RT5 pins.

Figure 12.6-25 Non-overlapping Signal Generation Using Normal Polarity RT1, RT3, and RT5

<Register setting>
 TCDTH, TCDTL : XXXXH OCCPH0 to 5, OCCPL0 to 5 : XXXXH (Compare value)
 TCCSH, TCCSL : XXXXXXXX X0X0XXXXB OCSH0 to 5, OCSL0 to 5 : -XX1XXXX XXXXXX11B
 CPCLRH, CPCLRL : XXXXH (Setting of cycle) DTCR0 to DTCR2 : 0XXXX100B
 TMRRH0 to TMRRH2, TMRRL0 to TMRRL2 : XXXXH (Setting of non-overlap timing)
 SIGCR1 : XXXXXXXXB (Setting of DTTI input and 16-bit dead timer count clock)
 Note: Be sure to set according to "X" operation.

16-bit dead timer 0



Pin name	Output signal
RTO0 (U)	Delayed signal is applied at rising edge of RT1.
RTO2 (V)	Delayed signal is applied at rising edge of RT3.
RTO4 (W)	Delayed signal is applied at rising edge of RT5.
RTO1 (X)	Delayed inverted signal is applied at falling edge of RT1.
RTO3 (Y)	Delayed inverted signal is applied at falling edge of RT3.
RTO5 (Z)	Delayed inverted signal is applied at falling edge of RT5.

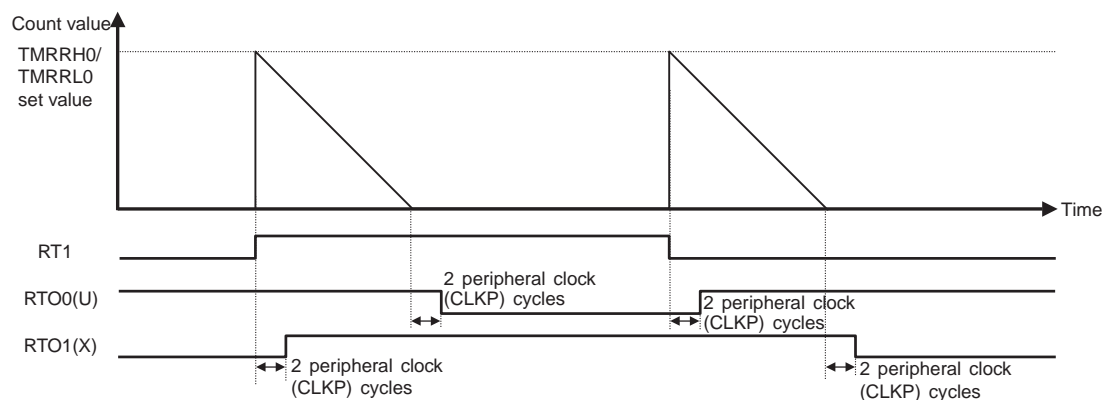
- This non-overlapping signal is generated via reverse-polarity RT1, RT3, and RT5 (16-bit dead timer control registers (DTCR0, DTCR1, and DTCR2) TMD8 to TMD0 (higher-order bits are 10 to 8; lower-order bits are 2 to 0) = 100_B)

When the DTCR0, DTCR1, and DTCR2 registers DMOD2 to DMOD0 (higher-order bit is 15, lower-order bit is 7) select the non-overlapping signal with a value of 1 (reverse polarity), a delay equivalent to the non-overlap time set in the 16-bit dead timer registers (TMRRH0 to TMRRH2, TMRRL0 to TMRRL2) is applied. This delay is generated on the rising edge and falling edge of the RT1, RT3, RT5 pins.

Figure 12.6-26 Non-overlapping Signal Generation Using Inverted Polarity RT1, RT3, and RT5

<Register setting>
TCDTH, TCDTL : XXXXH OCCPH0 to OCCPH5, OCCPL0 to OCCPL5 : XXXXH (Compare value)
TCCSH, TCCSL : XXXXXXXX X0X0XXXXB OCSH0 to OCSH5, OCSL0 to OCSL5 : -XX1XXXX XXXXXX11B
CPCLRH, CPCLRL : XXXXH (Setting of cycle) DTCR0 to DTCR2 : 1XXXX100B
TMRRH0 to TMRRH2, TMRRL0 to TMRRL2 : XXXXH (Setting of non-overlap timing)
SIGCR1 : XXXXXXXXB (Setting of DTTI input and 16-bit dead timer count clock)
Note: Be sure to set according to "X" operation.

16-bit dead timer 0



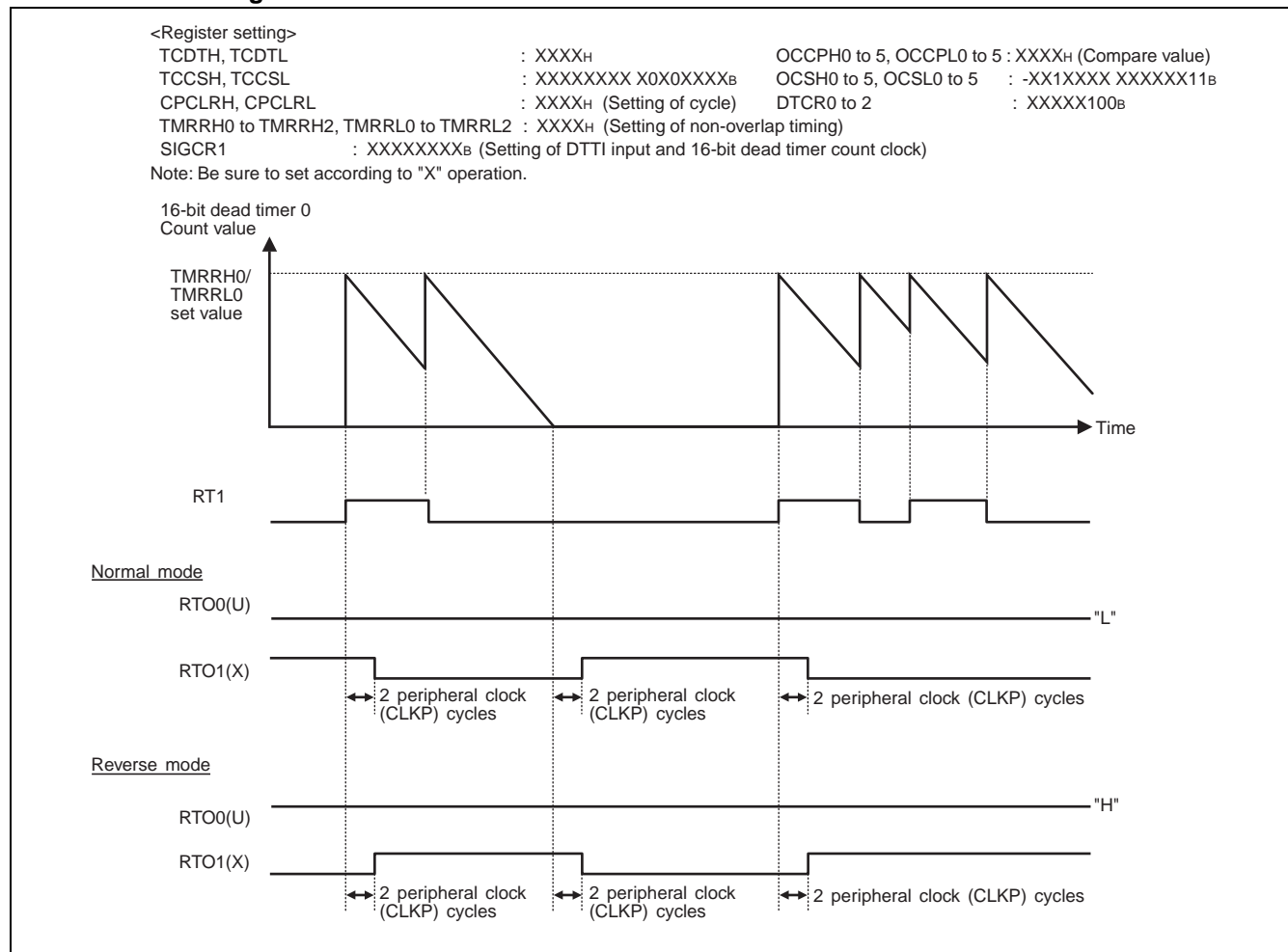
Pin name	Output signal
RTO0 (U)	Delayed inverted signal is applied at rising edge of RT1.
RTO2 (V)	Delayed inverted signal is applied at rising edge of RT3.
RTO4 (W)	Delayed inverted signal is applied at rising edge of RT5.
RTO1 (X)	Delayed signal is applied at falling edge of RT1.
RTO3 (Y)	Delayed signal is applied at falling edge of RT3.
RTO5 (Z)	Delayed signal is applied at falling edge of RT5.

■ Note on Using Dead Time Timer Mode

When the pulse width of RT1, RT3 or RT5 is smaller than the current setting of the non-overlap time, 16-bit dead timer reloads the value of TMRRH0 to TMRRH2 and TMRRL0 to TMRRL2 at the next RT edge, then restarts counting down.

If the compare output transition time is small and reloading is repeated before the dead timer underflow occurs, X and U are fixed to "L" in Normal mode, and X and U are fixed to "H" in Reverse mode. Therefore, set the 16-bit dead timer registers (TMRRH0 to TMRRH2, TMRRL0 to TMRRL2) not to be reloaded repeatedly.

Figure 12.6-27 When reloaded before the dead timer underflow occurs



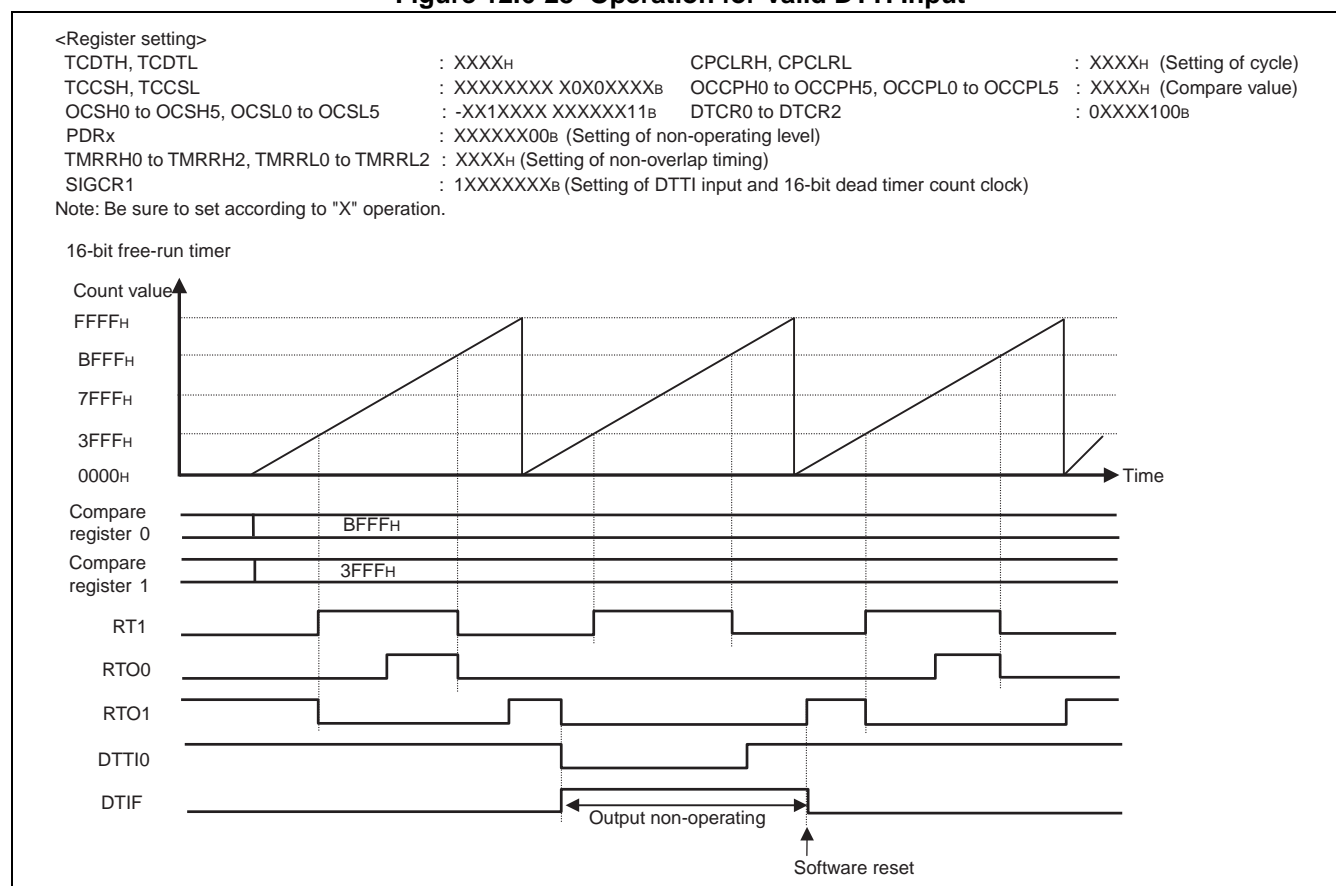
12.6.5.3 DTTI Pin Control Operation

You can control RTO0 to RTO5 output by means of the DTTI pins by setting "1" in the waveform control register 1 (SIGCR1) DTIE: bit7. When a DTTI pin "L" level is detected, RTO0 to RTO5 output is fixed at non-operating level until the interrupt flag (SIGCR register DTIF: bit6) is cleared. When RTO0 to RTO5 is at non-operating level, these pins can be set via software using the port data registers (PDR), which share them. Additionally, if they are used as input ports using the data direction register (DDR), Hi-Z is output.

■ DTTI Pin Input Operation

Even when "L" is detected in DTTI pin input, although the timer continues to operate while the waveform generator is operational, waveforms are not output to the external RTO0 to RTO5 pins.

Figure 12.6-28 Operation for Valid DTTI Input



■ DTTI Operation of Waveform Control Register 2 (SIGCR2)

The output of waveform control register 2's DTTI: bit0 with the DTTI pin input and OR is the DTTI input.

Consequently, when "0" is set in this register, control is permanently in DTTI input status, and the input from the DTTI pins has no meaning.

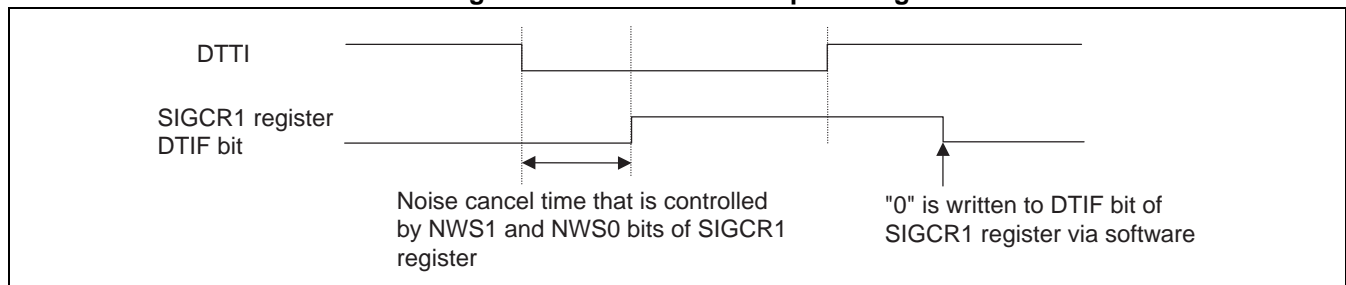
When this register is cleared by writing "1", the DTTI pin input value is used.

■ DTTI Pin Noise Cancellation Feature

The DTTI pin input noise cancellation feature is enabled when "1" is set in the waveform control register 1 (SIGCR1) NRSL: bit5. When the noise cancellation feature is enabled, there is a delay of 4, 8, 16, or 32 peripheral clock (CLKP) cycles (selected via the SIGCR1 register NWS1 and NWS0: bit1 and bit0), for the amount of time necessary to lock the output pins (RTO0 to RTO5) to non-operating level. Since the noise cancellation circuit uses resources, in modes where oscillation is stopped (e.g. stopped mode), input is disabled, even when DTTI input is enabled.

■ DTTI Interrupt

When DTTI "L" level is detected, after the noise cancellation time has elapsed, the DTTI interrupt flag (SIGCR1 register DTIF: bit6) is set to "1", and an interrupt request is sent to the interrupt controller.

Figure 12.6-29 DTTI Interrupt Timing**Notes:**

- The setting in PFRQ/PFRS are invalid when DTIF: bit6 of the waveform control register 1 (SIGCR1) is 1, and the respective ports are always used as general-purpose ports. See Section "5.2.5 Multi-Function Timer I/O Port" for details.
- If the SIGCR1 registers NWS1 and NWS0 bits' values change within the noise cancellation time, a larger (NWS1 and NWS0) noise cycle value is enabled.
- The SIGCR1 register DTIF: bit6 can only be cleared via software.

12.6.6 A/D Activation Compare Operation

An A/D activation can be performed when the value of 16-bit free-run timer reaches the specified value.

■ A/D Activation

Three A/D converter units can be activated.

■ A/D Compare Activation Enabled

If the compare register value is set, and other than "00_B" is set into the compare enable register (ADTGCE) CE00, CE01, CE10, CE11, CE20, CE21: bit0, bit1/bit2, bit3/bit4, bit5, when the free run timer and compare register value are matched, an A/D activation signal is generated.

When "00_B" is set into CE00, CE01, CE10, CE11, CE20, CE21, even if the free run timer and compare register value are matched, an A/D activation signal is not generated.

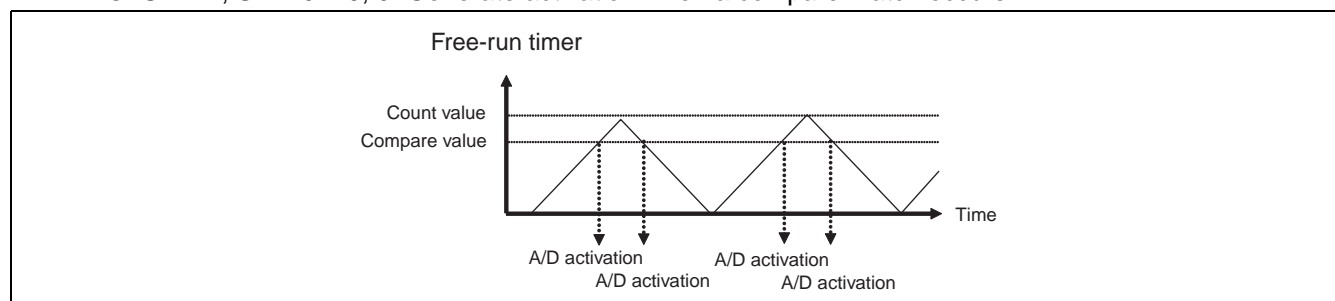
■ Setting of Free-run Timer Input Selection

The free-run timer input can be selected independently for the A/D activation compare by the compare enable register (ADTGCE) as well as the A/D activation compare enable control.

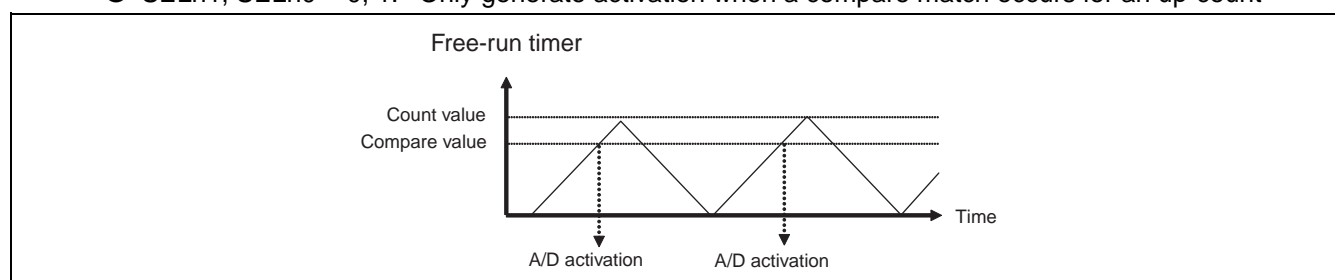
■ A/D Compare Activation Mode

The A/D activation mode is set in the SEL bits of the ADTGSEL register.

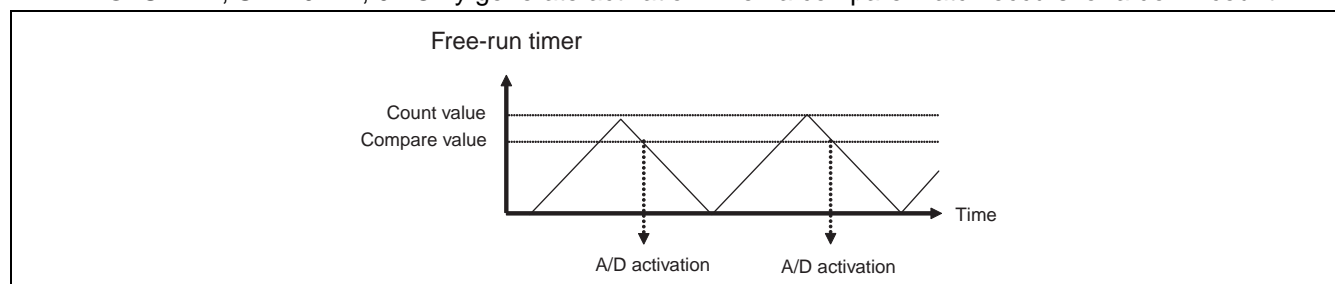
- SELn1, SELn0 = 0, 0: Generate activation when a compare match occurs



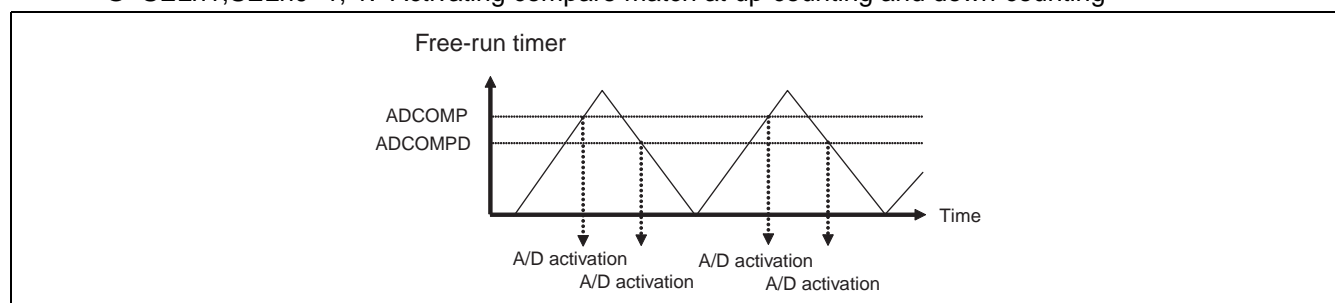
- SELn1, SELn0 = 0, 1: Only generate activation when a compare match occurs for an up-count



- SELn1, SELn0 = 1, 0: Only generate activation when a compare match occurs for a down-count



- SELn1, SELn0 = 1, 1: Activating compare match at up-counting and down-counting



■ Setting of Free-run Timer Count Direction Selection

The A/D activation compare register is compared with the free-run timer by the count direction selection register (ADTGSEL) in either of up-counting, down-counting or up/down-counting.

■ Compare Register Buffering

Writing "0" to the BUF_X bits (bit2 to bit0) of the buffer control register (ADTGBUF) enables the compare register buffering. If buffering is selected (by setting buffer control register (ADTGBUF), BTS bits (bit6 to bit4), the buffer is transferred the value of the compare register to the compare buffer register at the compare clear interruption (by setting ADTGBUF, BTS:bit6-to-bit4=1) or zero detection interruption (by setting ADTGBUF, BTS:bit6-to-bit4=0).

■ A/D Activation by Zero Detection of Free-run Timer or Compare Clear

If "1" is written to the AD2E to AD0E bits (bit2 to bit0) of the A/D trigger control register (ADTRGC), A/D can be activated when a zero detection of the free-run timer or a compare match interrupt occurs. When "0" is set to the SEL bit (bit6 to bit4) of the A/D trigger control register (ADTRGC), zero detection occurs. When "1" is set to it, compare match interrupt occurs.

■ Reload Timer (ch.1)

When A/D compare activation is disabled and the A/D activation by a zero detection of the free-run timer or a compare clear is disabled, the A/D activation by the 16-bit reload timer ch.1 is enabled. In MB91470 series, the activation by the 12-bit A/D converter 4 is enabled. In MB91480 series, the activation by the 10-bit A/D converter 1 is enabled.

Figure 12.6-30 Compare Register 0: Valid Buffer Function, and Compare Register 1: Invalid Buffer Function

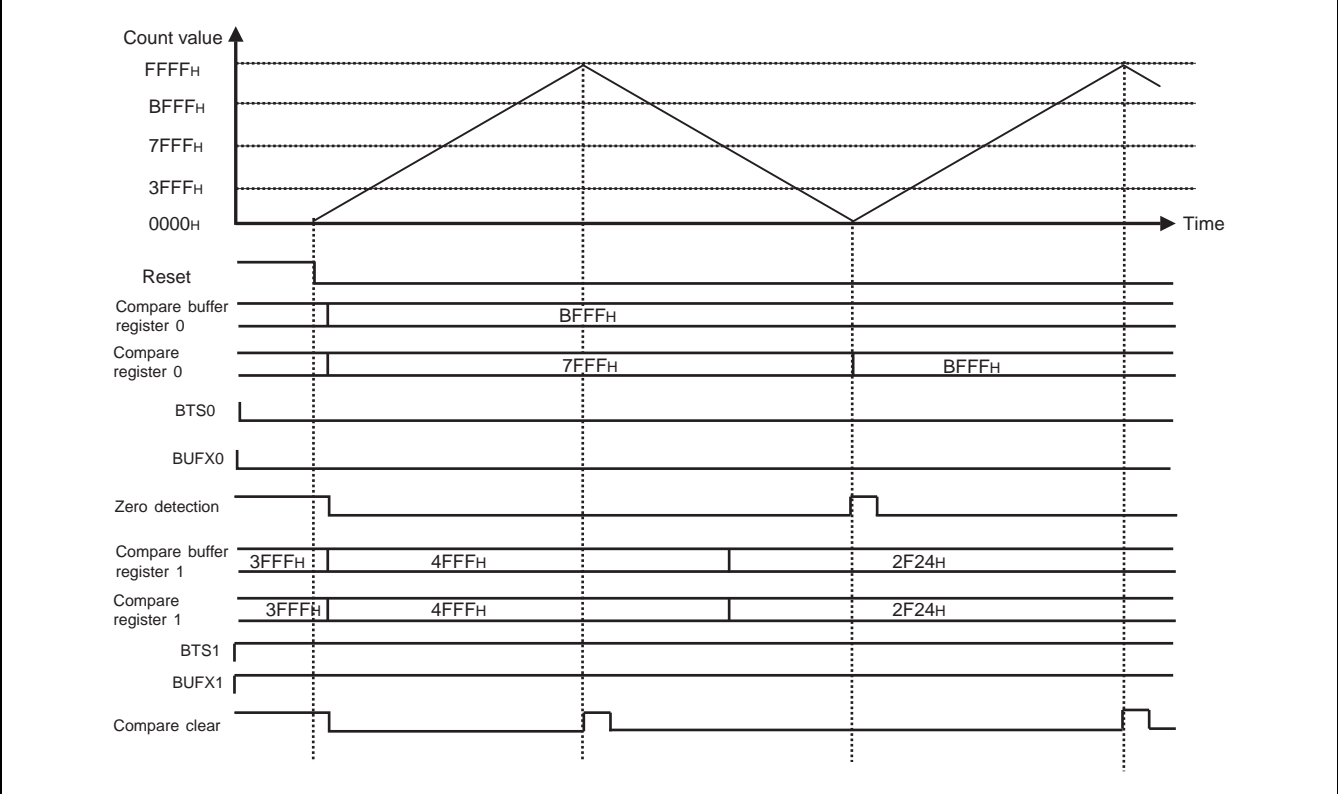


Figure 12.6-31 A/D Trigger 0 is Activating at Up/Down Counting, A/D Trigger 1 is Activating at Up Counting

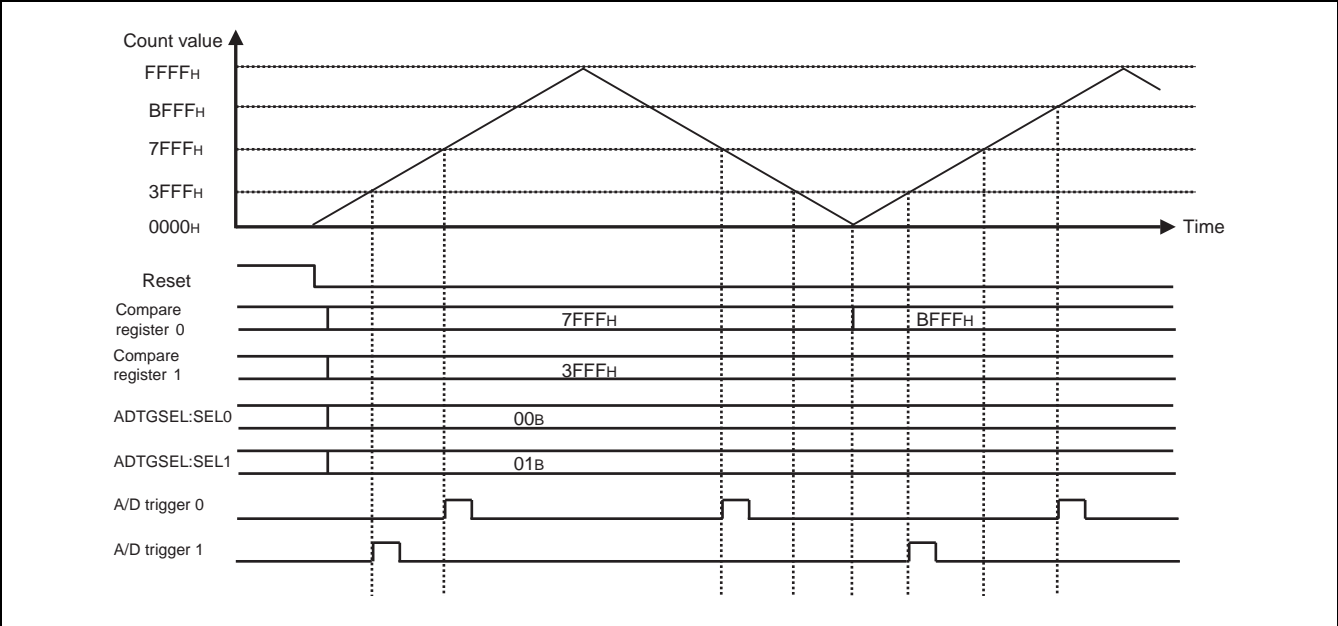


Figure 12.6-32 A/D Activation Compare Compare Clear Interrupt of Free-run Timer

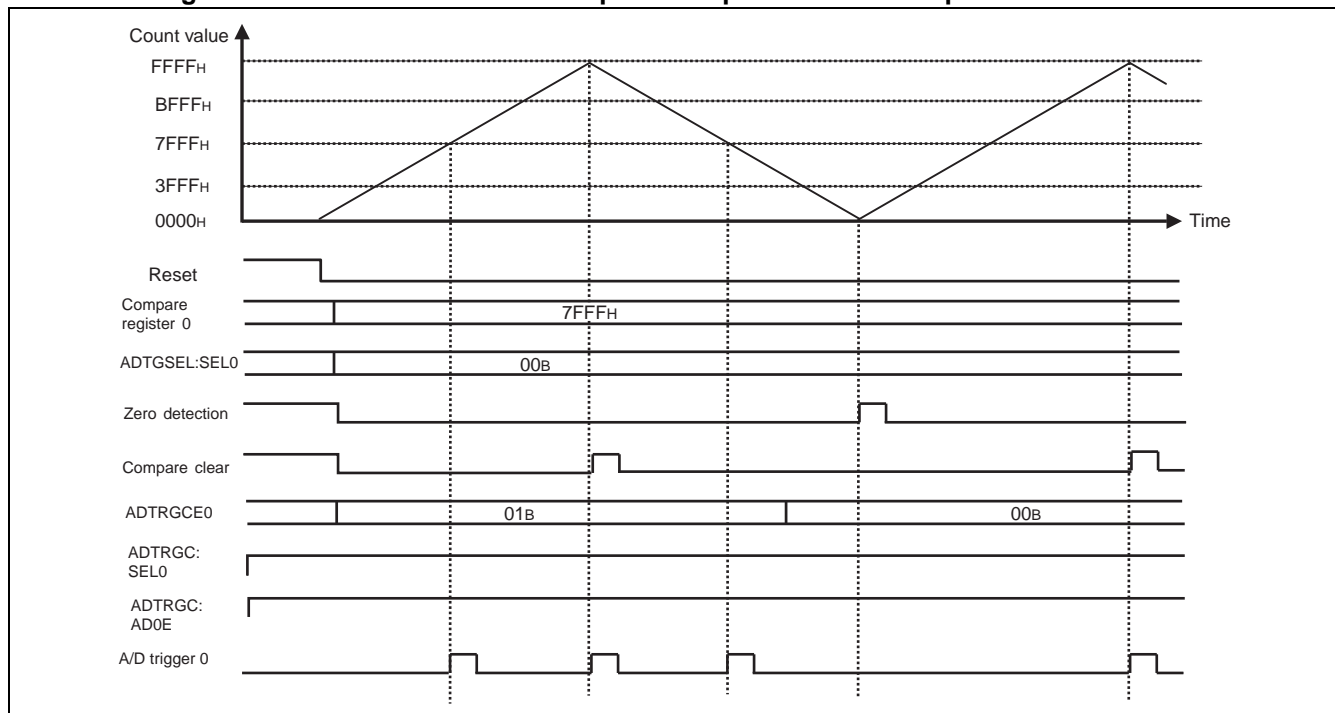


Figure 12.6-33 The Data Transfer Timing at Compare Match at Free-run timer Up-count Mode

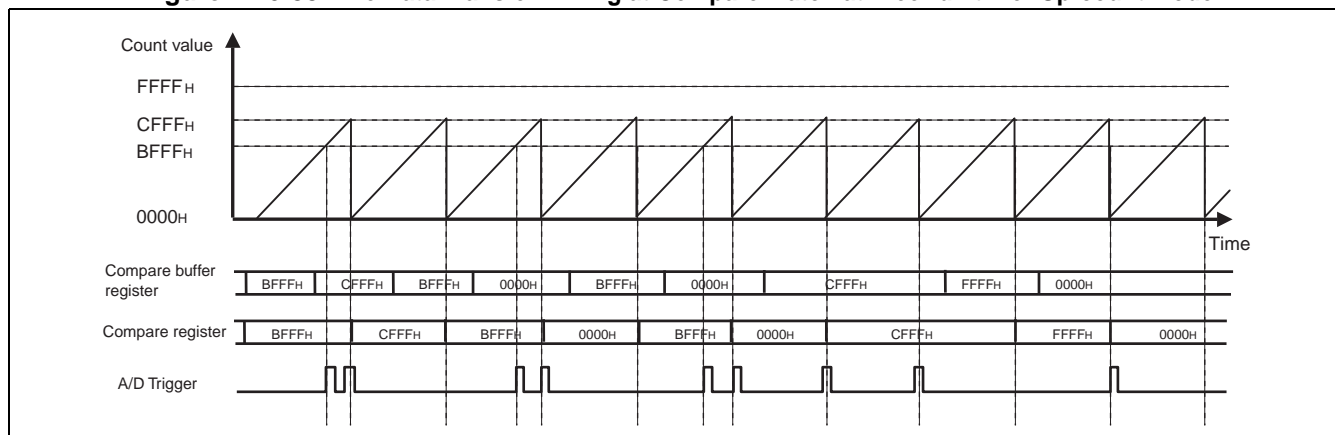


Figure 12.6-34 The Data Transfer Timing at Zero Detection at Free-run timer Up-count Mode

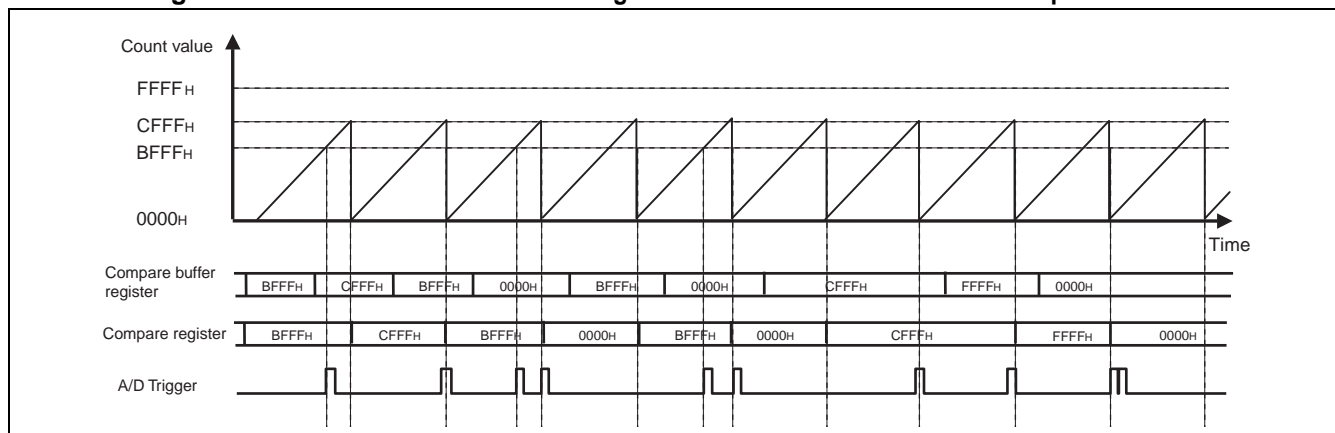


Figure 12.6-35 The Data Transfer Timing at Compare Match at Free-run timer Up/Down Count Mode

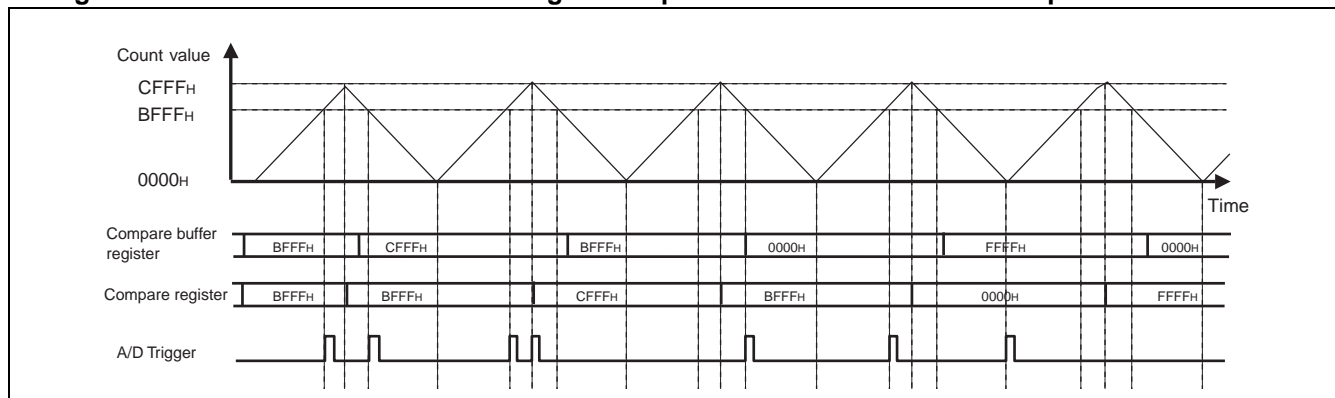
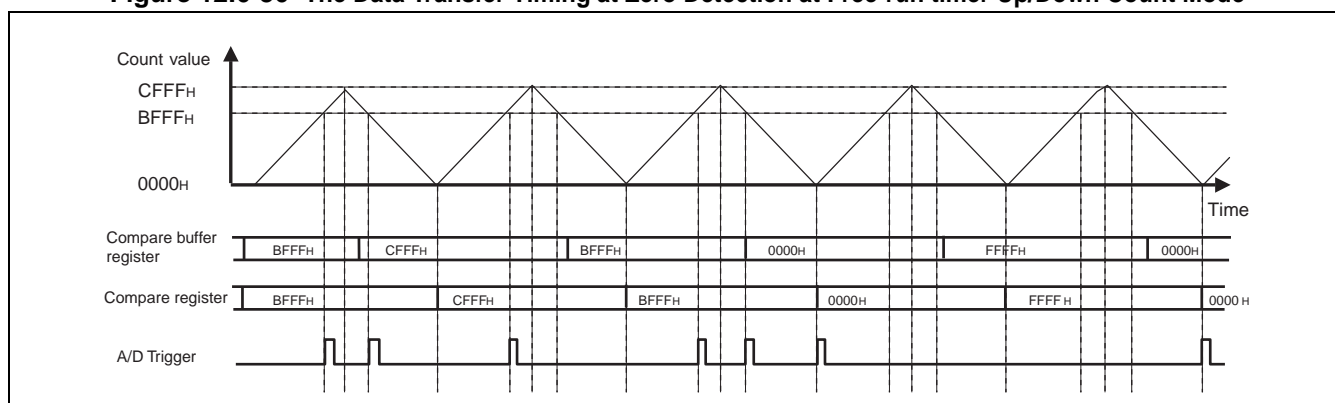


Figure 12.6-36 The Data Transfer Timing at Zero Detection at Free-run timer Up/Down Count Mode



Note:

See Section "12.7 Notes on Using the Multi-function Timer" for connection between the A/D activation compare and A/D converter.

12.7 Notes on Using the Multi-function Timer

Heed the following cautions when using the multi-function timer.

■ Notes on Using Different Series

Table 12.7-1 shows registers required to be set before using the multi-function timer.

Table 12.7-1 Comparison Table of Registers in Different Series (1 / 3)

Module	Register name	MB91470	MB91480
FT0 to FRT2 (Free-run timer unit 0 to unit 2 of multi-function timer0) Note: They are shared to multi-function timer1's input capture and output compare in MB91480 series.	CPCLRBH0/CPCLRH0 to CPCLRBH2/CPCLRH2	O	O
	CPCLRBL0/CPCLRL0 to CPCLRBL2/CPCLRL2	O	O
	TCDTH0 to TCDTH2	O	O
	TCDTL0 to TCDTL2	O	O
	TCCSH0 to TCCSH2	O	O
	TCCSL0 to TCCSL2	O	O
	TCCSM0 to TCCSM2	O	O
	ADTRGC0 to ADTRGC2	O	O
FRT3 to FRT5 (Free-run timer unit 3 to unit 5 of multi-function timer1) Note: They are shared to multi-function timer0's output compare and input capture in MB91480 series.	CPCLRBH3/CPCLRH3 to CPCLRBH5/CPCLRH5	X	O
	CPCLRBL3/CPCLRL3 to CPCLRBL5/CPCLRL5	X	O
	TCDTH3 to TCDTH5	X	O
	TCDTL3 to TCDTL5	X	O
	TCCSH3 to TCCSH5	X	O
	TCCSL3 to TCCSL5	X	O
	TCCSM3 to TCCSM5	X	O
	ADTRGC3 to ADTRGC5	X	O

Table 12.7-1 Comparison Table of Registers in Different Series (2 / 3)

Module	Register name	MB91470	MB91480
FRS0 to FRS9 (Free-run timer selector for multi-function timers)	FRS0 to FRS4	O	O
	FRS5 to FRS9	X	O
OCU0 to OCU2 (Output compare 0 to 2 of multi-function timer0)	OCCPBH0/OCCPH0 to OCCPBH5/OCCPH5	O	O
	OCCPBL0/OCCPL0 to OCCPBL5/OCCPL5	O	O
	OCSH1, OCSH3, OCSH5	O	O
	OCSL0, OCSL2, OCSL4	O	O
	OCMOD0	O	O
OCU3 to OCU5 (Output compare 3 to 5 of multi-function timer1)	OCCPBH6/OCCPH6 to OCCPBH11/OCCPH11	X	O
	OCCPBL6/OCCPL6 to OCCPBL11/OCCPL11	X	O
	OCSH7, OCSH9, OCSH11	X	O
	OCSL6, OCSL8, OCSL10	X	O
	OCMOD1	X	O
IC0 to IC3 (Input capture 0 to 3 of multi-function timer0)	IPCPH0 to IPCPH3	O	O
	IPCPL0 to IPCPL3	O	O
	ICSH23	O	O
	ICSL23	O	O
	PICSH01	O	O
	PICSL01	O	O
IC4 to IC7 (Input capture 4 to 7 of multi-function timer1)	IPCPH4 to IPCPH7	X	O
	IPCPL4 to IPCPL7	X	O
	ICSH67	X	O
	ICSL67	X	O
	PICSH45	X	O
	PICSL45	X	O

Table 12.7-1 Comparison Table of Registers in Different Series (3 / 3)

Module	Register name	MB91470	MB91480
WG0 (waveform generator 0 of multi-function timer0)	TMRRH0 to TMRRH2	O	O
	TMRRL0 to TMRRL2	O	O
	DTCR0 to DTCR2	O	O
	SIGCR10, SIGCR20	O	O
WG1 (waveform generator 1 of multi-function timer1)	TMRRH3 to TMRRH5	X	O
	TMRRL3 to TMRRL5	X	O
	DTCR3 to DTCR5	X	O
	SIGCR11, SIGCR21	X	O
ADTG0 (A/D activation compare 0 of multi-function timer0)	ADCOMPB0/ADCOMP0 to ADCOMP2	O	O
	ADTGCE0	O	O
	ADTGSEL0	O	O
	ADTGBUF0	O	O
ADTG1 (A/D activation compare 1 of multi-function timer1)	ADCOMPB3/ADCOMP3 to ADCOMP5	X	O
	ADTGCE1	X	O
	ADTGSEL1	X	O
	ADTGBUF1	X	O
PPG0/PPG2	GATEC0	O	O
PPG4	GATEC4	O	O
PPG8/PPG10	GATEC8	X	O
PPG12	GATEC12	X	O

Note:

O - installed, X - absent.

For PPG settings, refer to "CHAPTER 11 PPG".

■ Notes at Accessing the Buffer Registers

CPCLRL/Hn register in free-run timer, OCCPL/Hn register in output compare, and ADCOMPn/ADCOMPBn register in A/D activating compare have the buffer function. Do not access to these registers by the read-modify-write instruction.

■ Notes on Using the 16-bit Free-run Timers

● Cautions for setting via the program

- When a reset is executed, although the timer value becomes "0000_H", the zero-detect interrupt flag is not set.
- Since the timer-mode bit (TCCSL registers MODE: bit5) has a buffer, and so timer modes changed after zero-detect are enabled.
- A software clear (setting bit4 (SCLR) in the TCCSL register = 1) initializes the timer but does not generate a zero-detect interrupt.
- When the compare value and count value match, if the count starts, the compare-clear flag is not set.

● Cautions for interrupt

- If "1" is set in the timer state control register upper (TCCSH) IRQZF: bit14, then interrupt requests are enabled (TCCSH register's IRQZE: bit13 = 1), control cannot return from the interrupt processing. Be sure to clear the IRQZF: bit14.
- If "1" is set in the timer state control register upper (TCCSH) ICLR: bit9, then interrupt requests are enabled (TCCSH register's ICRE: bit8 = 1), control cannot return from the interrupt processing. Be sure to clear the ICLR: bit9.

● Cautions at accessing the TCCSH/TCCSM registers

- A set value is read from MSI2 to MSI0/MSI5 to MSI3 at the read-modify-write instruction.
- The counter value is read from MSI2 to MSI0/MSI5 to MSI3 at the normal reading.

■ Cautions for Use of Free-run Timer Selector

Be sure to select the setting while the free-run timer is stopped.

■ Notes on Using the 16-bit Output Compare

● Cautions for interrupt

If "11_B" is set in the compare control registers lower-order(OCSL0, OCSL2, and OCSL4) IOP1, IOP0 : bit7 and bit6, then interrupt requests are enabled (OCSL register's IOE1, IOE0 : bit6 and bit5=11_B), control cannot return from the interrupt processing. Be sure to clear the IOP0, IOP1 bits.

■ Cautions for Use of 16-bit Input Capture

● Cautions for interrupt

- If "1" is set in the input capture state control registers lower-order (PICSL01 and ICSL23) ICP3, ICP2, ICP1, and ICP0 (both bit7 and bit6), then interrupt requests are enabled (PICSL01 and ICSL23 register's ICE3, ICE2, ICE1, and ICE0 (both bit5 and bit4) =11_B), control cannot return from the interrupt processing. Be sure to clear ICP 3, ICP 2, ICP 1, and ICP 0 (both bit7 and bit6).
- When the input capture pin (IC) level changes the time between setting the bit for ICP3, ICP2, ICP1, and ICP0 and processing of the interrupt routine, the valid edge indication bits of the ICP3, ICP2, ICP1, and ICP0 (ICSH23 register's IEI3 and IEI2: bit9 and bit8 and PICSH01 register's IEI1 and IEI0: bit9, bit8) indicates the newly detected edge.

■ Notes on Using the Waveform Generator

● Cautions for setting via the program

- Confirm that the trigger source and 16-bit dead timer are not counting when the bit values of TMD8, TMD5, TMD2 (higher-order bit is 10; lower-order bit is 2), TMD7, TMD4, TMD1 (higher-order bit is 9; lower-order bit is 1), and TMD6, TMD3, TMD0 (higher-order bit is 8; lower-order bit is 0) in the 16-bit dead timer control registers (DTCR0, DTCR1, and DTCR2) are changed while the waveform generator is operating (DTCR0, DTCR1, and DTCR2 register's TMD2 to TMD0, TMD5 to TMD3, TMD8 to TMD6 are "001_B", "010_B", or "100_B"). If this operation is not performed, an unintended waveform will be output from the RTO pin due to the output scheduled by the previous trigger. However, the RTO output returns to normal operation when a timer underflow occurs or when triggered again by the new trigger source.
- The trigger source is at "H" level for RT when TMD8 to TMD0 (higher-order bits are 10 to 8; lower-order bits are 2 to 0) in the DTCR0, DTCR1, and DTCR2 registers are "001_B"; it is rising edge of RT when these bits are "010_B"; it is rising or falling edge of RT when these bits are "100_B".

For example, if TMD bit8 to bit0 change from "100_B" to "010_B", the following steps can be executed.

1. Set the 16-bit dead timer register (TMRRH0 to TMRRH2, TMRRL0 to TMRRL2) to an extremely small value like 0001_H.
 2. Set the RTO1, RTO3, or RTO5 output to "L" or "H" and wait for an underflow on timer 0, 1, or 2.
 3. Change the mode bits (TMD 8 to TMD 0) and corresponding settings.
 4. A corrected output waveform appears at the RTO pins after 1 machine cycle.
- If the value of the 16-bit dead timer register (TMRRH0 to TMRRH2, TMRRL0 to TMRRL2) is modified while the timer is counting, the new value is not used until the next timer trigger. When accessing the timer registers, be sure to use half-word or word data transfer commands.
 - Only change the waveform control register 1's (SIGCR1) DCK2 to DCK0: bit4 to bit2 when the timers are not counting.
 - Only change the waveform control register 1's (SIGCR1) NWS1 and NWS0: bit1 and bit0 when the noise cancellation feature is disabled.

● Cautions for interrupt

- If "1" is set in the 16-bit dead timer control register (DTCR0, DTCR1, and DTCR2) TMIF2 to TMIF0 (higher-order bit is 12; lower-order bit is 4), then interrupt requests are enabled (DTCR0, DTCR1, and DTCR2 register's TMIE2 to TMIE0 (higher-order bit is 11; lower-order bit is 3) =1), control cannot return from the interrupt processing. Be sure to clear the TMIF bit.
- Control cannot return from interrupt processing after setting 1 in the waveform control register 1 (SIGCR1)DTIF:bit6. Be sure to clear the DTIF bit.

■ Notes on Using the A/D Activation Compare

Be sure to select the setting while the free-run timer is stopped.

12.8 Example Program for Multi-function Timer

Below is a sample multi-function timer program.

■ Example Program for the 16-bit Free-run Timer

● Processing

- When the 16-bit free-run timer is 4 ms, generate a compare-clear interrupt.
- This timer is used to re-generate a compare-clear timer during up-count mode.
- 16 MHz is for the peripheral clock (CLKP), and 62.5 ns is for the count clock.

● Coding example

```

ICR33      .EQU      000461H    ; Compare clear interrupt control register for the 16-bit free-run
                                   ; timer 0

TCCSH      .EQU      0000B8H    ; Timer control status register
CPCLRBH    .EQU      0000B4H    ; Compare-clear buffer register

; ----- Main Program -----
                ORG      C0000H

START:
;               :                               ; Assumes that the stack pointer (SP) has already been
                                   ; initialized.

                ANDCCR    #0EFH    ; Disables the interrupt.
                LDI       #ICR32,r0

LDI          #00H,r1
STB          r1,@r0                ; Interrupt levels 16 (the highest priority)
LDI          #CPCLRBH,r0            ; Set value to the compare clear buffer register so that
LDI          #0FA00H,r1            ; compare clear interrupts will be generated at 4ms
                STH        r1,@r0    ; intervals in 16-bit free-run timer up-count mode
                LDI        #TCCSH,r3 ; Set up-down count mode,
                LDI        #0110H,r1 ; set 62.5ns count clock,
                STH        r1,@r3    ; enable compare clear interrupt,
                                   ; clear compare clear interrupt flag bit,
                                   ; disable interrupt mask,
                                   ; clear timer, and enable operation

                STILM      #14H      ; Set the ILM in PS to level 20
                ORCCR      #10H      ; Interruption permission

LOOP         LDI          #00H,r0    ; Infinite loop
                LDI        #01H,r1
                BRA        LOOP      ;

; ----- Interrupt Program -----

```

```

WARI    LDI        #0100H,r1
        ANDH       r1,@r3    ; Clear interrupt request flag.
;
;
;      User processing
;
;      RETI          ; Returns from interrupt.

; ----- Vector Settings -----
VECT    .ORG FFFF8H
        .DATA.W    WARI      ; Set interrupt routine.
        .ORG       FFFF8H
        .DATA.W    0x07000000 ; Set single-chip mode.
        .DATA.W    START     ; Set reset vectors
        .END

```

■ Example Program for the 16-bit Output Compare

● Processing

- When the 16-bit free-run timers count value matches the output compare value, an output compare match is generated.
- Use when the 16-bit free-run timer is in up/down count mode.

● Coding example

```

ICR44   .EQU       00046CH    ; Output compare 0/1 interrupt register
TCCSH   .EQU       0000B8H    ; Timer control status register
CPCLRBH .EQU       0000B4H    ; Compare-clear buffer register
OCCPBH0 .EQU       0000A0H    ; Output compare buffer register 0
OCCPBH1 .EQU       0000A2H    ; Output compare buffer register 1
OCSH1   .EQU       0000ACH    ; Compare control register

; ----- Main Program -----
START:
;      :                               ; Assumes that the stack pointer (SP) has already been
;                               ; initialized.
        ANDCCR    #0EFH        ; Disables the interrupt.
        LDI       #ICR44,r0
        LDI       #00H,r1
        STB       r1,@r0        ; Interrupt levels 16 (the highest priority)
        LDI       #CPCLRBH,r0   ; Set compare clear buffer register
        LDI       #0FFFFH,r1    ; for 16-bit free-run timer
        STH       r1,@r0

```

```

        LDI        #OCCPBH0,r0 ; Set the output compare register 0.
        LDI        #0BFFFH,r1
        STH        r1,@r0
        LDI        #OCCPBH1,r0 ; Set the output compare register 1.
        LDI        #07FFFH,r1
        STH        r1,@r0
        LDI        #OCSH1,r3    ; Enable output compare output.
        LDI        #6C33H,r2    ; Enable compare match interrupts 0/1.
        STH        r2,@r3      ; Clear the interrupt flag bit.

        LDI        #TCCSH,r0    ; Set up-down count mode,
        LDI        #0010H,r1    ; clear timer, and enable operation
        STH        r1,@r0
        STILM      #14H        ; Set the ILM in PS to level 20
        ORCCR      #10H        ; Interruption permission
LOOP    LDI        #00H,r0      ; Infinite loop
        LDI        #01H,r1
        BRA        LOOP        ;

; ----- Interrupt Program -----
WARI :
        ANDH       r2,@r3      ; Clear interrupt register flag.
        ;          :
        ;          : User processing
        ;          :
        RETI        ; Returns from interrupt.

; ----- Vector Settings -----
VECT    .ORG FFFF8H
        .DATA.W    WARI        ; Set interrupt routine.
        .ORG       FFFF8H
        .DATA.W    0x07000000 ; Set single-chip mode.
        .DATA.W    START      ; Set reset vectors.
        .END

```


CHAPTER 13

BASE TIMER

This chapter provides an overview of the base timer, summarizes its register configuration and functions, and describes its operations.

- 13.1 Overview of the Base Timer
- 13.2 Block Diagrams of the Base Timer
- 13.3 Base Timer's Registers
- 13.4 Operations of the Base Timer
- 13.5 32-bit Mode Operations
- 13.6 Notes of Using the Base Timer
- 13.7 Base Timer Interrupts
- 13.8 Base Timer Description by Function Mode

13.1 Overview of the Base Timer

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section outlines the base timer in each function mode available.

■ Function Mode Bit Settings and Timer Function Modes Assigned

FMD2/FMD1/FMD0 bit Settings	Timer Function Mode
000 _B	Reset mode
001 _B	16-bit PWM timer
010 _B	16-bit PPG timer
011 _B	16/32-bit reload timer
100 _B	16/32-bit PWC timer

■ Reset Mode

Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance.

■ 16-bit PWM Timer

The 16-bit PWM timer mainly consists of a 16-bit down counter, a 16-bit data register buffered for period setting, a 16-bit compare register buffered for duty cycle setting, and a pin controller.

Period data and duty cycle data can be updated during timer operation as they are held in their buffered respective registers.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (CLKP) by 1, 4, 16, 128, and 256).

The PWM timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PWM timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16-bit PPG Timer

The 16-bit PPG timer mainly consists of a 16-bit down counter, a 16-bit data register for "H"-width setting, a 16-bit data register for "L"-width setting, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (CLKP) by 1, 4, 16, 128, and 256).

The PPG timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PPG timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16/32-bit Reload Timer

The 16/32-bit reload timer mainly consists of a 16-bit down counter, a 16-bit reload register, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (CLKP) by 1, 4, 16, 128, and 256).

The reload timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the reload timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16/32-bit PWC Timer

The 16/32-bit PWC timer mainly consists of a 16-bit up counter, a measurement input pin, and control registers.

The PWC timer measures the time between arbitrary events based on the pulse input from an external source.

The reference count clock can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (CLKP) by 1, 4, 16, 128, and 256).

Measurement modes "H" pulse width (\uparrow to \downarrow) / "L" pulse width (\downarrow to \uparrow)
 Rising period (\uparrow to \uparrow) / Falling period (\downarrow to \downarrow)
 Inter-edge measurement (\uparrow or \downarrow to \downarrow or \uparrow)

The PWC timer can generate an interrupt request upon completion of measurement.

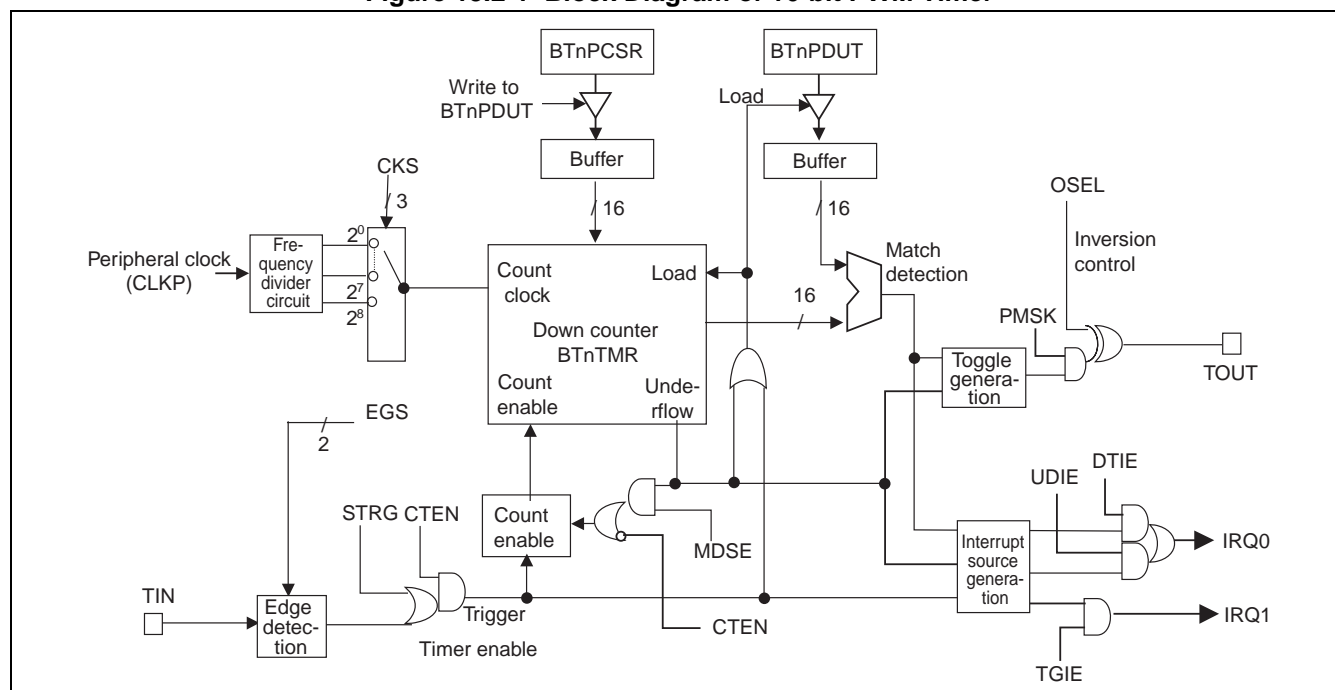
The PWC timer can select one-shot measurement or continuous measurement.

13.2 Block Diagrams of the Base Timer

This section provides a block diagram of the base timer in each function mode.

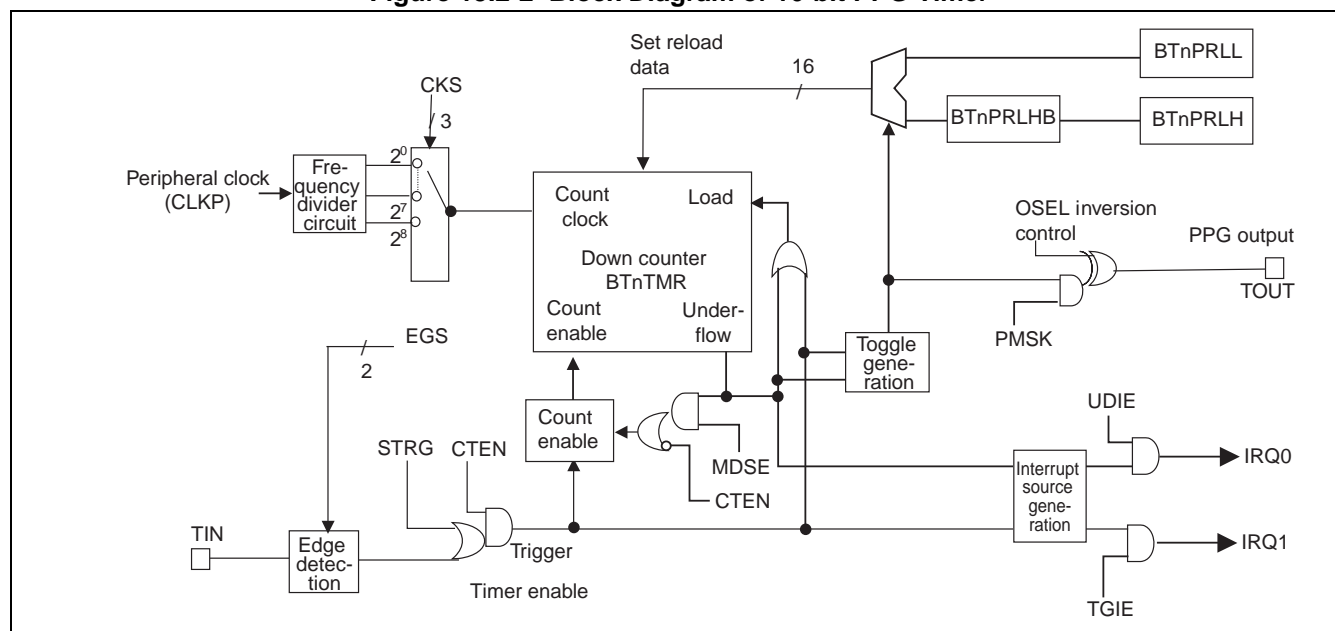
■ Block Diagram of 16-bit PWM Timer

Figure 13.2-1 Block Diagram of 16-bit PWM Timer



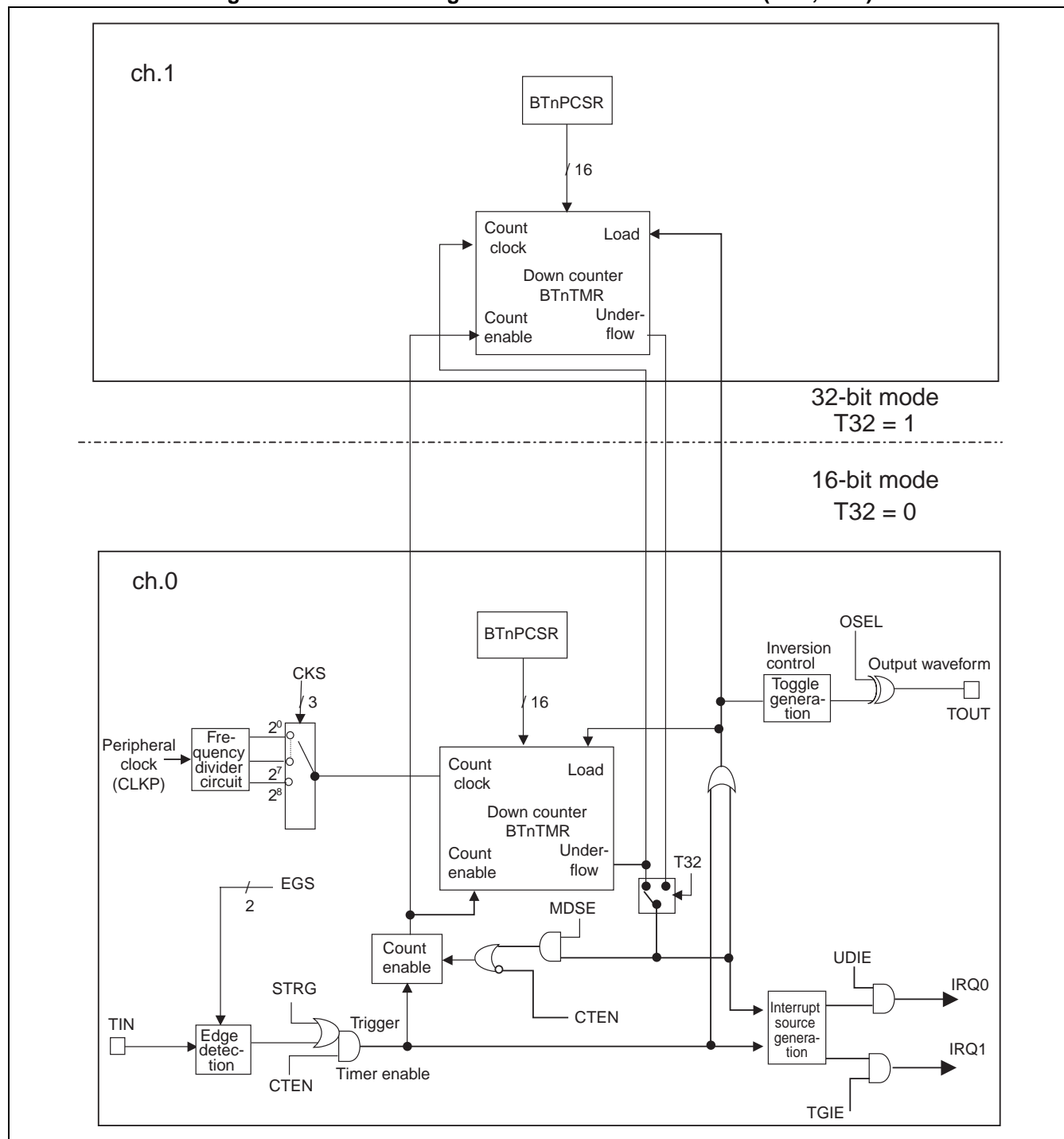
■ Block Diagram of 16-bit PPG Timer

Figure 13.2-2 Block Diagram of 16-bit PPG Timer



■ Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)

Figure 13.2-3 Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)

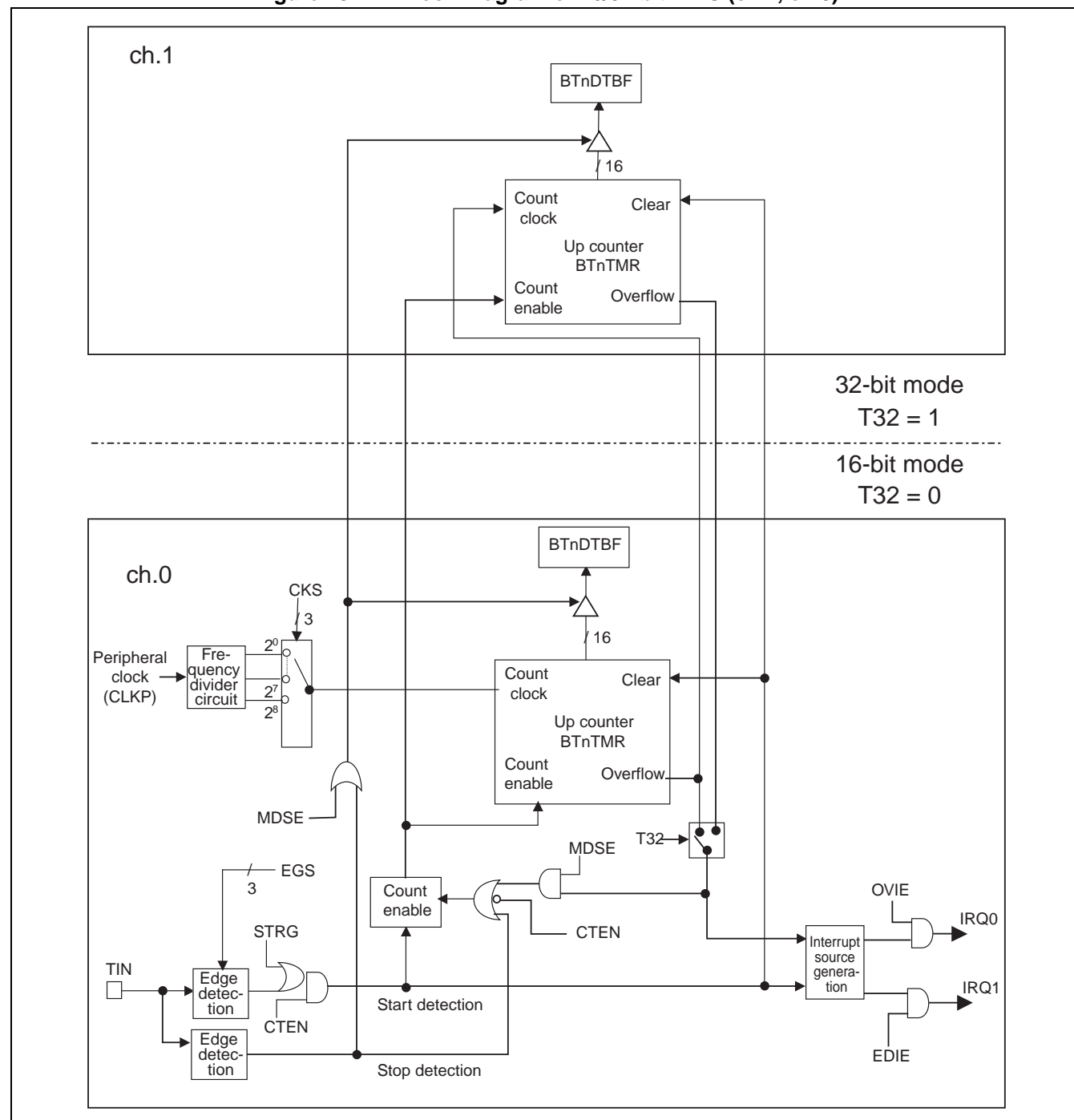


Note:

The reload timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, and between ch.4 and ch.5. No 32-bit operation is applicable to any other combination of channels.

■ Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)

Figure 13.2-4 Block Diagram of 16/32-bit PWC (ch.1, ch.0)



Note:

The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, and between ch.4 and ch.5. No 32-bit operation is applicable to any other combination of channels.

13.3 Base Timer's Registers

This section lists the registers used for the base timer and their bit configurations in each timer function mode.

■ List of Base Timer's Registers

Table 13.3-1 List of Base Timer's Registers

Function mode settings (FMD2, FMD1, FMD0)	Addresses		bit15	bit8	bit7	bit0
All modes	000162 _H 000582 _H 000592 _H 0005A2 _H	000163 _H 000583 _H 000593 _H 0005A3 _H	BTnTMCR (timer control register)			
All modes	-	000165 _H 000585 _H 000595 _H 0005A5 _H	-		BTnSTC (status control register)	
001 _B /010 _B /011 _B	000160 _H 000580 _H 000590 _H 0005A0 _H	000161 _H 000581 _H 000591 _H 0005A1 _H	BTnTMR (timer register)			
100 _B			-			
001 _B /011 _B	000168 _H 000588 _H 000598 _H 0005A8 _H	000169 _H 000589 _H 000599 _H 0005A9 _H	BTnPCSR (period setting register)			
010 _B			BTnPRL ("L"-width setting reload register)			
100 _B			-			
001 _B	00016A _H 00058A _H 00059A _H 0005AA _H	00016B _H 00058B _H 00059B _H 0005AB _H	BTnPDUT (duty setting register)			
010 _B			BTnPRLH ("H"-width setting reload register)			
011 _B			-			
100 _B			BTnDTBF (data buffer register)			

■ Bit Configurations in Each Timer Function Mode

Figure 13.3-1 Registers for 16-bit PWM Timer

Function mode setting								FMD=001 _B
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMCR
-	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0	(timer control register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
-	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG	
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	BTnSTC
-	TGIE	DTIE	UDIE	-	TGIR	DTIR	UDIR	(status control register)
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMR
								(timer register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnPCSR
								(period setting register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnPDUT
								(duty setting register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Figure 13.3-2 Registers for 16-bit PPG Timer

Function mode setting FMD=010 _B								
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMCR
-	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0	(timer control register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
-	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG	
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	BTnSTC
-	TGIE	-	UDIE	-	TGIR	-	UDIR	(status control register)
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMR
								(timer register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnPRLL
								("L"-width setting reload register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnPRLH
								("H"-width setting reload register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Figure 13.3-3 Registers for Reload Timer

Figure 10-3-3: Register for period timer

Function mode setting FMD=011_B

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMCR
-	CKS2	CKS1	CKS0	-	-	EGS1	EGS0	(timer control register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
T32	FMD2	FMD1	FMD0	OSEL	MDSE	CTEN	STRG	
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	BTnSTC
-	TGIE	-	UDIE	-	TGIR	-	UDIR	(status control register)
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMR
								(timer register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnPCSR
								(period setting register)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Figure 13.3-4 Registers for PWC Timer

Function mode setting FMD=100_B

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnTMCR
-	CKS2	CKS1	CKS0	-	EGS2	EGS1	EGS0	(timer control register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T32	FMD2	FMD1	FMD0	-	MDSE	CTEN	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	BTnSTC
ERR	EDIE	-	OVIE	-	EDIR	-	OVIR	(status control register)

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	BTnDTBF
								(data buffer register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

13.4 Operations of the Base Timer

This section introduces how the base timer operates in each timer function mode.

■ Operations of the Base Timer

● Reset mode

Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance. If you set this mode for even-numbered channels in 32-bit mode, odd-numbered channels are reset as well at the same time. Thus you do not have to set the reset mode for odd-numbered channels.

● 16-bit PWM timer

The 16-bit PWM timer starts decrementing its counter by the value set as a period when triggered to start. The PWM timer then sets the output to the "L" level first and, if the 16-bit down counter value matches the value set in the duty setting register, inverts the output to the "H" level. Then it inverts the output back to the "L" level when the counter causes an underflow subsequently. This generates a waveform with an arbitrary period and duty cycle.

● 16-bit PPG timer

The 16-bit PPG timer starts decrementing its counter by the value set in the "L"-width setting reload register when triggered to start. The PPG timer then sets the output to the "L" level first and inverts the output back to the "H" level when the counter causes an underflow. The PPG timer continuously decrements the counter by the value set in the "H"-width setting reload register and inverts the output level to "L" when the counter causes an underflow. This generates a waveform with arbitrary "L" and "H" widths.

● 16-bit reload timer

The 16-bit reload timer starts decrementing its 16-bit down counter by the value set as a period when triggered to start. When the down counter causes an underflow, the interrupt flag is set. Depending on the MDSE bit setting, the output level either toggles, or is inverted, between "H" and "L" each time the counter causes an underflow or becomes "H" when the counter starts counting and "L" when it causes an underflow.

● 32-bit reload timer

The 32-bit reload timer is the same in basic operation as the 16-bit reload timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit timers, respectively, interrupt control and output wave control follow their respective settings for the even-numbered channel. To set the period, write the value to the upper register (odd-numbered channel) first and then to the lower register (even-numbered channel).

To obtain the timer value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

Note:

The PWM, PPG, and reload timers can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, and between ch.4 and ch.5. No 32-bit operation is applicable to any other combination of channels.

● 16-bit PWC timer

The 16-bit PWC timer starts the 16-bit up counter upon input of a pre-set measurement start edge and stops the counter upon detection of a measurement stop edge. The count value between the two edges is written to the data buffer register as a pulse width.

● 32-bit PWC timer

The 32-bit PWC timer is the same in basic operation as the 16-bit PWC timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit counters, respectively, interrupt control follows the setting for the even-numbered channel. To obtain the measured value or count value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

Note:

The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, and between ch.4 and ch.5. No 32-bit operation is applicable to any other combination of channels.

13.5 32-bit Mode Operations

The reload timer and PWC timer can operate in 32-bit mode using a pair of channels. This section describes the basic functions and operations of 32-bit mode.

■ Functions of 32-bit Mode

The 32-bit mode combines two channels of base timer into a 32-bit data reload timer or PWC timer. Either 32-bit timer allows the timer/counter value to be read even during operation as it takes the upper 16-bit timer/counter value of the odd-numbered channel also when reading the lower 16-bit timer/counter value of the even-numbered channel.

■ Setting the 32-bit Mode

First, set the FMD2, FMD1, and FMD0 bits in the BTnTMCR register for the even-numbered channel to "000_B" to reset in reset mode. Then, select the reload timer or PWC timer and set its operations in the same way as in 16-bit mode. At this time, write "1" to the T32 bit in the BTnTMCR register to enter the 32-bit operation mode. The T32 bit for the odd-numbered channel must be left containing "0". Neither the reset mode setting is required for the odd-numbered channel. To use the base timer as the reload timer, set the period setting register for the odd-numbered channel to the upper 16-bit reload value among 32 bits and set the period setting register for the even-numbered channel to the lower 16-bit reload value.

As the transition to 32-bit operation mode takes place the moment is written to the T32 bit, the setting must be changed with counting halted on both of the channels.

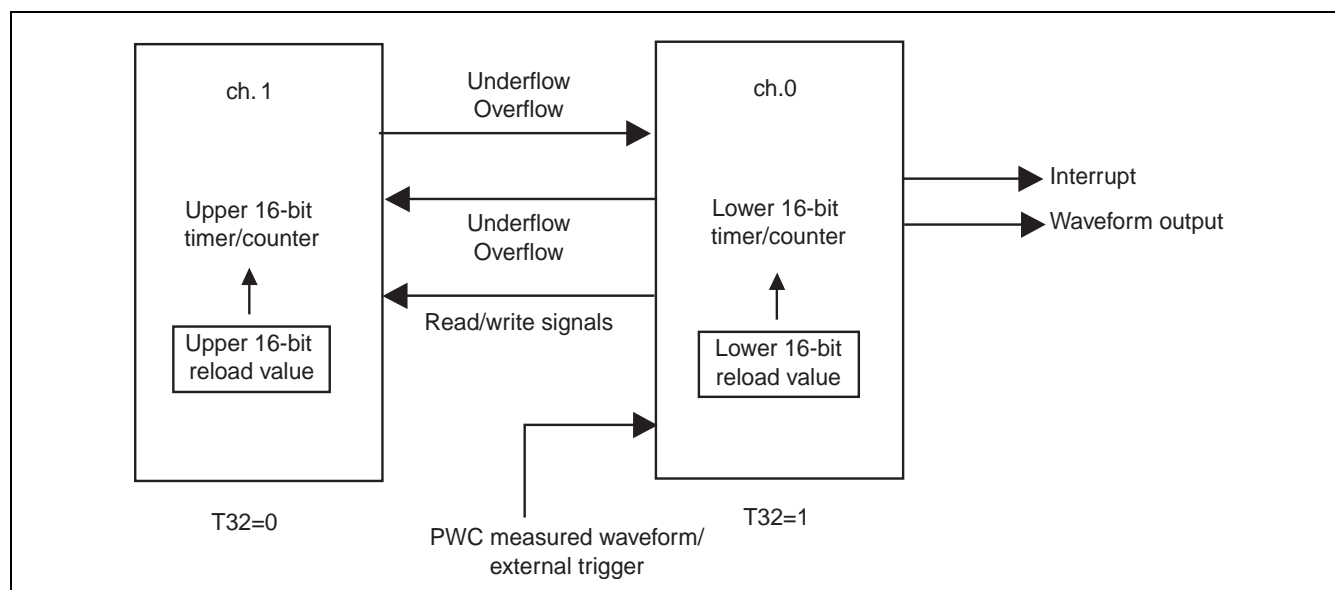
To switch from 32-bit mode to 16-bit mode, set the FMD2, FMD1, and FMD0 bits in the BTnTMCR register for the even-numbered channel to "000_B" to reset the states of both of the even-numbered and odd-numbered channels in reset mode. Then set each channel for operation in 16-bit mode.

■ Operations in 32-bit Mode

When the reload timer or PWC timer is started in 32-bit mode under control of the even-numbered channel, the timer/counter of the even-numbered channel operates as the lower 16-bit timer/counter and the timer/counter of the odd-numbered channel operates as the upper 16-bit one.

In 32-bit mode, the base timer follows the settings for the even-numbered channel while ignoring those for the odd-numbered channel (except the period setting register when serving as the reload timer). Even for the timer start, waveform output, and interrupt signal settings, the even-numbered channel overrides the odd-numbered channel (odd-numbered channel is always masked at "L").

The following example shows a PWC configuration using ch.0 and ch.1.



Note:

The reload timer or PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, and between ch.4 and ch.5. No 32-bit operation is applicable to any other combination of channels.

13.6 Notes of Using the Base Timer

This section summarizes the notes on using the base timer.

■ Common Notes on Using Each Type of Timer

● Notes on setting through programming

- The following bits in the BTnTMCR register must not be updated during operation. Be sure to update them before starting the base timer or after stopping it.

[bit14, bit13, bit12]	CKS2, CKS1, CKS0	: Clock select bits
[bit10, bit9, bit8]	EGS2, EGS1, EGS0	: Measurement edge select bits
[bit7]	T32	: 32-bit timer select bit
		(Used with the reload timer or PWC timer selected)
[bit6, bit5, bit4]	FMD2, FMD1, FMD0	: Timer function mode select bits
[bit2]	MDSE	: Measurement mode (one-shot/continuous) select bit
- If you set the FMD2, FMD1, and FMD0 bits in the BTnTMCR register to "000_B" to enter the reset mode, all the registers of the base timer are initialized and thus they must be set all over again.
- If you set the FMD2, FMD1, and FMD0 bits in the BTnTMCR register to "000_B" to enter the reset mode, the other bits in the BTnTMCR register are initialized with their settings ignored.

■ Notes on Using the 16-bit PWM/PPG/Reload Timer

● Notes on setting through programming

- When the interrupt request flag is attempted to be set and cleared at the same timing, the flag set action overrides the flag clear action.
- When the down counter is attempted to load and count at the same timing, the load action overrides the count action.
- Set the FMD2, FMD1, and FMD0 bits in the BTnTMCR register to select the timer function mode before setting the period, duty cycle, "H" width, and "L" width.
- If a restart is detected when counting is completed in one-shot mode, the counter is restarted with the count value reloaded.

■ Notes on Using the PWC Timer

● Notes on setting through programming

- Writing "1" to the counting enable bit (CTEN) clears the counter, nullifying the data existing in the counter before counting is enabled.
- If you set the PWC mode (FMD = 100_B) after a system reset or in reset mode and enables measurement (CTEN = 1) at the same time, the timer may operate according to the immediately preceding measurement signal.
- If a measurement start edge is detected the moment a restart is set in continuous measurement mode, the timer immediately starts counting from "0001_H".
- An attempt to restart the timer after starting counting can result as follows, depending on that timing:
 - If the attempt is made at a measurement end edge in one-shot pulse width measurement mode:
Although the timer is restarted and waits for an measurement start edge, the measurement end flag (EDIR) is set.
 - If the attempt is made at a measurement end edge in continuous pulse width measurement mode:
Although the timer is restarted and waits for a measurement start edge, the measurement end flag (EDIR) is set and the current measurement result is transferred to the BTnDTBF register.

When restarting the timer during operation, control interrupts while paying attention to the behaviors of flags.

13.7 Base Timer Interrupts

This section lists the interrupt request flags, interrupt enable bits, and interrupt factors for the base timer in each timer function mode.

■ Interrupt Control Bits and Interrupt Factors by Timer Function Mode

Table 13.7-1 lists the interrupt control bits and interrupt factors for the base timer in each timer function mode.

Table 13.7-1 Interrupt Control Bits and Interrupt Factors in Each Timer Function Mode

	Status control register (BTnSTC)			
	Interrupt request flag bits	Interrupt request enable bits	Interrupt factors	IRQ
PWM timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	DTIR: bit1	DTIE: bit5	Duty match detection	
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PPG timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
Reload timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PWC timer function	OVIR: bit0	OVIE: bit4	Overflow detection	IRQ0
	EDIR: bit2	EDIE: bit6	Measurement end detection	IRQ1

13.8 Base Timer Description by Function Mode

This section describes each function of the base timer.

■ Base Timer Function

- PWM function
- PPG function
- Reload timer function
- PWC function

13.8.1 PWM Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWM timer.

- Timer Control Register (BTnTMCR) for PWM Timer
- PWM Period Setting Register (BTnPCSR)
- PWM Duty Setting Register (BTnPDUT)
- Timer Register (BTnTMR)
- 16-bit PWM Timer Operation
- One-shot Operation
- Interrupt Factors and Timing Chart
- Output Waveforms

13.8.1.1 Timer Control Register (BTnTMCR) for PWM Timer

The timer control register (BTnTMCR) controls the PWM timer. Keep in mind that the register contains bits which cannot be updated with the PWM timer operating.

■ Timer Control Register (BTnTMCR Upper Byte)

Figure 13.8-1 Timer Control Register (BTnTMCR Upper Byte)

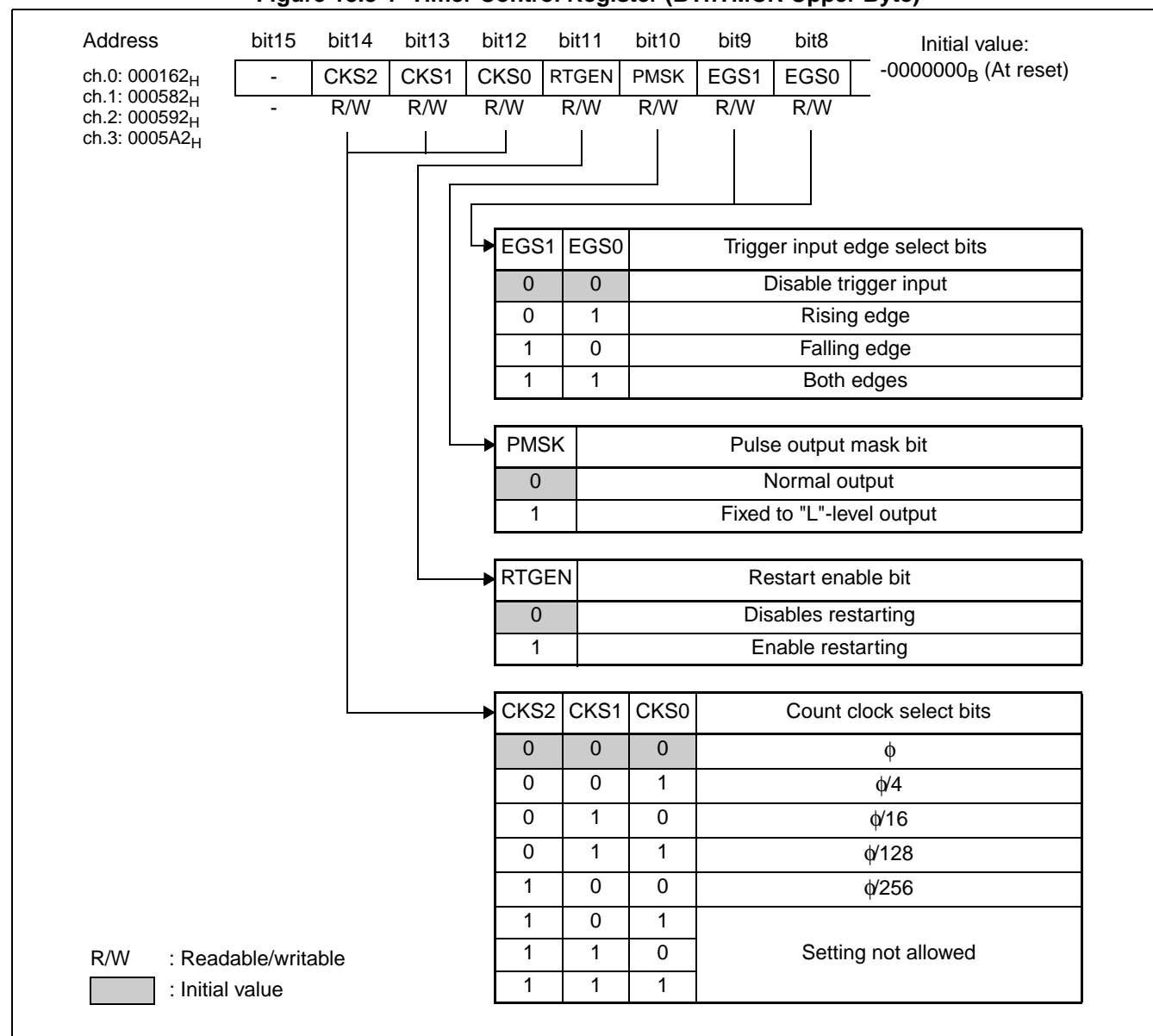


Table 13.8-1 Timer Control Register (BTnTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value of this bit is undefined. Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	<ul style="list-style-type: none"> Select the count clock for the 16-bit down counter. The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	Enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	<ul style="list-style-type: none"> Controls the PWM output waveform level. When this bit is "0", the PWM waveform is output as it is. When the bit is "1", the PWM output is masked to the "L" level irrespective of the period and duty cycle. <p>Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PWM output to the "H" level.</p>
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	<ul style="list-style-type: none"> Select the effective edge of the input waveform as an external trigger to set the trigger condition. When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ **Timer Control Register (BTnTMCR Lower Byte)**

Figure 13.8-2 Timer Control Register (BTnTMCR Lower Byte)

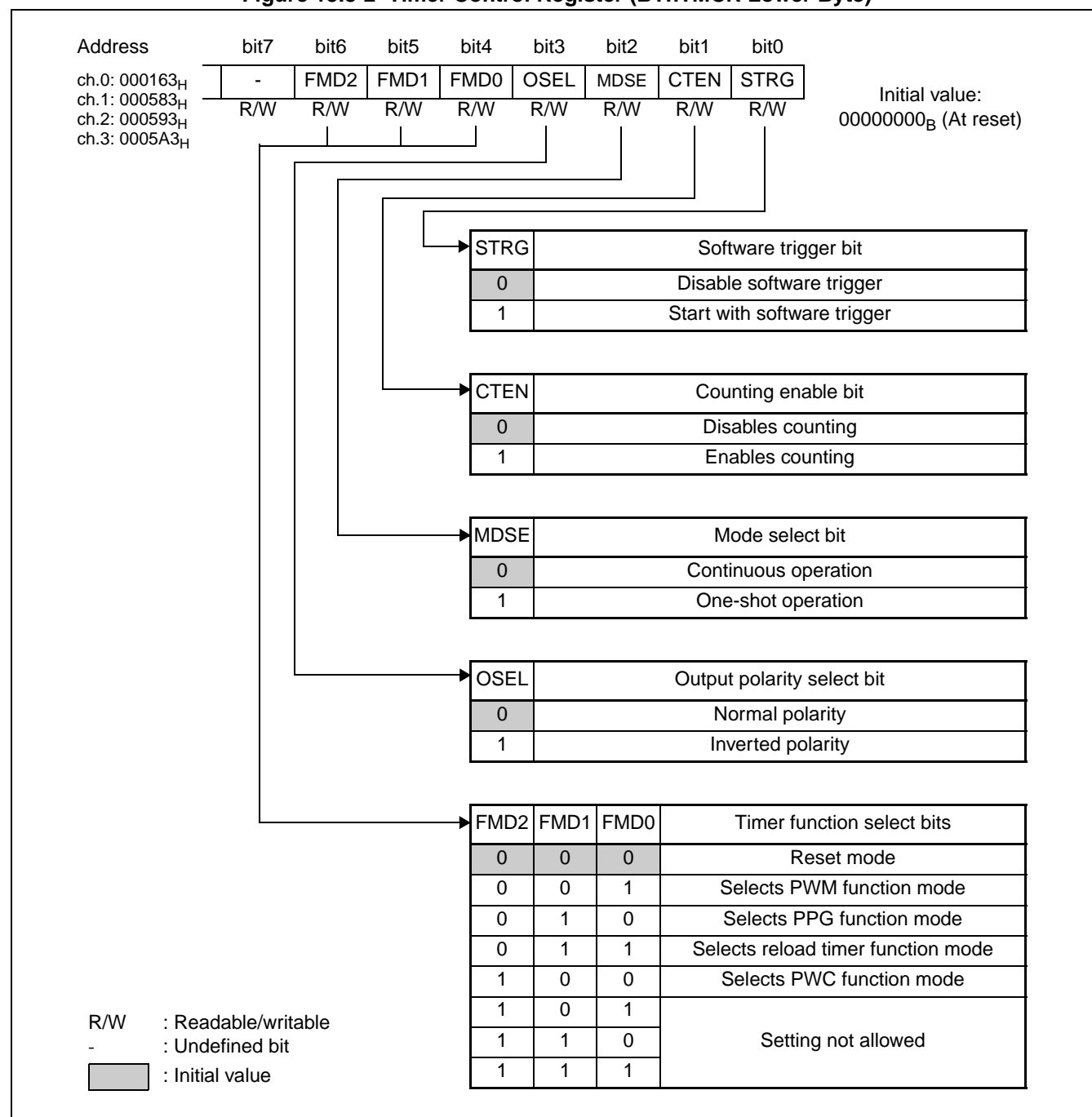














Table 13.8-2 Timer Control Register (BTnTMCR Lower Byte)

Bit name		Function												
bit7	Undefined bit	<ul style="list-style-type: none">The value read is "0"When writing to this bit, write "0".												
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none">These bits select the timer function mode.Setting the FMD2, FMD1, and FMD0 bits to "001_B" selects the PWM function mode.The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.												
bit3	OSEL: Output polarity select bit	<p>Selects the polarity of PWM output.</p> <table><tr><th>Polarity</th><th>After reset</th><th>Duty match</th><th>Underflow</th></tr><tr><td>Normal</td><td>"L" output</td><td></td><td></td></tr><tr><td>Inverted</td><td>"H" output</td><td></td><td></td></tr></table>	Polarity	After reset	Duty match	Underflow	Normal	"L" output			Inverted	"H" output		
Polarity	After reset	Duty match	Underflow											
Normal	"L" output													
Inverted	"H" output													
bit2	MDSE: Mode select bit	<ul style="list-style-type: none">Selects continuous pulse output or one-shot pulse output.The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.												
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none">This bit enables the down counter.Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.												
bit0	STRG: Software trigger bit	<ul style="list-style-type: none">Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <p>Note:</p> <p>Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</p> <ul style="list-style-type: none">The value read from the STRG bit is always "0". <p>Note:</p> <p>Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</p>												

■ **Status Control Register (BTnSTC)**

Figure 13.8-3 Status Control Register (BTnSTC)

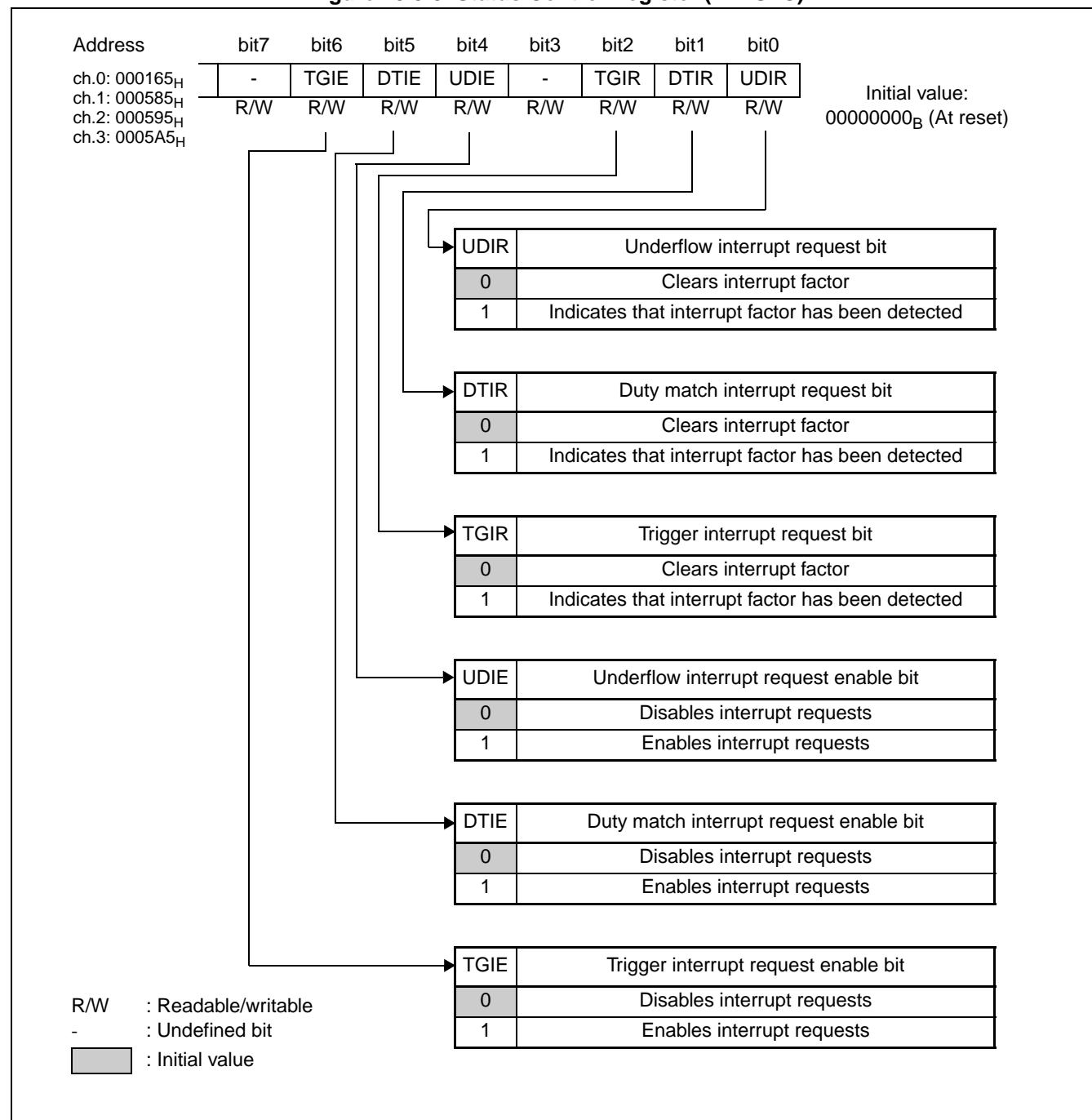


Table 13.8-3 Status Control Register (BTnSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> Controls bit2: TGIR interrupt requests. Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	DTIE: Duty match interrupt request enable bit	<ul style="list-style-type: none"> Controls bit1: DTIR interrupt requests. Setting the DTIR bit (bit1) with the DTIE bit enabling duty match interrupt requests generates an interrupt request to the CPU.
bit4	UDIE: Underflow interrupt request enable bit	<ul style="list-style-type: none"> Controls bit0: UDIR interrupt requests. Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> The TGIR bit is set to "1" upon detection of a software trigger or trigger input. Writing "0" to the TGIR bit clears it. Writing "1" to the TGIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	DTIR: Duty match interrupt request bit	<ul style="list-style-type: none"> The DTIR bit is set to "1" when the count value matches the duty cycle setting. Writing "0" to the DTIR bit clears it. Writing "1" to the DTIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit0	UDIR: Underflow interrupt request bit	<ul style="list-style-type: none"> The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to FFFF_H. Writing "0" to the UDIR bit clears it. Writing "1" to the UDIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

13.8.1.2 PWM Period Setting Register (BTnPCSR)

The PWM period setting register (BTnPCSR) is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

■ Bit Configuration of the PWM Period Setting Register (BTnPCSR)

Figure 13.8-4 shows the bit configuration of the PWM period setting register (BTnPCSR).

Figure 13.8-4 Bit Configuration of the PWM Period Setting Register (BTnPCSR)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 000168 _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 000588 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 000598 _H									
ch.3: 0005A8 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: XXXXXXXX _B (At reset)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable
X : Undefined value

The BTnPCSR register is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

After writing to the period setting register to initially set or update it, be sure to write to the duty setting register.

- Access the BTnPCSR register using 16-bit data.
- Set the PWM period using the BTnPCSR register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTnTMCR register.

13.8.1.3 PWM Duty Setting Register (BTnPDUT)

The PWM duty setting register (BTnPDUT) is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

■ Bit Configuration of the PWM Duty Setting Register (BTnPDUT)

Figure 13.8-5 shows the bit configuration of the PWM duty setting register (BTnPDUT).

Figure 13.8-5 Bit Configuration of the PWM Duty Setting Register (BTnPDUT)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 00016A _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 00058A _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 00059A _H									
ch.3: 0005AA _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: XXXXXXXX _B (At reset)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable									
X : Undefined value									

The BTnPDUT register is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

If you set the period setting and duty setting registers to the same value, the output level is all "H" in normal polarity or all "L" in inverted polarity.

Do not set the BTnPDUT register to a value greater than the value of the PSCR register, or PWM output will be undefined.

- Access the BTnPDUT register using 16-bit data.
- Set the PWM duty cycle using the BTnPDUT register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTnTMCR register.

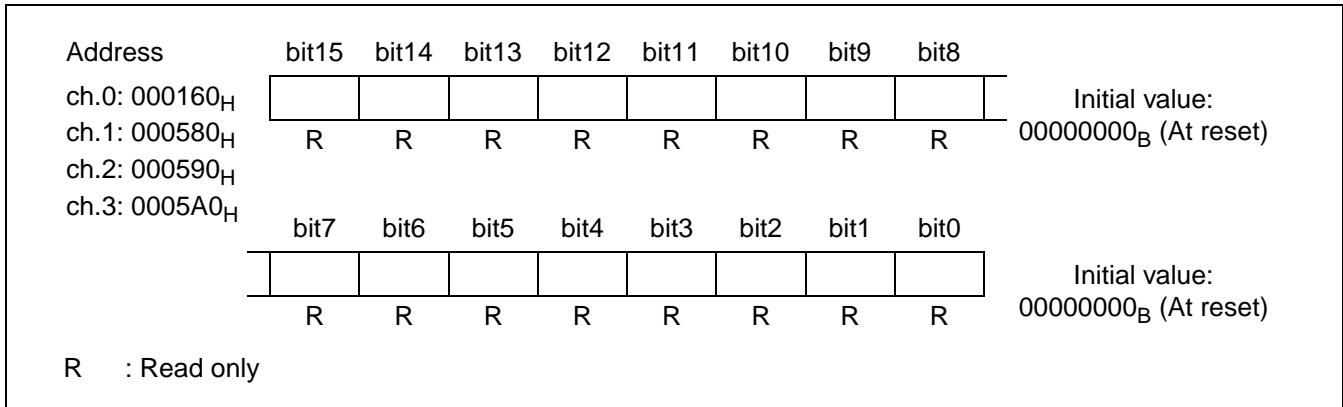
13.8.1.4 Timer Register (BTnTMR)

The timer register (BTnTMR) allows the value of the 16-bit down counter to be read from.

■ Bit Configuration of the Timer Register (BTnTMR)

Figure 13.8-6 shows the bit configuration of the PWM timer register (BTnTMR).

Figure 13.8-6 Bit Configuration of the Timer Register (BTnTMR)



The BTnTMR register allows the value of the 16-bit down counter to be read from.

Note:

Access the BTnTMR register using 16-bit data.

13.8.1.5 16-bit PWM Timer Operation

In PWM timer mode, a waveform having a specified period can be output either in single shots or continuously after detection of a trigger.

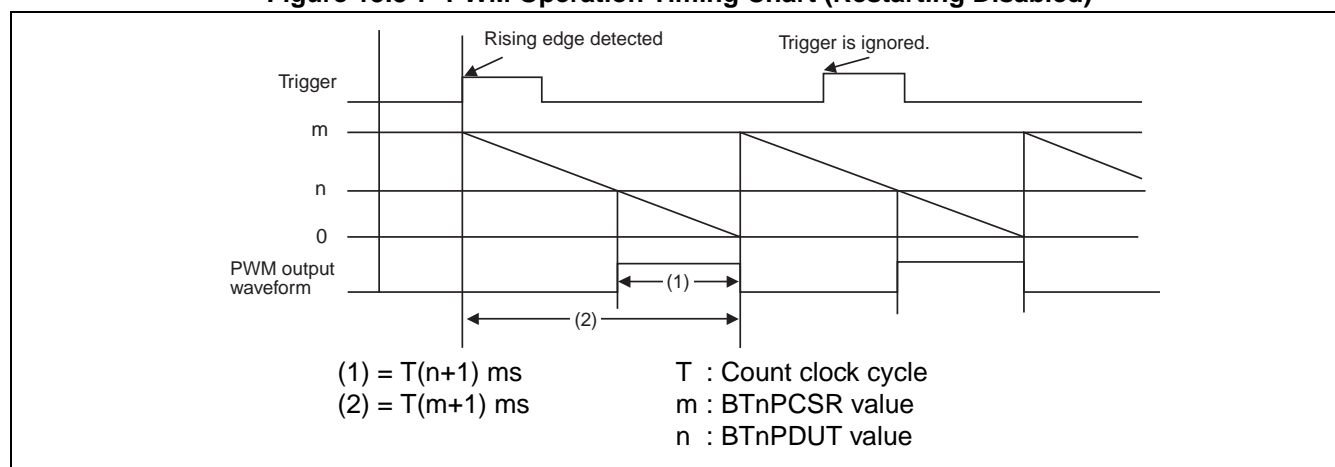
The period of output pulses can be controlled by changing the BTnPCSR value.

The duty ratio can be controlled by changing the BTnPDUT value. After writing data to the BTnPCSR register, be sure to write to the BTnPDUT register as well.

■ Continuous Operation

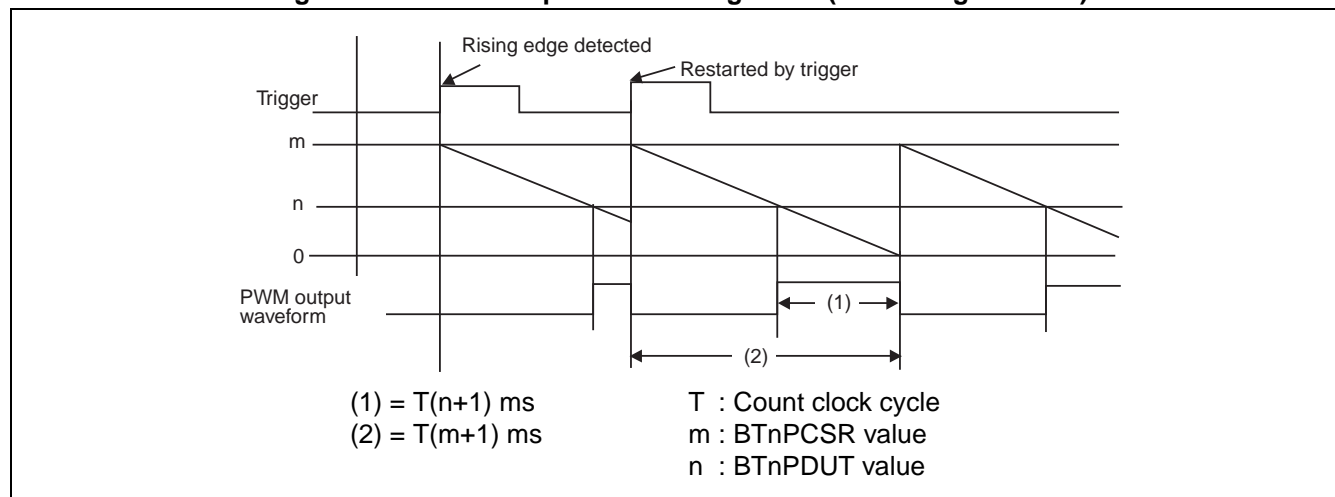
- When restarting is disabled (RTGEN = 0)

Figure 13.8-7 PWM Operation Timing Chart (Restarting Disabled)



- When restarting is enabled (RTGEN = 1)

Figure 13.8-8 PWM Operation Timing Chart (Restarting Enabled)



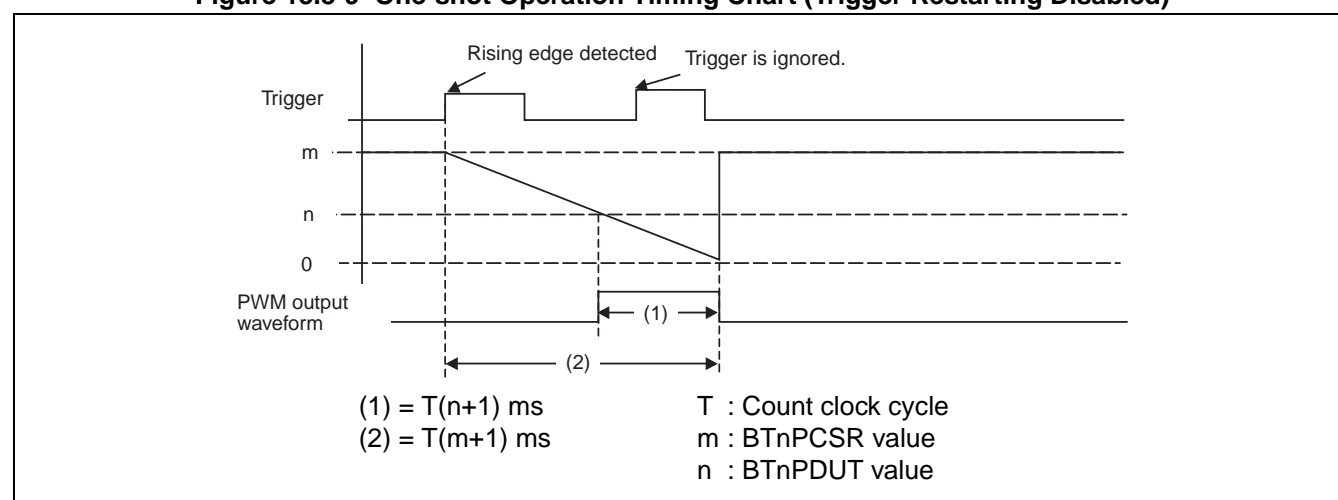
13.8.1.6 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ One-shot Operation

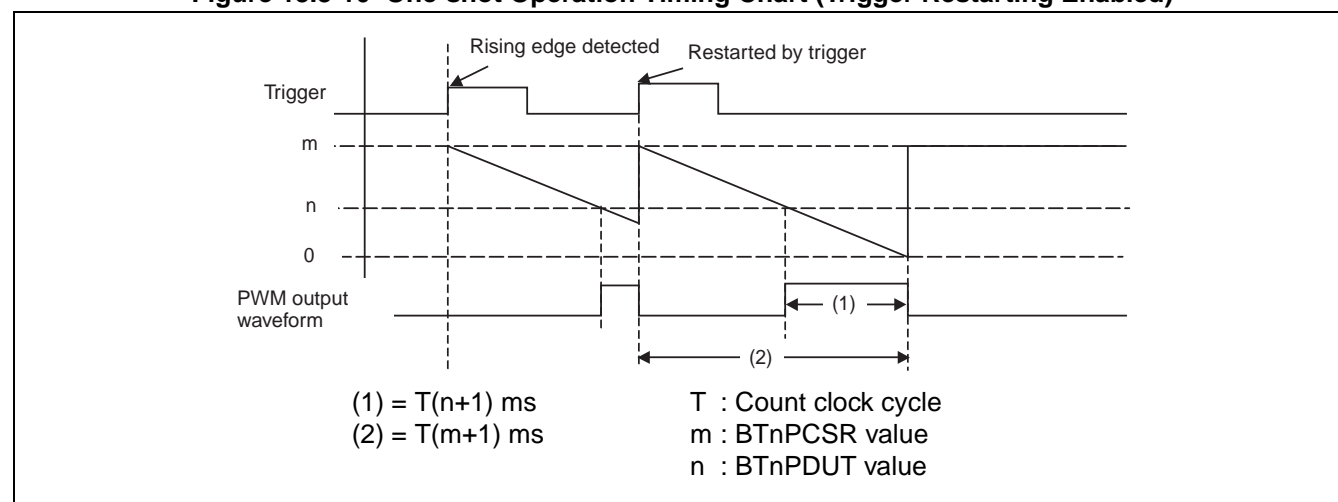
- When restarting is disabled (RTGEN = 0)

Figure 13.8-9 One-shot Operation Timing Chart (Trigger Restarting Disabled)



- When restarting is enabled (RTGEN = 1)

Figure 13.8-10 One-shot Operation Timing Chart (Trigger Restarting Enabled)



13.8.1.7 Interrupt Factors and Timing Chart

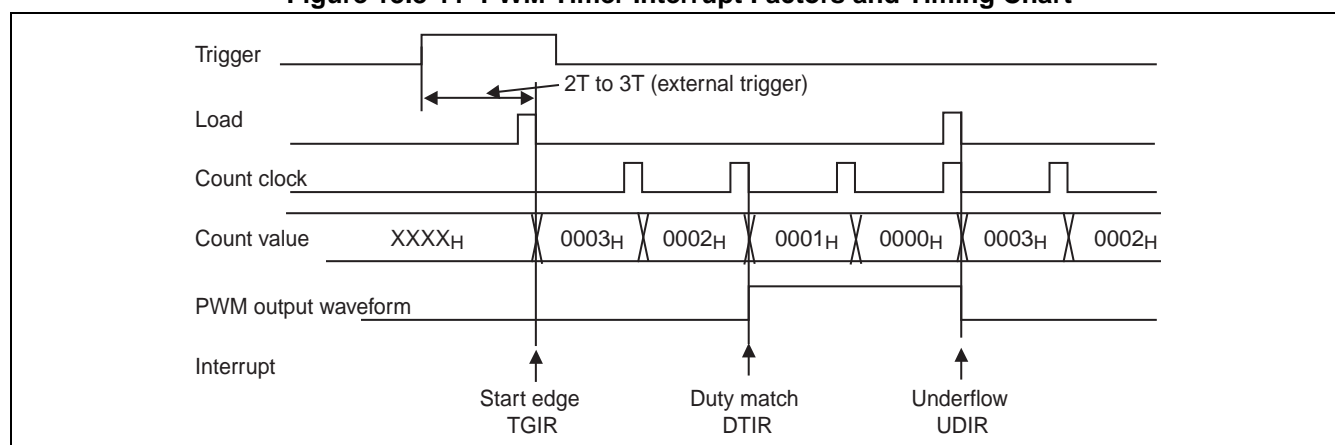
This section provides the interrupt factors and timing chart.

■ Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (CLKP) cycle) until the counter value is loaded after the input of the trigger.

Figure 13.8-11 shows the interrupt factors and timing chart, assuming "period setting" = 3 and "duty value" = 1.

Figure 13.8-11 PWM Timer Interrupt Factors and Timing Chart



13.8.1.8 Output Waveforms

This section illustrates PWM output.

■ PWM Output at All "L" or All "H" Level

Figure 13.8-12 and Figure 13.8-13 illustrate how to provide PWM output at all "L" and all "H" levels, respectively.

Figure 13.8-12 Example of PWM Output at All "L" Level

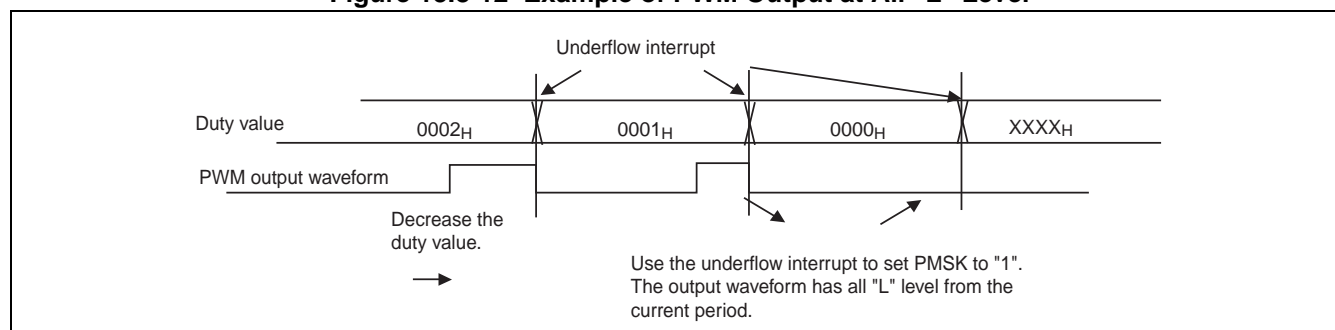
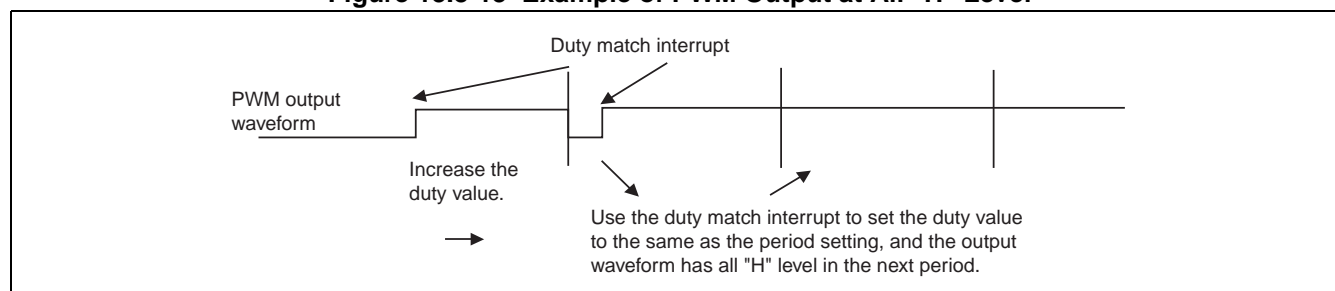


Figure 13.8-13 Example of PWM Output at All "H" Level



13.8.2 PPG Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PPG timer.

- Timer Control Register (BTnTMCR) for PPG Timer
- "L"-width Setting Reload Register (BTnPRLL)
- "H"-width Setting Reload Register (BTnPRLH)
- Timer Register (BTnTMR)
- 16-bit PPG Timer Operation
- Continuous Operation
- One-shot Operation
- Interrupt Factors and Timing Chart

13.8.2.1 Timer Control Register (BTnTMCR) for PPG Timer

The timer control register (BTnTMCR) controls the PPG timer. Keep in mind that the register contains bits which cannot be updated with the PPG timer operating.

■ Timer Control Register (BTnTMCR Upper Byte)

Figure 13.8-14 Timer Control Register (BTnTMCR Upper Byte)

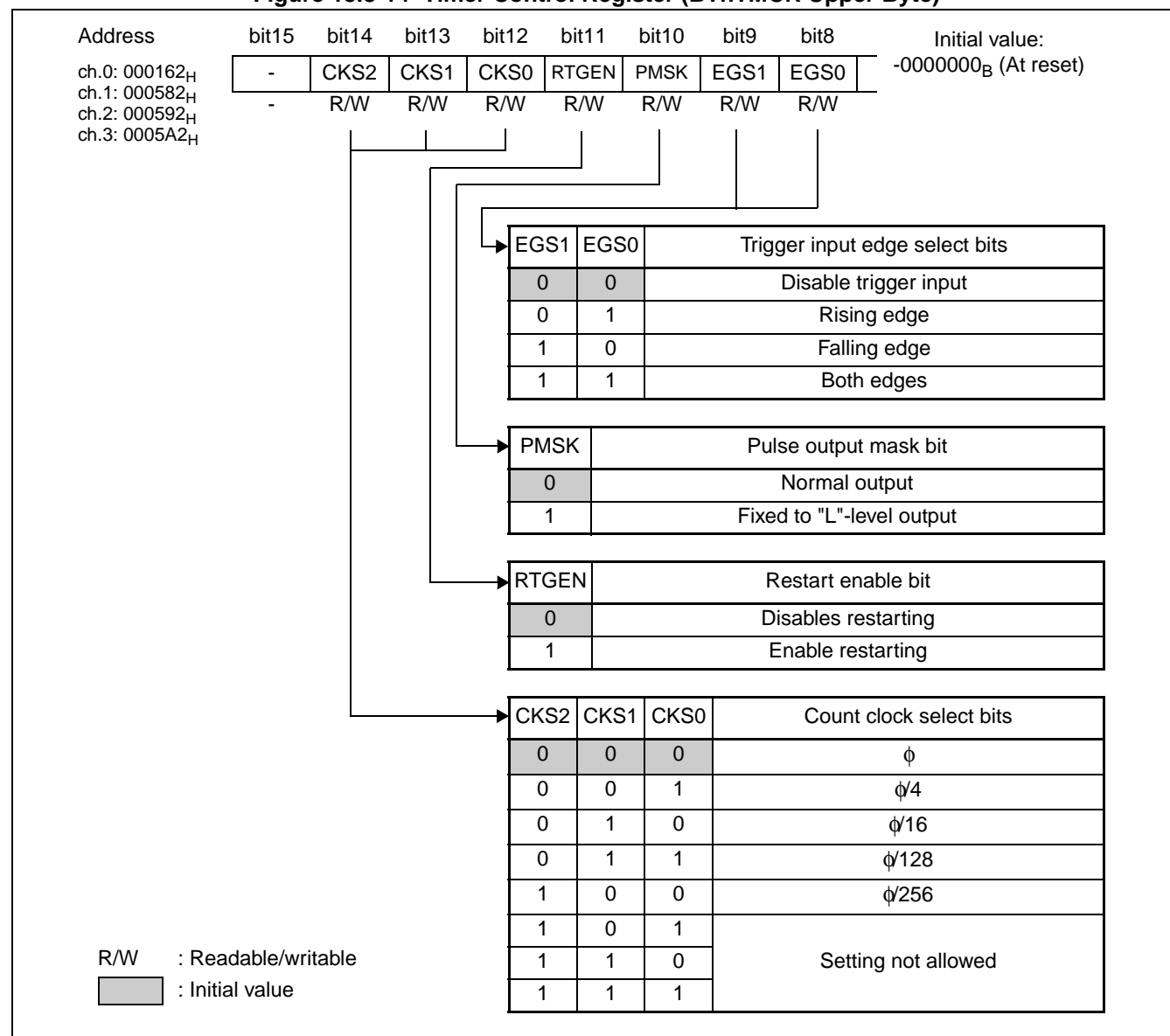


Table 13.8-4 Timer Control Register (BTnTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value of this bit is undefined. Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	<ul style="list-style-type: none"> Select the count clock for the 16-bit down counter. The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	This bit enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	<ul style="list-style-type: none"> Controls the PPG output waveform level. When this bit is "0", the PPG waveform is output as it is. When the bit is "1", the PPG output is masked to the "L" level irrespective of the "H" and "L" width settings. <p>Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PPG output to the "H" level.</p>
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	<ul style="list-style-type: none"> Select the effective edge of the input waveform as an external trigger to set the trigger condition. When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ **Timer Control Register (BTnTMCR Lower Byte)**

Figure 13.8-15 Timer Control Register (BTnTMCR Lower Byte)

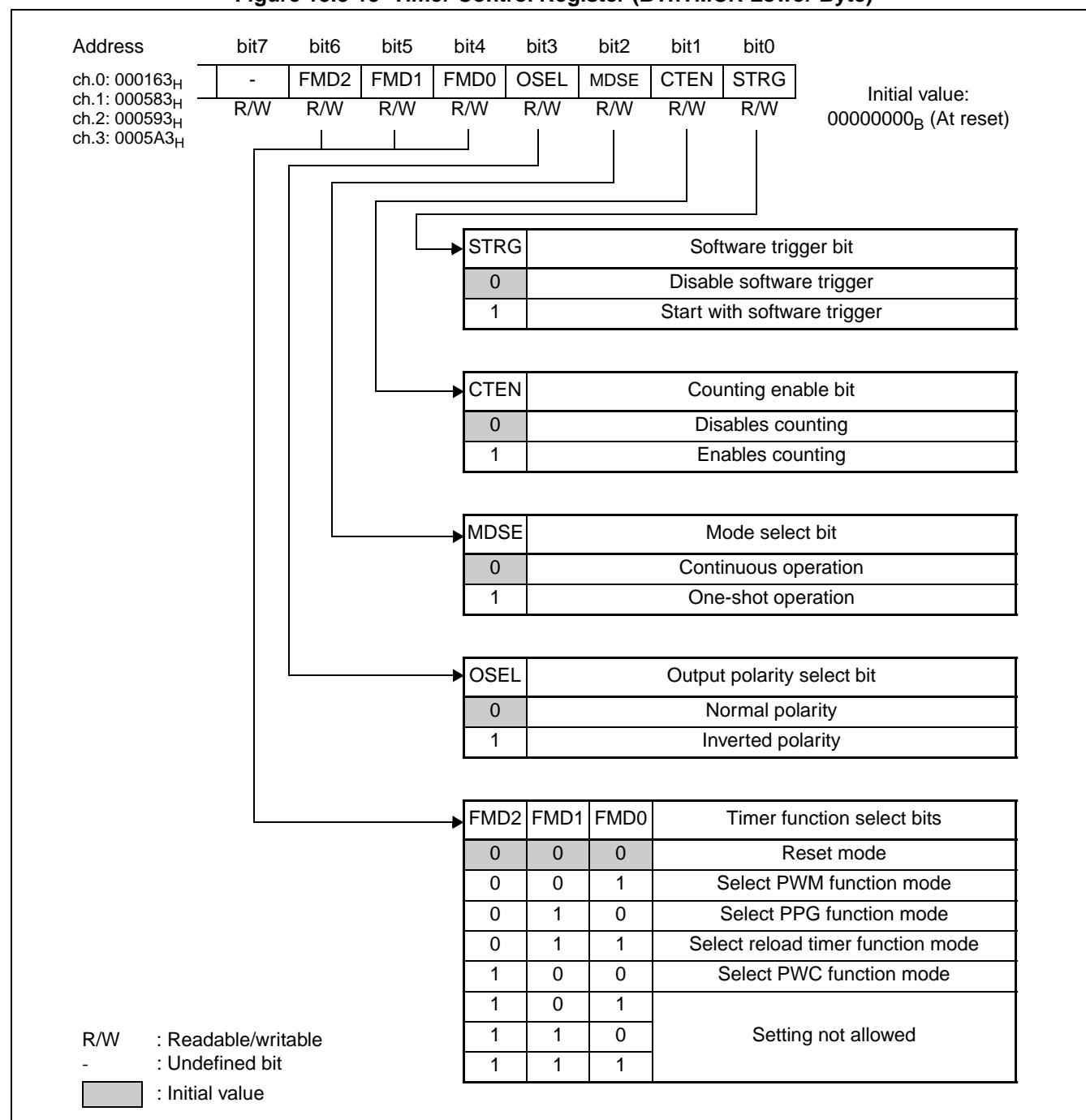














Table 13.8-5 Timer Control Register (BTnTMCR Lower Byte)

Bit name		Function												
bit7	Undefined bit	<ul style="list-style-type: none">The value read is "0"When writing to this bit, write "0".												
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none">These bits select the timer function mode.Setting the FMD2, FMD1, and FMD0 bits to "010_B" selects the PPG function mode.The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.												
bit3	OSEL: Output polarity select bit	<ul style="list-style-type: none">Selects the polarity of PPG output. <table><tr><th>Polarity</th><th>After reset</th><th>End of "L"-width counting</th><th>End of "H"-width counting</th></tr><tr><td>Normal</td><td>"L" output</td><td></td><td></td></tr><tr><td>Inverted</td><td>"H" output</td><td></td><td></td></tr></table>	Polarity	After reset	End of "L"-width counting	End of "H"-width counting	Normal	"L" output			Inverted	"H" output		
Polarity	After reset	End of "L"-width counting	End of "H"-width counting											
Normal	"L" output													
Inverted	"H" output													
bit2	MDSE: Mode select bit	<ul style="list-style-type: none">Selects continuous pulse output or one-shot pulse output.The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.												
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none">This bit enables the down counter.Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.												
bit0	STRG: Software trigger bit	<ul style="list-style-type: none">Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <p>Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</p> <ul style="list-style-type: none">The value read from the STRG bit is always "0". <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</p>												

■ **Status Control Register (BTnSTC)**

Figure 13.8-16 Status Control Register (BTnSTC)

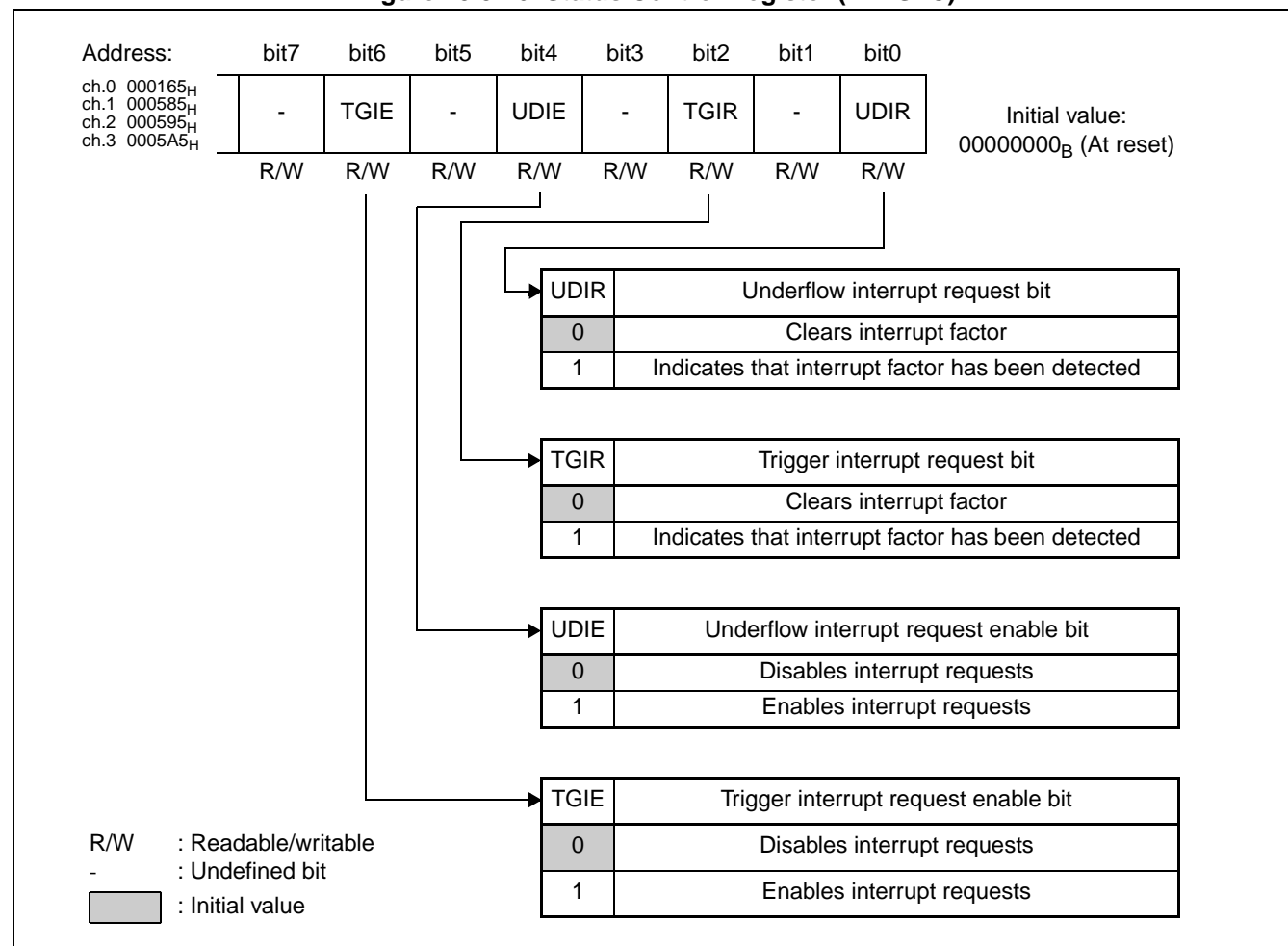


Table 13.8-6 Status Control Register (BTnSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> Controls bit2: TGIR interrupt requests. Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	<ul style="list-style-type: none"> Controls bit0: UDIR interrupt requests. Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> The TGIR bit is set to "1" upon detection of a software trigger or trigger input. Writing "0" to the TGIR bit clears it. Writing "1" to the TGIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	<ul style="list-style-type: none"> The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to FFFF_H during counting from the value set as the "H" width. Writing "0" to the UDIR bit clears it. Writing "1" to the UDIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

13.8.2.2 "L"-width Setting Reload Register (BTnPRLL)

The "L"-width setting reload register (BTnPRLL) is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

■ Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)

Figure 13.8-17 shows the bit configuration of the "L"-width setting reload register (BTnPRLL).

Figure 13.8-17 Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 000168 _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 000588 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 000598 _H									
ch.3: 0005A8 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: XXXXXXXX _B (At reset)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable
X : Undefined value

The BTnPRLL register is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

- Access the BTnPRLL register using 16-bit data.
- Set the "L" width using the BTnPRLL register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTnTMCR register.

13.8.2.3 "H"-width Setting Reload Register (BTnPRLH)

The "H"-width setting reload register (BTnPRLH) is a buffered register for setting the "H" width of PPG output waveforms. Transfer from the BTnPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

■ Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)

Figure 13.8-18 shows the bit configuration of the "H"-width setting reload register (BTnPRLH).

Figure 13.8-18 Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 00016A _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 00058A _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 00059A _H									
ch.3: 0005AA _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: XXXXXXXX _B (At reset)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable									
X : Undefined value									

The BTnPRLH register is used to set the "H" width of PPG output waveforms. Transfer from the BTnPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

- Access the BTnPRLH register using 16-bit data.
- Set the "H" width using the BTnPRLH register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTnTMCR register.

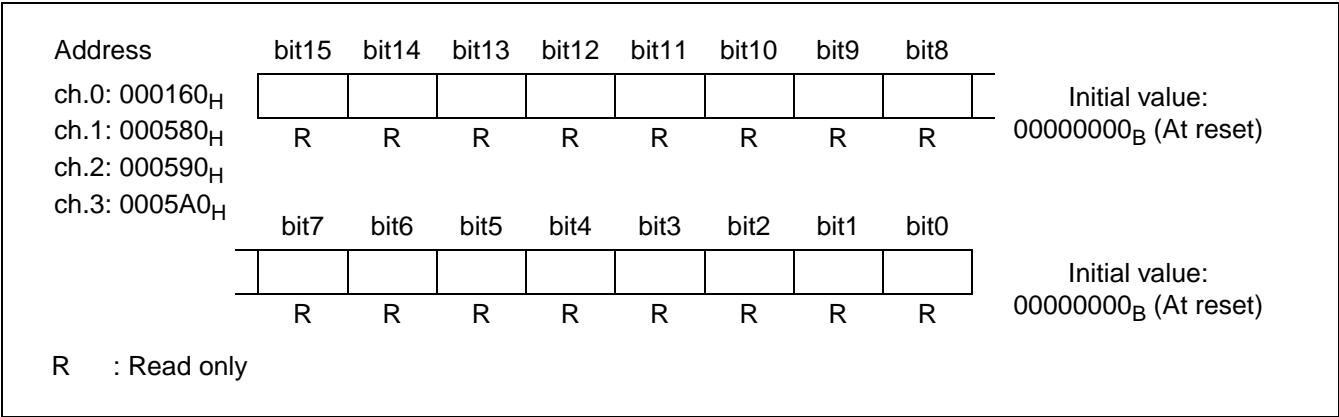
13.8.2.4 Timer Register (BTnTMR)

The timer register (BTnTMR) allows the value of the 16-bit down counter to be read from.

■ Bit Configuration of the Timer Register (BTnTMR)

Figure 13.8-19 shows the bit configuration of the PPG timer register (BTnTMR).

Figure 13.8-19 Bit Configuration of the Timer Register (BTnTMR)



The BTnTMR register allows the value of the 16-bit down counter to be read from.

Note:

Access the BTnTMR register using 16-bit data.

13.8.2.5 16-bit PPG Timer Operation

In PPG timer mode, an arbitrary output pulse can be controlled by setting its "L" and "H" widths in their respective reload registers.

■ Principles of Operation

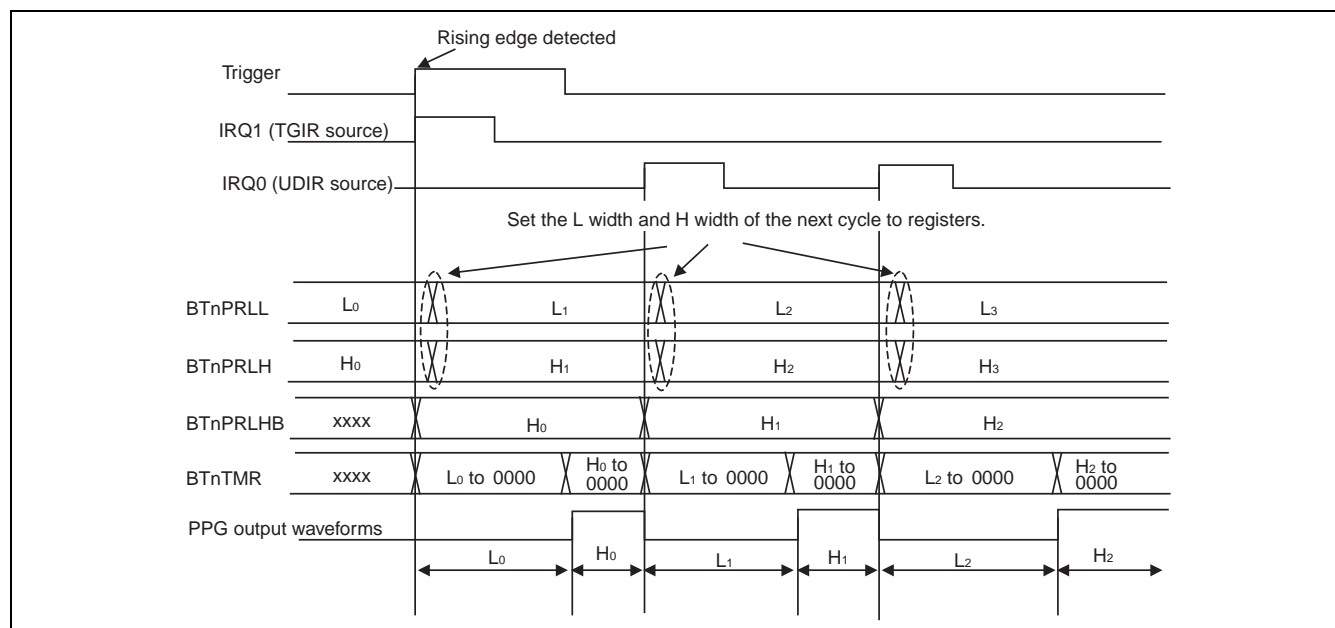
The PPG timer has two 16-bit reload registers for setting the "L" and "H" widths respectively and one "H" width setting buffer (BTnPRLL, BTnPRLH, PRLHB).

In response to the start trigger, the 16-bit down counter loads the BTnPRLL value and the BTnPRLH value is transferred to the BTnPRLHB buffer at the same time. The counter is decremented every count clock with the PPG output at the "L" level. When an underflow is detected, the counter reloads the BTnPRLHB value and is decremented with the PPG output waveform inverted. When an underflow is detected again, the PPG output waveform is inverted, the counter reloads the BTnPRLL set value, and the BTnPRLH set value is transferred to the BTnPRLHB buffer.

Through these steps, the output waveform becomes the pulse output with the "L" and "H" widths corresponding to their respective reload register values.

■ Reload Register Write Timing

Data is written to the BTnPRLL and BTnPRLH reload registers upon detection of a start trigger and between when the underflow interrupt request bit (UDIR) is set and when the next period begins. The data set then becomes the setting for the next period. The BTnPRLL and BTnPRLH settings are automatically transferred to the BTnTMR and BTnPRLHB, respectively, upon detection of a start trigger and when an underflow occurs at the end of "H" width counting. The data transferred to the BTnPRLHB is automatically reloaded to the BTnTMR when an underflow occurs at the end of "L" width counting.



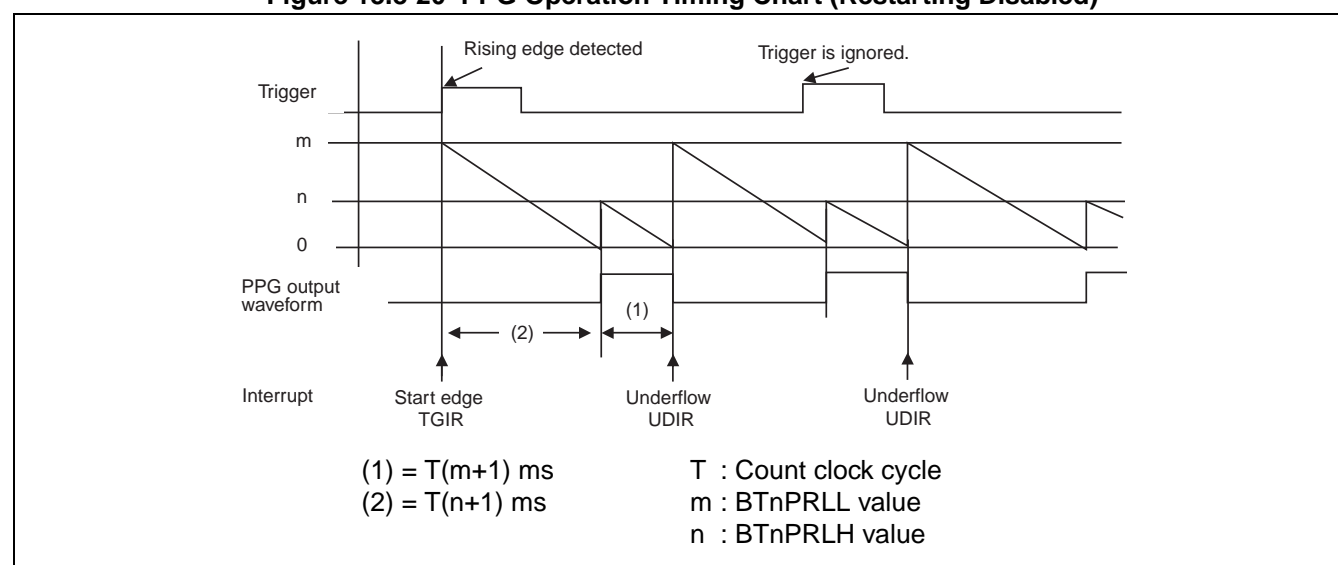
13.8.2.6 Continuous Operation

In continuous operation mode, an arbitrary pulse can be output continuously by updating the "L" and "H" widths at the set timing of each interrupt. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ Continuous Operation

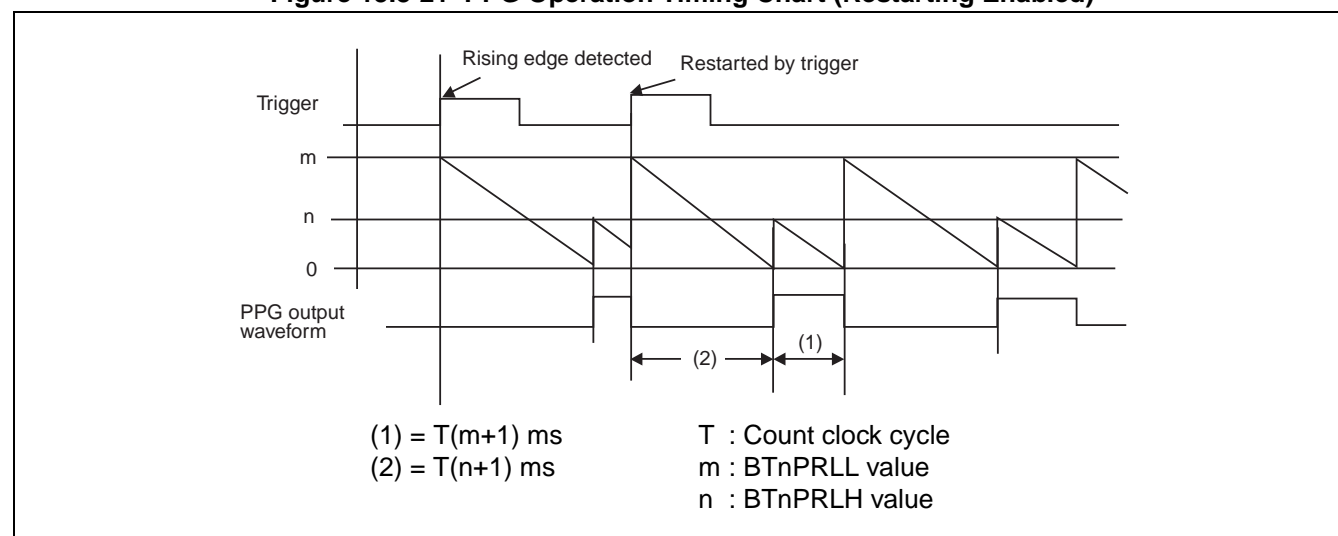
- When restarting is disabled (RTGEN = 0)

Figure 13.8-20 PPG Operation Timing Chart (Restarting Disabled)



- When restarting is enabled (RTGEN = 1)

Figure 13.8-21 PPG Operation Timing Chart (Restarting Enabled)



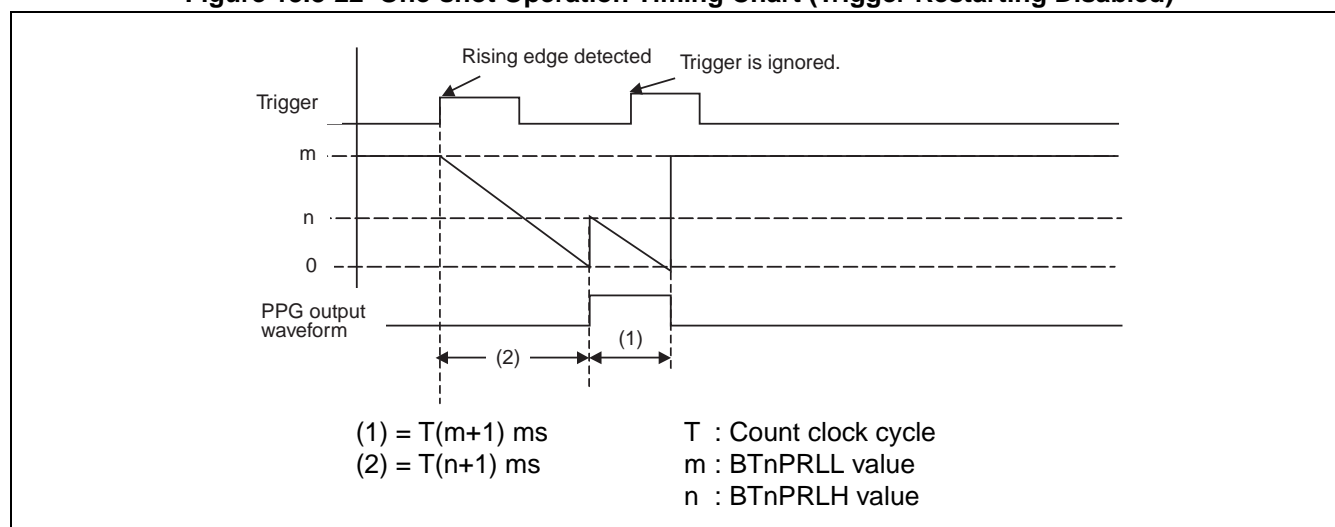
13.8.2.7 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ One-shot Operation

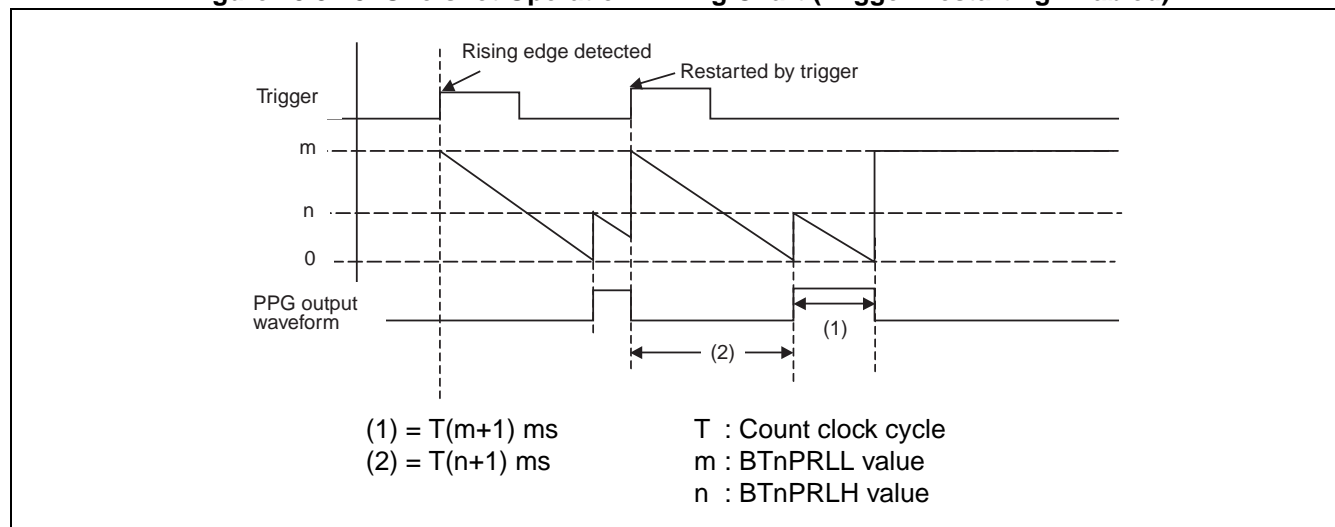
- When restarting is disabled (RTGEN = 0)

Figure 13.8-22 One-shot Operation Timing Chart (Trigger Restarting Disabled)



- When restarting is enabled (RTGEN = 1)

Figure 13.8-23 One-shot Operation Timing Chart (Trigger Restarting Enabled)



■ Relationship between Reload Value and Pulse Width

The output pulse width is obtained by adding 1 to the value written in the 16-bit reload register and multiplying the result by the count clock cycle. When the reload register value is 0000_H, therefore, the output has a pulse width of one count clock cycle. When the reload register value is FFFF_H, the output has a pulse width of 65536 count clock cycles. The pulse width is calculated from the following equation.

$PL = T \times (L+1)$	PL : "L" pulse width
$PH = T \times (H+1)$	PH : "H" pulse width
	T : Count clock cycle
	L : BTnPRLL value
	H : BTnPRLH value

13.8.2.8 Interrupt Factors and Timing Chart

This section provides the interrupt factors and timing chart.

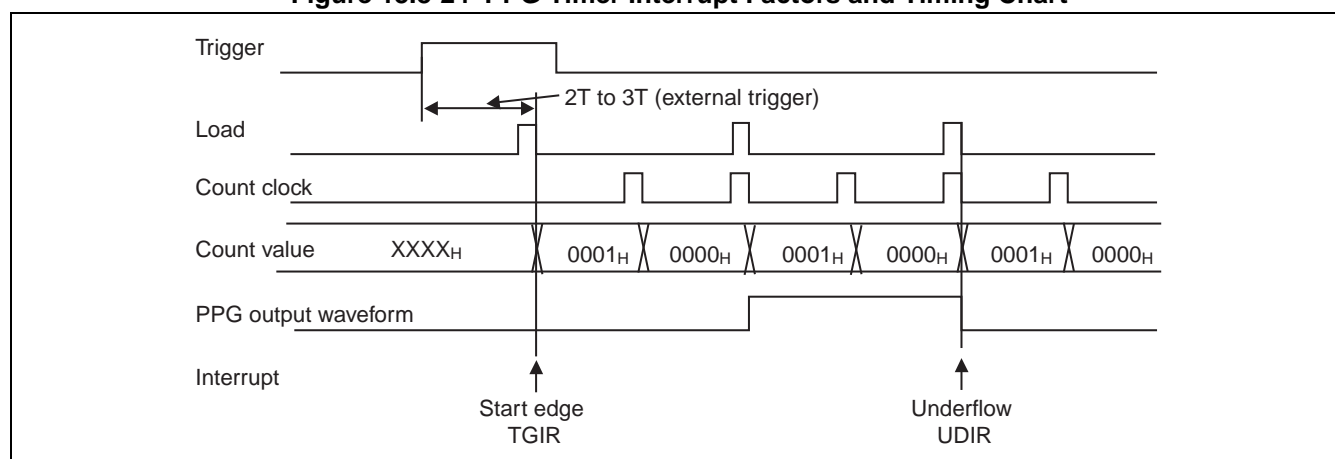
■ Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (CLKP) cycle) until the counter value is loaded after the trigger is generated.

Interrupt factors are set when the PPG start trigger is detected and when an underflow is detected during "H" level output.

Figure 13.8-24 shows the interrupt factors and timing chart, assuming "L" width setting = 3 and "H" width setting = 1.

Figure 13.8-24 PPG Timer Interrupt Factors and Timing Chart



13.8.3 Reload Timer Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the reload timer.

- Timer Control Register (BTnTMCR) for Reload Timer
- Period Setting Register (BTnPCSR)
- Timer Register (BTnTMR)
- 16-bit Reload Timer Operation

13.8.3.1 Timer Control Register (BTnTMCR) for Reload Timer

The timer control register (BTnTMCR) controls the reload timer.

■ Timer Control Register (BTnTMCR Upper Byte)

Figure 13.8-25 Timer Control Register (BTnTMCR Upper Byte)

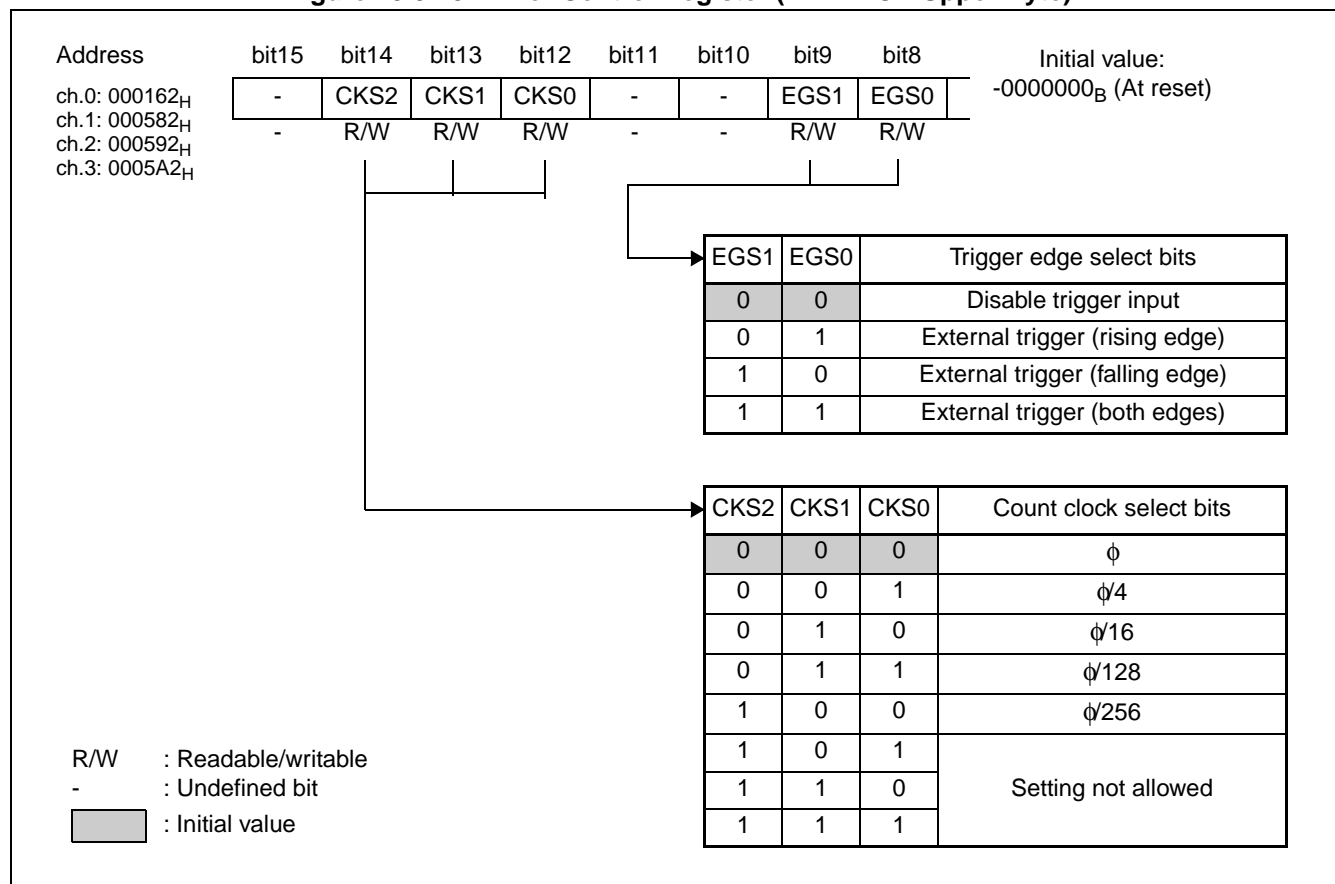


Table 13.8-7 Timer Control Register (BTnTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value of this bit is undefined. Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	<ul style="list-style-type: none"> Select the count clock for the 16-bit down counter. The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11, bit10	Undefined bits	<ul style="list-style-type: none"> The value read is "0" When writing to these bits, write "0".
bit9, bit8	EGS1, EGS0: Trigger edge select bits	<ul style="list-style-type: none"> Select the effective edge of the input waveform as an external trigger to set the trigger condition. When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTnTMCR Lower Byte)

Figure 13.8-26 Timer Control Register (BTnTMCR Lower Byte)

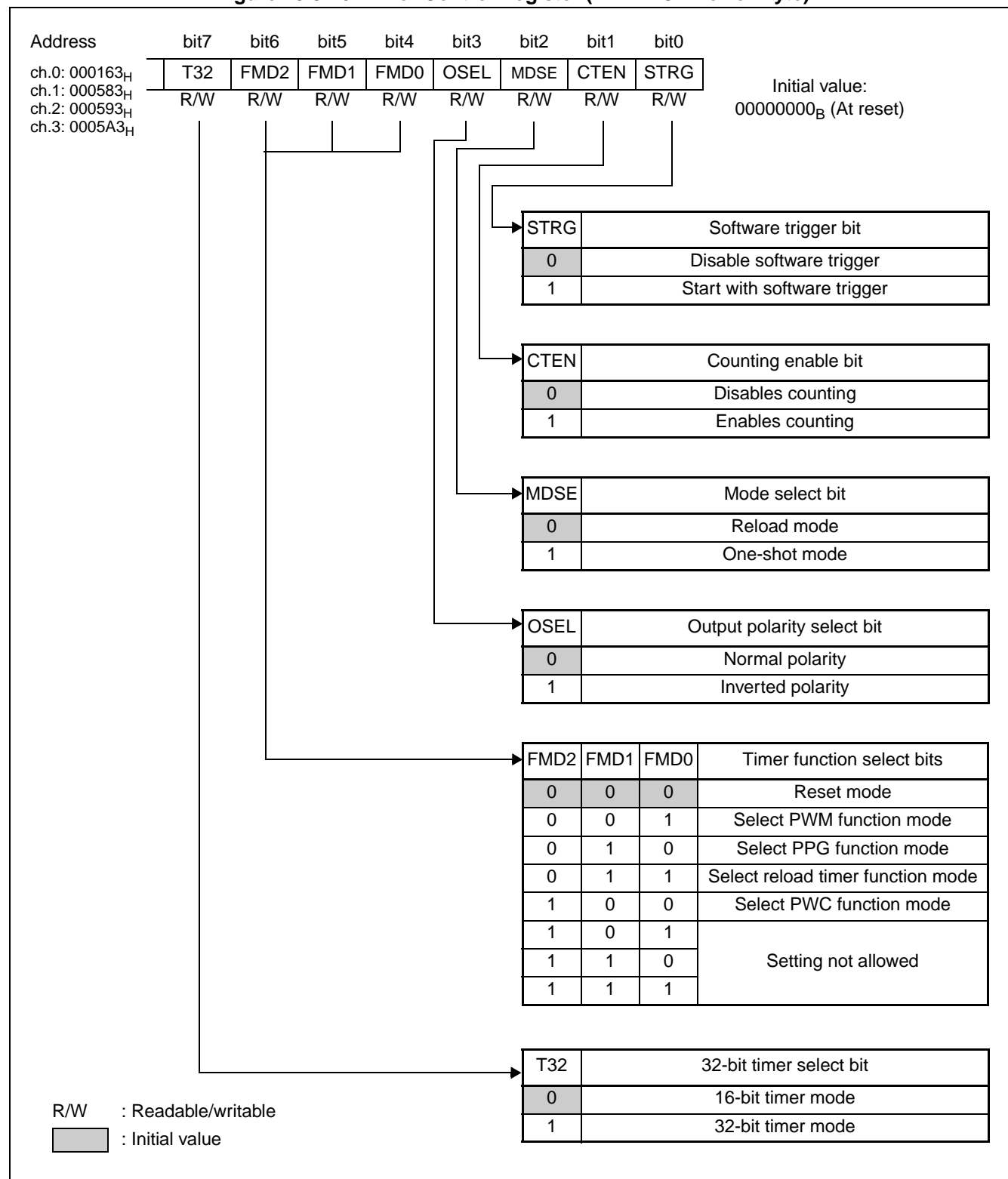


Table 13.8-8 Timer Control Register (BTnTMCR Lower Byte)

Bit name		Function															
bit7	T32: 32-bit timer select bit	<ul style="list-style-type: none"> This bit selects the 32-bit timer mode. When the FMD2, FMD1, and FMD0 bits contain "011_B" to select the reload timer, setting the T32 bit to "1" places the timer in 32-bit timer mode. The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. →See Section "13.5 32-bit Mode Operations". 															
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none"> These bits select the timer function mode. Setting the FMD2, FMD1, and FMD0 bits to "011_B" selects the reload timer function mode. The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. 															
bit3	OSEL: Output polarity select bit	<ul style="list-style-type: none"> Selects the timer output at normal level or inverted level. The output waveform is generated as follows depending on the combination with the MDSE bit (bit2): <table border="1"> <thead> <tr> <th>MDSE</th><th>OSEL</th><th>Output Waveforms</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Toggle output of "L" at the count start</td></tr> <tr> <td>0</td><td>1</td><td>Toggle output of "H" at the count start</td></tr> <tr> <td>1</td><td>0</td><td>Rectangular wave of "H" during count</td></tr> <tr> <td>1</td><td>1</td><td>Rectangular wave of "L" during count</td></tr> </tbody> </table>	MDSE	OSEL	Output Waveforms	0	0	Toggle output of "L" at the count start	0	1	Toggle output of "H" at the count start	1	0	Rectangular wave of "H" during count	1	1	Rectangular wave of "L" during count
MDSE	OSEL	Output Waveforms															
0	0	Toggle output of "L" at the count start															
0	1	Toggle output of "H" at the count start															
1	0	Rectangular wave of "H" during count															
1	1	Rectangular wave of "L" during count															
bit2	MDSE: Mode select bit	<ul style="list-style-type: none"> Setting the MDSE bit to "0" selects reload mode, in which the counter loads the reload register value to continue counting the moment a count value underflow occurs from 0000_H to FFFF_H. Setting the MDSE bit to "1" selects one-shot mode, in which the counter stops operation the moment a count value underflow occurs from 0000_H to FFFF_H. The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. 															
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none"> This bit enables the down counter. Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter. 															
bit0	STRG: Software trigger bit	<ul style="list-style-type: none"> Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <p>Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</p> <ul style="list-style-type: none"> The value read from the STRG bit is always "0". <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</p>															

■ Status Control Register (BTnSTC)

Figure 13.8-27 Status Control Register (BTnSTC)

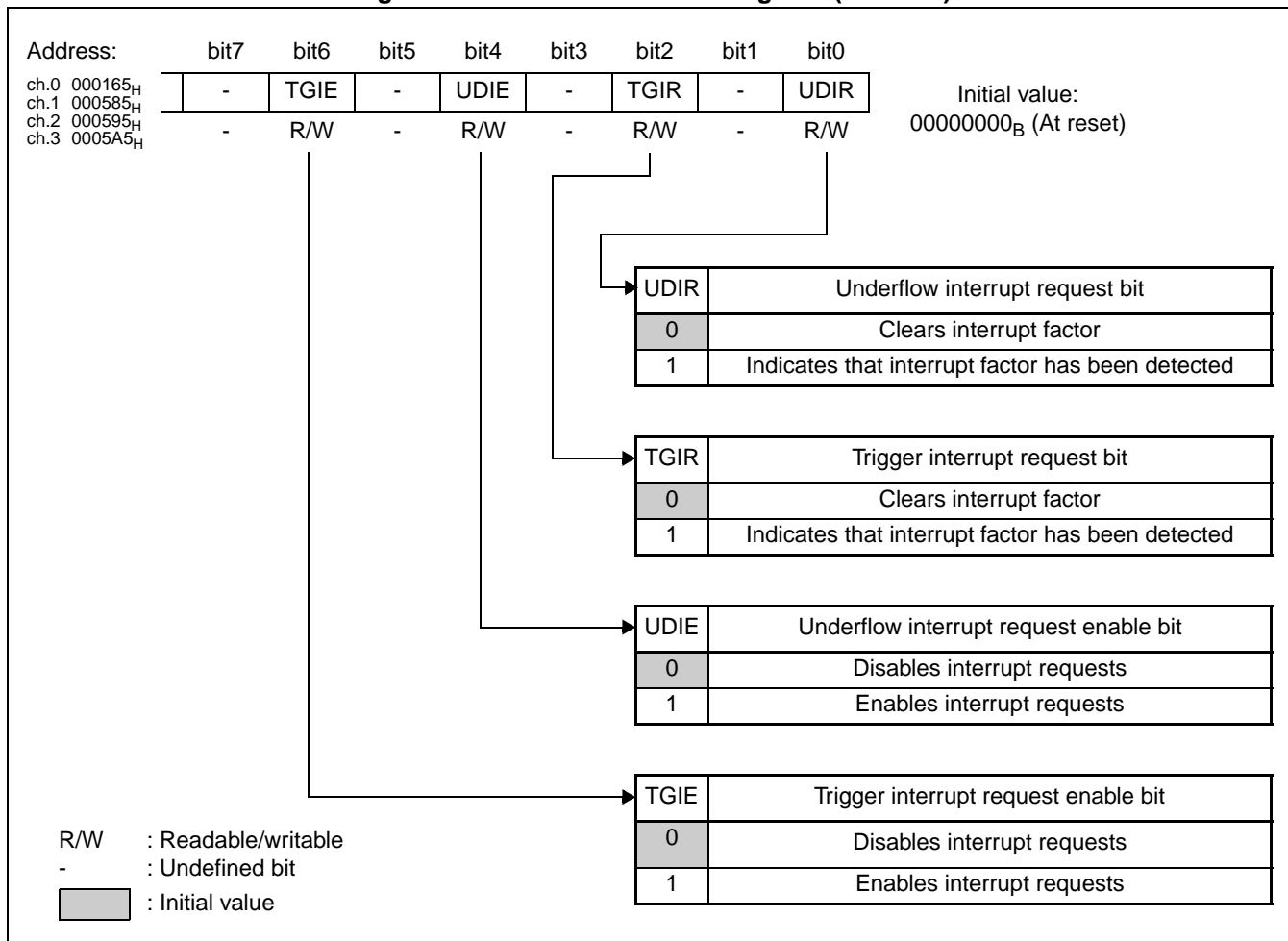


Table 13.8-9 Status Control Register (BTnSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> Controls bit2: TGIR interrupt requests. Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	<ul style="list-style-type: none"> Controls bit0: UDIR interrupt requests. Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> The TGIR bit is set to "1" upon detection of a software trigger or trigger input. Writing "0" to the TGIR bit clears it. Writing "1" to the TGIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	<ul style="list-style-type: none"> The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to FFFF_H. Writing "0" to the UDIR bit clears it. Writing "1" to the UDIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

13.8.3.2 Period Setting Register (BTnPCSR)

The period setting register (BTnPCSR) holds the initial count value. In 32-bit mode, the register holds the initial count value of the lower 16 bits for the even-numbered channel or the initial count value of the upper 16 bits for the odd-numbered channel. The initial value immediately after a reset is undefined. To access this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Period Setting Register (BTnPCSR)

Figure 13.8-28 shows the bit configuration of the period setting register (BTnPCSR).

Figure 13.8-28 Bit Configuration of the Period Setting Register (BTnPCSR)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 000168 _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 000588 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2: 000598 _H									Initial value: XXXXXXXX _B (At reset)
ch.3: 0005A8 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable
X : Undefined value

The BTnPCSR register is used to set the period. Transfer to the timer register takes place when an underflow occurs.

- Access the BTnPCSR register using 16-bit data.
- Set the period using the BTnPCSR register after selecting the reload timer function mode using the FMD2, FMD1, and FMD0 bits in the BTnTMCR register.
- To write data to the BTnPCSR register in 32-bit mode, access its upper 16-bit data (data for the odd-numbered channel) first and then the lower 16-bit data (data for the even-numbered channel).

13.8.3.3 Timer Register (BTnTMR)

The timer register (BTnTMR) allows the count value of the timer to be read from. In 32-bit mode, the register holds the count value of the lower 16 bits for the even-numbered channel or the count value for the upper 16 bits for the odd-numbered channel. The initial value is undefined.

To read this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Timer Register (BTnTMR)

Figure 13.8-29 shows the bit configuration of the timer register (BTnTMR).

Figure 13.8-29 Bit Configuration of the Timer Register (BTnTMR)

Figure 10-20 Bit Configuration of the Filter Register (Continued)									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 000160 _H									Initial value: 00000000 _B (At reset)
ch.1: 000580 _H	R	R	R	R	R	R	R	R	
ch.2: 000590 _H									
ch.3: 0005A0 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: 00000000 _B (At reset)
	R	R	R	R	R	R	R	R	
R : Read only									

The BTnTMR register allows the value of the 16-bit down counter to be read from.

Notes:

- Access the BTnTMR register using 16-bit data.
- To read data from the BTnTMR register in 32-bit mode, access its lower 16-bit data (data for the even-numbered channel) first and then the upper 16-bit data (data for the odd-numbered channel).

13.8.3.4 16-bit Reload Timer Operation

In reload timer mode, the timer decrements the counter from the value set in the period setting register in synchronization with the count clock, and finishes counting when the count value reaches "0" or continues operation with the period setting loaded automatically until the counter stops being decremented.

■ Counting with the Internal Clock Selected

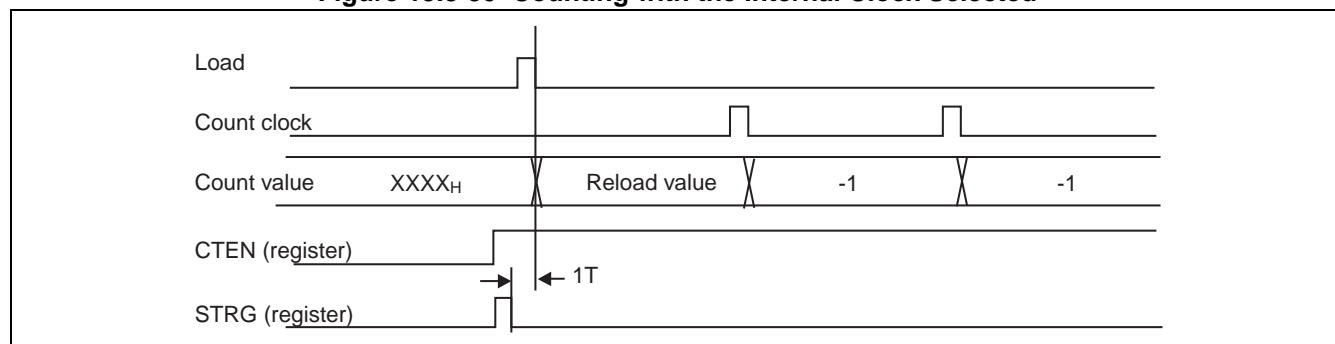
To start counting the moment counting is enabled, write "1" to both of the CTEN and STRG bits in the timer control register. The STRG bit maintains the trigger input always enabled irrespective of the operation mode as long as the timer is active (CNTE = 1).

Enable counting and start the timer using a software trigger or external trigger, and the timer loads the period setting register value to the counter to start decrementing the counter.

It takes 1T (T: peripheral clock (CLKP) cycle) for data in the period setting register to be loaded into the counter after the counter start trigger is set.

Figure 13.8-30 illustrates how the counter is started by the software trigger and operates.

Figure 13.8-30 Counting with the Internal Clock Selected



■ Underflow Operation

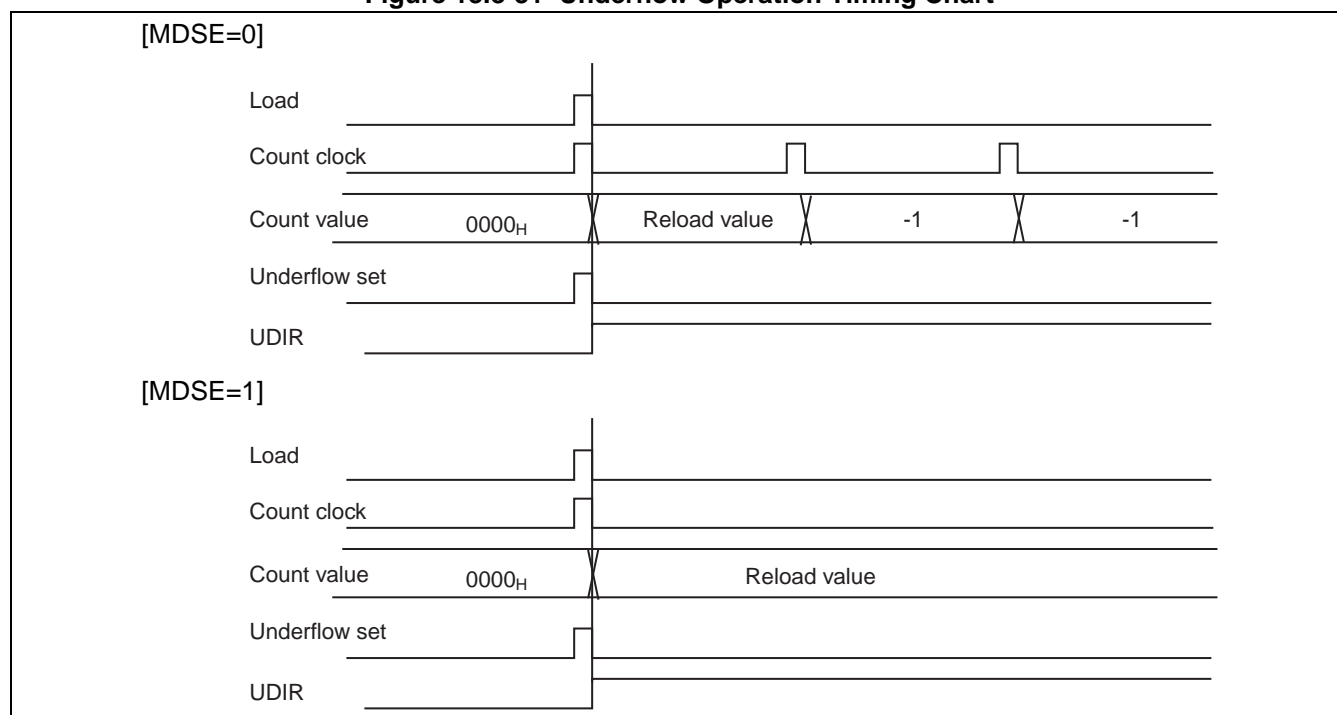
When the counter value changes from "0000_H" to "FFFF_H", the transition is detected as an underflow. When the counter counts [period setting register value + 1], therefore, an underflow occurs.

When an underflow occurs, the content of the period setting register (BTnPCSR) is loaded into the counter, and the counter continues counting if the MDSE bit in the timer control register (BTnTMCR) is "0". If the MDSE bit is "1", the counter stops operation with the loaded counter value left unchanged.

When an underflow occurs, the UDIR bit in the status control register (BTnSTC) is set and an interrupt request occurs if the UDIE bit is "1".

Figure 13.8-31 is a timing chart of underflow operation.

Figure 13.8-31 Underflow Operation Timing Chart



13.8.4 PWC Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWC timer.

- Timer Control Register (BTnTMCR) for PWC Timer
- Data Buffer Register (BTnDTBF)
- PWC Operation

13.8.4.1 Timer Control Register (BTnTMCR) for PWC Timer

The timer control register (BTnTMCR) controls the PWC timer.

■ Timer Control Register (BTnTMCR Upper Byte)

Figure 13.8-34 Timer Control Register (BTnTMCR Upper Byte)

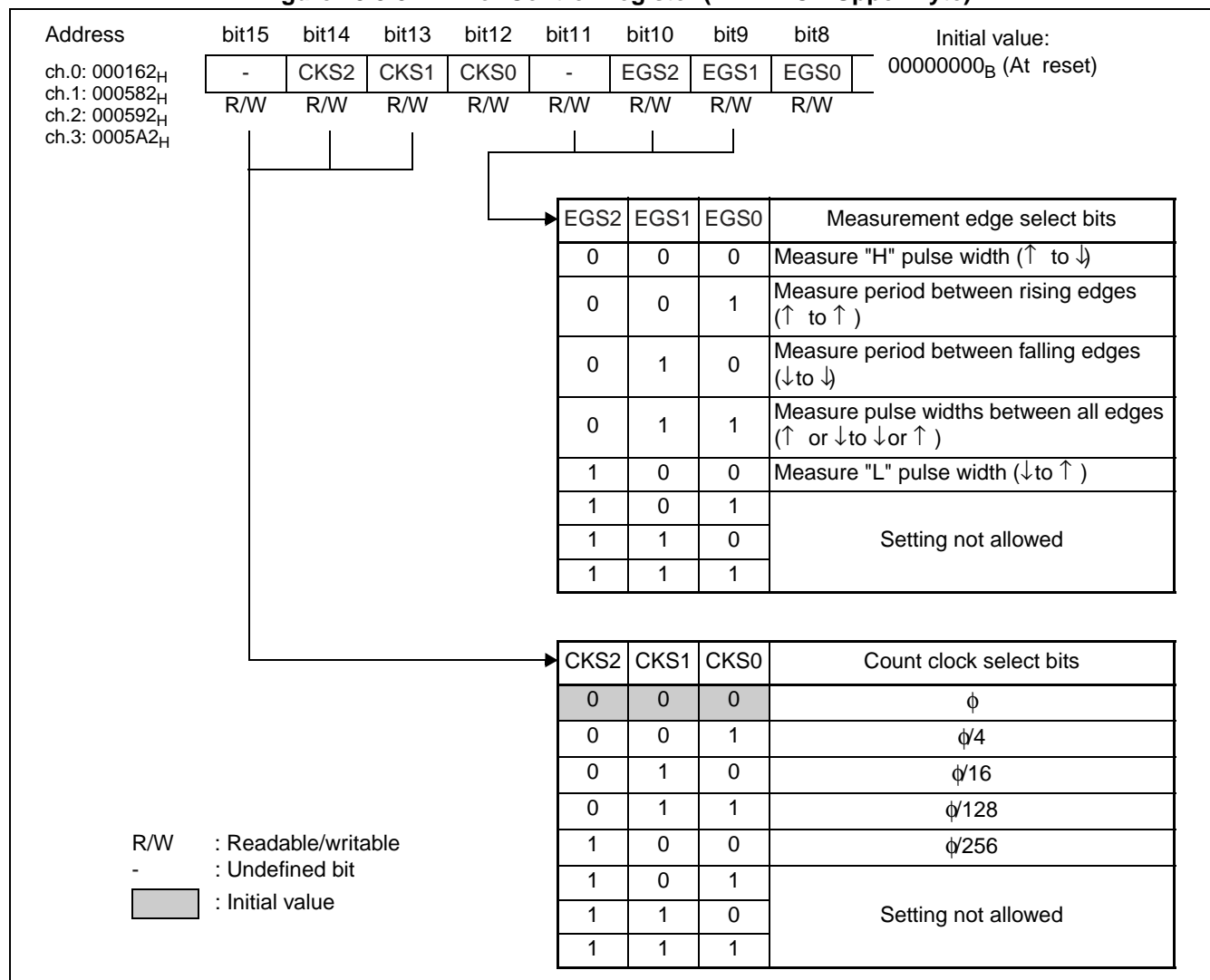


Table 13.8-10 Timer Control Register (BTnTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	<ul style="list-style-type: none"> Select the count clock for the 16-bit up counter. The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit10 to bit8	EGS2, EGS1, EGS0: Measurement edge select bits	<ul style="list-style-type: none"> Set the measurement edge condition. EGS2, EGS1, and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTnTMCR Lower Byte)

Figure 13.8-35 Timer Control Register (BTnTMCR Lower Byte)

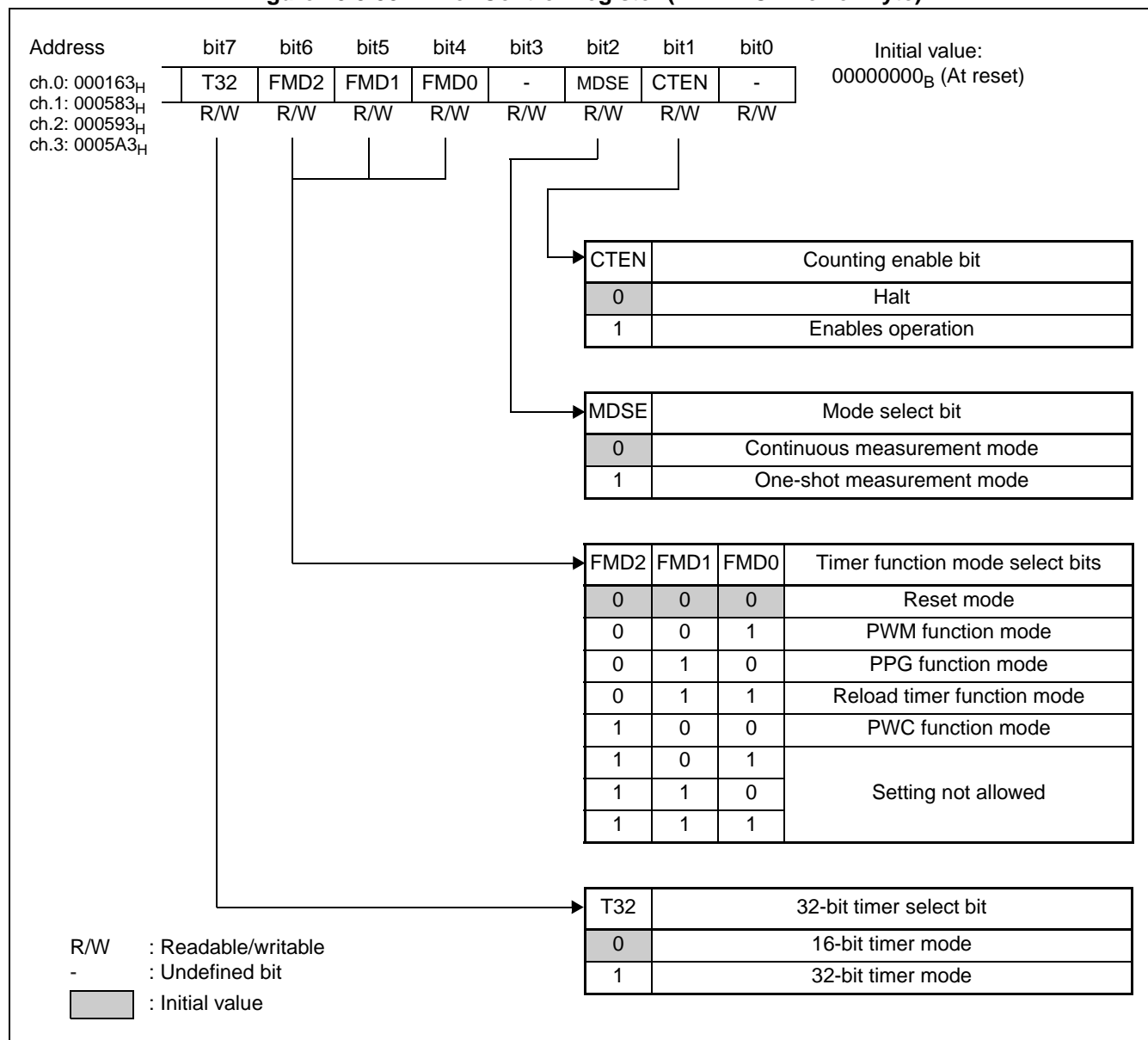


Table 13.8-11 Timer Control Register (BTnTMCR Lower Byte)

Bit name		Function									
bit7	T32: 32-bit timer select bit	<ul style="list-style-type: none"> This bit selects the 32-bit timer mode. When the FMD2, FMD1, and FMD0 bits contain "100_B" to select the PWC timer, setting the T32 bit to "1" places the timer in 32-bit PWC mode. The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. →See Section "13.5 32-bit Mode Operations". 									
bit6 to bit4	FMD2, FMD1, FMD0: Timer function mode select bits	<ul style="list-style-type: none"> These bits select the timer function mode. Setting the FMD2, FMD1, and FMD0 bits to "100_B" selects the PWC timer function mode. The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. 									
bit3	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0". 									
bit2	MDSE: Mode select bit	<ul style="list-style-type: none"> Selects measurement mode as follows. <table border="1"> <thead> <tr> <th>MDSE</th><th>Mode</th><th>Operation</th></tr> </thead> <tbody> <tr> <td>0</td><td>Continuous measurement</td><td>Continuous measurement: buffer register enabled</td></tr> <tr> <td>1</td><td>One-shot measurement</td><td>Halts after each measurement</td></tr> </tbody> </table> <ul style="list-style-type: none"> The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. 	MDSE	Mode	Operation	0	Continuous measurement	Continuous measurement: buffer register enabled	1	One-shot measurement	Halts after each measurement
MDSE	Mode	Operation									
0	Continuous measurement	Continuous measurement: buffer register enabled									
1	One-shot measurement	Halts after each measurement									
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none"> This bit enables the starting or restarting of the up counter. Writing "1" to this bit with the counter enabled for operation (CTEN bit = 1) causes a restart, resulting in the counter cleared and waiting for the measurement start edge. Writing "0" to the bit with the counter enabled for operation (CTEN bit = 1 stops the counter. 									
bit0	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0". 									

■ Status Control Register (BTnSTC)

Figure 13.8-36 Status Control Register (BTnSTC)

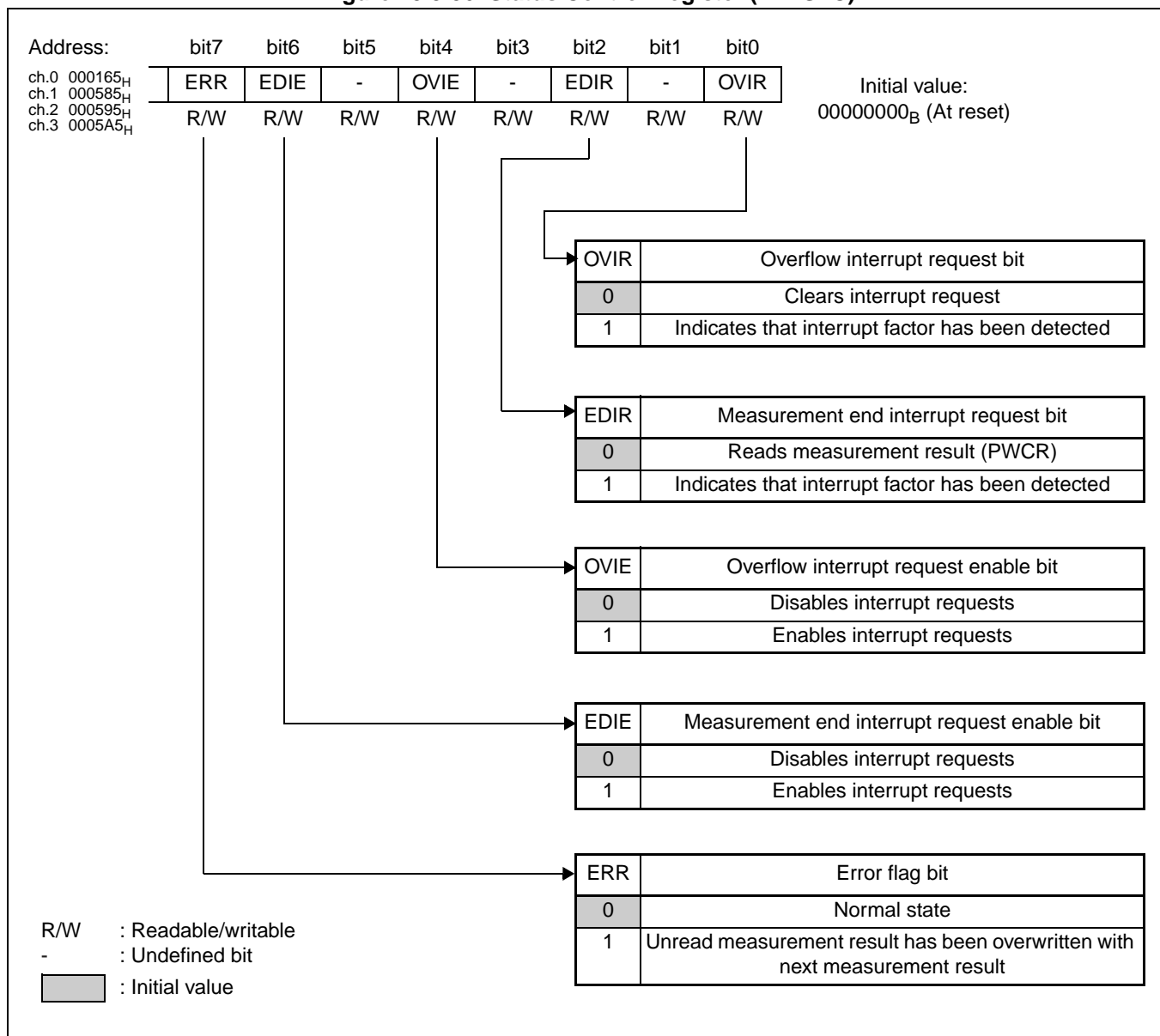


Table 13.8-12 Status Control Register (BTnSTC)

Bit name		Function
bit7	ERR: Error flag bit	<ul style="list-style-type: none"> This flag indicates that the next measurement has been completed before reading the current measurement result from the BTnDTBF register in continuous measurement mode. In this case, the BTnDTBF register is updated with the new measurement result, discarding the preceding measurement result. Measurement continues irrespective of the ERR bit value. The ERR bit can only be read; an attempt to write to it has no effect on the bit value. The ERR bit is cleared by reading the measurement result (BTnDTBF).
bit6	EDIE: Measurement end interrupt request enable bit	<ul style="list-style-type: none"> Controls bit2: EDIR interrupt requests. Setting the EDIR bit (bit2) with the EDIE bit enabling measurement end interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	OVIE: Overflow interrupt request enable bit	<ul style="list-style-type: none"> Controls bit0: OVIR interrupt requests. Setting the OVIR bit (bit0) with the OVIE bit enabling overflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	EDIR: Measurement end interrupt request bit	<ul style="list-style-type: none"> Indicates that measurement has been completed. The flag is set to "1" upon completion. The EDIR bit is cleared by reading the measurement result (BTnDTBF). The EDIR bit can only be read; an attempt to write to it has no effect on the bit value.
bit1	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	OVIR: Overflow interrupt request bit	<ul style="list-style-type: none"> The flag is set to "1" when a count value overflow occurs from FFFF_H to 0000_H. Writing "0" to the OVIR bit clears it. Writing "1" to the OVIR bit has no effect on the bit value. When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

13.8.4.2 Data Buffer Register (BTnDTBF)

The data buffer register (BTnDTBF) allows the measured value or count value of the PWC timer to be read from. In 32-bit mode, the register holds the value of the lower 16 bits for the even-numbered channel or the value of the upper 16 bits for the odd-numbered channel.

To read this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Data Buffer Register (BTnDTBF)

Figure 13.8-37 shows the bit configuration of the data buffer register (BTnDTBF).

Figure 13.8-37 Bit Configuration of the Data Buffer Register (BTnDTBF)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
ch.0: 00016A _H									Initial value: XXXXXXXX _B (At reset)
ch.1: 00058A _H	R	R	R	R	R	R	R	R	
ch.2: 00059A _H									
ch.3: 0005AA _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
									Initial value: XXXXXXXX _B (At reset)
	R	R	R	R	R	R	R	R	
R : Read only									

- The BTnDTBF register can only be read in both of the continuous and one-shot measurement modes. An attempt to write to the register makes no change to the register value.
- In continuous measurement mode (BTnTMCR: bit3 MDSE = 1), the BTnDTBF register serves as a buffer register holding the preceding measurement result.
- In one-shot measurement mode (BTnTMCR: bit3 MDSE = 0), the BTnDTBF register directly accesses the up counter. Even during counting, the count value can be read from this register. When the measurement is completed, the register preserved the measurement result as it is.
- Access the BTnDTBF register using 16-bit data.

13.8.4.3 PWC Operation

The PWC timer has a pulse width measurement feature, capable of selecting the count clock from among five types and measuring the time between arbitrary events of the input pulse and their cycle. The following outlines the basic functions and operations of the pulse width measurement feature.

■ Pulse Width Measurement Feature

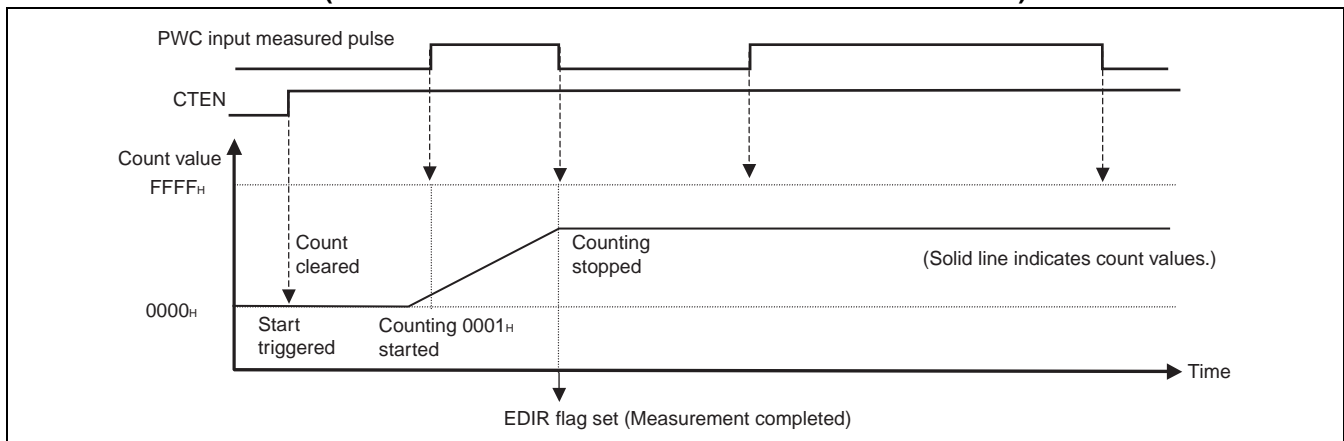
When started, the timer clears the counter to "0000_H" but does not perform counting until the pre-set measurement start edge is input. Upon detection of the measurement start edge, the timer increments the counter from "0001_H". Upon detection of the measurement end edge, the timer stops the counter. The timer saves the count value between the two events as the pulse width to the register.

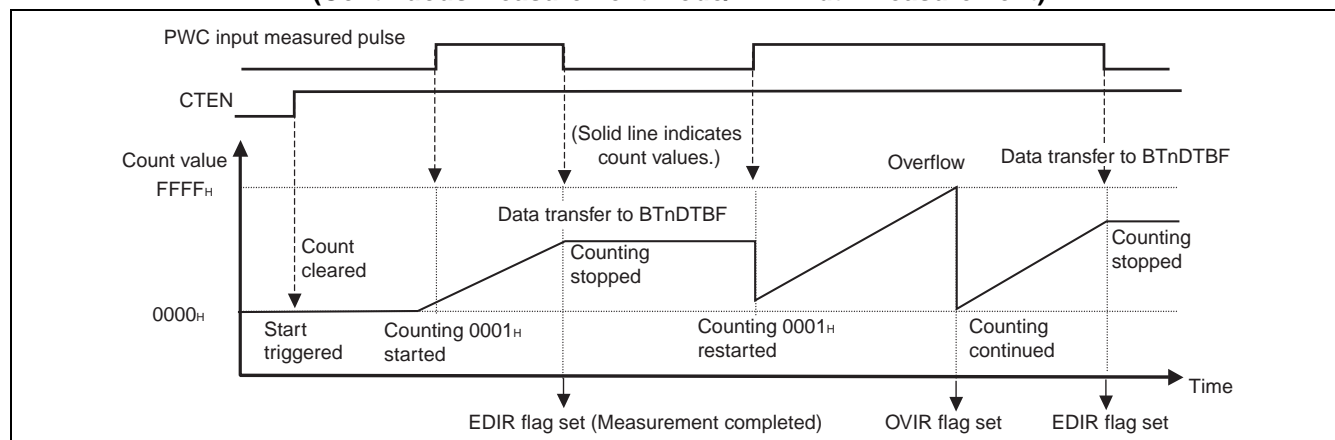
An interrupt request can be generated upon completion of measurement or when an overflow occurs.

After measurement, the timer acts as follows depending on the measurement mode:

- In one-shot measurement mode: The timer stops operation.
- In continuous measurement mode: The timer transfers the counter value to the buffer register and stops counting until the measurement start edge is input again.

**Figure 13.8-38 Pulse Width Measurement Operation
(One-shot Measurement Mode/"H" Width Measurement)**



**Figure 13.8-39 Pulse Width Measurement Operation
(Continuous Measurement Mode/"H" Width Measurement)**

■ Selecting the Count Clock

The count clock for the counter can be selected from among five types, depending on the settings of the CKS2 (bit6), CKS1 (bit5), and CKS0 (bit4) in the BTnTMCR registers.

The following count clocks can be selected:

BTnTMCR Register	Internal count clock selected
CKS2, CKS1, CKS0 bits	
000 _B	Peripheral clock (CLKP) [Initial value]
001 _B	Peripheral clock (CLKP) divided by 4
010 _B	Peripheral clock (CLKP) divided by 16
011 _B	Peripheral clock (CLKP) divided by 128
100 _B	Peripheral clock (CLKP) divided by 256
101 _B	Setting not allowed
110 _B	
111 _B	

The initial value immediately after a reset selects the peripheral clock (CLKP).

Note: Be sure to select the count clock before starting the counter.

■ Selecting the Operation Mode

Operation and measurement modes are selected depending on their settings in the BTnTMCR register.

Operation mode setting BTnTMCR bit10 to bit8: EGS2, EGS1, EGS0

(Selecting the measurement edge)

Measurement mode setting . . . BTnTMCR bit2: MDSE

(Selecting one-shot/continuous measurement)

Listed below are the selectable operation modes and their respective bit settings.

Operation mode		MDSE	EGS2	EGS1	EGS0
↑ to ↓ "H" pulse width measurement	Continuous measurement: Buffer enabled	0	0	0	0
	One-shot measurement: Buffer disabled	1	0	0	0
↑ to ↑ measurement of period between rising edges	Continuous measurement: Buffer enabled	0	0	0	1
	One-shot measurement: Buffer disabled	1	0	0	1
↓ to ↓ measurement of period between falling edges	Continuous measurement: Buffer enabled	0	0	1	0
	One-shot measurement: Buffer disabled	1	0	1	0
↑ or ↓ to ↓ or ↑ measurement between all edges	Continuous measurement: Buffer enabled	0	1	1	1
	One-shot measurement: Buffer disabled	1	1	1	1
↓ to ↑ "L" pulse width measurement	Continuous measurement: Buffer enabled	0	1	0	0
	One-shot measurement: Buffer disabled	1	1	0	0
Setting not allowed		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

The initial value immediately after a reset selects "H" pulse width/one-shot measurement mode.

Be sure to select the operation mode before starting the counter.

■ Starting and Stopping Pulse Width Measurement

Each type of measurement can be started, restarted, and aborted by the CTEN bit (bit1) in the BTnTMCR register.

You can start/restart pulse width measurement by writing "1" to the CTEN bit. You can abort it by writing "0" to the CTEN bit.

CTEN	Function
1	Starts/restarts pulse width measurement
0	Aborts pulse width measurement

■ Operation after being Started

The timer operation after the pulse width measurement mode has been started does not start counting until the measurement start edge is input. Upon detection of the measurement start edge, the 16-bit up counter starts counting from "0001_H".

■ Restarting

Restarting the timer means starting the timer during operation again while it has already been started (by writing "1" again to the CTEN bit already containing "1"). When restarted, the timer behaves as follows:

- If restarted the timer waiting for the measurement start edge: No effect on its operation.
- If restarted during measurement: The timer clears the counter to "0000_H" and waits for the measurement start edge again. If the restart and measurement end edge detection occur at the same time, the measurement end flag (EDIR) is set. In continuous measurement mode, the measurement result is transferred to the BTnDTBF register.

■ Stopping

In one-shot measurement mode, the timer stops counting automatically when the counter causes an overflow or when measurement is completed, requiring no special attention. To stop the timer either in continuous measurement mode or before it stops automatically, you have to abort it.

■ Clearing the Counters and Their Initial Values

The 16-bit up counter is cleared to "0000_H" when:

- a reset occurs
- "1" is written to the CTEN bit (bit1) in the BTnTMCR register (including the case of restarting).

The 16-bit up counter is initialized to "0001_H" when:

- measurement start edge is detected.

■ Details of Pulse Width Measurement Operation

● One-shot measurement and continuous measurement

There are two modes of pulse width measurement: one is to perform measurement only once and the other is to perform measurement continuously. Each mode is selected by using the MDSE bit in the BTnTMCR register (see "■ Selecting the Operation Mode" in "13.8.4.3 PWC Operation"). The two modes have the following differences:

One-shot measurement mode:

When the measurement end edge is input once, the counter stops counting and the measurement end flag (EDIR) in the BTnSTC register is set, finishing the current measurement session. If the counter is restarted at the same time, however, it waits for the measurement start edge.

Continuous measurement mode:

When the measurement end edge is input, the counter stops counting, the measurement end flag (EDIR) in the BTnSTC register is set, and the counter remains idle until the measurement start edge is input again. Next time the measurement start edge is input, the counter is initialized to "0001_H" to start measurement. Upon completion of measurement, the measurement result in the counter is transferred to the BTnDTBF register.

Be sure to select or change the measurement mode with the counter stopped.

● Measurement result data

The one-shot measurement and continuous measurement modes are different in the handling of measurement results and counter values and the BTnDTBF function. The differences in measurement results between the two modes are as follows:

One-shot measurement mode:

When the BTnDTBF register is read during operation, the count value being measured can be obtained.

When the BTnDTBF register is read after measurement is completed, measurement result data is obtained.

Continuous measurement mode:

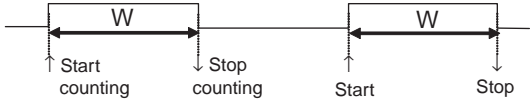
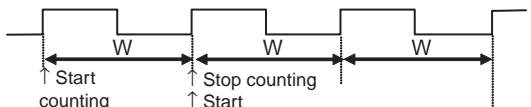
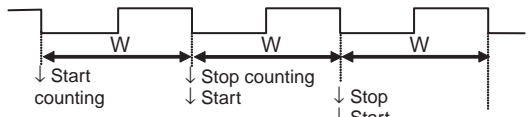
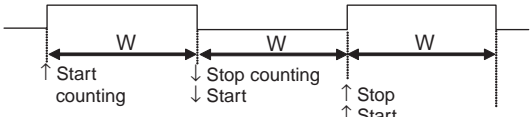
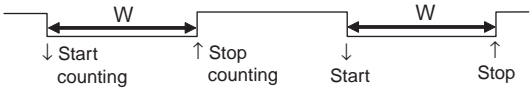
When measurement is completed, the measurement result in the counter is transferred to the BTnDTBF register.

When the BTnDTBF register is read, the last measurement result is obtained. During measurement operation, the BTnDTBF register holds the result of preceding measurement. The count value being measured cannot be read.

If the current measurement is completed before the preceding measurement result is read in continuous measurement mode, the preceding measurement result is overwritten by the new measurement result. In this case, the error flag (ERR) in the BTnSTC register is set. The error flag (ERR) is cleared automatically when the BTnDTBF register is read.

■ Measurement Mode and Counting

Measurement mode can be selected from among five types, depending on what part of the input pulse is measured. The following table summarizes each measurement mode and its target.

Measurement mode	EGS2, EGS1, EGS0	Measurement target (W: Pulse width to be measured)
"H" pulse width measurement	000 _B	 <p>Measure the width of "H" period. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of falling edge</p>
Measurement of period between rising edges	001 _B	 <p>Measure the period between rising edges. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of rising edge</p>
Measurement of period between falling edges	010 _B	 <p>Measure the period between falling edges. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of falling edge</p>
Measurement of pulse widths between all edges	011 _B	 <p>Measure the width between continuously input edges. Start counting (measurement) : upon detection of edge Stop counting (measurement) : upon detection of edge</p>
Measurement of "L" pulse width	100 _B	 <p>Measure the width of the "L" period. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of rising edge</p>

In any measurement mode, the counter started for measurement is cleared to "0000_H" and remains idle without counting until the measurement start edge is input. When the measurement start edge is input, the counter is incremented every count clock until the measurement end edge is input.

When measurement of pulse widths between all edges or period measurement is performed in continuous measurement mode, the end edge becomes the next measurement start edge.

● Pulse width/period calculation method

The following equation can be used to calculate the measured pulse width/period from measurement result data obtained from the BTnDTBF register after measurement is completed:

$T_W = n \times t$ [ms]	T_W : Measured pulse width/period [ms] n : Measurement result data in BTnDTBF t : Count clock cycle [ms]
-------------------------	--

● Generating interrupt requests

Interrupt requests can be generated in two ways.

- Interrupt request in response to counter overflow

When the counter is incremented to cause an overflow during measurement, the overflow flag (OVIR) is set and generates an interrupt request if overflow interrupt requests have been enabled.

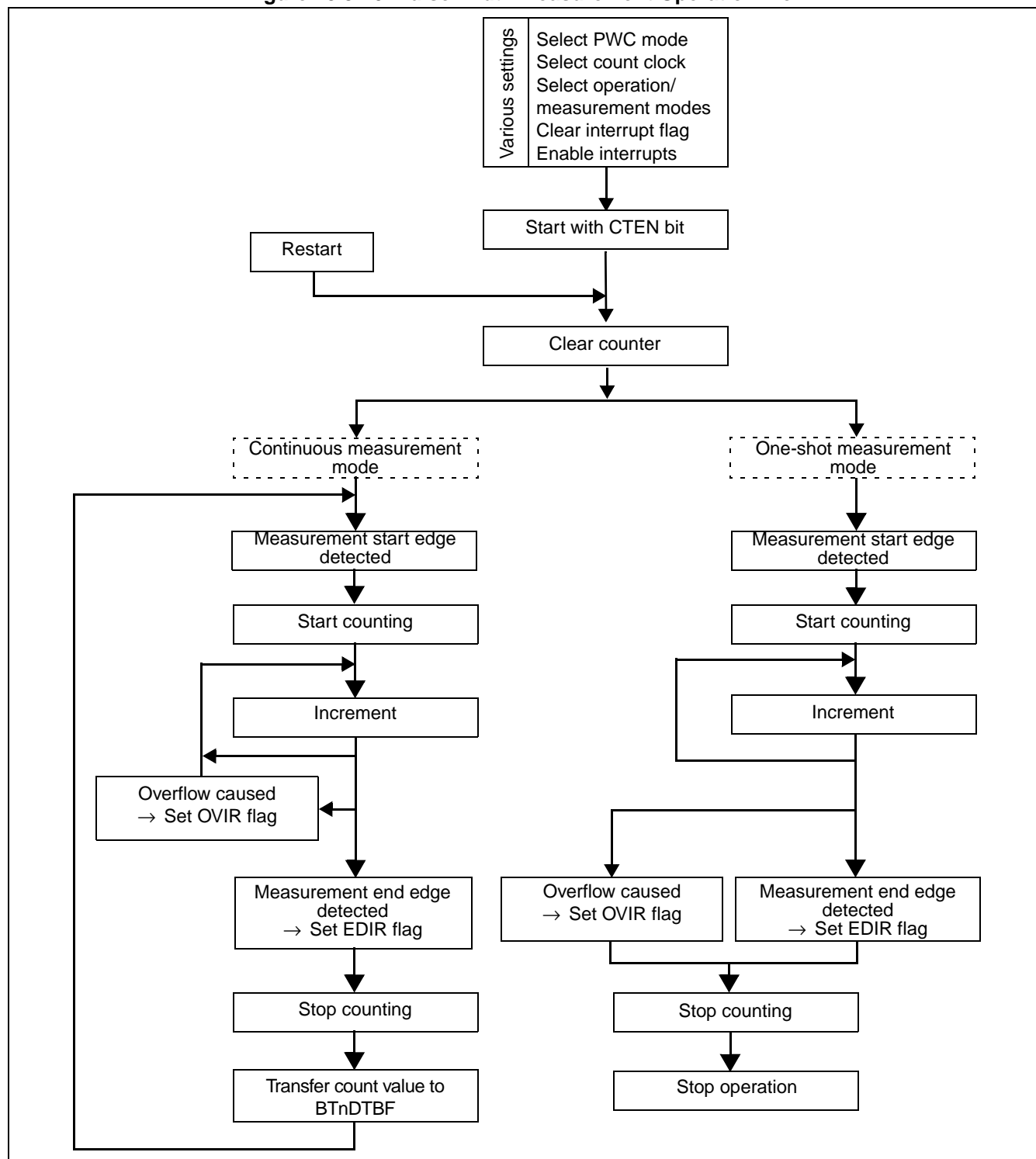
- Interrupt request upon completion of measurement

When the measurement end edge is detected, the measurement end flag (EDIR) in the BTnSTC register is set and generates an interrupt request if measurement end interrupt requests have been enabled.

The measurement end flag (EDIR) is cleared automatically when the measurement result is read from the BTnDTBF register.

■ Pulse Width Measurement Operation Flow

Figure 13.8-40 Pulse Width Measurement Operation Flow



CHAPTER 14

UP/DOWN COUNTER

This chapter describes the function and operation of 8/16-bit up/down counter.

- 14.1 Overview of Up/Down Counter
- 14.2 Block Diagram of Up/Down Counter
- 14.3 Register of Up/Down Counter
- 14.4 Operation of Up/Down Counters

14.1 Overview of Up/Down Counter

The 8/16-bit up/down counter is the up/down counter/timer which consists of three event input pins, 16-bit up/down counters, 16-bit reload/compare registers, and their control circuits.

The operating mode can switch one channel of 8-bit counter or one channel of 16-bit by setting.

■ Features of Up/Down Counter

- With the 16-bit count register, counting can be performed in a range between 0_D to 65535_D .
- The following four count modes can be selected for the count clock:
 - Timer mode
 - Up/down counter mode
 - Phase difference count mode (multiply-by-2)
 - Phase difference count mode (multiply-by-4)
- In timer mode, the count clock can be selected from two internal clocks and input from an internal circuit.
Count clocks available for selection (for operation at 40MHz)
 - 50ns (20MHz: divide-by-2)
 - 200ns (5MHz: divide-by-8)
- The detection edge of the external pin input signal can be selected in up and in down counting mode.
 - Detection of falling edge
 - Detection of rising edge
 - Detection of both rising and falling edges
 - Edge detection disabled
- The phase difference counting mode is suitable for counting for an encoder, such as for a motor. Using one of A phase output, B phase output, and Z phase output for the encoder as input allows to count rotation angle and number of rotations easily and with high precision.
- Two different functions can be selected for the ZIN pin (this applies for all modes).
 - Counter clear function
 - Gate function
- The compare function and reload function are available. These functions can be used separately or combined. By combining these functions, counting up or down can be performed with an arbitrary width.
 - Compare function (compare interrupt request output)
 - Compare function (compare interrupt request output and counter clearing)
 - Reload function (underflow interrupt request output and reloading)
 - Compare and reload function (compare interrupt request output, counter clearing, underflow interrupt request output, and reloading)
 - Compare and reload disabled

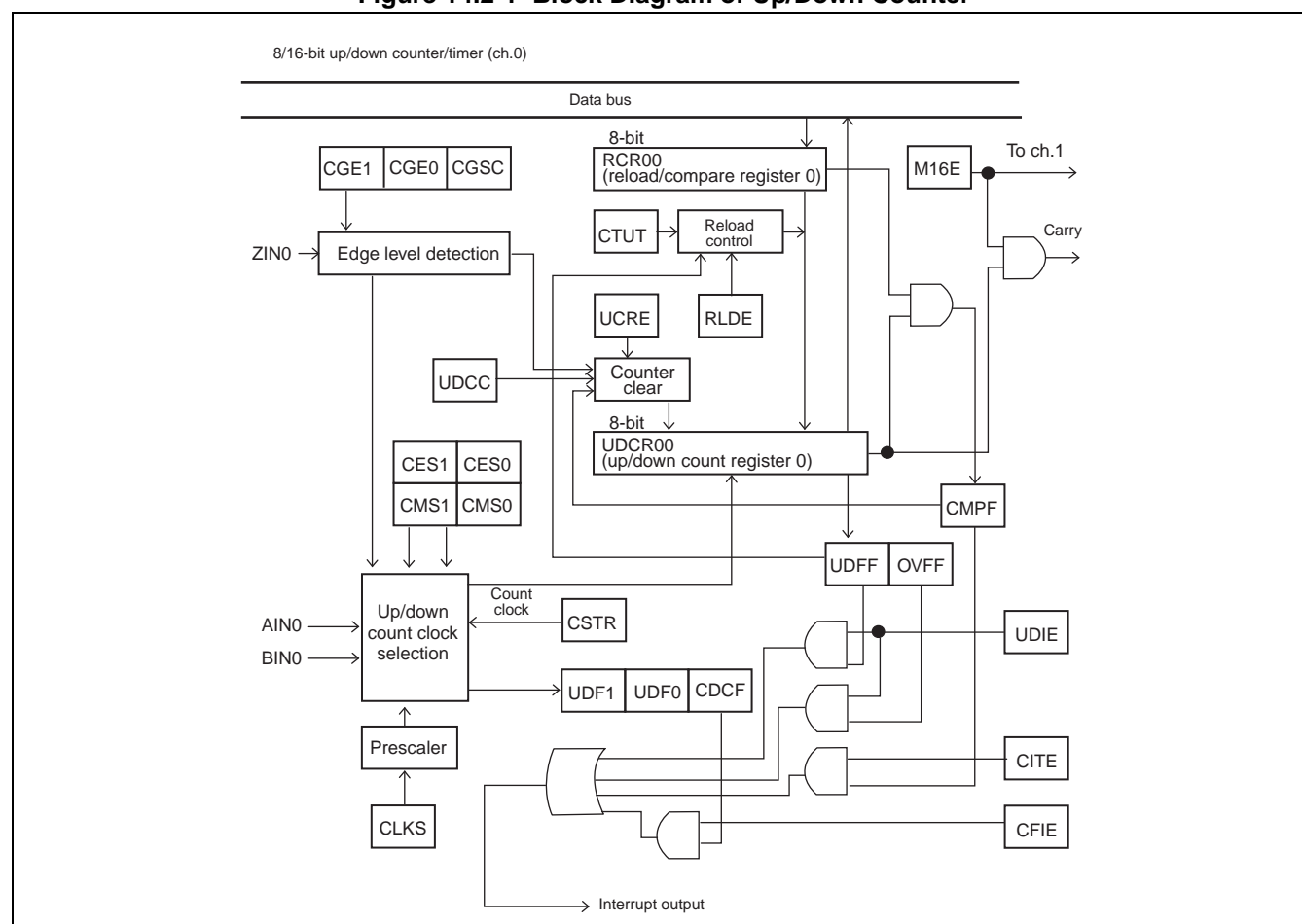
- With the count direction flag, the counting direction immediately before the current count can be identified.
- The generation of interrupts when a compare match occurs, at reload (underflow), at overflow, or when the counting direction changes, can be controlled individually.

14.2 Block Diagram of Up/Down Counter

This section explains the block diagram of up/down counter.

■ Block Diagram of Up/Down Counter

Figure 14.2-1 Block Diagram of Up/Down Counter



MB91470/480 Series

14.3 Register of Up/Down Counter

The up/down counter has up/down count register (UDCR), reload compare register (RCR), counter status register (CSR), and counter control register (CCR).
This section explains these registers.

■ List of Registers of Up/Down Counter

Figure 14.3-1 List of Registers of Up/Down Counter

UDCR10									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000542 _H	D15	D14	D13	D12	D11	D10	D09	D08	00000000 _H
	R	R	R	R	R	R	R	R	
UDCR00									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000543 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _H
	R	R	R	R	R	R	R	R	
RCR10									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000540 _H	D15	D14	D13	D12	D11	D10	D09	D08	00000000 _H
	W	W	W	W	W	W	W	W	
RCR00									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000541 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _H
	W	W	W	W	W	W	W	W	
CSR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000547 _H	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	00000000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R	R	
CCR10									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000544 _H	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	00000000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
CCRL0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000545 _H	-	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	-0000000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R	R	
R/W: Readable/writable									
R: Read only									
W: Write only									

14.3.1 Up/Down Count Register (UDCR)

Up/down count register (UDCR) is 8-bit count register. Up/down counting is performed by an input from the internal circuit, an internal prescaler, or an input of AIN pin and BIN pin. Also, in 16-bit count mode, this register operates as 16-bit count register.

■ Up/Down Count Register (UDCR)

Figure 14.3-2 Up/Down Count Register (UDCR)

UDCR10									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000542 _H	D15	D14	D13	D12	D11	D10	D09	D08	00000000 _H
	R	R	R	R	R	R	R	R	
UDCR00									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000543 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _H
	R	R	R	R	R	R	R	R	
R: Read only									

Values cannot be written to this register directly. To write a value to this register, the RCR must be used. First write the value to write to this register to the RCR, then set the CTUT bit of the CCRL register to "1". The value will then be transferred from the RCR to this register (in a reload-operation by software).

Note:

In 16-bit mode, perform a 16-bit read operation for this register once.
In 8-bit mode, only UDCR00 value is effective.

MB91470/480 Series

14.3.2 Reload Compare Register (RCR)

Reload compare register (RCR) is 8-bit reload/compare register. The reload value and the compare value are set by this register. The reload value and the compare value are the same and up/down count is enabled in 00_H to the value of this register (16-bit operation mode: 0000_H to the value of this register) by activating the function of reload and compare.

■ Reload Compare Register (RCR)

Figure 14.3-3 Reload Compare Register (RCR)

RCR10									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000540 _H	D15	D14	D13	D12	D11	D10	D09	D08	00000000 _H
	W	W	W	W	W	W	W	W	
RCR00									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000541 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _H
	W	W	W	W	W	W	W	W	
W: Write only									

This register is enabled to write only and disabled to read. By setting the CTUT bit of the CCR register to "1" while counting is stopped, the value of this register can be transferred to the UDCR (reloaded by software).

Note:

In 16-bit mode (when M16E = 1), write a 16-bit value to this register once.
In 8-bit mode (when M16E = 0), write a 8-bit value to RCR00.

14.3.3 Counter Status Register (CSR)

Counter status register (CSR) can check the state of up/down counter and control the interrupt.

■ Bit Configuration of Counter Status Register (CSR)

Figure 14.3-4 Bit Configuration of Counter Status Register (CSR)

CSR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	00000000 _H
00000547 _H	R/W	R/W	R/W	R/W	R/W	R/W	R	R	

R/W: Readable/writable
R: Read only

[bit7] CSTR: Count start bit

This bit controls start and stop of UDCR counting operation.

CSTR	Count operation
0	Stops the counting operation [initial value].
1	Starts the counting operation.

[bit6] CITE: Compare interrupt enable bit

This bit controls whether to enable or disable interrupt output to the CPU when a compare detection flag (CMPF) is set (during a compare operation).

CITE	Compare interrupt enable
0	Disables compare interrupt [initial value].
1	Enables compare interrupt.

[bit5] UDIE: Overflow/underflow interrupt enable bit

This bit controls whether to enable or disable interrupt output to the CPU when OVFF/UDFF is set (when overflow or underflow occurs).

UDIE	Overflow/underflow interrupt enable
0	Disables overflow/underflow interrupt [initial value].
1	Enables overflow/underflow interrupt.

[bit4] CMPF: Compare detection flag

This flag indicates that the comparison result of the UDCR value and RCR value are equal.

In write operations, the flag can only be set to "0", not to "1".

CMPF	Meaning of compare detection flag
0	Comparison result does not match [initial value].
1	Comparison result matches.

[bit3] OVFF: Overflow detection flag

This flag indicates the occurrence of an overflow.

In write operations, this flag can only be set to "0", not to "1".

OVFF	Meaning of overflow detection flag
0	No overflow [initial value]
1	Overflow

[bit2] UDF: Underflow detection flag

This flag indicates the occurrence of an underflow.

In write operations, this flag can only be set to "0", not to "1".

UDFF	Meaning of underflow detection flag
0	No underflow [initial value]
1	Underflow

[bit1, bit0] UDF1, UDF0: Up/down flags

These bits indicate the type of a counting operation (up or down) immediately preceding the current operation.

Only reading is allowed. No writing is allowed.

UDF1	UDF0	Detection edge
0	0	No input [initial value]
0	1	Down count
1	0	Up count
1	1	Both up and down counting were performed simultaneously.

14.3.4 Counter Control Register (CCR)

Counter control register (CCR) is the register which controls the operation mode of up/down counter. The function of bit15 (M16E) is different in odd channel and even channel.

■ Bit Configuration of Counter Control Register (CCR)

Figure 14.3-5 Bit Configuration of Counter Control Register (CCR)

CCRHO	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
Address 000544 _H	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	00000000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
CCRL0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address 000545 _H	-	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	-0001000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R	R	

R/W: Readable/writable
R: Read only

[bit15] M16E: 16-bit mode permission setting bit

8-bit/16-bit operation mode selection (switching) bit

M16E	16-bit mode enable setting
0	8-bit operation mode [initial value]
1	16-bit operation mode

[bit14] CDCF: Count direction change flag

This flag sets when the count direction is changed. When the count direction is changed up to down or down to up during counting, "1" is set to this bit.

Writing "0" clears the setting.

Writing "1" is ignored. The value of this bit is not changed.

CDCF	Direction change detection
0	Direction has not been changed [initial value].
1	Direction has been changed once or more.

The count direction is set to down immediately after a reset. Therefore, CDCF is set to "1" when up counting is performed immediately after a reset.

[bit13] CFIE: Count direction change interrupt enable bit

This bit controls the interrupt output for the CPU when CDCF is set. An interrupt occurs if the count direction is changed at least once during counting.

CFIE	Direction change interrupt enable
0	Disables direction change interrupt [initial value].
1	Enables direction change interrupt.

[bit12] CLKS: Internal prescaler selection bit

When timer mode is selected, this bit selects the frequency of the internal prescaler.

This bit is effective only in timer mode and only for down counting.

CLKS	Selected internal clock
0	Two peripheral clock (CLKP) cycles [initial value]
1	Eight peripheral clock (CLKP) cycles

[bit11, bit10] CMS1, CMS0: Counting mode selection bits

These bits select counting mode.

CMS1	CMS0	Counting mode
0	0	Timer mode (down count) [initial value]
0	1	Up or down counting mode
1	0	Phase difference counting mode, 2 multiplication
1	1	Phase difference counting mode, 4 multiplication

[bit9, bit8] CES1, CES0: Count clock edge selection bits

In up/down counting mode, these bits select the input of internal circuit or the detection edge of external pins AIN and BIN.

This setting is invalid in modes other than up or down counting mode.

CES1	CES0	Selection edge
0	0	Disables edge detection [initial value].
0	1	Detects falling edge.
1	0	Detects rising edge.
1	1	Detects rising and falling edges.

[bit7] Reserved: Reserved bit

This bit is reserved. Be sure to set this bit to "0".

[bit6] CTUT: Counter write bit

This bit transfers data from RCR to UDCR.

When this bit is set to "1", data is transferred from RCR to UDCR.

Writing "0" to this bit has no effect. The read value is always "0".

Do not set this bit to "1" during counting (when the CSTR bit of the CSR is "1").

[bit5] UCRE: UDCR clear enable bit

This bit controls the compare operation that clears UDCR.

UDCR clear functions other than clearing due to comparing (such as due to the ZIN pin), are not affected.

UCRE	Counter clear by compare
0	Disables counter clear [initial value].
1	Enables counter clear.

[bit4] RLDE: Reload enable bit

This bit controls the start of the reload function. When the reload function is started, if UDCR leads the underflow, this bit transfers the value of RCR to UDCR.

RLDE	Reload function
0	Disables the reload function [initial value].
1	Enables the reload function.

[bit3] UDCC: UDCR clear bit

This bit clears the UDCR. When this bit is set to "0", the UDCR is cleared to "0000_H".

Writing "1" to this bit has no effect. The read value is always "1".

[bit2] CGSC: Counter clear/gate selection bit

This bit selects the function of the external pin ZIN.

CGSC	ZIN pin function
0	Counter clear function [initial value]
1	Gate function

[bit1, bit0] CGE1, CGE0: Counter clear/gate edge selection bits

These bits select the detection edge/level of the external pin ZIN.

CGE1	CGE0	When counter clear function is selected	When gate function is selected
0	0	Disables edge detection [initial value].	Disables level detection [initial value] (count disable)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Setting disabled	Setting disabled

14.4 Operation of Up/Down Counters

This section describes the up/down counter operation.

■ Selecting Counting Mode

These counters/timers have four counting modes. The CMS1 and CMS0 bits of the CCR register are used to select the counting modes.

CMS1	CMS0	Counting mode
0	0	Timer mode (down count) [initial value]
0	1	Up/down counting mode
1	0	Phase difference counting mode, 2 multiplication
1	1	Phase difference counting mode, 4 multiplication

● Timer mode [down count]

In timer mode, the output of the internal prescaler is used for counting down. For the internal prescaler, either two peripheral clock (CLKP) cycles or eight peripheral clock (CLKP) cycles can be selected with the CLKS bit of the CCRH0 register.

● Up/down counting mode

In up/down counting mode, counting up/down is performed by counting the input through external pins AIN and BIN. The input through the AIN pin controls counting up and the input through the BIN pin controls counting down.

The inputs through the AIN pin and BIN pin are subject to edge-detected. The edge detection can be selected by the CES1 and CES0 bits of the CCRH register.

CES1	CES0	Selection edge
0	0	Disables the edge detection. [initial value]
0	1	Detects falling edge.
1	0	Detects rising edge.
1	1	Detects both falling and rising edges.

● Phase difference counting mode (two multiplication/four multiplication)

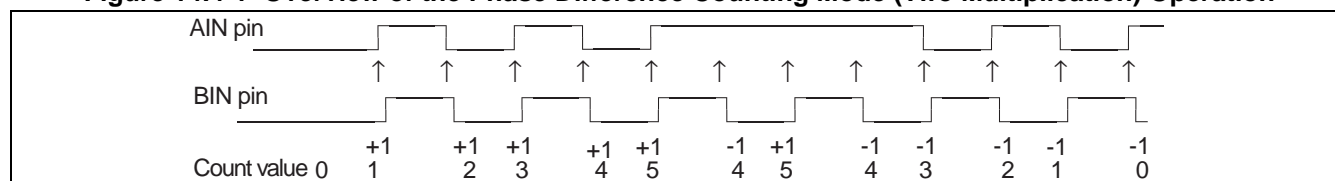
In phase difference counting mode, to count the phase difference between phase A and phase B of the output signal for the encoder, detect the input level of the BIN pin at input edge detection of the AIN pin.

For the phase difference between AIN pin input and BIN pin input in two multiplication or four multiplication mode, count up if the AIN is faster, and count down if the BIN is faster.

In two multiplication mode, counting is performed by detecting the value of the AIN pin in the period between the rising and falling edges of the BIN pin. In this case, counting is performed as follows:

Edge of the BIN pin	Level of the AIN pin	Count
Rising ↑	"H" level	Count up
Rising ↑	"L" level	Count down
Falling ↓	"H" level	Count down
Falling ↓	"L" level	Count up

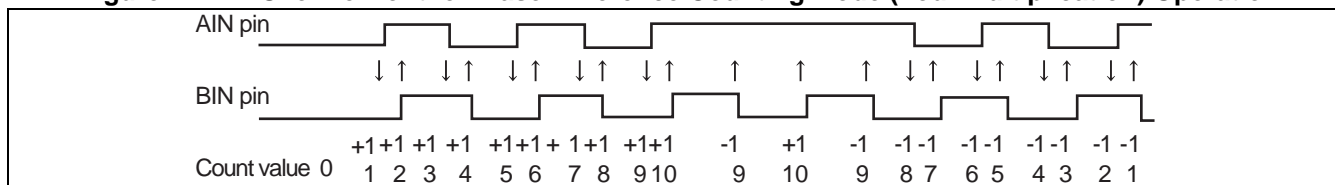
Figure 14.4-1 Overview of the Phase Difference Counting Mode (Two Multiplication) Operation



In four-multiplication mode, counting is performed by detecting the value of the AIN pin at the timing between the rising and falling edges of the BIN pin and detecting the value of the BIN pin at the timing between the rising and falling edges of the AIN pin. In this case, counting is performed as follows:

Edge input	Edge	Level input	Level	Count
BIN	Rising ↑	AIN	"H" level	Count up
	Rising ↑		"L" level	Count down
	Falling ↓		"H" level	Count down
	Falling ↓		"L" level	Count up
AIN	Rising ↑	BIN	"H" level	Count down
	Rising ↑		"L" level	Count up
	Falling ↓		"H" level	Count up
	Falling ↓		"L" level	Count down

Figure 14.4-2 Overview of the Phase Difference Counting Mode (Four Multiplication) Operation



For counting the encoder output, by inputting the A phase to the AIN pin, the B phase to the BIN pin, and the Z phase to the ZIN pin, a highly precise count of the rotation angle and number of rotations can be obtained and the rotation direction can be detected as well.

When this counting mode is selected, the detection edge selection with the CES1 and CES0 bits is invalid.

■ Reload/Compare Function

This counters have reload and compare clear functions, which can be combined for processing.

The examples of setting are shown in following.

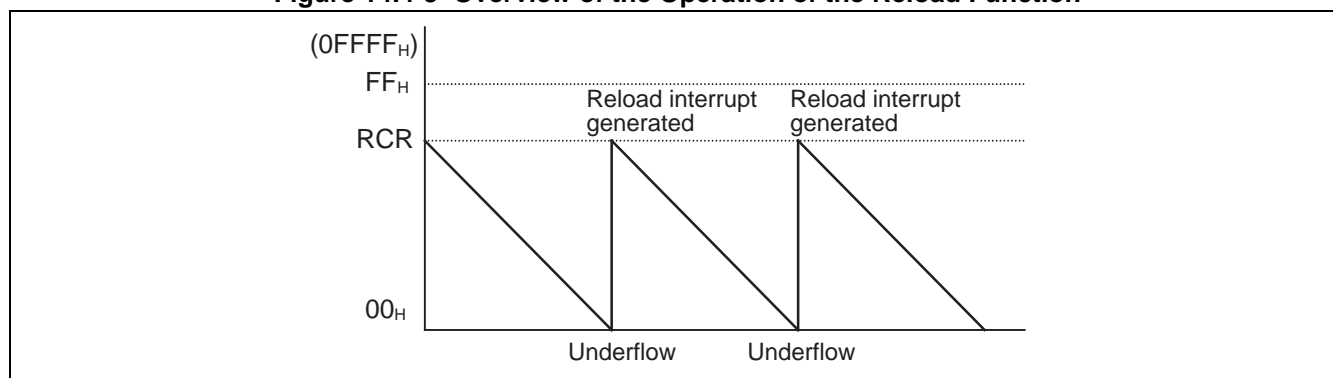
RLDE	UCRE	Reload/Compare function
0	0	Disables clearing by reload/compare [initial value].
0	1	Enables clearing by compare.
1	0	Reload is enabled.
1	1	Enables clearing by reload/compare.

● Reload function

When the reload function is started, the value of the RCR is transferred to the UDCR with the timing of the down count clock after an underflow. In this case, when UDFB bit is set, an interrupt request is generated.

In a mode in which down counting is not performed, starting this function is invalid.

Figure 14.4-3 Overview of the Operation of the Reload Function

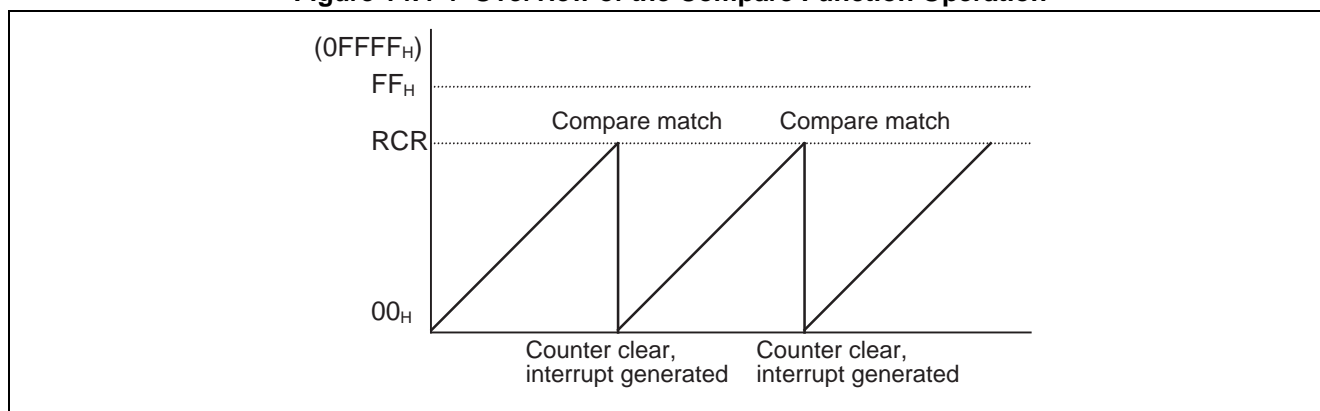


● Compare clear function

When the compare clear function is enabled, the compare function can be used in all modes other than timer mode. When the compare function is started, if the value of RCR and the value of UDCR match, CMPF bit is set and an interrupt request is generated. When the compare clear function is started, the UDCR is cleared with the timing of the next up count clock. (The UDCR is not cleared when counting down is performed.)

In a mode in which up counting is not performed, starting this function is invalid.

Figure 14.4-4 Overview of the Compare Function Operation

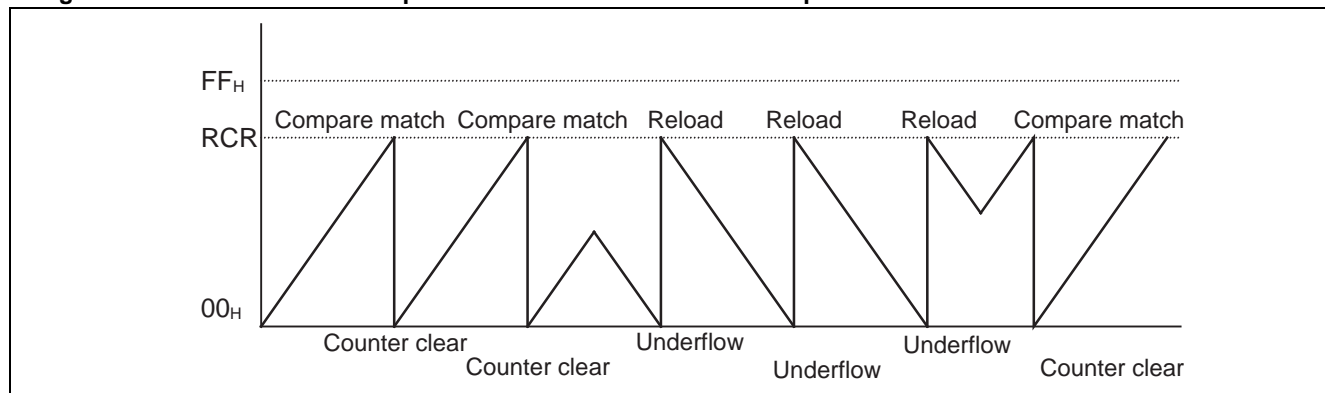


■ Synchronous Start of Reload/Compare Function

When the reload/compare function is started, counting up or down can be performed with an arbitrary width.

The reload function is started at an underflow and transfers the value of the RCR to the UDCR. When the values of RCR and UDCR match, the compare function clears the UDCR. By using these functions, counting up or down is performed for values between "0000_H" and the value of the RCR.

Figure 14.4-5 Overview of the Operation when the Reload and Compare Functions are Started at the Same Time

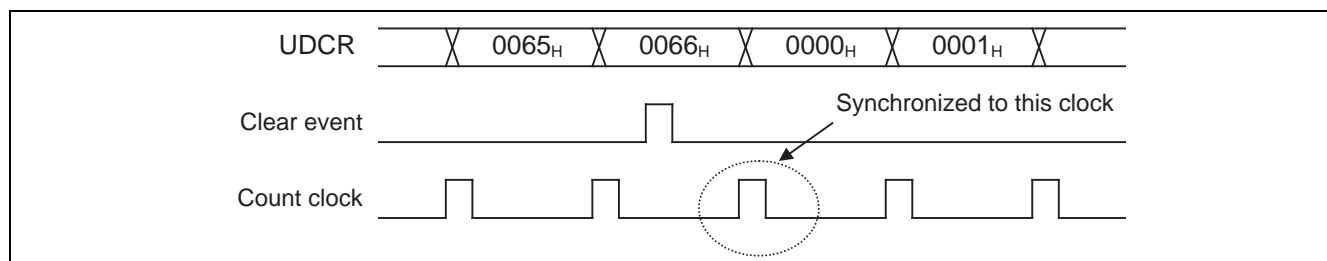


An interrupt to the CPU can be generated at a compare match or at reload (underflow). These interrupt outputs can be enabled separately.

The timing for clearing the UDCR is different during counting and when counting is stopped.

Reloading (writing "1" to the CTUT bit) by software is not allowed during counting.

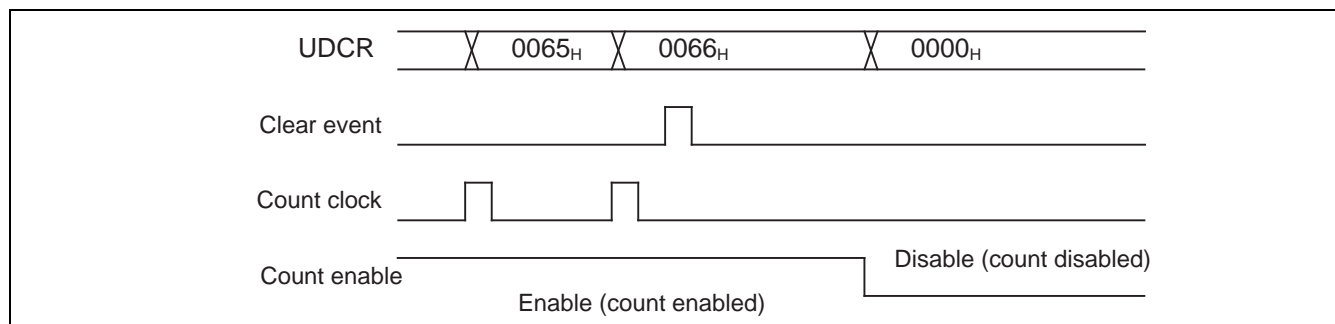
- During counting, if an event for clearing occurs, all the events are synchronized with the count clock.



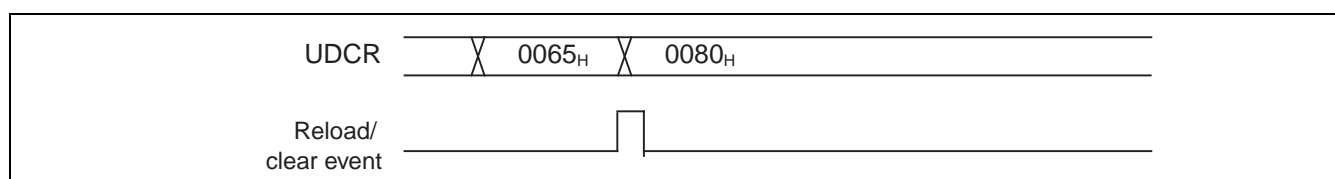
Reference:

During counting, reloading due to an underflow is performed in synchronization with all count clocks.

- When an event for clearing occurs during counting, if counting is stopped in count clock synchronization wait state (state of waiting for the count input for synchronization), the clear operations are performed when counting is stopped.



- If the events for reloading and clearing occur during counting, reload and clear are performed when the event occurs.



Clear by compare is performed when the values of the UDCR and the RCR match and while counting up. If down counting is performed or counting is stopped, the clear operation is not performed even when the values of the UDCR and the RCR match.

As for the timing of clearing and reloading, the clear operation follows the above timing for all events other than reset input, and reloading also uses the above timing for all events.

When the events for clearing and reloading occur at the same time, the clear event takes priority.

■ Writing Data to UDCR

Data cannot be written to the UDCR directly from the data bus. To write arbitrary value to the UDCR, follow the procedure below.

- Write the data that is to be written to the UDCR first to the RCR (Note that this means that the original data in the RCR will be lost).
 - By setting the CTUT bit of the CCR to 1, data is transferred from the RCR to the UDCR.
- Perform the above operation while counting is stopped (when the CSTR bit of the CSR is 0).

Reference:

If 1 is written to the CTUT bit by mistake during counting, the value of the RCR is transferred to the UDCR at the timing for a write.

Besides the above procedure, the following procedure can also be applied to clear the counter.

- Clearing by reset input.
- Clearing by edge input through the ZIN pin.
- Clearing by writing 0 to UDCC bit of the CCR.
- Clearing by compare function.

The above can be performed regardless of whether counting is performed or stopped.

■ Count Clear/Gate Function

The ZIN pin can be used after selecting the count clear function or gate function based on the CGSC bit of the CCR register.

When the count clear function is started, the ZIN pin clears the counter. The CGE1 and CGE0 bits of the CCRL register can control which edge input of the ZIN pin to use for counting.

When the gate function is started, the ZIN pin enables or disables counting. The CGE1 and CGE0 bits of the CCR register can control which level input of the ZIN pin enables counting.

This function is effective for all modes.

Table 14.4-1 ZIN Pin Function

CGSC	ZIN pin function
0	Counter clear function [initial value]
1	Gate function

Table 14.4-2 Count Clear/Gate Function

CGE1	CGE0	When counter clear function is used	When gate function is used
0	0	Disables edge detection. [initial value]	Disables level detection. [initial value] (count disable)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Setting disabled	Setting disabled

■ Count Direction Flag

The count direction flag (UDF1 and UDF0) indicates at the time of up/down counting whether the counting operation preceding the current operation was counting up or down. Based on the count clock signal from the input of the AIN and BIN pins, the value of this flag changes for each count. Current rotation direction, such as control of motor, can be determined by referring to this flag.

Table 14.4-3 Count Direction Flag

UDF1	UDF0	Count direction
0	0	Without input [initial value]
0	1	Down count
1	0	Up count
1	1	Up/down occurs simultaneously (no counting operation is performed).

■ Count Direction Change Flag

The CDCF is set when the counting direction changes between up and down. Simultaneously to setting this flag, an interrupt request to the CPU can be generated. By referring to the interrupt and count direction flag, the direction to which counting is changed can be determined. However, note that when the period of direction change is short and multiple direction changes are performed in succession, the direction that the flag indicates after the direction change may return to the original direction so that it appears as if the counting direction has not changed at all in between.

Table 14.4-4 Count Direction Change Flag

CDCF	Direction change detection
0	No direction change [initial value]
1	Direction has changed (at least once).

■ Compare Detection Flag

The CMPF is set when the values of UDCR and RCR match during counting. This flag is set for a match during counting up or down, match by occurrence of a reloading event, as well as when the values already match when counting started.

■ Operations for 8-bits and 16-bit

This module can be used as an 8-bit up/down counter or a 16-bit up/down counter. Setting the M16E bit of the CCR register to 0 sets 8-bit mode. Setting the bit to "1" sets 16-bit mode.

■ **Interrupt Generation Timing**

Interrupt flag	Flag setting interrupt	Reload	Clear
CDCF (Count direction change flag)	An interrupt is generated simultaneously with setting of the flag when counting starts immediately after the counting direction is changed.		
CMPF (Compare detection flag)	An interrupt is generated simultaneously with setting of the flag when the values of RCR and UDCR match when up or down counting, or reload counting is initiated.		UDCR is cleared at the timing of the first up count after RCR and UDCR match. (UDCR is not cleared for down counting).
OVFF (Overflow detection flag)	An interrupt is generated simultaneously with setting of the flag at the timing of the first up count after the count reaches "FFFF _H ".		UDCR is cleared at the timing of the first count after the count reaches "FFFF _H ".
UDFF (Underflow detection flag)	An interrupt is generated simultaneously with setting of the flag at the timing of the first down count after the count reaches "0000 _H ".	The value of RCR is transferred to UDCR at the timing of the first count after the count reaches "0000 _H ".	

- Because the value of RCR is used for both the reload and compare values, the compare flag is set always when reloading is performed.
- If the clear function is enabled, clearing occurs when up counting is performed after the compare match during down counting.

■ **Note**

The count direction is set to down when the count is reset. Therefore, at the first up count after resetting, CDCF bit is set to 1 to indicate that the counting direction has been changed.

After the up count register (UDCR) reaches the maximum count that the register can hold, counting continues without a carry-over. It therefore appears that counting is continuing with the up/down count register cleared.

CHAPTER 15

MULTI-FUNCTION SERIAL INTERFACE

This chapter describes the functions and operations of the multi-function serial interface.

- 15.1 Characteristics of Multi-function Serial Interface
- 15.2 UART (Asynchronous Serial Interface)
- 15.3 Overview of UART (Asynchronous Serial Interface)
- 15.4 Registers of UART (Asynchronous Serial Interface)
- 15.5 Interrupts of UART
- 15.6 Operation of UART
- 15.7 Dedicated Baud Rate Generator
- 15.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
- 15.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)
- 15.10 Notes on UART Mode
- 15.11 CSIO (Clock Synchronous Serial Interface)
- 15.12 Overview of CSIO (Clock Synchronous Serial Interface)
- 15.13 Registers of CSIO (Clock Synchronous Serial Interface)
- 15.14 Interrupts of CSIO (Clock Synchronous Serial Interface)
- 15.15 Operation of CSIO (Clock Synchronous Serial Interface)
- 15.16 Dedicated Baud Rate Generator
- 15.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)
- 15.18 Notes on CSIO Mode
- 15.19 I²C Interface
- 15.20 Overview of I²C Interface
- 15.21 Registers of I²C Interface
- 15.22 Interrupts of I²C Interface
- 15.23 Dedicated Baud Rate Generator
- 15.24 Notes on I²C Mode

15.1 Characteristics of Multi-function Serial Interface

This multi-function serial interface has the following characteristics.

■ Interface Mode

The following interface modes are selectable for the multi-function serial interface depending on the operation mode settings.

- UART0 (Asynchronous normal serial interface)
- UART1 (Asynchronous multi-processor serial interface)
- CSIO (Clock synchronous serial interface) (SPI can be supported)
- I²C (I²C bus interface)

■ Switching the Interface Mode

To communicate through each serial interface, the registers shown in Table 15.1-1 should be used to set the operation mode before starting the communication.

Figure 15.1-1 Bit Structure of SMR Register

SMR		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Address:									
ch.0 000063 _H	ch.1 000073 _H	MD2	MD1	MD0	-	SBL	BDS	SCKE	SOE
ch.2 000083 _H	ch.3 000093 _H								
ch.4 000563 _H	ch.5 000573 _H								
Read/Write		(R/W)	(R/W)	(R/W)	(-)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value		(0)	(0)	(0)	(-)	(0)	(0)	(0)	(0)

Table 15.1-1 Switching Interface Mode

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (Asynchronous normal serial interface)
0	0	1	UART1 (Asynchronous multi-processor serial interface)
0	1	0	CSIO (Clock synchronization serial interface) (SPI can be supported)
1	0	0	I ² C (I ² C bus interface)

Note: Settings other than above are prohibited.

Notes:

- Transmission and reception cannot be guaranteed when the operation mode is switched while one of the serial interfaces is still in use for transmission or reception operation.
- The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

■ Transmission/Reception FIFO

This UART has a 16-byte transmission FIFO and 16-byte reception FIFO. The FIFO steps should be converted to 16 bytes when reading through this text.

15.2 UART (Asynchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes those supported in operation modes 0 and 1.

- UART (Asynchronous Serial Interface)
- Overview of UART (Asynchronous Serial Interface)
- Registers of UART (Asynchronous Serial Interface)
 - Serial Control Register (SCR)
 - Serial Mode Register (SMR)
 - Serial Status Register (SSR)
 - Extended Communication Control Register (ESCR)
 - Reception Data Register/Transmission Data Register (RDR0/RDR1/TDR0/TDR1)
 - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
 - FIFO Control Register 1 (FCR1)
 - FIFO Control Register 0 (FCR0)
 - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of UART
 - Occurrence of Reception Interrupts and Flag Set Timing
 - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
 - Occurrence of Transmission Interrupts and Flag Set Timing
 - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of UART
- Dedicated Baud Rate Generator
 - Setting Baud Rate
- Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
- Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

15.3 Overview of UART (Asynchronous Serial Interface)

UART (asynchronous serial interface) is a general-purpose serial data communication interface to perform asynchronous communication (start-stop synchronization) with an external unit. The UART supports a two-way communication function (normal mode) and a master/slave communication function (multi-processor mode: the master and slaves both supported). The UART also has transmission/reception FIFO.

■ Functions of UART (Asynchronous Serial Interface)

		Function
1	Data	<ul style="list-style-type: none"> Full-duplex double buffer (when FIFO is not used) Transmission/reception FIFO (maximum size: 16 bytes each) (when FIFO is used)
2	Serial input	Oversampling is performed for three times to determine the reception value by the majority of the sampling values achieved.
3	Transfer system	Asynchronous
4	Baud rate	<ul style="list-style-type: none"> Dedicated baud rate generator (15-bit reload counter configuration) The reload counter can be used to adjust the external clock input.
5	Data length	5 to 9 bits (in normal mode), 7 or 8 bits (in multi-processor mode)
6	Signaling system	NRZ (Non Return to Zero), inverted NRZ
7	Start bit detection	<ul style="list-style-type: none"> Synchronized with the falling edge of a start bit (NRZ) Synchronized with the rising edge of a start bit (inverted NRZ)
8	Reception error detection	<ul style="list-style-type: none"> Framing error Overrun error Parity error *
9	Interrupt request	<ul style="list-style-type: none"> Reception interrupt (completion of reception, framing error, overrun error, parity error*) Transmission interrupt (transmission data empty, transmission bus idle) Transmission FIFO interrupt (when transmission FIFO is empty) The extended intelligent I/O service (EI²OS) and DMA function are available for both transmission and reception.
10	Master/slave communication function (multi-processor mode)	Communication between 1 (master) and n (slaves) is enabled. (The master and slave systems are both supported.)
11	FIFO options	<ul style="list-style-type: none"> Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes) Transmission FIFO or reception FIFO selectable Transmission data can be resent. The interrupt timing for reception FIFO can be modified by software. FIFO reset is supported separately.

*: The detection of a parity error is enabled only in normal mode.

15.4 Registers of UART (Asynchronous Serial Interface)

This section lists the registers of UART (asynchronous serial interface).

■ List of Registers of UART (Asynchronous Serial Interface)

Figure 15.4-1 List of Registers of UART (Asynchronous Serial Interface)

Address		bit15	bit8 bit7	bit0
UART	000062 _H	000063 _H	SCR (serial control register)	SMR (serial mode register)
	000072 _H	000073 _H		
	000082 _H	000083 _H		
	000092 _H	000093 _H		
	000562 _H	000563 _H		
	000572 _H	000573 _H		
	000060 _H	000061 _H	SSR (serial status register)	ESCR (extended communication control register)
	000070 _H	000071 _H		
	000080 _H	000081 _H		
	000090 _H	000091 _H		
	000560 _H	000561 _H		
	000570 _H	000571 _H		
	000066 _H	000067 _H	RDR/TDR (transmission/reception data register)	
	000076 _H	000077 _H		
	000086 _H	000087 _H		
	000096 _H	000097 _H		
	000566 _H	000567 _H		
	000576 _H	000577 _H		
	000064 _H	000065 _H	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)
	000074 _H	000075 _H		
	000084 _H	000085 _H		
	000094 _H	000095 _H		
	000564 _H	000565 _H		
	000574 _H	000575 _H		
000068 _H	000069 _H	-	-	
000078 _H	000079 _H			
000088 _H	000089 _H			
000098 _H	000099 _H			
000568 _H	000569 _H			
000578 _H	000579 _H			

(Continued)

(Continued)

Address		bit15	bit8 bit7	bit0
FIFO	00006E _H 00006F _H	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)	
	00007E _H 00007F _H			
	00008E _H 00008F _H			
	00009E _H 00009F _H			
	00056E _H 00056F _H			
	00057E _H 00057F _H			
	00006C _H 00006D _H	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)	
	00007C _H 00007D _H			
	00008C _H 00008D _H			
	00009C _H 00009D _H			
	00056C _H 00056D _H			
	00057C _H 00057D _H			

Table 15.4-1 Bit Assignment of UART (Asynchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SBL	BDS	SCKE	SOE
SSR/ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	-	-	INV	PEN	P	L2	L1	L0
TDR1/ TDR0 (RDR1/ RDR0)	-							D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	-	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

■ Operation Mode

The UART (asynchronous serial interface) operates in two different modes. The mode selection is determined by MD2, MD1 and MD0 in the serial mode register (SMR).

Table 15.4-2 Operation Modes of UART (Asynchronous Serial Interface)

Operation mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multi-processor mode)

15.4.1 Serial Control Register (SCR)

The serial control register (SCR) enables or disables transmission/reception, transmission/reception interrupts, and transmission bus idle interrupts. SCR can also reset the UART.

Serial Control Register (SCR)

Figure 15.4-2 shows the bit structure of the serial control register (SCR), and Table 15.4-3 describes the function of each bit.

Figure 15.4-2 Bit Structure of Serial Control Register (SCR)

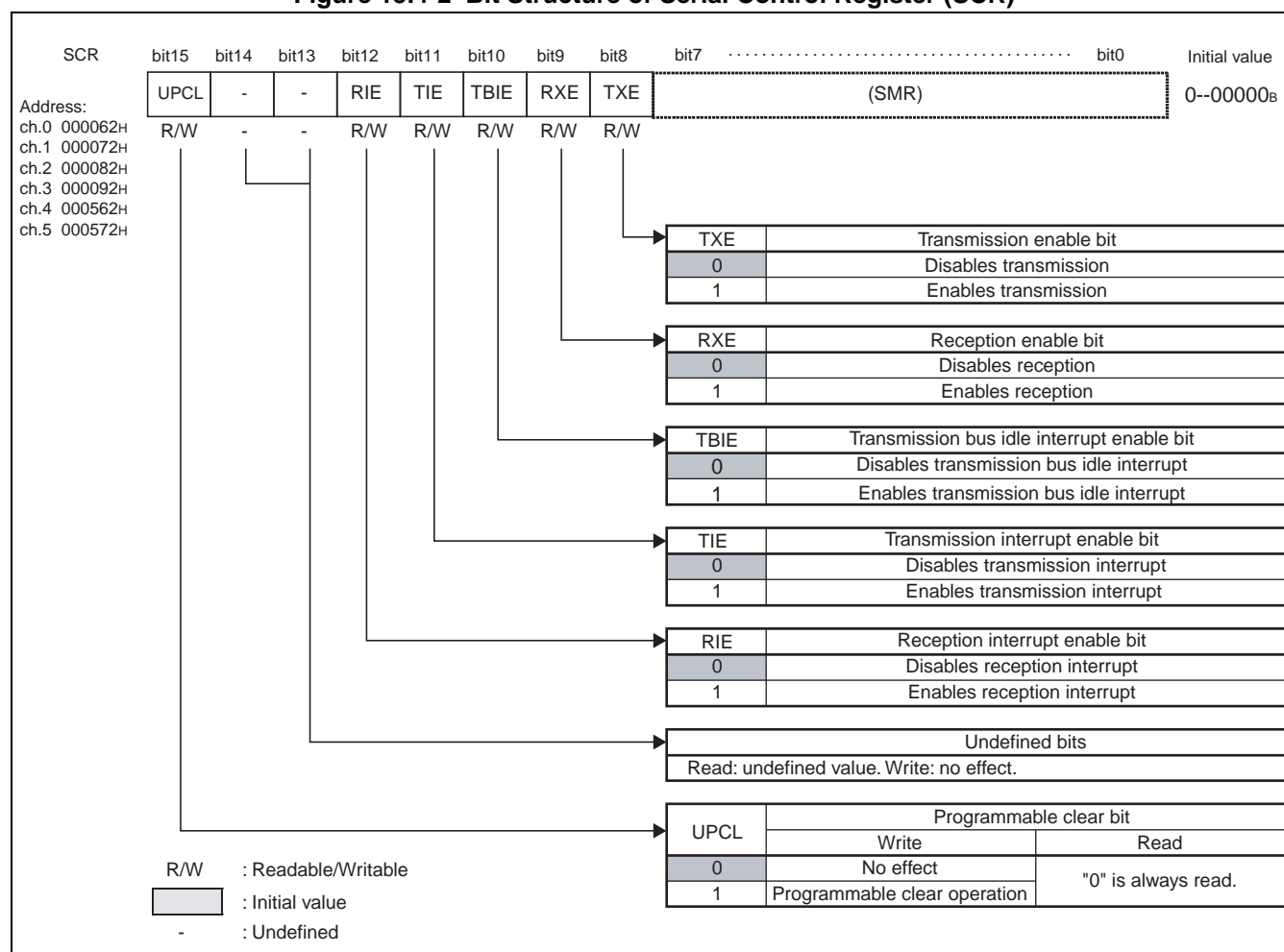


Table 15.4-3 Functional Description of Each Bit of Serial Control Register (SCR) (1 / 2)

Bit name		Function
bit15	UPCL: Programmable clear bit	<p>This bit is used to initialize the internal state of the UART.</p> <p>Setting the bit to "1":</p> <ul style="list-style-type: none"> The UART will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. All the transmission/reception interrupt sources (PE, FRE, ORE, RDRF, TDRE and TBI) will be initialized (000011_B). <p>Setting the bit to "0": No effect on the operation</p> <p>Reading this bit always returns "0".</p> <p>Note:</p> <p>Execute the programmable clear operation after disabling interrupts.</p> <p>Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.</p>
bit14, bit13	Undefined bits	<p>Read: undefined value</p> <p>Write: no effect</p>
bit12	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of reception interrupt requests to the CPU. A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (PE, ORE or FRE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of transmission interrupt requests to the CPU. A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	<p>This bit is used to enable/disable UART reception operation.</p> <ul style="list-style-type: none"> Setting the bit to "0" disables the reception operation. Setting the bit to "1" enables the reception operation. <p>Note:</p> <p>Even when the reception operation is enabled (RXE = 1), such operation does not start until the falling edge of a start bit (in NRZ format: INV = 0) is input. (When the inverted NRZ format is selected (INV = 1), the reception operation does not start until the rising edge is input.)</p> <p>If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.</p>

Table 15.4-3 Functional Description of Each Bit of Serial Control Register (SCR) (2 / 2)

Bit name		Function
bit8	TXE: Transmission enable bit	<p>This bit is used to enable/disable UART transmission operation.</p> <ul style="list-style-type: none">• Setting the bit to "0" disables the transmission operation.• Setting the bit to "1" enables the transmission operation. <p>Note: If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.</p>

15.4.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction, data length and stop bit length, and enables or disables the output to the serial data and serial clock pins.

Serial Mode Register (SMR)

Figure 15.4-3 shows the bit structure of the serial mode register (SMR), and Table 15.4-4 describes the function of each bit.

Figure 15.4-3 Bit Structure of Serial Mode Register (SMR)

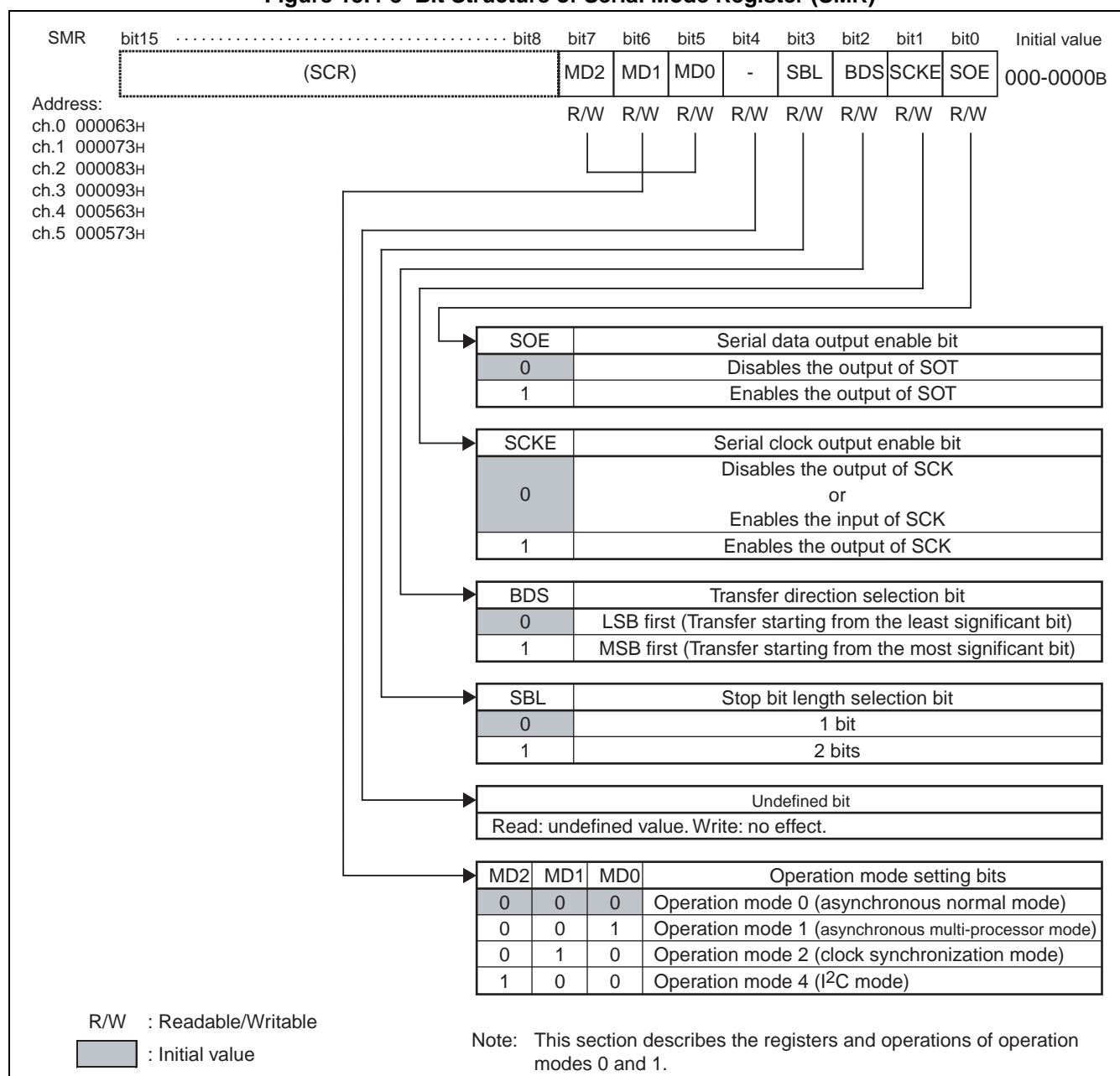


Table 15.4-4 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2, MD1, MD0: Operation mode setting bits	<p>These bits set the operation mode for the asynchronous serial interface.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode).</p> <p>Note:</p> <p>Settings other than above are prohibited.</p> <p>To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.</p>
bit4	Undefined bit	<p>Read: undefined value</p> <p>Write: no effect</p>
bit3	SBL: Stop bit length selection bit	<p>This bit is used to select a bit length for a stop bit (frame end mark of transmission data).</p> <p>Setting the bit to "0" sets the stop bit to 1 bit in length.</p> <p>Setting the bit to "1" sets the stop bit to 2 bits in length.</p> <p>Note:</p> <p>In reception, only the first bit of each stop bit is always detected.</p> <p>Set this bit when transmission is disabled (TXE = 0).</p>
bit2	BDS: Transfer direction selection bit	<p>This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1).</p> <p>Note:</p> <p>Set this bit when transmission and reception are disabled (TXE = RXE = 0).</p>
bit1	SCKE: Serial clock output enable bit	<p>This bit is used to control the I/O port of the serial clock.</p> <p>Setting the bit to "0":</p> <p>The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port. Also select the external clock (BGR:EXT = 1) using the external clock selection bit.</p> <p>Setting the bit to "1" enables the output of SCK.</p>
bit0	SOE: Serial data output enable bit	<p>This bit is used to enable/disable the output of serial data.</p> <p>Setting the bit to "0" disables the output.</p> <p>Setting the bit to "1" enables the output of SOT.</p>

Note:

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

15.4.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

■ Serial Status Register (SSR)

Figure 15.4-4 shows the bit structure of the serial status register (SSR) and Table 15.4-5 describes the function of each bit.

Figure 15.4-4 Bit Structure of Serial Status Register (SSR)

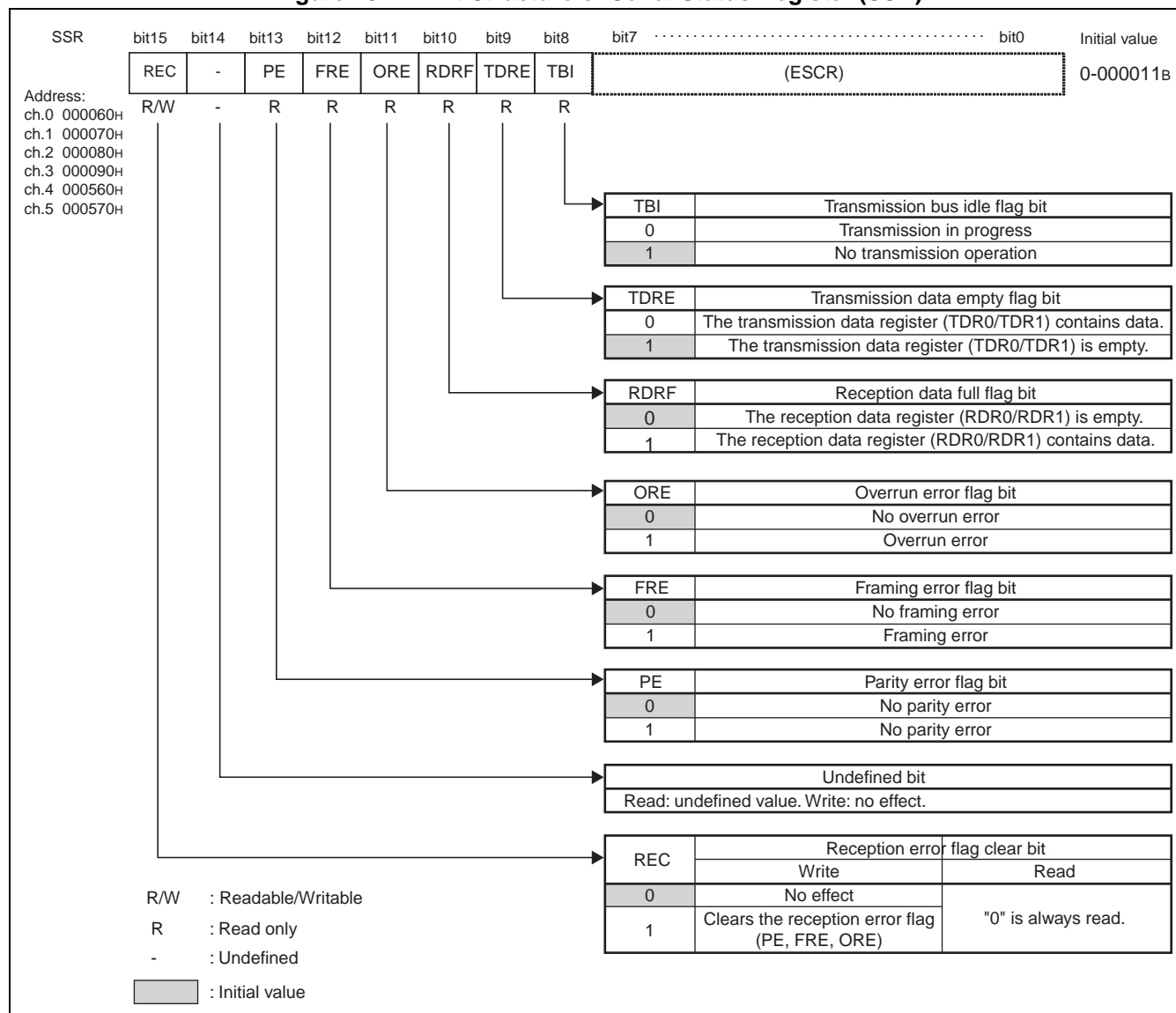


Table 15.4-5 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	<p>This bit is used to clear the PE, FRE or ORE flag in the serial status register (SSR).</p> <ul style="list-style-type: none"> • Writing "1" clears the error flag. • Writing "0" has no effect. <p>Reading this bit always returns "0".</p>
bit14	Undefined bit	<p>Read: undefined value Write: no effect</p>
bit13	PE: Parity error flag bit (only available in operation mode 0)	<ul style="list-style-type: none"> • The bit is set to "1" when a parity error occurs during reception (SMR:PEN = 1). The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the PE bit and the SCR:RIE bit are set to "1". • When this flag is set, the data in the reception data register (RDR0/RDR1) is invalid. • If this flag is set during the use of reception FIFO, the reception FIFO enable bit will be cleared and no reception data will be stored to the reception FIFO.
bit12	FRE: Framing error flag bit	<ul style="list-style-type: none"> • This bit is set to "1" when a framing error occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the FRE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR0/RDR1) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit11	ORE: Overrun error flag bit	<ul style="list-style-type: none"> • This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR0/RDR1) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

Table 15.4-5 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit10	RDRF: Reception data full flag bit	<ul style="list-style-type: none"> This flag indicates the status of the reception data register (RDR0/RDR1). The bit is set to "1" when reception data is loaded to RDR0/RDR1. The bit is cleared to "0" when the reception data register (RDR0/RDR1) is read. A reception interrupt request is output when the RDRF and RIE bits are set to "1". RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. When the reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" during the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR0/RDR1 is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	<ul style="list-style-type: none"> This flag indicates the status of the transmission data register (TDR0/TDR1). When transmission data is written to TDR0/TDR1, the bit becomes "0", indicating that TDR0/TDR1 contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR0/TDR1 no longer contains any valid data. A transmission interrupt request is output when the TDRE and TIE bits are set to "1". The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "15.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
bit8	TBI: Transmission bus idle flag bit	<ul style="list-style-type: none"> This bit indicates that the UART is not performing transmission operation. The bit is set to "0" when transmission data is written to the transmission data register (TDR0/TDR1). The bit is set to "1" when the transmission data register is empty (TDRE =1) and no transmission operation is in progress. The TBI bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

15.4.4 Extended Communication Control Register (ESCR)

The extended communication control register (ESCR) can be used to set the transmission/reception data length, enable/disable the parity bit, select the parity bit, and invert the serial data format.

■ Bit Structure of the Extended Communication Control Register (ESCR)

Figure 15.4-5 shows the bit structure of the extended communication control register (ESCR) and Table 15.4-6 describes the function of each bit.

Figure 15.4-5 Bit Structure of Extended Communication Control Register (ESCR)

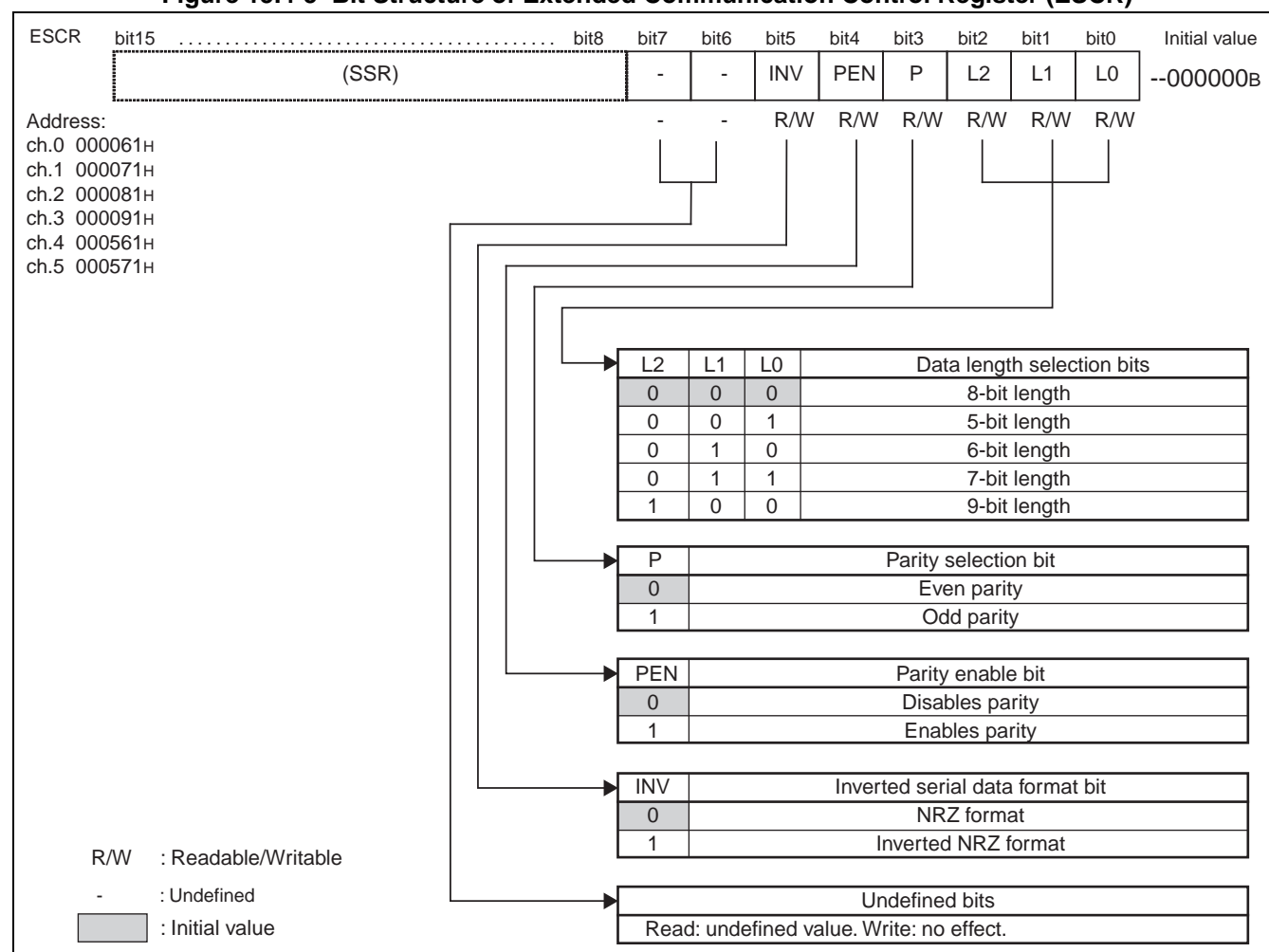


Table 15.4-6 Functional Description of Each Bit of Extended Communication Control Register (ESCR)

Bit name		Function
bit7, bit6	Undefined bits	Read: undefined value Write: no effect
bit5	INV: Inverted serial data format bit	This bit is used to select the NRZ format or the inverted NRZ format as the serial data format.
bit4	PEN: Parity enable bit (only available in operation mode 0)	This bit is used to determine whether the parity bit should be added (in transmission) or detected (in reception). <ul style="list-style-type: none"> When this bit is set to "0", the parity bit is not added. When this bit is set to "1", the parity bit is added. Note: This bit is fixed to "0" internally in operation mode 1.
bit3	P: Parity selection bit (only available in operation mode 0)	This bit is used to select odd parity "1" or even parity "0" when parity is enabled (ESCR:PEN = 1). <ul style="list-style-type: none"> Setting the bit to "0" selects even parity. Setting the bit to "1" selects odd parity.
bit2 to bit0	L2, L1, L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. <ul style="list-style-type: none"> Selecting "000_B" sets the data length to 8 bits. Selecting "001_B" sets the data length to 5 bits. Selecting "010_B" sets the data length to 6 bits. Selecting "011_B" sets the data length to 7 bits. Selecting "100_B" sets the data length to 9 bits. Note: Settings other than above are prohibited. For operation mode 1, set the data length to 7 or 8 bits. Any other setting is prohibited.

15.4.5 Reception Data Register / Transmission Data Register (RDR0/RDR1/TDR0/TDR1)

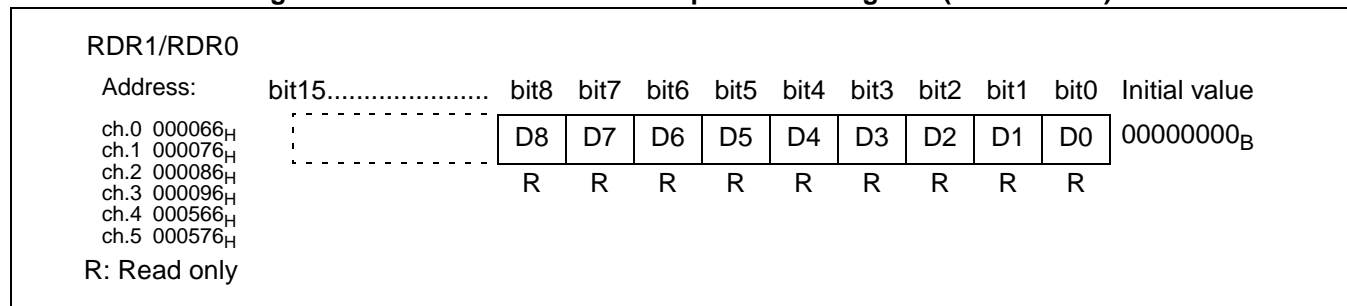
The reception and transmission data registers are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

When FIFO operation is enabled, the RDR0/RDR1/TDR0/TDR1 address becomes the read/write address for the FIFO.

■ Reception Data Register (RDR0/RDR1)

Figure 15.4-6 illustrates the bit structure of the serial reception register (RDR0/RDR1).

Figure 15.4-6 Bit Structure of Reception Data Register (RDR0/RDR1)



The reception data register (RDR0/RDR1) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR0/RDR1).
- "0" is placed in one of the upper bits, depending on the data length, as shown below.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR0/RDR1). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).
- Read the reception data register (RDR0/RDR1) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the reception data register (RDR0/RDR1) is read.
- If a reception error occurs (one of SSR:PE, ORE, or FRE is "1"), the data in the reception data register (RDR0/RDR1) becomes invalid.

- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the received AD bit is stored in bit D8.
 - 16-bit access is used to read RDR0/RDR1 for a 9-bit transfer in operation mode 1.
-

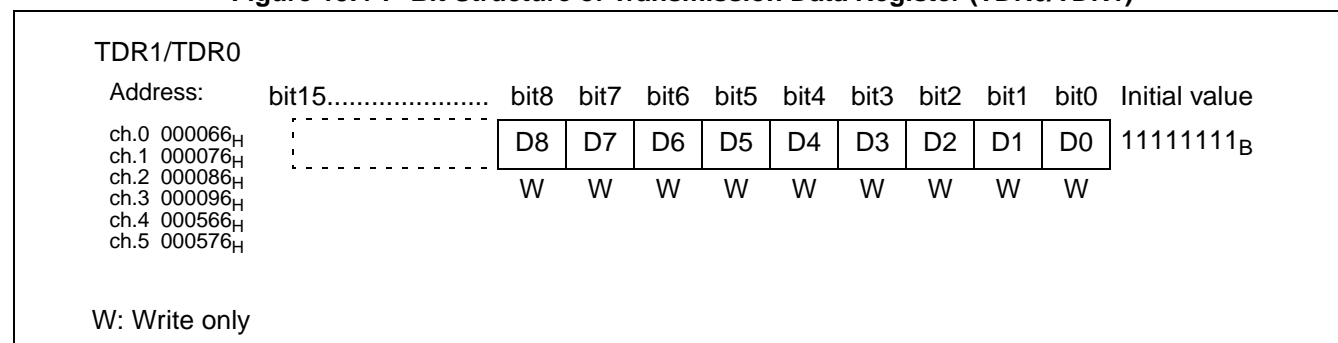
Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
 - RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
 - If a reception error occurs (one of SSR:PE, ORE, or FRE is "1") when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
-

■ Transmission Data Register (TDR0/TDR1)

Figure 15.4-7 illustrates the bit structure of the transmission data register.

Figure 15.4-7 Bit Structure of Transmission Data Register (TDR0/TDR1)



The transmission data register (TDR0/TDR1) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR0/TDR1) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR0/TDR1).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- Transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".
- Transmission data cannot be written when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.
- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the AD bit is sent by writing to bit D8.
- 16-bit access is used to write to TDR0/TDR1 for a 9-bit transfer in operation mode 1.

Notes:

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. These registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
 - For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "15.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
-

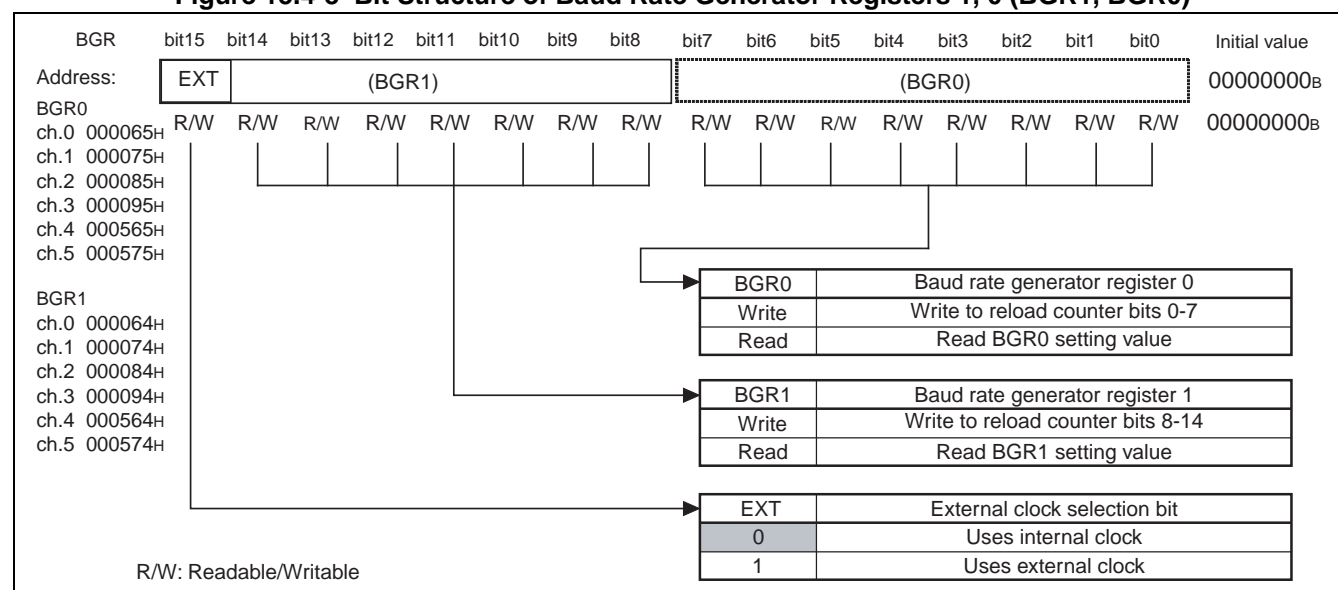
15.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock. They also allow an external clock to be selected as the clock source for the reload counter.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 15.4-8 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 15.4-8 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



- The baud rate generator registers are used to set a division ratio for the serial clock.
- BGR1 and BGR0 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers (BGR1/BGR0).
- The EXT bit (bit15) is used to determine whether the internal clock or external clock should be used as the clock source for the reload counter. Setting EXT to "0" selects the internal clock, while setting EXT to "1" selects the external clock.

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
 - When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a programmable clear (UPCL) operation after changing the BGR1/BGR0 setting value.
 - When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (CLKP) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
 - Select 4 or a larger value for BGR1/BGR0. However, data may not be able to be received properly, due to a baud rate error or reload settings.
 - To change the setting to the external clock (EXT = 1) during the operation of the baud rate generator, write "0" to baud rate generator registers 1, 0 (BGR1, BGR0), execute a programmable clear (UPCL) operation, and then set to the external clock (EXT = 1).
-

15.4.7 FIFO Control Register 1 (FCR1)

The FIFO control register (FCR1) sets a FIFO test, selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 15.4-9 shows the bit structure of the FIFO control register 1 (FCR1) and Table 15.4-7 describes the function of each bit.

Figure 15.4-9 Bit Structure of FIFO Control Register 1 (FCR1)

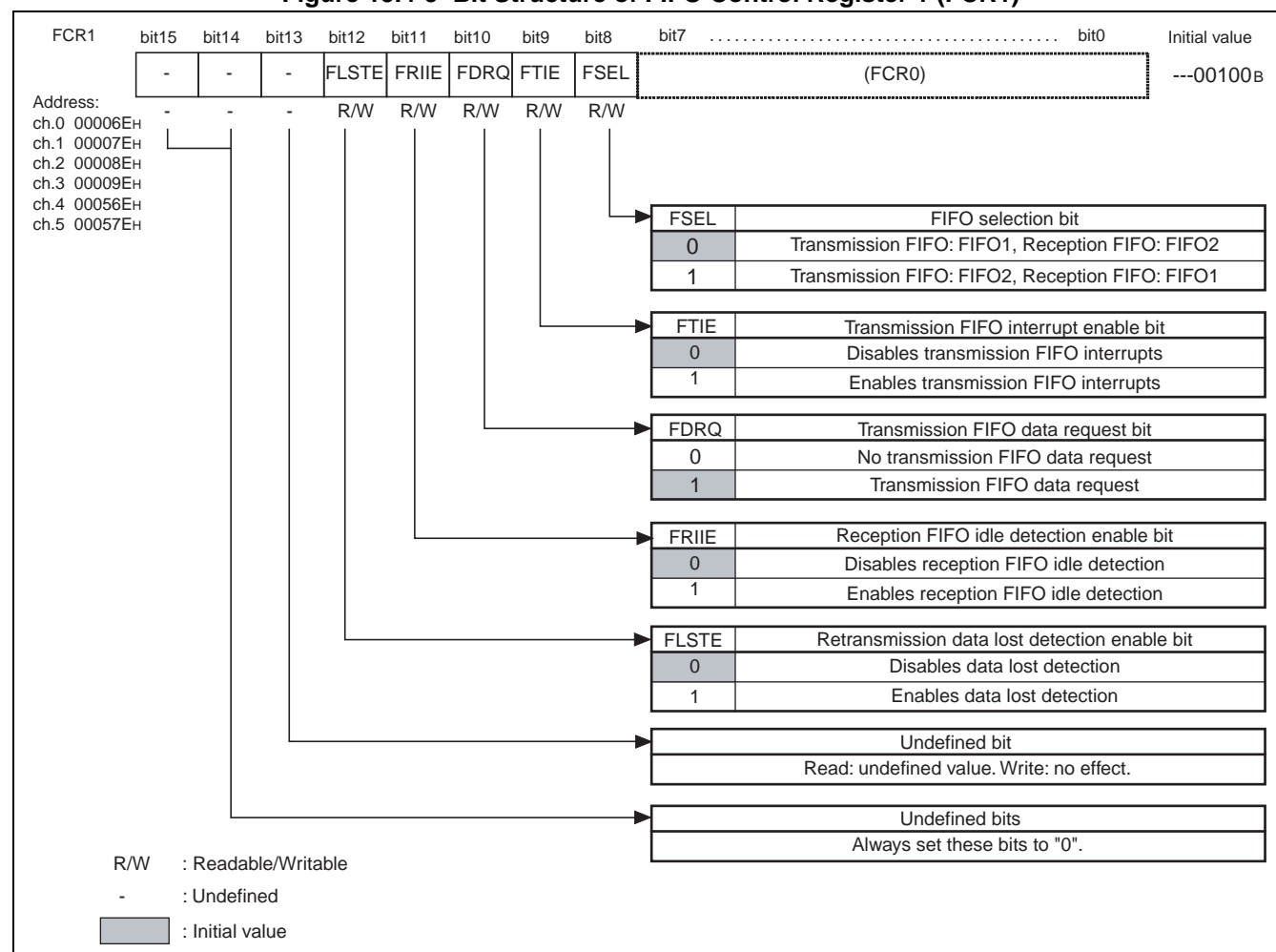


Table 15.4-7 Functional Description of Each Bit of FIFO Control Register 1 (FCR1) (1 / 2)

Bit name		Function
bit15, bit14	Undefined bits	Always set these bits to "0".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8-bit time or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) FDRQ reset condition <ul style="list-style-type: none"> • Writing "0" to this bit • When the transmission FIFO is full. Note: It is valid to write "0" when transmission FIFO has been enabled. It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".

Table 15.4-7 Functional Description of Each Bit of FIFO Control Register 1 (FCR1) (2 / 2)

Bit name		Function
bit8	FSEL: FIFO selection bit	<p>This bit is used to select transmission/reception FIFO.</p> <p>Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2.</p> <p>Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1.</p> <p>Note:</p> <p>This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1).</p> <p>To modify this bit, disable FIFO (FCR:FE2, FE1=0) and I²C interface (ISMK:EN=0) operations beforehand.</p>

Note:

There are two types of requests available for transmission interrupts: transmission FIFO interrupt requests; and transmission buffer interrupt requests.

15.4.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 15.4-10 shows the bit structure of the FIFO control register 0 (FCR0) and Table 15.4-8 describes the function of each bit.

Figure 15.4-10 Bit Structure of FIFO Control Register 0 (FCR0)

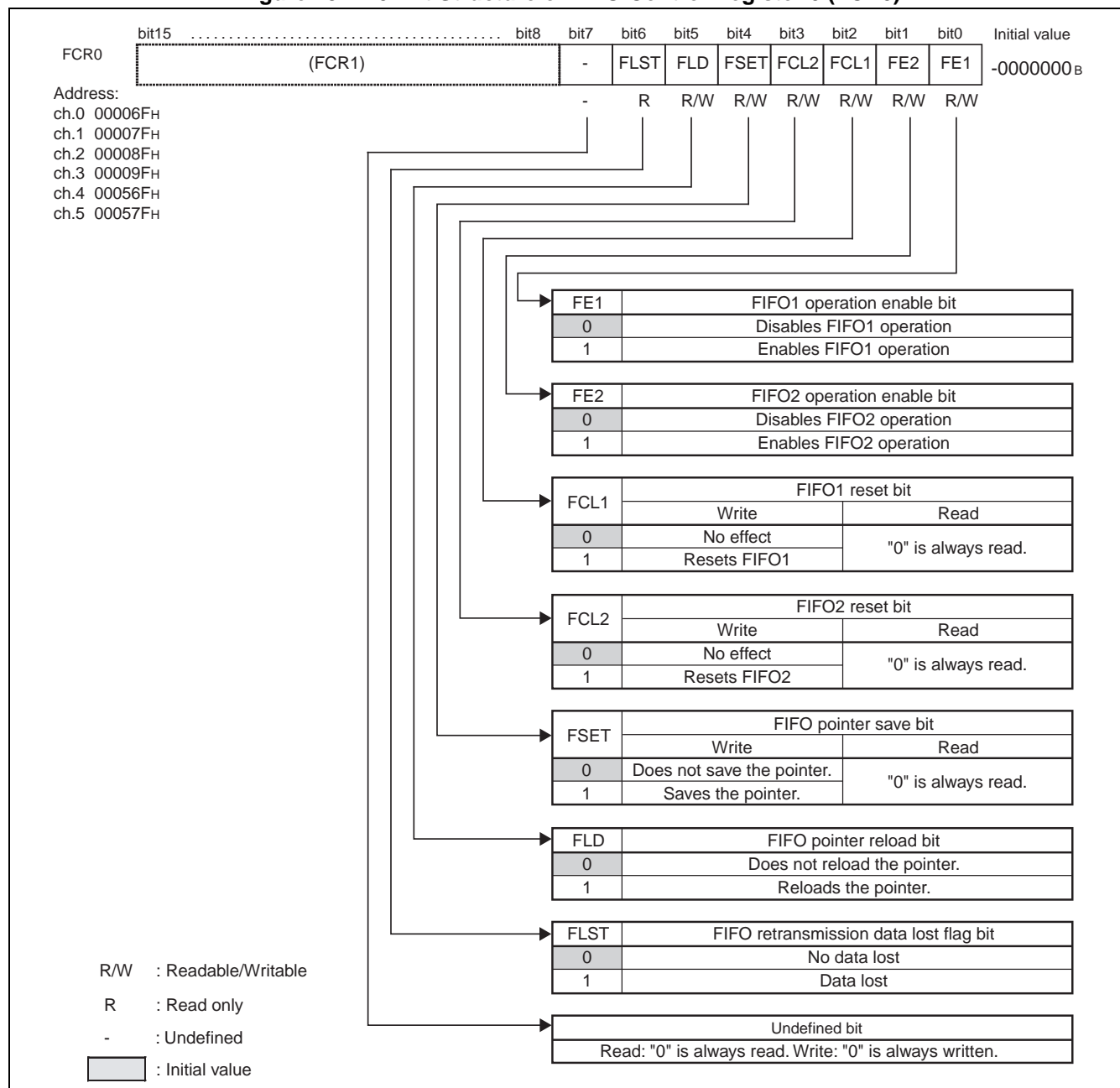


Table 15.4-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	<p>This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition Writing (overwriting) to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions</p> <ul style="list-style-type: none"> FIFO reset (writing "1" to FCL) Writing "1" to the FSET bit <p>Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.</p>
bit5	FLD: FIFO pointer reload bit	<p>This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. Note: Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.</p>
bit4	FSET: FIFO pointer save bit	<p>This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to communication will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. Note: Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".</p>
bit3	FCL2: FIFO2 reset bit	<p>This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR1:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO2. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".</p>

Table 15.4-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR1:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained.</p> <p>Note: Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".</p>
bit1	FE2: FIFO2 operation enable bit	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> To use FIFO2, set this bit to "1". When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> To use FIFO1, set this bit to "1". When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. Even when FIFO1 is disabled, its status is retained.

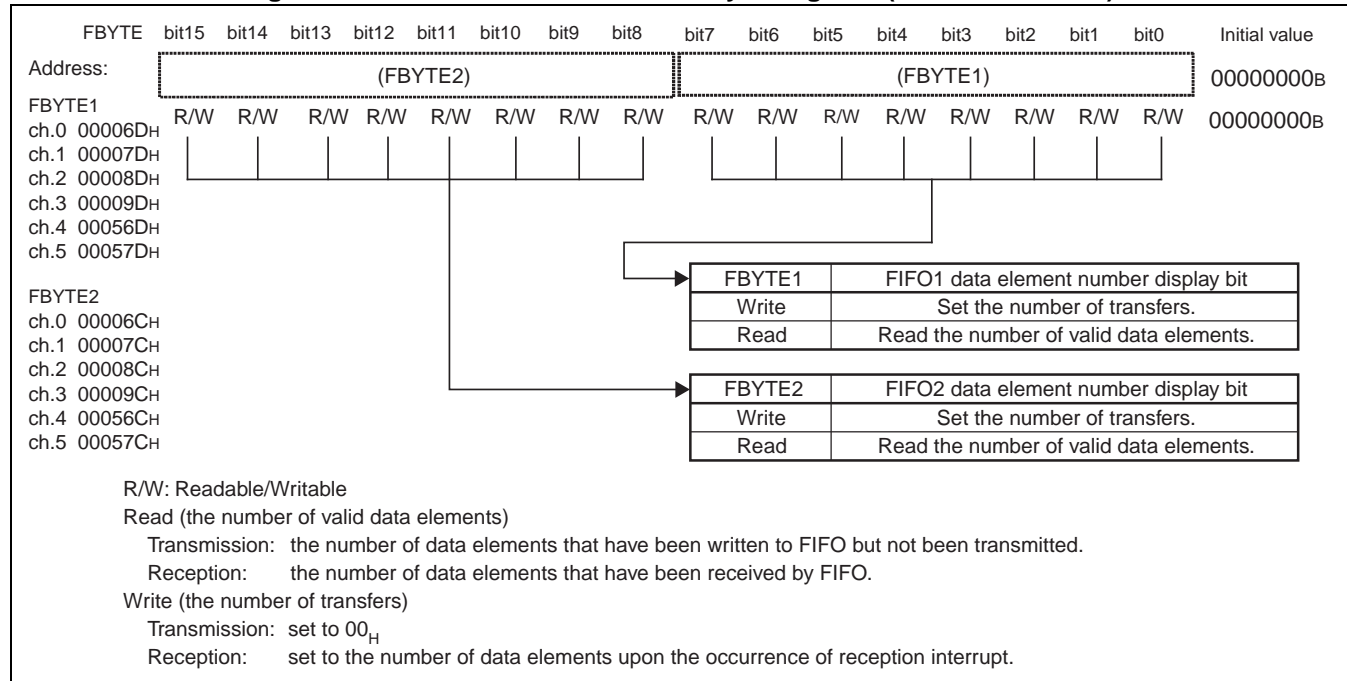
15.4.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. It can also determine whether a reception interrupt should occur when the reception FIFO has received a specified number of data elements.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 15.4-11 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 15.4-11 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)



The FBYTE1/FBYTE2 register indicates the number of valid data elements written to or received by FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 15.4-9 Displaying the Number of Data Elements

FSEL	FIFO selection	Number of data elements displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (SSR:RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR0/RDR1 is read while the eight clocks are still being counted, the counter will be reset

to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.

Notes:

- Set 00_H to the FBYTE1/FBYTE2 register of the transmission FIFO.
 - Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
 - Change this register after prohibiting receiving.
 - Read modify write (RMW) instructions cannot be used for this register.
 - Settings that will exceed the capacity of FIFO are prohibited.
-

15.5 Interrupts of UART

The UART has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR0/RDR1) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR0/TDR1) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

■ Interrupts of UART

Table 15.5-1 shows the interrupt control bits and interrupt sources of the UART.

Table 15.5-1 Interrupt Control Bits and Interrupt Sources of UART (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Reception	RDRF	SSR	○	○	Reception of 1 byte	SCR:RIE	Reading reception data (RDR0/RDR1)
					Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR0/RDR1) until reception FIFO becomes empty
					Detection of the idle state of reception for 8-bit timer or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	○	○	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	○	○	Framing error		
	PE	SSR	○	×	Parity error		

Table 15.5-1 Interrupt Control Bits and Interrupt Sources of UART (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Trans- mission	TDRE	SSR	○	○	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR0/TDR1), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	TBI	SSR	○	○	No transmission operation	SCR:TBIE	Writing to transmission data (TDR0/TDR1), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	FDRQ	FCR1	○	○	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

15.5.1 Occurrence of Reception Interrupts and Flag Set Timing

Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:PE, ORE, FRE).

■ Occurrence of Reception Interrupts and Flag Set Timing

Reception data is stored to the reception data register (RDR0/RDR1) when the first stop bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:PE, ORE, FRE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

Note:

If a reception error occurs, the data in the reception data register (RDR0/RDR1) will become invalid.

Figure 15.5-1 Timing for Setting RDRF (Reception Data Full) Flag Bit

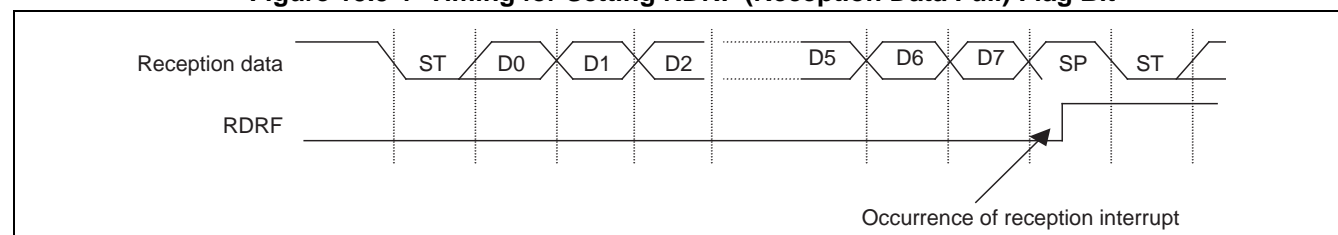


Figure 15.5-2 Timing for Setting FRE (Framing Error) Flag Bit

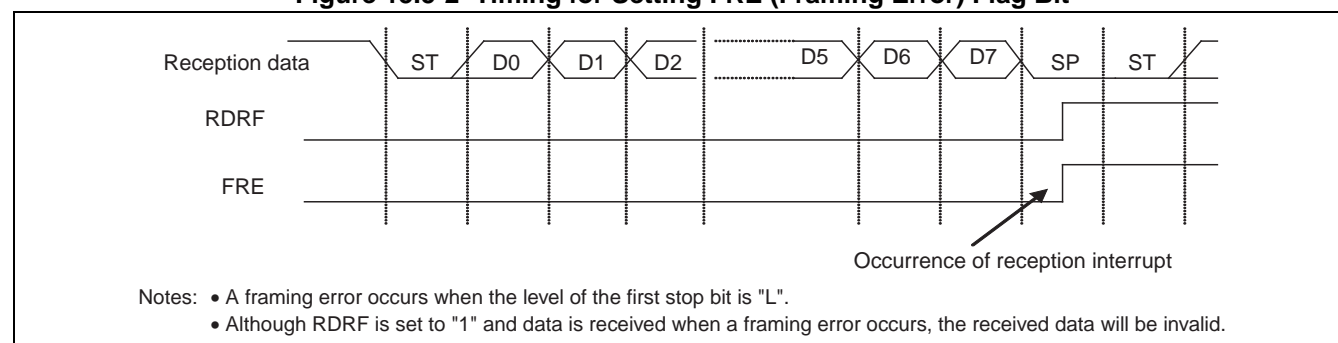
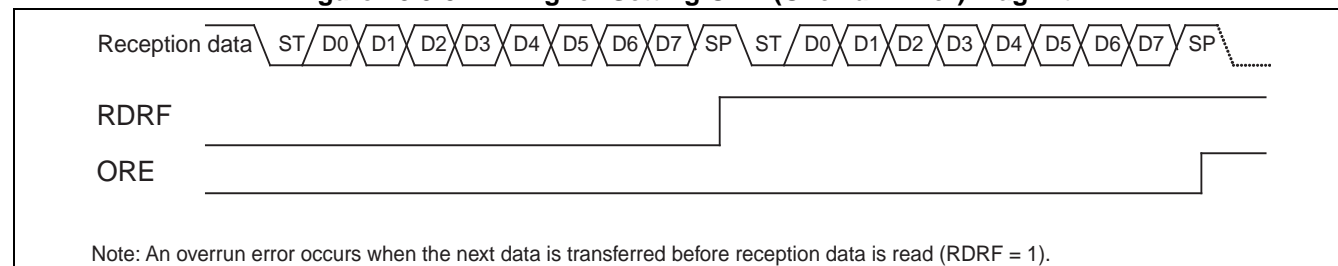


Figure 15.5-3 Timing for Setting ORE (Overrun Error) Flag Bit



MB91470/480 Series**15.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing**

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1 register (FBYTE1/FBYTE2) is received.

■ Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR0/RDR1 is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR0/RDR1) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 15.5-4 Timing for Generating Reception Interrupt when Reception FIFO is Used

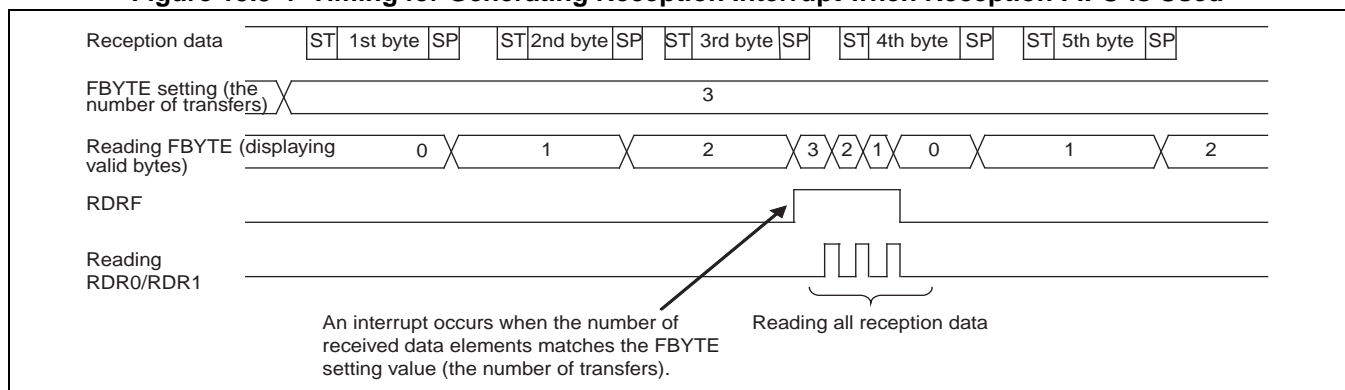
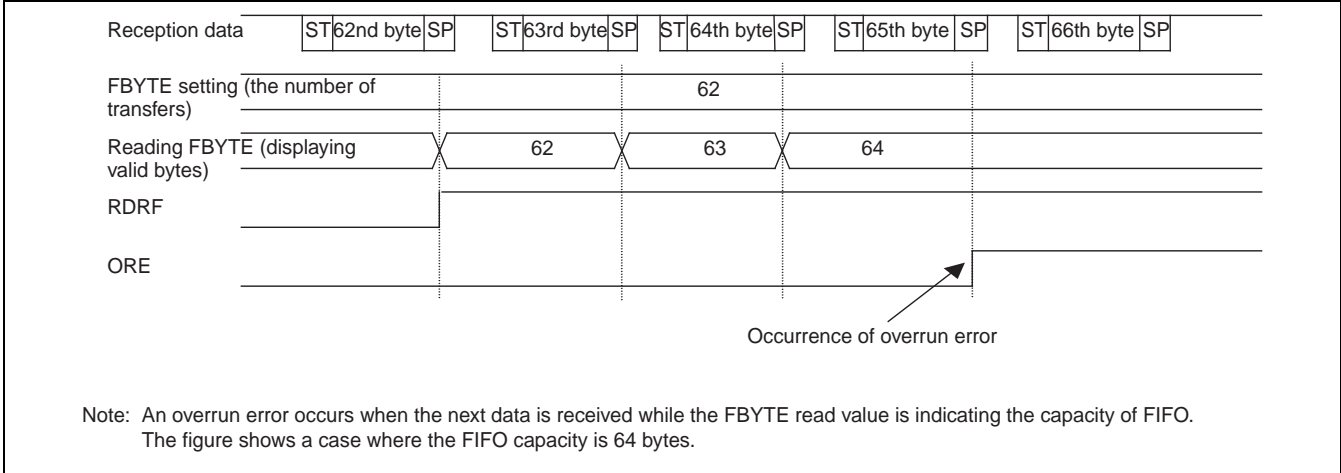


Figure 15.5-5 Timing for Setting ORE (Overrun Error) Flag Bit



MB91470/480 Series**15.5.3 Occurrence of Transmission Interrupts and Flag Set Timing**

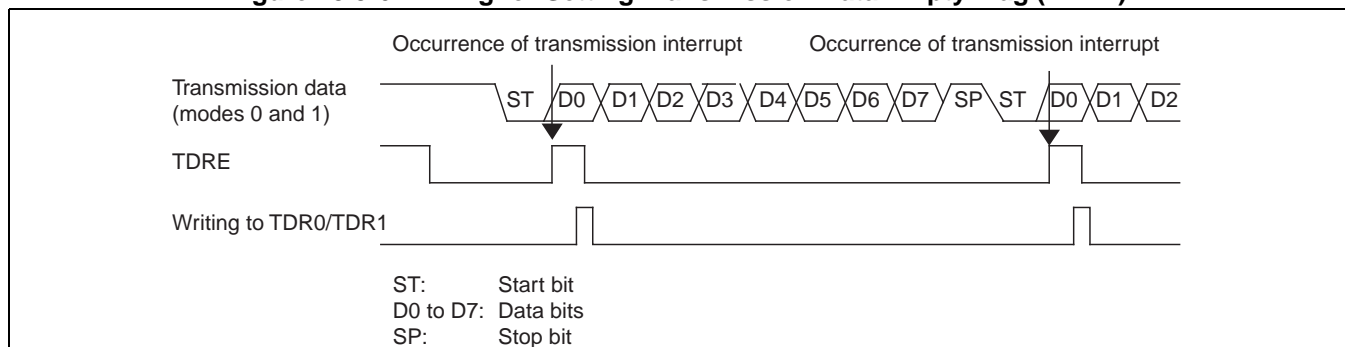
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR0/TDR1) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ Occurrence of Transmission Interrupts and Flag Set Timing

● Timing for setting the transmission data empty flag (TDRE)

It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR0/TDR1) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR0/TDR1).

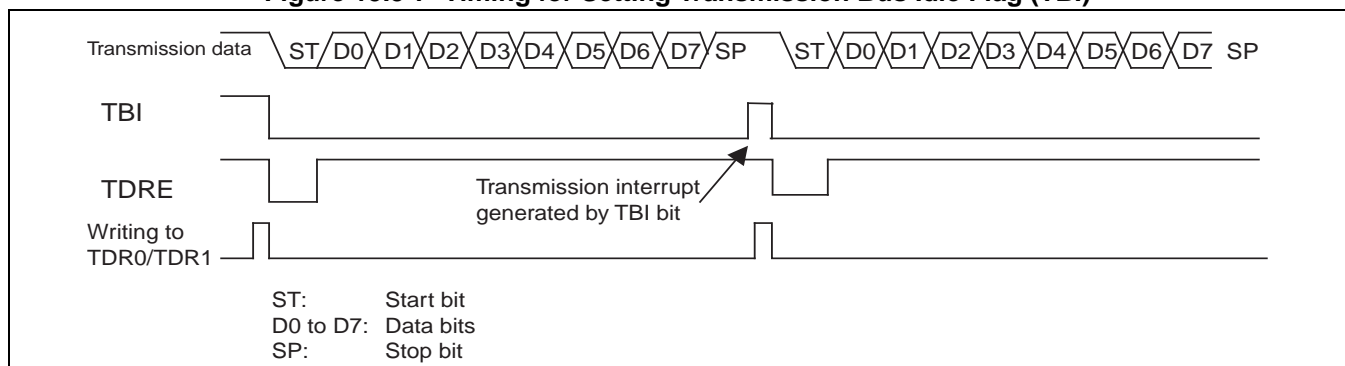
Figure 15.5-6 Timing for Setting Transmission Data Empty Flag (TDRE)



● Timing for setting the transmission bus idle flag (TBI)

The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR0/TDR1).

Figure 15.5-7 Timing for Setting Transmission Bus Idle Flag (TBI)



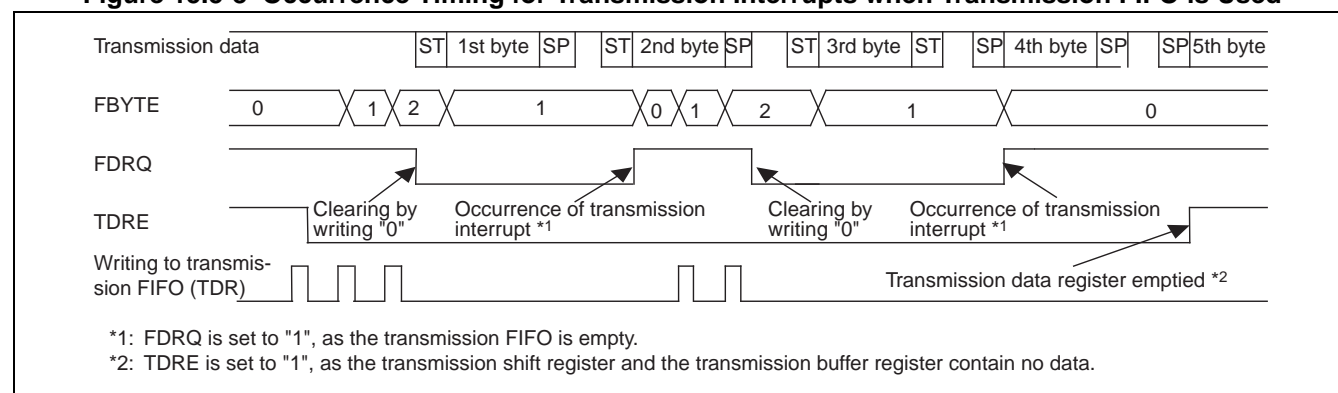
15.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

■ Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data.
At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data.
FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 15.5-8 Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



MB91470/480 Series**15.6 Operation of UART**

The UART operates in two-way serial asynchronous communications for mode 0 and in master/slave multi-processor communications for mode 1.

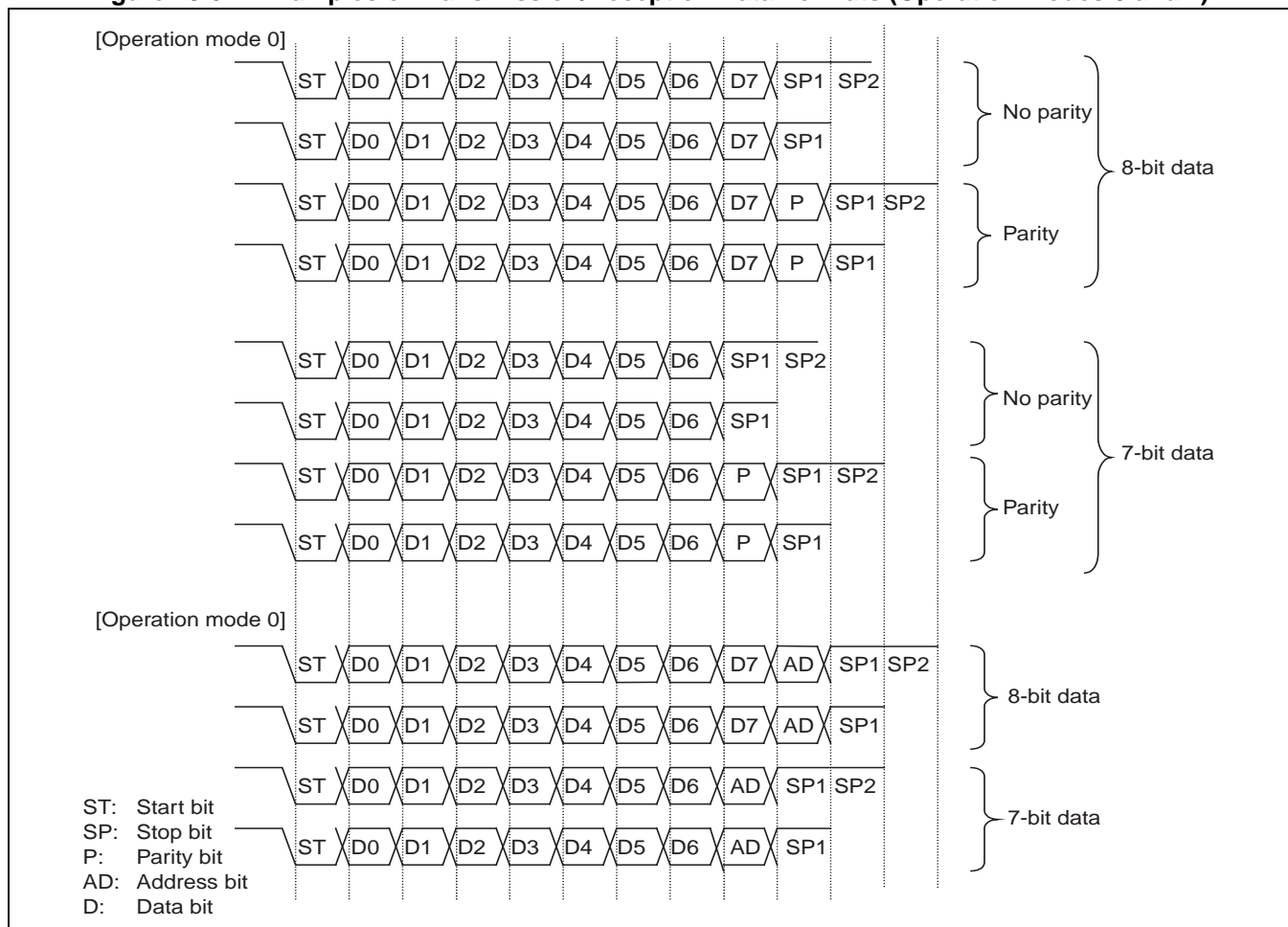
■ Operation of UART

● Transmission/reception data format

- Transmission and reception data is transmitted or received for a specified data bit length, always starting from the start bit and finishing with the stop bit (at least 1 bit).
- The data transfer direction (LSB or MSB first) is determined by the BDS bit in the serial mode register (SMR). When the addition of parity is selected, the parity bit is always placed between the last data bit and the first stop bit.
- The addition or omission of parity can be selected in operation mode 0 (normal mode).
- In operation mode 1 (multi-processor mode), the AD bit is added rather than parity.

Figure 15.6-1 shows transmission/reception data formats for operation modes 0 and 1.

Figure 15.6-1 Examples of Transmission/Reception Data Formats (Operation Modes 0 and 1)



Notes:

- Figure 15.6-1 shows cases where the data length is set to 7 or 8 bits. (The data length can be set to 5-9 bits for operation mode 0.)
 - When the BDS bit in the serial mode register (SMR) is set to "1" (MSB first), the bits are processed in the following order: D7, D6, D5...D1, D0 (P).
 - When the data length is set to X bits, the lower X bits of the transmission/reception data register (RDR0/RDR1/TDR0/TDR1) become valid.
-

● Transmission operation

- Transmission data can be written to the transmission data register (TDR0/TDR1) when the transmission data empty flag bit (TDRE) in the serial status register (SSR) is set to "1". (If the transmission FIFO is enabled, transmission data can be written even when TDRE is set to "0".)
 - Writing transmission data to the transmission data register (TDR0/TDR1) sets the transmission data empty flag bit (TDRE) to "0".
 - When the transmission operation enable bit in the serial control register (SCR:TXE) is set to "1", transmission data is loaded to the transmission shift register and the transmission starts from the start bit.
 - Once transmission starts, the transmission data empty flag bit (TDRE) is set back to "1". At this point, a transmission interrupt will occur if transmission interrupts have been enabled (SCR:TIE = 1). The next transmission data can be written to the transmission data register through interrupt processing.
-

Notes:

- The initial value of the transmission data empty flag bit (SSR:TDRE) is "1". Therefore, a transmission interrupt occurs immediately after transmission interrupts are enabled (SCR:TIE).
 - The initial value of the FIFO transmission data request bit (FCR1:FDRQ) is "1". Therefore, a transmission interrupt occurs immediately after FIFO transmission interrupts are enabled (FCR1:FTIE = 1).
-

● Reception operation

- Reception operation starts when such operation is enabled (SCR:RXE = 1).
- When a start bit is detected, one frame of data is received according to the data format set in the extended communication control register (ESCR:PEN, P, L2, L1, L0) and the serial mode register (SMR:BDS).
- Once one frame of data has been received, the reception data full flag bit (SSR:RDRF) is set to "1". At this point, a reception interrupt will occur if reception interrupts have been enabled (SCR:RIE = 1).
- Read reception data after one frame of data has been received, and then check the error flag status of the serial status register (SSR). If a reception error is occurring, the error must be treated.
- Reading reception data clears the reception data full flag bit (SSR:RDRF) to "0".
- When the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be set to "1", if the received data is equivalent of the number of frames specified in the reception FBYTE1/FBYTE2.
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR0/RDR1 is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- If the error flag in the serial status register (SSR) is set to "1" when the reception FIFO has been enabled, the data in which the error has occurred will not be stored to the reception FIFO. At the same time, the reception data full flag bit (SSR:RDRF) will not be set to "1". (In case of an overrun error, however, the RDRF flag will be set to "1".) The reception FBYTE1/FBYTE2 indicates the number of data elements that was successfully received before the error occurs. The reception FIFO will not be enabled unless the error flag in the serial status register (SSR) is cleared to "0".
- If the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be cleared to "0" when the reception FIFO no longer has data.

Note:

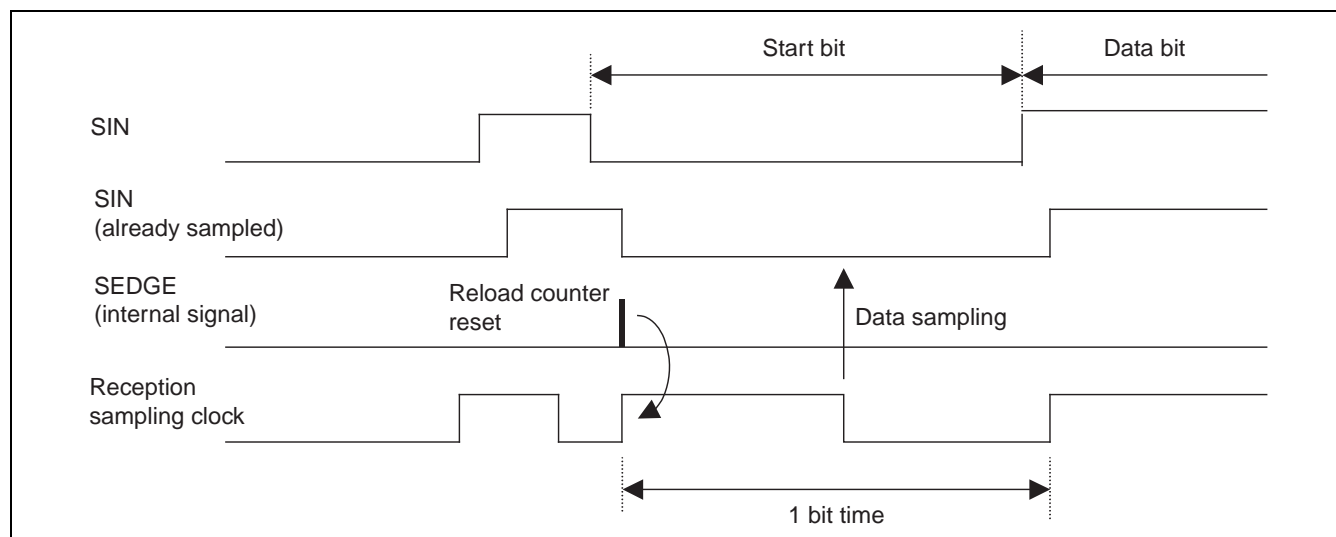
The data in the reception data register (RDR0/RDR1) will become valid, if no reception error occurs (SSR:PE, ORE, FRE = 0) when the reception data register full flag bit (SSR:RDRF) is set to "1".

● Clock selection

- The internal clock or external clock can be used.
- To use the external clock, set BGR:EXT to "1". In this case, the external clock is divided by the baud rate generator.

● Detection of the start bit

- In asynchronous mode, a start bit is identified by the falling edge of a SIN signal. Therefore, even when reception operation has been enabled (SCR:RXE = 1), such operation will not start unless the falling edge of a SIN signal is input.
- When the falling edge of a start bit is detected, the reception reload counter of the baud rate generator is reset and reloaded to start counting down. This allows sampling to be always performed using the central part of data.



● Stop bit

- 1 bit or 2 bits can be selected for the bit length.
- The reception data full flag bit (SSR:RDRF) is set to "1" when the first stop bit is detected.

● Detection of errors

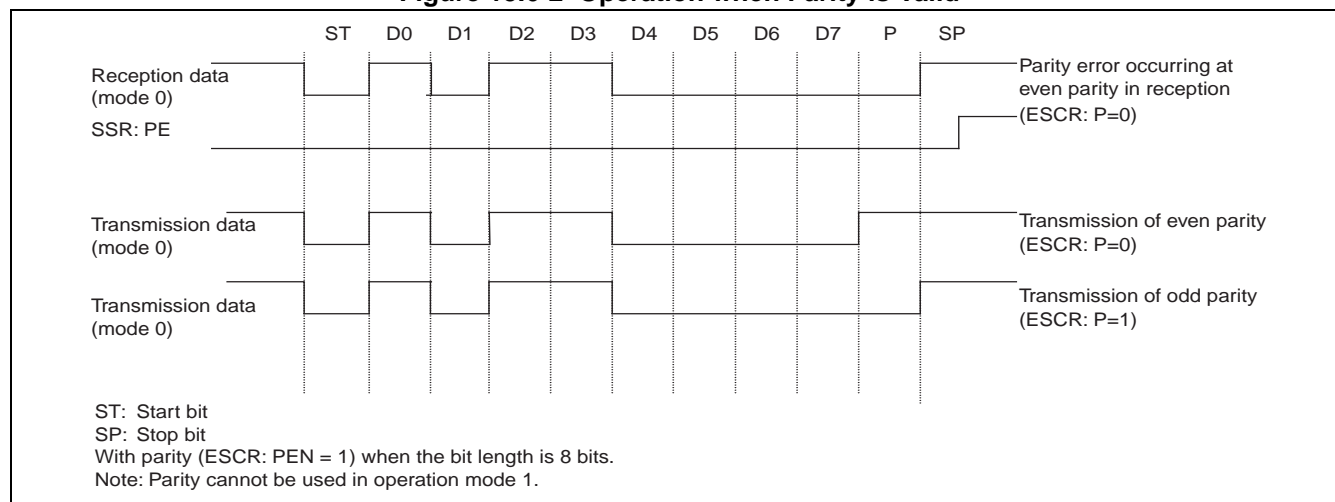
- In operation mode 0, parity errors, overrun errors and frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors, on the other hand, cannot be detected.

● Parity bit

- Addition of the parity bit can be selected only in operation mode 0. The parity enable bit (ESCR:PEN) can be used to determine the addition or omission of parity, while the parity selection bit (ESCR:P) can be used to select even parity or odd parity.
- Parity cannot be used in operation mode 1.

Figure 15.6-2 shows transmission/reception data when parity is valid.

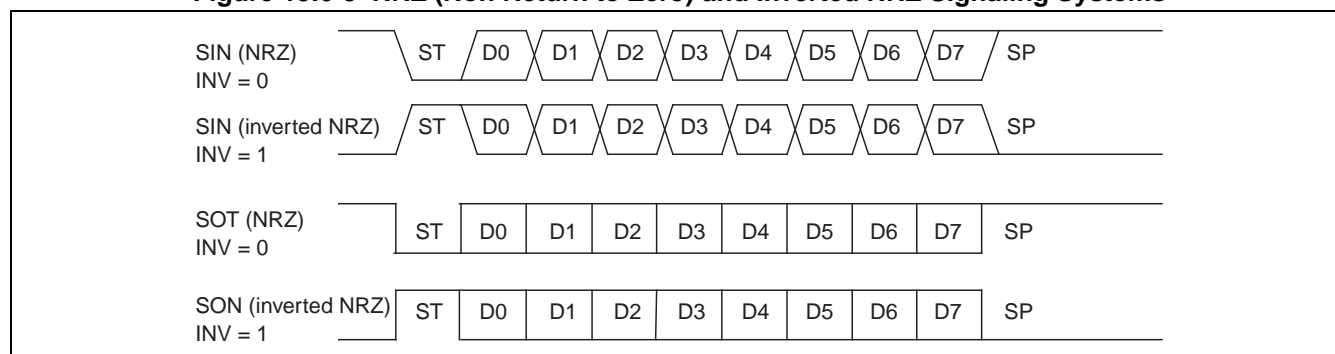
Figure 15.6-2 Operation when Parity is Valid



● Data signaling system

The NRZ (Non Return to Zero) signaling system (ESCR:INV = 0) or the inverted NRZ signaling system (ESCR:INV = 1) can be selected by setting the INV bit in the extended communication control register. Figure 15.6-3 shows the NRZ and inverted NRZ signaling systems.

Figure 15.6-3 NRZ (Non Return to Zero) and Inverted NRZ Signaling Systems



● Data transfer system

LSB-first or MSB-first data bit transfer system can be selected.

15.7 Dedicated Baud Rate Generator

One of the following options can be selected for the transmission/reception clock source of the UART.

- **Dedicated baud rate generator (reload counter)**
 - **External clock input to the baud rate generator (reload counter)**
-

■ UART Baud Rate Selection

One of the following two options can be selected for the baud rate.

- Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock, according to the set value.

To set the clock source, select the internal clock (SMR:EXT = 0).

- Baud rate achieved by dividing the external clock using the dedicated baud rate generator (reload counter)

The external clock is used as the clock source for the reload counter.

The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the external clock, according to the set value.

To set the clock source, select the external clock and the baud rate generator clock (SMR:EXT = 1).

This mode is made available on the assumption that an oscillator with a special frequency is divided for use.

Notes:

- Set the external clock (EXT = 1) while the reload counter is stopped (BGR1/BGR0 = 15'h00).
 - When the external clock has been selected (EXT = 1), the "H" and "L" widths of the external clock must be two or more peripheral clocks (CLKP).
-

15.7.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value

b: Baud rate

ϕ : Peripheral clock (CLKP), external clock frequency

(2) Example of calculation

If the peripheral clock (CLKP) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:

Reload value:

$$V = (16 \times 1000000) / 19200 - 1 = 832$$

As a result, the baud rate is:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

(3) Baud rate error

The following formula is used to calculate a baud rate error.

$$\text{Error}(\%) = (\text{calculated value} - \text{target value}) / \text{target value} \times 100$$

Example: peripheral clock (CLKP) = 20MHz, target baud rate = 153600bps

$$\text{Reload value} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{Baud rate (calculated value)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error}(\%) = (153846 - 153600) / 153600 \times 100 = 0.16 \text{ (\%)}$$

Notes:

- The reload counter halts when the reload value is set to "0".
 - When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (CLKP) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
 - Select 4 or a larger value for the reload value. However, data may not be able to be received properly, due to a baud rate error or reload settings.
-

■ **Reload Values and Baud Rates for Different Peripheral Clock (CLKP) Frequencies**

Table 15.7-1 Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	0	-	-	-	-	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

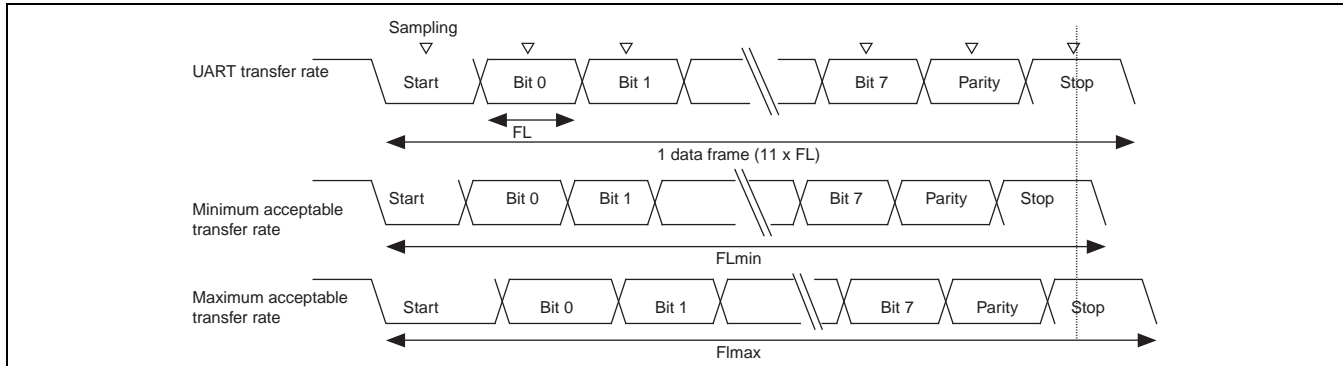
- Value: the value set in BGR1/BGR0 registers (decimal)
- ERR: baud rate error (%)

■ Acceptable Baud Rate Range for Reception

The following figure shows the range of acceptable baud rate differences at the transmission destination during reception.

The following calculation formula must be used to set a baud rate error for reception within the acceptable error range.

Figure 15.7-1 Acceptable Baud Rate Range for Reception



As shown in the figure, the sampling timing for reception data is determined by the counter selected by the BGR1/BGR0 registers after a start bit is detected. If all data including the last data (stop bit) can fit in this sampling timing, reception can be performed successfully.

In theory, the following is expected when this is applied to 11-bit reception.

When the sampling timing margin is equivalent of two clocks of the peripheral clock (CLKP) (ϕ), the minimum acceptable transfer rate (FLmin) is as follows:

$$FL_{min} = (11 \text{ bits} \times (V + 1) - (V + 1)/2 + 2)/\phi = (21V + 25)/2\phi \text{ (s)}$$

V: reload value ϕ : peripheral clock (CLKP)

Consequently, the maximum receivable baud rate at the transmission destination (BGmax) is as follows:

$$BG_{max} = 11/FL_{min} = 22\phi/(21V+25) \text{ (bps)}$$

V: reload value ϕ : peripheral clock (CLKP)

Likewise, the maximum acceptable transfer rate (FLmax) can be calculated as shown below:

$$FL_{max} = (11 \text{ bits} \times (V + 1) + (V + 1)/2 - 2)/\phi = (23V + 19)/2\phi \text{ (s)}$$

V: reload value ϕ : peripheral clock (CLKP)

Consequently, the minimum receivable baud rate at the transmission destination (BGmin) is as follows:

$$BG_{min} = 11/FL_{max} = 22\phi/(23V+19) \text{ (bps)}$$

V: reload value ϕ : peripheral clock (CLKP)

Based on the aforementioned calculation formulas for the minimum/maximum baud rates, the acceptable baud rate error between the UART and transmission destination can be calculated as shown below.

Table 15.7-2 Acceptable Baud Rate Error

Reload value (V)	Maximum acceptable baud rate error	Minimum acceptable baud rate error
3	0%	0
10	+2.98%	-2.81%
50	+4.37%	-4.02%
100	+4.56%	-4.18%
200	+4.66%	-4.26%
32767	+4.76%	-4.35%

Note:

The accuracy of reception depends on the number of bits per frame, the peripheral clock (CLKP) and the reload value. The accuracy becomes higher as the peripheral clock (CLKP) and the division ratio become higher.

■ External Clock

The baud rate generator divides the external clock, when "1" is written to the EXT bit in the baud rate generator register 1, 0 (BGR1, BGR0).

Note:

The UART synchronizes external clock signals with the internal clock. Therefore, the operation becomes unstable when an external clock which cannot be synchronized is used.

■ Functions of Reload Counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the external or internal clock.

■ Starting a Count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

■ Restart

The reload counter restarts under the following conditions.

- For both transmission and reception reload counters
 - Programmable reset (SCR:UPCL bit)
- For reception reload counter
 - Detecting the falling edge of a start bit in asynchronous mode

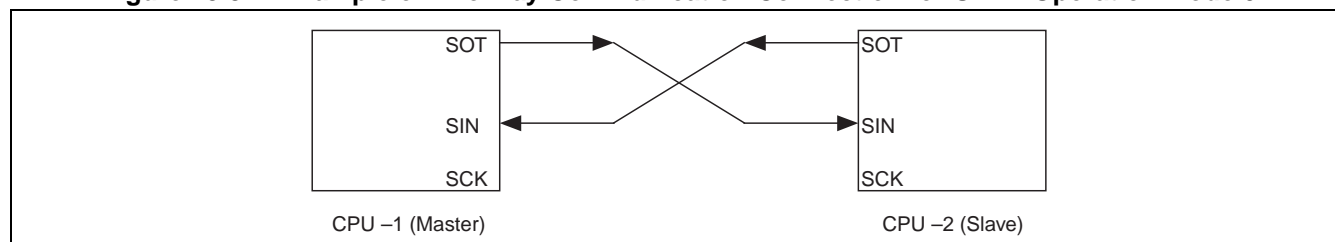
15.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

Asynchronous serial two-way communication is enabled in operation mode 0.

■ Connection between CPUs

Two-way communication should be selected for operation mode 0 (normal mode). Two CPUs are connected to each other, as shown in Figure 15.8-1.

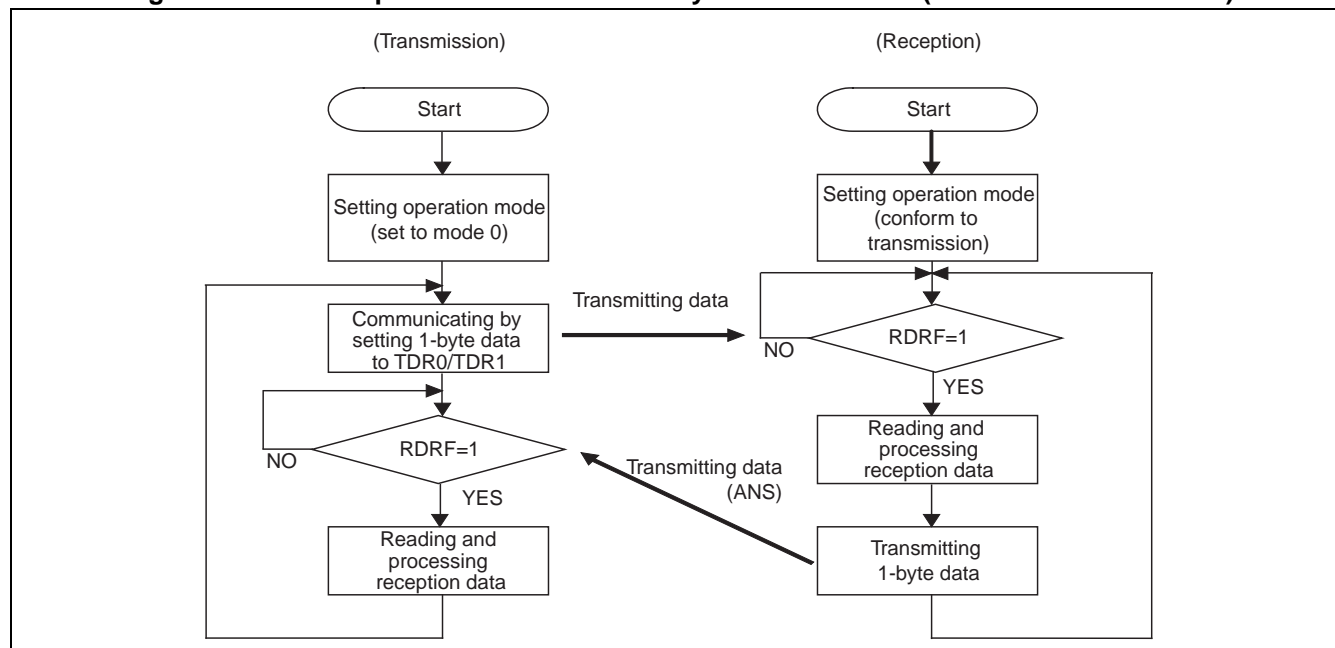
Figure 15.8-1 Example of Two-way Communication Connection for UART Operation Mode 0



■ Flowchart

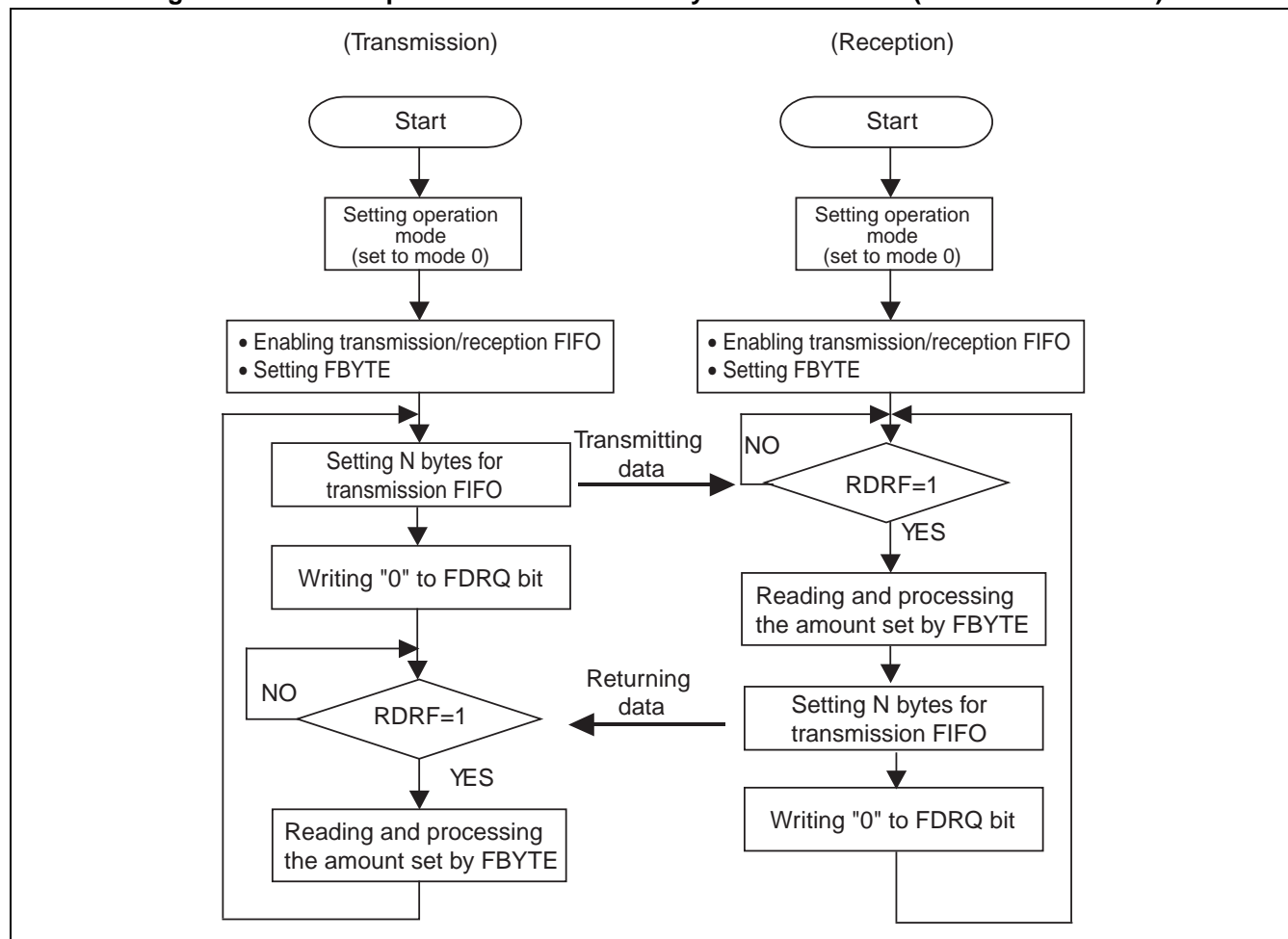
- When FIFO is not used

Figure 15.8-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



- When FIFO is used

Figure 15.8-3 Example Flowchart for Two-way Communication (When FIFO is Used)



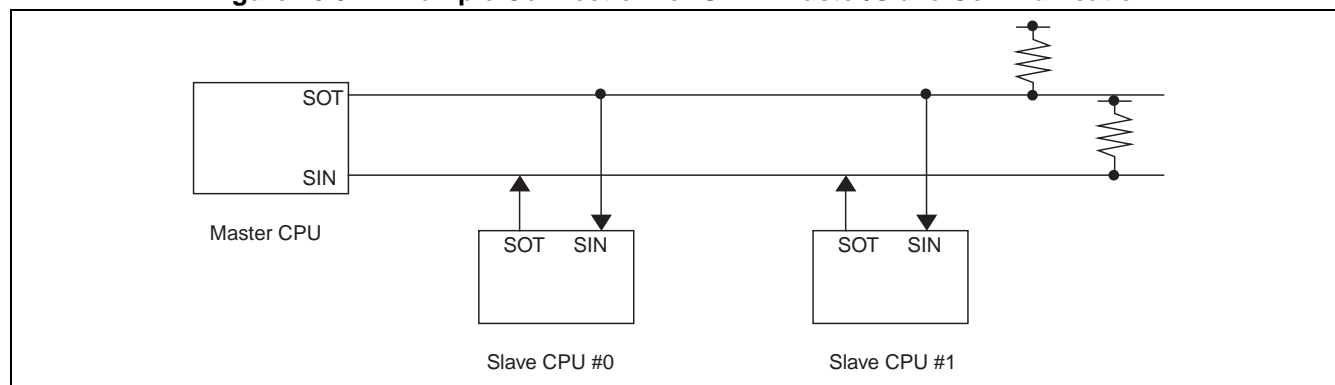
15.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

In operation mode 1 (multi-processor mode), communication among multiple CPUs is enabled through master/slave connection. The connected CPUs can be used as a master/slave.

■ Connection among CPUs

In this master/slave communication, one master CPU and more than one slave CPU are connected to two common communication lines, as shown in Figure 15.9-1, to configure a communication system. The UART can be used by both the master and slaves.

Figure 15.9-1 Example Connection for UART Master/Slave Communication



■ Function Selection

For master/slave communication, select the operation mode and data transfer system shown in Table 15.9-1.

Table 15.9-1 Selection of Master/Slave Communication Function

	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission/reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = 1 + 7-bit or 8-bit address	None	1 bit or 2 bits	LSB first or MSB first
Data transmission/reception			AD = 0 + 7-bit or 8-bit data			

Note:

Use half word access for transmission/reception data (TDR/RDR) in operation mode 1.

● Communication procedure

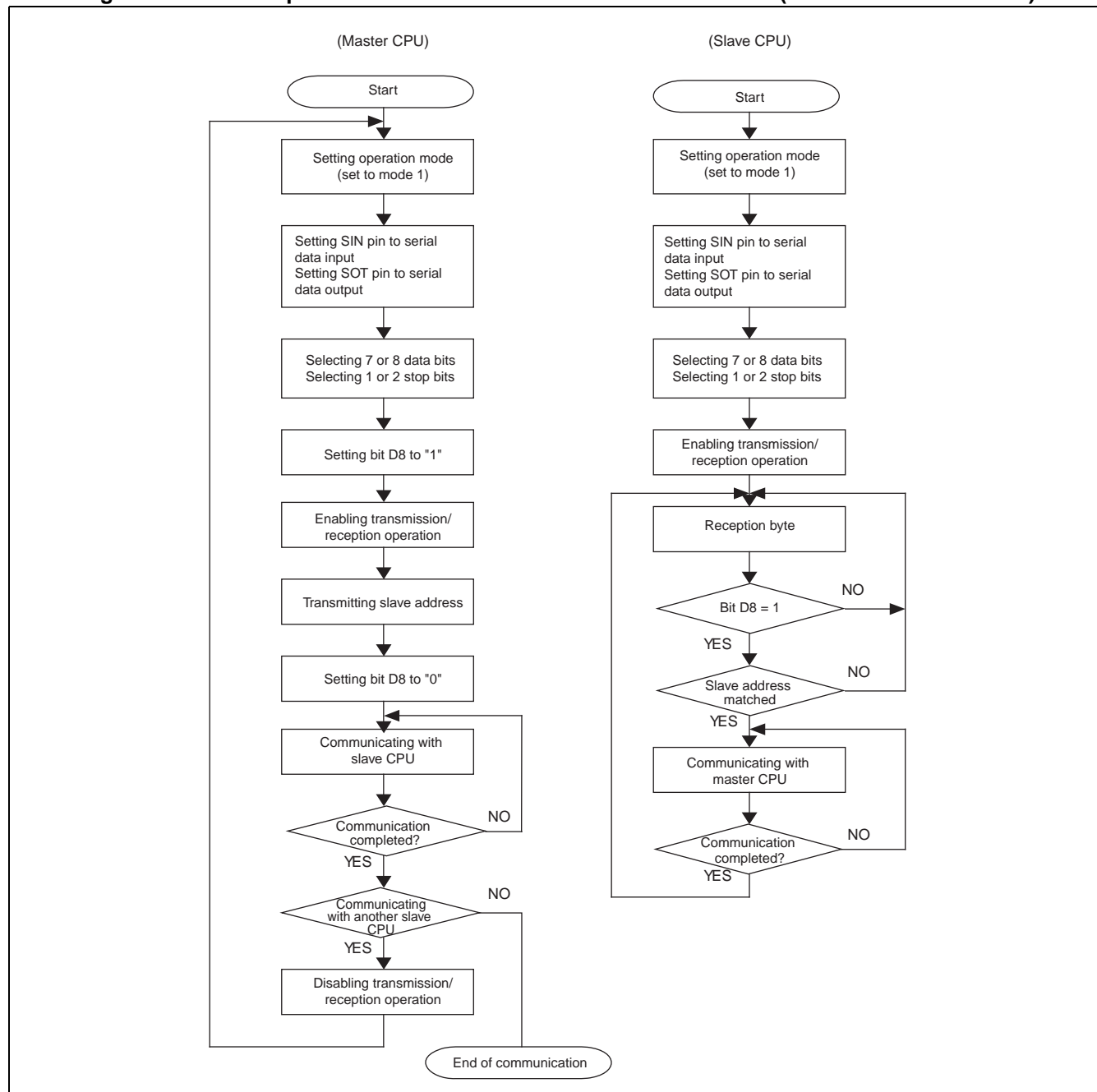
Communication is started when the master CPU transmits address data, where bit D8 is treated as "1". This data is used to select a slave CPU which will be the communication destination. Each slave CPU judges the address data on a program, and communicates (normal data) with the master CPU when the data matches its assigned address.

Figure 15.9-2 and Figure 15.9-3 show flowcharts for the master/slave communication (multi-processor mode).

■ Flowcharts

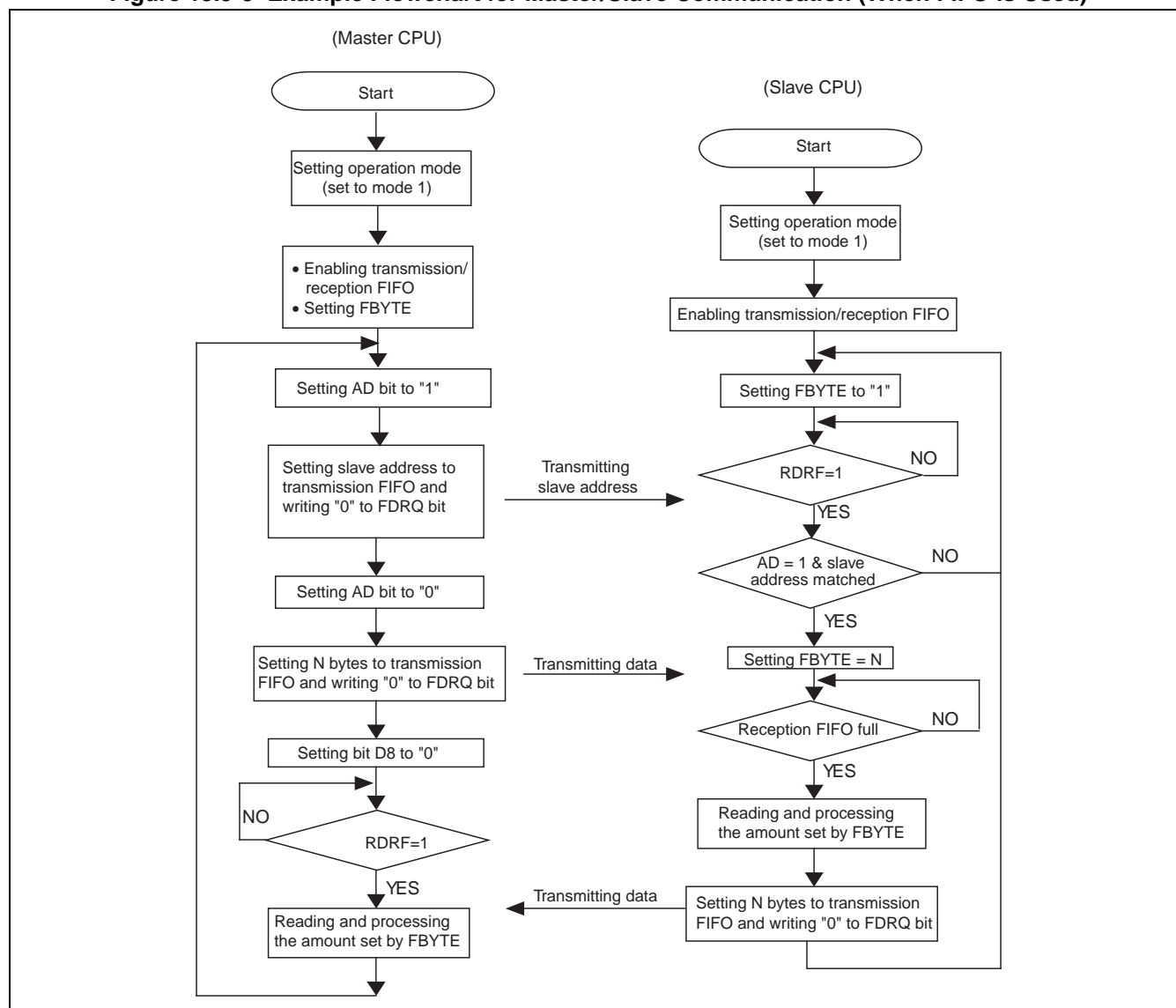
- When FIFO is not used:

Figure 15.9-2 Example Flowchart for Master/Slave Communication (When FIFO is Not Used)



● When FIFO is used

Figure 15.9-3 Example Flowchart for Master/Slave Communication (When FIFO is Used)



15.10 Notes on UART Mode

The notes for when you use the UART mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.

15.11 CSIO (Clock Synchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes the CSIO functions that are supported in operation mode 2.

- CSIO (Clock Synchronous Serial Interface)
- Overview of CSIO (Clock Synchronous Serial Interface)
- Registers of CSIO (Clock Synchronous Serial Interface)
 - Serial Control Register (SCR)
 - Serial Mode Register (SMR)
 - Serial Status Register (SSR)
 - Extended Communication Control Register (ESCR)
 - Reception Data Register / Transmission Data Register (RDR/TDR)
 - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
 - FIFO Control Register 1 (FCR1)
 - FIFO Control Register 0 (FCR0)
 - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of CSIO (Clock Synchronous Serial Interface)
 - Occurrence of Reception Interrupts and Flag Set Timing
 - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
 - Occurrence of Transmission Interrupts and Flag Set Timing
 - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of CSIO (Clock Synchronous Serial Interface)
- Dedicated Baud Rate Generator
 - Setting Baud Rate
- Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

15.12 Overview of CSIO (Clock Synchronous Serial Interface)

CSIO (Clock Synchronous Serial Interface) is a general-purpose interface for serial data communication, which allows synchronous communications with external units (SPI supported). In addition, this interface comes with transmission/reception FIFO (up to 16 bytes each)

■ Functions of CSIO (Clock Synchronous Serial Interface)

		Function
1	Data buffer	<ul style="list-style-type: none"> Full-duplex double buffer (when FIFO is not used) Transmission/reception FIFO (up to 16 bytes each) (when FIFO is used)
2	Transfer system	<ul style="list-style-type: none"> Clock synchronization (no start bit / no stop bit) Master/slave function SPI supported (both master & slaves supported)
3	Baud rate	<ul style="list-style-type: none"> Dedicated baud rate generator available (15-bit reload counter configuration, in master operation) External clock can be input (in slave operation)
4	Data length	Variable from 5 bits to 9 bits
5	Reception error detection	Overrun error
6	Interrupt request	<ul style="list-style-type: none"> Reception interrupt (completion of reception, overrun error) Transmission interrupt (transmission data empty, transmission bus idle) Transmission FIFO interrupt (when transmission FIFO is empty) Extended intelligent I/O service (EI²OS) and DMA transfer support function are available for both transmission and reception.
7	Synchronous mode	Master or slave function
8	Pin access	Serial data output pin can be set to "1".
9	FIFO options	<ul style="list-style-type: none"> Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes) Transmission FIFO or reception FIFO selectable Transmission data can be resent. The interrupt timing for reception FIFO can be modified by software. FIFO reset is supported separately.

15.13 Registers of CSIO (Clock Synchronous Serial Interface)

This section lists the registers of CSIO (clock synchronous serial interface).

■ List of Registers of CSIO (Clock Synchronous Serial Interface)

Figure 15.13-1 List of Registers of CSIO (Clock Synchronous Serial Interface)

Address		bit15	bit8 bit7	bit0
CSIO	000062 _H 000063 _H 000072 _H 000073 _H 000082 _H 000083 _H 000092 _H 000093 _H 000562 _H 000563 _H 000572 _H 000573 _H	SCR (serial control register)		SMR (serial mode register)
	000060 _H 000061 _H 000070 _H 000071 _H 000080 _H 000081 _H 000090 _H 000091 _H 000560 _H 000561 _H 000570 _H 000571 _H	SMR (serial status register)		ESCR (extended communication control register)
	000066 _H 000067 _H 000076 _H 000077 _H 000086 _H 000087 _H 000096 _H 000097 _H 000566 _H 000567 _H 000576 _H 000577 _H	RDR/TDR (transmission/reception data register)		
	000064 _H 000065 _H 000074 _H 000075 _H 000084 _H 000085 _H 000094 _H 000095 _H 000564 _H 000565 _H 000574 _H 000575 _H	BGR1 (baud rate generator register 1)		BGR0 (baud rate generator register 0)
	000068 _H 000069 _H 000078 _H 000079 _H 000088 _H 000089 _H 000098 _H 000099 _H 000568 _H 000569 _H 000578 _H 000579 _H	-		-

(Continued)

(Continued)

Address		bit15	bit8 bit7	bit0
FIFO	00006E _H 00006F _H	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)	
	00007E _H 00007F _H			
	00008E _H 00008F _H			
	00009E _H 00009F _H			
	00056E _H 00056F _H			
	00057E _H 00057F _H			
	00006C _H 00006D _H	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)	
	00007C _H 00007D _H			
	00008C _H 00008D _H			
	00009C _H 00009D _H			
	00056C _H 00056D _H			
	00057C _H 00057D _H			

Table 15.13-1 Bit Assignment of CSIO (Clock Synchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

15.13.1 Serial Control Register (SCR)

The serial control register (SCR) enables/disables transmission/reception interrupts, transmission idle interrupts and transmission/reception operations. This register can also set SPI connection and reset CSIO.

Serial Control Register (SCR)

Figure 15.13-2 shows the bit structure of the serial control register (SCR), and Table 15.13-2 describes the function of each bit.

Figure 15.13-2 Bit Structure of Serial Control Register (SCR)

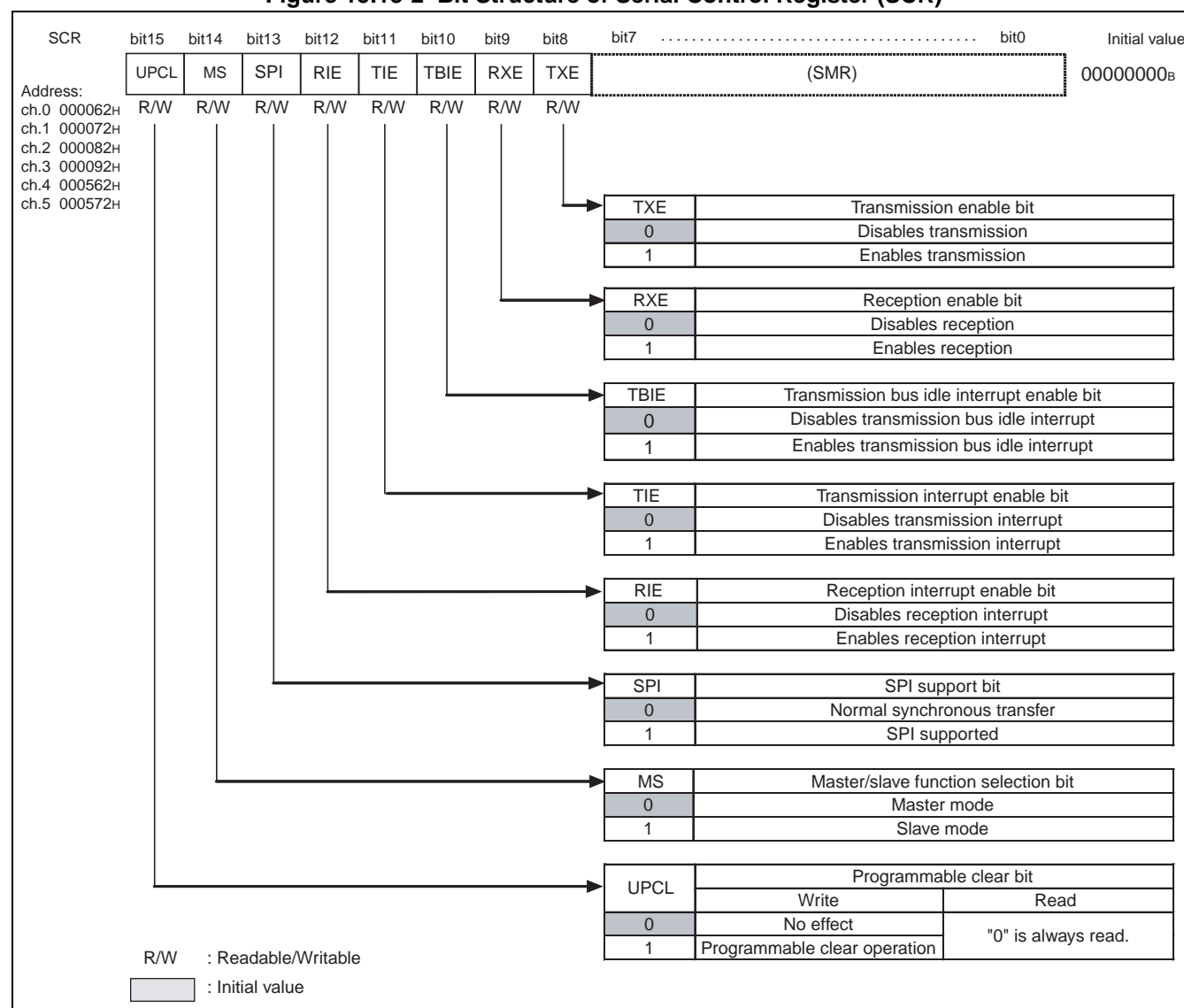


Table 15.13-2 Functional Description of Each Bit of Serial Control Register (SCR) (1 / 2)

Bit name		Function
bit15	UPCL: Programmable clear bit	<p>This bit is used to initialize the internal state of the CSIO. Setting the bit to "1":</p> <ul style="list-style-type: none"> The CSIO will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. All the transmission/reception interrupt sources (TDRE, TBI, RDRF and ORE) will be initialized. Setting the bit to "0": No effect on the operation Reading this bit always returns "0". <p>Note: Execute the programmable clear operation after disabling interrupts. Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.</p>
bit14	MS: Master/slave function selection bit	<p>This bit is used to select master or slave mode. Setting the bit to "0" selects master mode. Setting the bit to "1" selects slave mode.</p> <p>Note: The external clock will be input directly, if SMR:SCKE is set to "0" when slave mode is selected.</p>
bit13	SPI: SPI support bit	<p>This bit is used to enable communication supporting SPI. Setting the bit to "0" enables normal synchronous communication. Setting the bit to "1" enables SPI support.</p>
bit12	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of reception interrupt requests to the CPU. A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (ORE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of transmission interrupt requests to the CPU. A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	<p>This bit is used to enable/disable CSIO reception operation. Setting the bit to "0" disables data frame reception operation. Setting the bit to "1" enables data frame reception operation.</p> <p>Note: If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.</p>

Table 15.13-2 Functional Description of Each Bit of Serial Control Register (SCR) (2 / 2)

Bit name		Function
bit8	TXE: Transmission enable bit	<p>This bit is used to enable/disable CSIO transmission operation. Setting the bit to "0" disables data frame transmission operation. Setting the bit to "1" enables data frame transmission operation.</p> <p>Note: If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.</p>

15.13.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction, data length and serial clock inversion, and enables or disables the output to the serial data and serial clock pins.

■ Serial Mode Register (SMR)

Figure 15.13-3 shows the bit structure of the serial mode register (SMR), and Table 15.13-3 describes the function of each bit.

Figure 15.13-3 Bit Structure of Serial Mode Register (SMR)

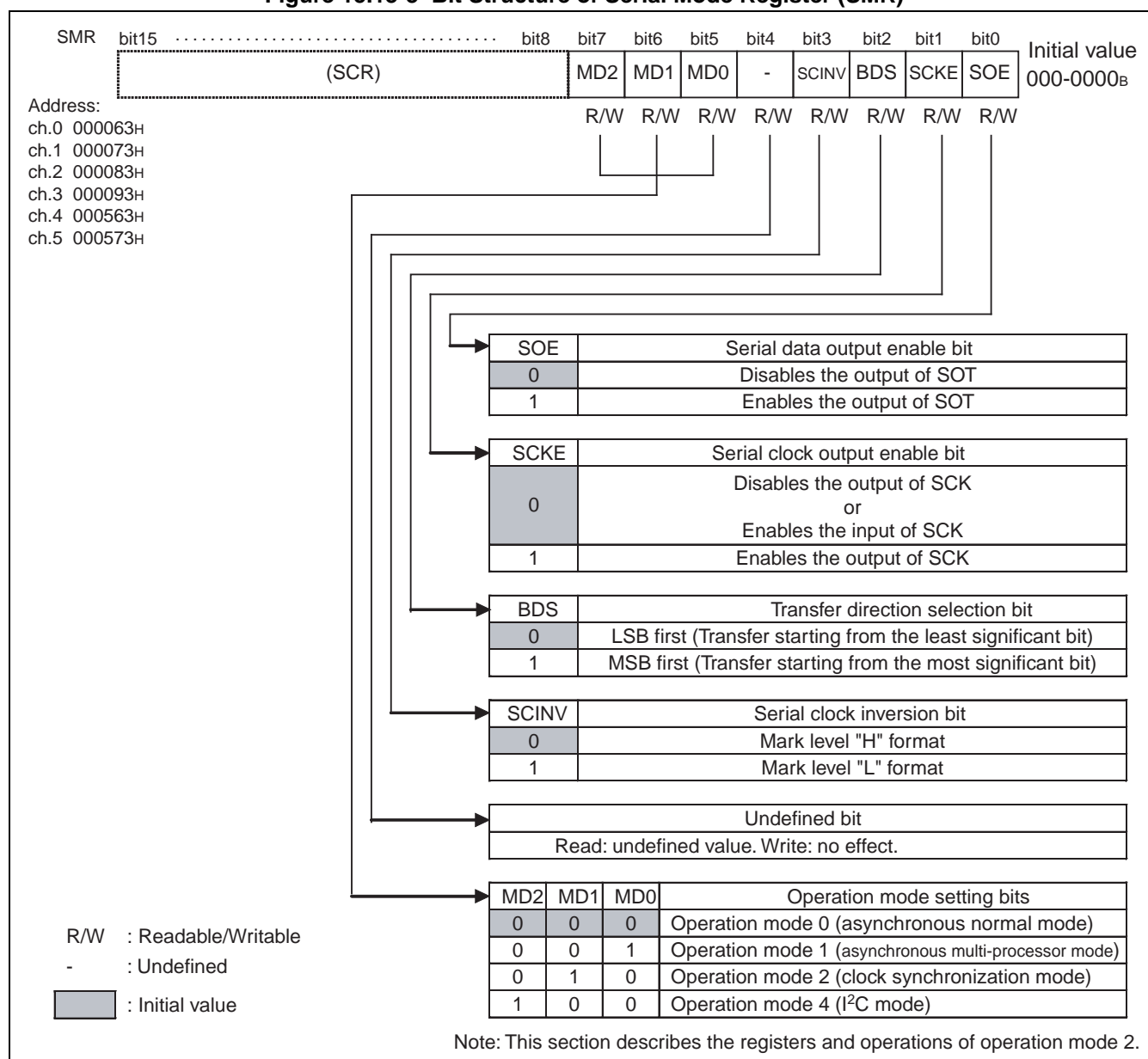


Table 15.13-3 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	<p>These bits are used to select the operation mode.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 2 (clock synchronization mode).</p> <p>Note:</p> <ul style="list-style-type: none"> Settings other than above are prohibited. To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.
bit4	Undefined bit	<p>Read: undefined value</p> <p>Write: no effect</p>
bit3	SCINV: Serial clock inversion bit	<p>This bit is used to invert the serial clock format.</p> <p>Setting the bit to "0":</p> <ul style="list-style-type: none"> Changes the mark level of the serial clock output to "H". Transmission data is output, being synchronized with the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. Reception data is sampled at the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. <p>Setting the bit to "1":</p> <ul style="list-style-type: none"> Changes the mark level of the serial clock output to "L". Transmission data is output, being synchronized with the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. Reception data is sampled at the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. <p>Note:</p> <ul style="list-style-type: none"> Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit2	BDS: Transfer direction selection bit	<p>This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1).</p> <p>Note:</p> <ul style="list-style-type: none"> Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit1	SCKE: Serial clock output enable bit	<p>This bit is used to control the I/O port of the serial clock.</p> <p>Setting the bit to "0":</p> <p>The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port.</p> <p>Setting the bit to "1" enables the output of SCK.</p>
bit0	SOE: Serial data output enable bit	<p>This bit is used to enable/disable the output of serial data.</p> <p>Setting the bit to "0" enables the output of the "H" level of SOT.</p> <p>Setting the bit to "1" enables the output of SOT.</p>

Note:

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

15.13.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

Serial Status Register (SSR)

Figure 15.13-4 shows the bit structure of the serial status register (SSR) and Table 15.13-4 describes the function of each bit.

Figure 15.13-4 Bit Structure of Serial Status Register (SSR)

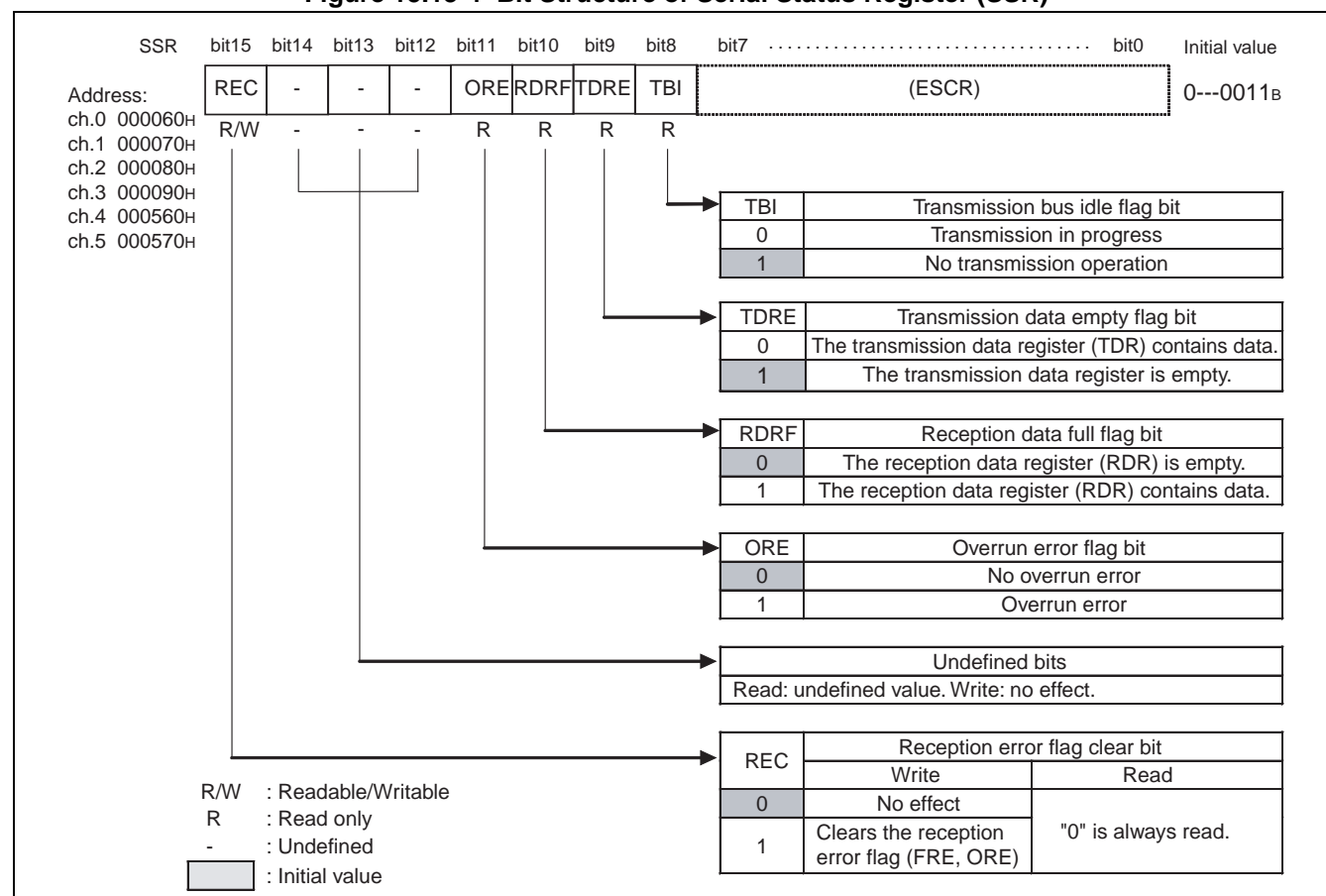


Table 15.13-4 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the ORE flag in the serial status register (SSR). <ul style="list-style-type: none"> • Writing "1" clears the error flag. • Writing "0" has no effect. Reading this bit always returns "0".
bit14 to bit12	Undefined bits	Read: undefined value Write: no effect
bit11	ORE: Overflow error flag bit	<ul style="list-style-type: none"> • This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR0/RDR1) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	<ul style="list-style-type: none"> • This flag indicates the status of the reception data register (RDR0/RDR1). • The bit is set to "1" when reception data is loaded to RDR0/RDR1. The bit is cleared to "0" when the reception data register (RDR0/RDR1) is read. • A reception interrupt request is output when the RDRF and RIE bits are set to "1". • RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. • During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR0/RDR1 is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. • This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	<ul style="list-style-type: none"> • This flag indicates the status of the transmission data register (TDR0/TDR1). • When transmission data is written to TDR0/TDR1, the bit becomes "0", indicating that TDR0/TDR1 contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR0/TDR1 no longer contains any valid data. • A transmission interrupt request is output when the TDRE and TIE bits are set to "1". • The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "15.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".

Table 15.13-4 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit8	TBI: Transmission bus idle flag bit	<ul style="list-style-type: none"> • This bit indicates that the CSIO is not performing transmission operation. • The bit is set to "0" when data is written to the transmission data register (TDR0/TDR1). • The bit is set to "1" when the transmission data register (TDR0/TDR1) is empty (TDRE = 1) and no transmission operation is in progress. • The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

15.13.4 Extended Communication Control Register (ESCR)

The extended communication control register (ESCR) can be used to set the transmission/reception data length and fix the serial output to "H".

■ Bit Structure of the Extended Communication Control Register (ESCR)

Figure 15.13-5 shows the bit structure of the extended communication control register (ESCR) and Table 15.13-5 describes the function of each bit.

Figure 15.13-5 Bit Structure of Extended Communication Control Register (ESCR)

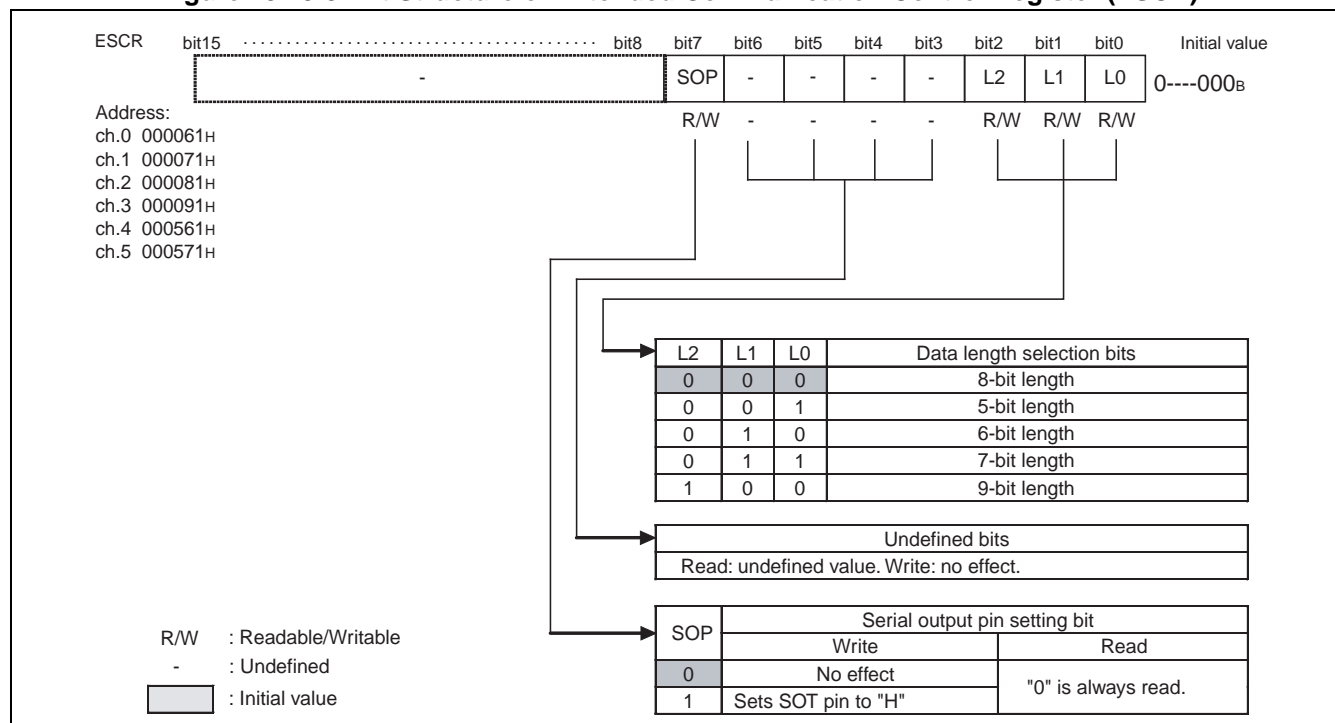


Table 15.13-5 Functional Description of Each Bit of Extended Communication Control Register (ESCR)

Bit name		Function
bit7	SOP: Serial output pin setting bit	<ul style="list-style-type: none"> This bit is used to set the serial output pin to "H". The SOT pin is set to "H" when "1" is written to this bit. It is not necessary to write "0" to this bit after that. Reading this bit always returns "0". Note: Do not set this bit during serial data transmission.
bit6 to bit3	Undefined bits	Read: undefined value Write: no effect
bit2 to bit0	L2 to L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. Selecting "000 _B " sets the data length to 8 bits. Selecting "001 _B " sets the data length to 5 bits. Selecting "010 _B " sets the data length to 6 bits. Selecting "011 _B " sets the data length to 7 bits. Selecting "100 _B " sets the data length to 9 bits. Note: Settings other than above are prohibited.

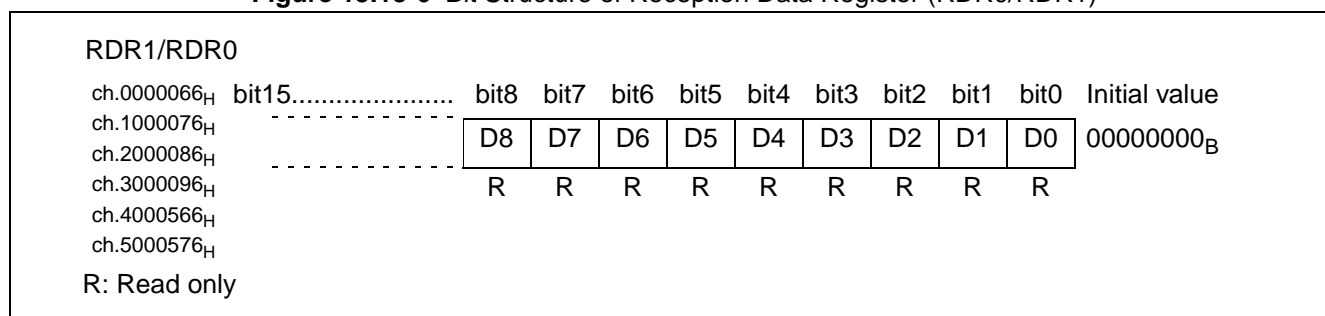
15.13.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception and transmission data registers are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

■ Reception Data Register (RDR0/RDR1)

Figure 15.13-6 illustrates the bit structure of the serial reception register (RDR0/RDR1).

Figure 15.13-6 Bit Structure of Reception Data Register (RDR0/RDR1)



The reception data register (RDR0/RDR1) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR0/RDR1).
- "0" is placed in upper bits, as shown below, in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR0/RDR1). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).
- Read the reception data register (RDR0/RDR1) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the serial reception data register (RDR0/RDR1) is read.
- If a reception error occurs (SSR:ORE), the data in the reception data register (RDR0/RDR1) becomes invalid.
- 16-bit access is used to read RDR0/RDR1 for a 9-bit transfer.

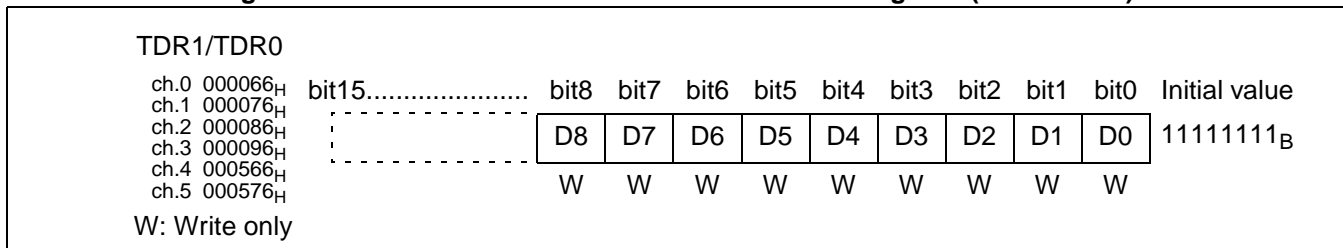
Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
- If a reception error occurs (SSR:ORE = 1) when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

■ Transmission Data Register (TDR0/TDR1)

Figure 15.13-7 illustrates the bit structure of the transmission data register.

Figure 15.13-7 Bit Structure of Transmission Data Register (TDR0/TDR1)



The transmission data register (TDR0/TDR1) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR0/TDR1) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Table 15.13-6 Invalid Data of Transmission Data Register (TDR0/TDR1)

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

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR0/TDR1).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- The next transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write the next transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".
- Transmission data cannot be written to the transmission data register (TDR0/TDR1) when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.

- 16-bit access is used to write to TDR0/TDR1 for a 9-bit transfer.
-

Notes:

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
 - For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "15.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
-

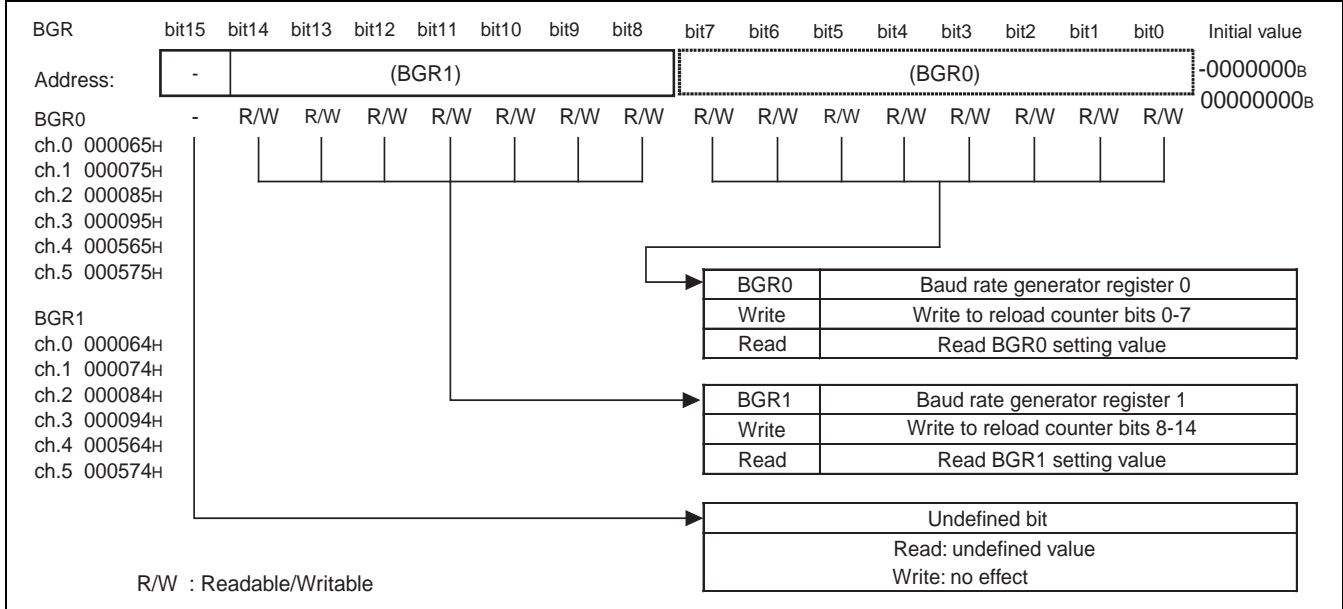
15.13.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 15.13-8 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 15.13-8 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



- A value is set to the baud rate generator registers 1, 0 (BGR1, BGR0).
- BGR0 and BGR1 correspond to the lower bits and upper bits respectively and they can write a reload value to be counted as well as read BGR0/BGR1 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
 - When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the setting of the SCINV bit. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
 - SCINV = 0: The "H" width of the serial clock is 1 peripheral clock (CLKP) cycle longer.
 - SCINV = 1: The "L" width of the serial clock is 1 peripheral clock (CLKP) cycle longer.
 - Select 1 or a larger number for the reload value. However, select 3 or a larger value for the reload value of the CSIO which will become the master, when using these CSIO's as the master and slave.
 - When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a CSIO reset (UPCL) after changing the BGR0/BGR1 setting value.
 - Set a baud rate to BGR0/BGR1 during the use of reception FIFO to set the reception FIFO idle detection enable bit (FCR1:FRIIE) to "1" and operate in slave mode.
-

15.13.7 FIFO Control Register 1 (FCR1)

The FIFO control register (FCR1) sets a FIFO test, selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 15.13-9 shows the bit structure of the FIFO control register 1 (FCR1) and Table 15.13-7 describes the function of each bit.

Figure 15.13-9 Bit Structure of FIFO Control Register 1 (FCR1)

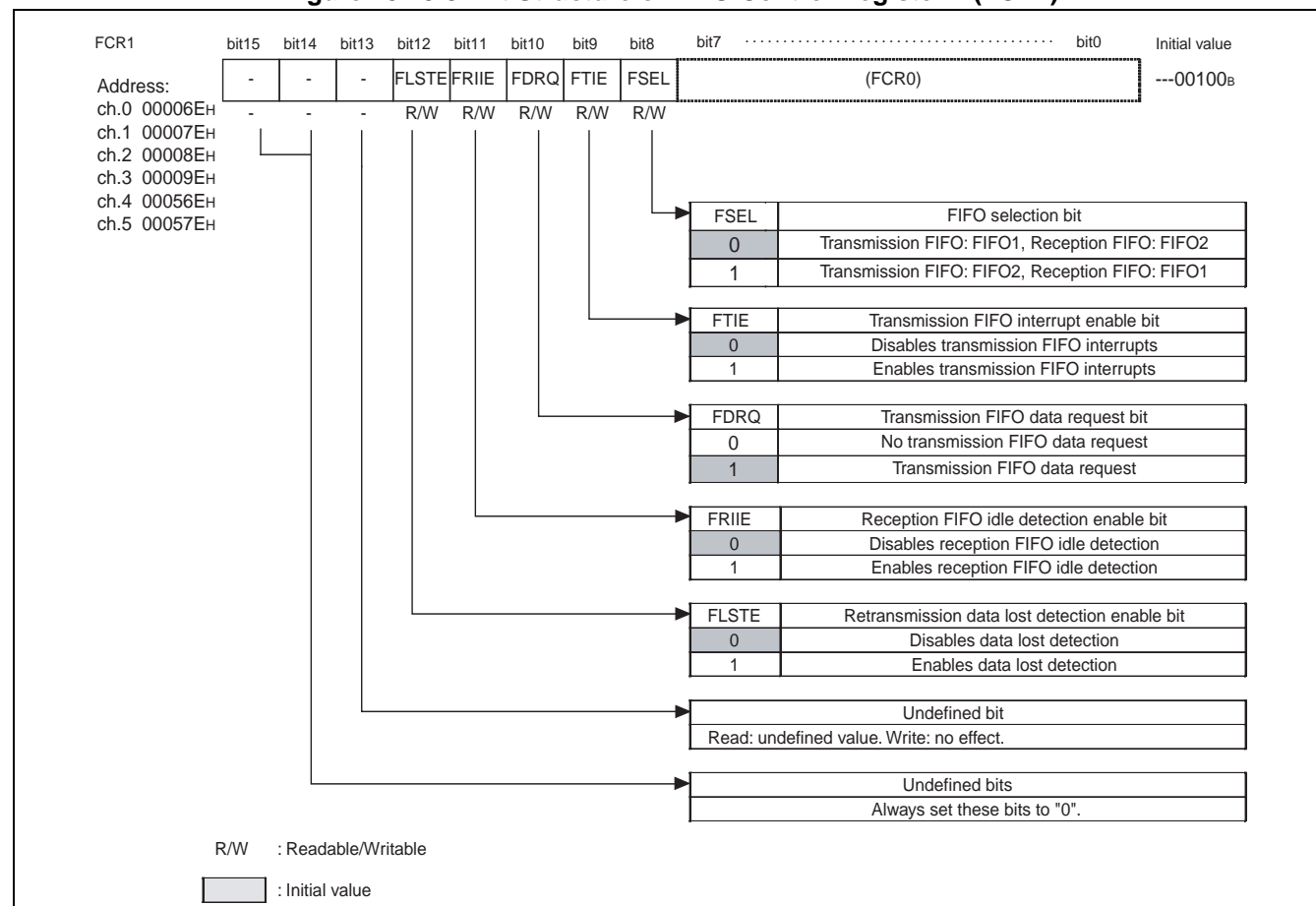


Table 15.13-7 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Undefined bits	Always set these bits to "0".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8-bit timer or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition <ul style="list-style-type: none"> • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition <ul style="list-style-type: none"> • Writing "0" to this bit • When the transmission FIFO is full. Note: It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. Note: This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FE2, FE1 = 0) and transmission/reception (TXE = RXE = 0) in advance.

15.13.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 15.13-10 shows the bit structure of the FIFO control register 0 (FCR0) and Table 15.13-8 describes the function of each bit.

Figure 15.13-10 Bit Structure of FIFO Control Register 0 (FCR0)

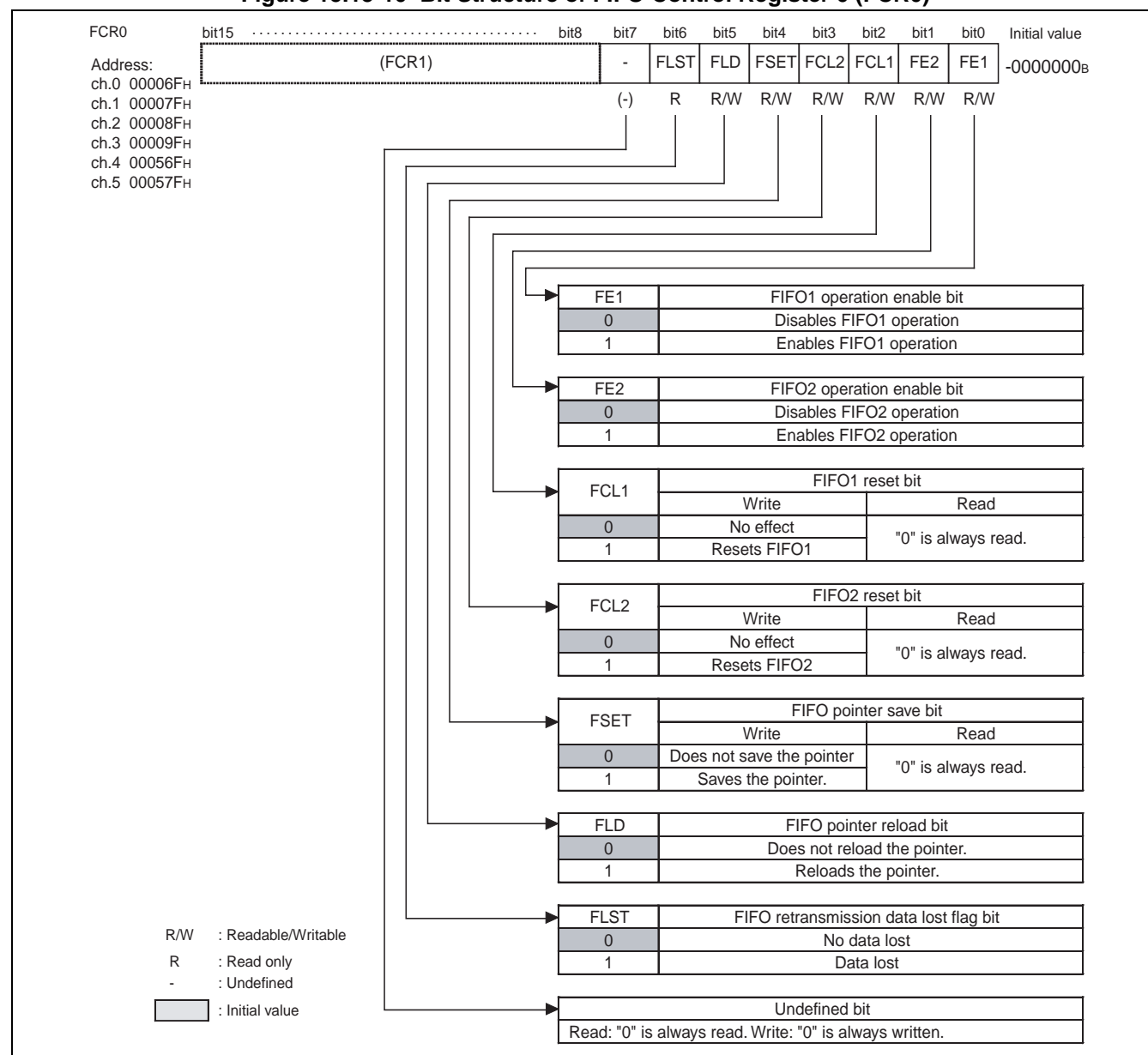


Table 15.13-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	<p>This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition</p> <ul style="list-style-type: none"> Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. <p>FLST reset conditions</p> <ul style="list-style-type: none"> FIFO reset (writing "1" to FCL) Writing "1" to the FLST bit <p>Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.</p>
bit5	FLD: FIFO pointer reload bit	<p>This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs.</p> <p>The bit becomes "0" when retransmission has been set.</p> <p>Note:</p> <p>Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset.</p> <p>It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress.</p> <p>Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.</p>
bit4	FSET: FIFO pointer save bit	<p>This bit is used to save the read pointer of the transmission FIFO.</p> <p>If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs.</p> <p>Setting the bit to "1" saves the current read pointer value.</p> <p>Setting the bit to "0" has no effect.</p> <p>Note:</p> <p>Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".</p>
bit3	FCL2: FIFO2 reset bit	<p>This bit is used to reset FIFO2.</p> <p>Setting this bit to "1" initializes the internal state of FIFO2.</p> <p>Only the FCR0: FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained.</p> <p>Note:</p> <p>Disable transmission/reception before resetting FIFO2.</p> <p>Set the transmission FIFO interrupt enable bit to "0" before the reset.</p> <p>The number of valid data elements for the FBYTE2 register will become "0".</p>

Table 15.13-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR1:FLST1 bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".</p>
bit1	FE2: FIFO2 operation enable bit	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> • To use FIFO2, set this bit to "1". • When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> • To use FIFO1, set this bit to "1". • When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO1 is disabled, its status is retained.

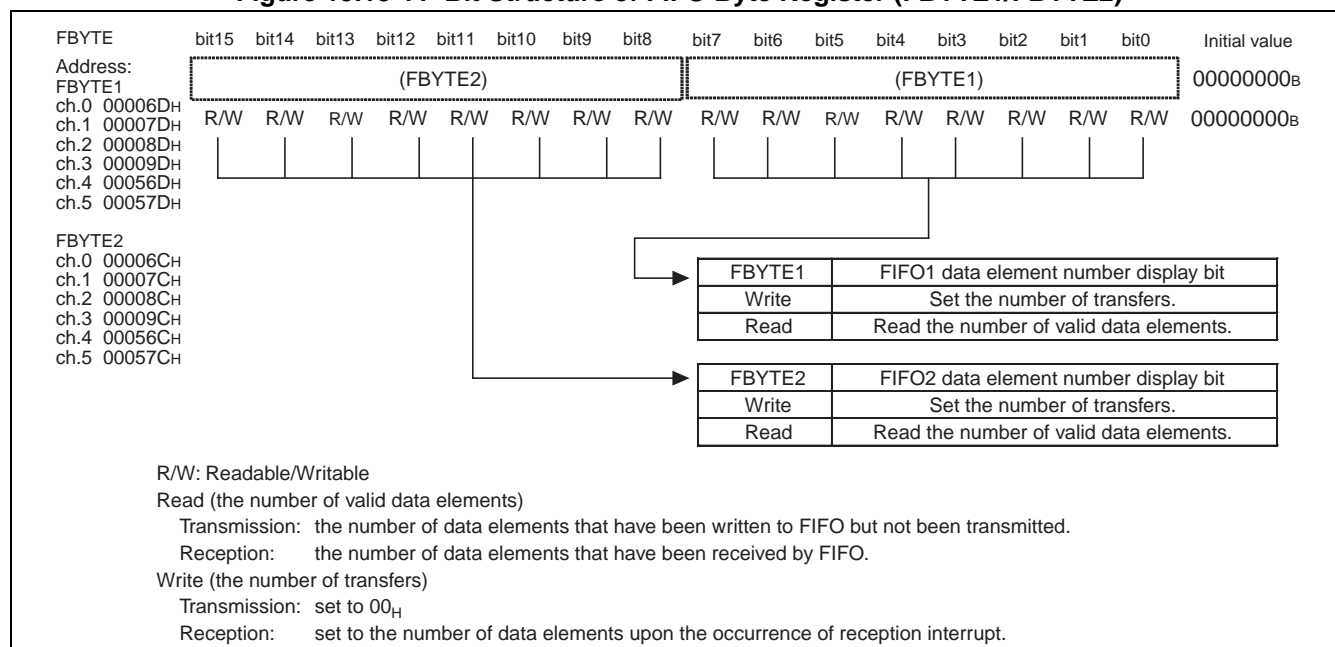
15.13.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 15.13-11 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 15.13-11 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)



The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 15.13-9 Displaying the Number of Data Elements

FSEL	FIFO selection	Number of bytes displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR0/RDR1 is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- To receive data in master operation (master reception), set the TIE and TBIE bits to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. When the TXE bit is set to "1", the serial clock will be output for a specified amount of data so that the specified amount of data can be received. To set the TIE and TBIE bits to "1", wait until the FDRQ becomes "1".

Notes:

- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
 - Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE and TBIE bits are set to "0".
 - To disable reception (RXE = 0) while receiving data in master operation, disable transmission/reception after disabling the transmission FIFO.
 - Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
 - To modify the FBYTE1/FBYTE2 of the reception FIFO, disable reception beforehand.
 - Read modify write (RMW) instructions cannot be used for this register.
 - Settings that will exceed the capacity of FIFO are prohibited.
-

15.14 Interrupts of CSIO (Clock Synchronous Serial Interface)

CSIO (clock synchronous serial interface) has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR0/RDR1) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR0/TDR1) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

■ Interrupts of CSIO

Table 15.14-1 lists the interrupt control bits and interrupt sources of the CSIO.

Table 15.14-1 Interrupt Control Bits and Interrupt Sources of CSIO

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Reception	RDRF	SSR	Reception of 1 byte	SCR:RIE	Reading reception data (RDR0/RDR1)
			Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR0/RDR1) until reception FIFO becomes empty
			Detection of the idle state of reception for 8-bit timer or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR0/TDR1), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to transmission data (TDR0/TDR1), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

15.14.1 Occurrence of Reception Interrupts and Flag Set Timing

Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:ORE).

■ Occurrence of Reception Interrupts and Flag Set Timing

Reception data is stored to the reception data register (RDR0/RDR1) when the last data bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:ORE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

Note:

If a reception error occurs, the data in the reception data register (RDR0/RDR1) will become invalid.

Figure 15.14-1 Reception Operation and Flag Set Timing

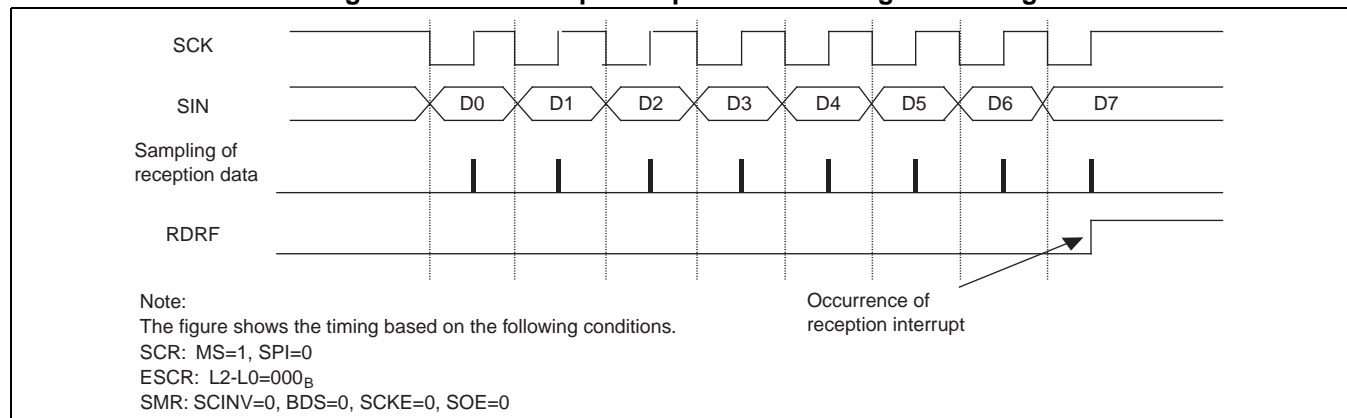
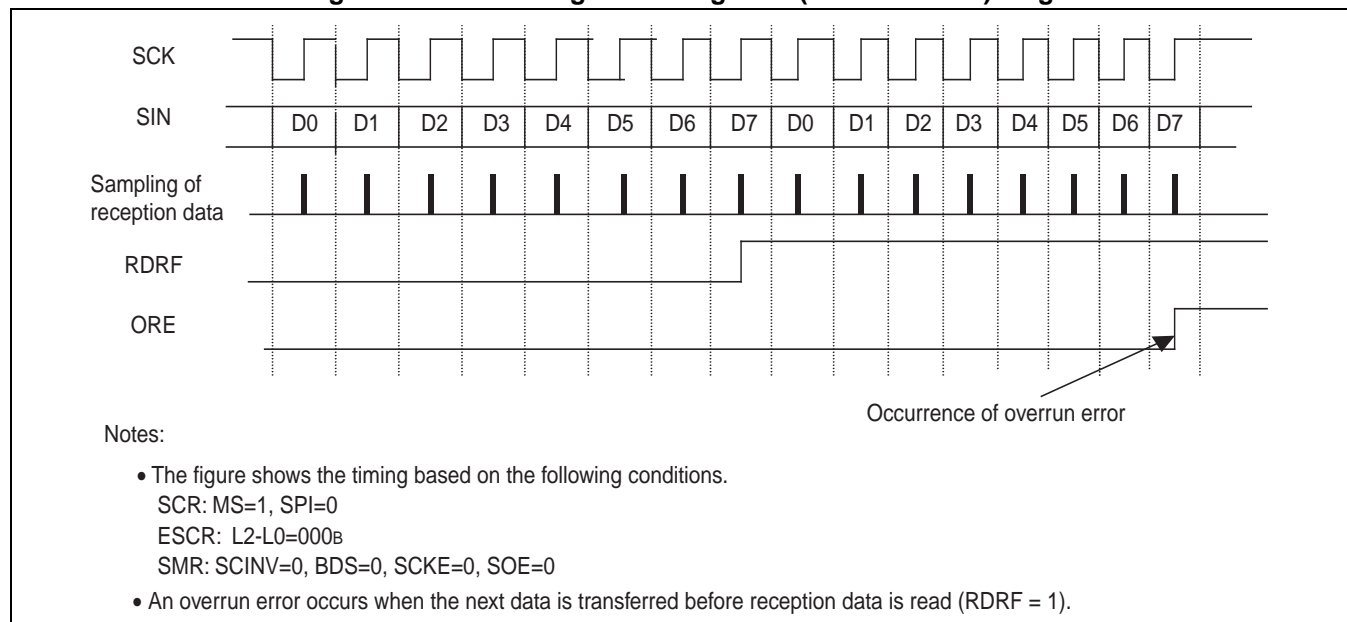


Figure 15.14-2 Timing for Setting ORE (Overrun Error) Flag



15.14.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1/FBYTE2 register (FBYTE1/FBYTE2) is received.

■ Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR0/RDR1 is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR0/RDR1) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 15.14-3 Timing for Generating Reception Interrupt when Reception FIFO is Used

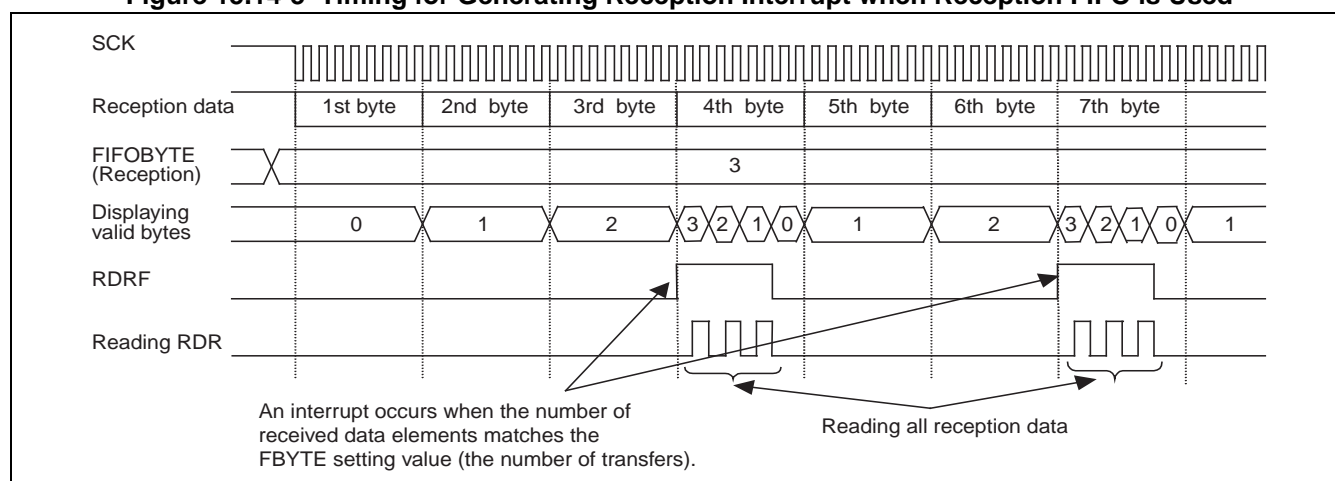
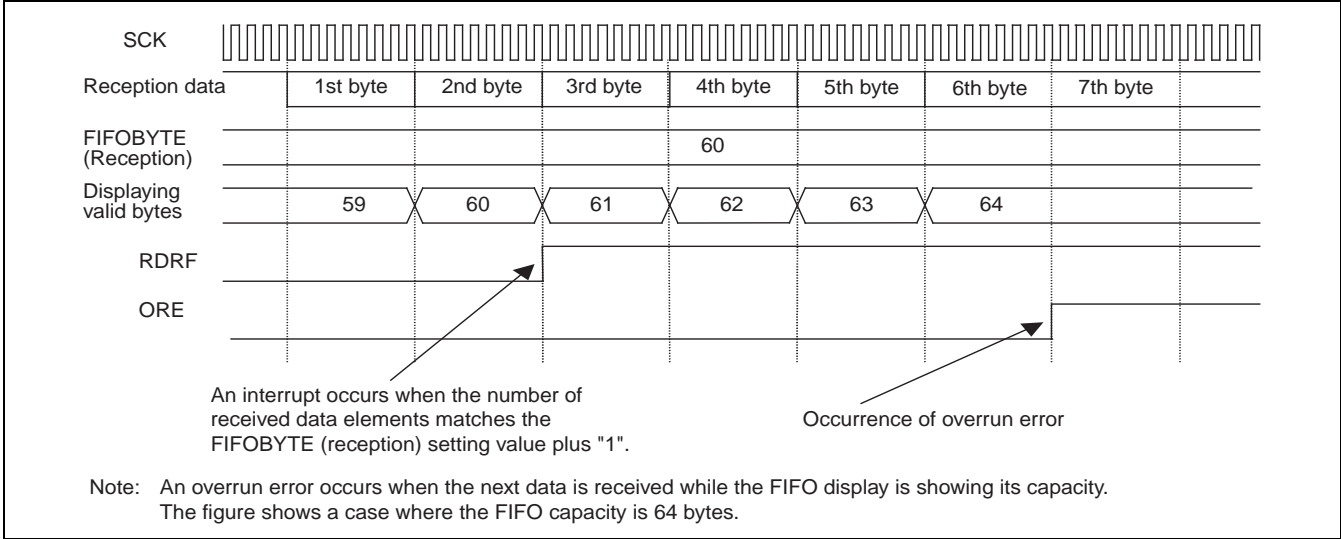


Figure 15.14-4 Timing for Setting ORE (Overrun Error) Flag Bit



15.14.3 Occurrence of Transmission Interrupts and Flag Set Timing

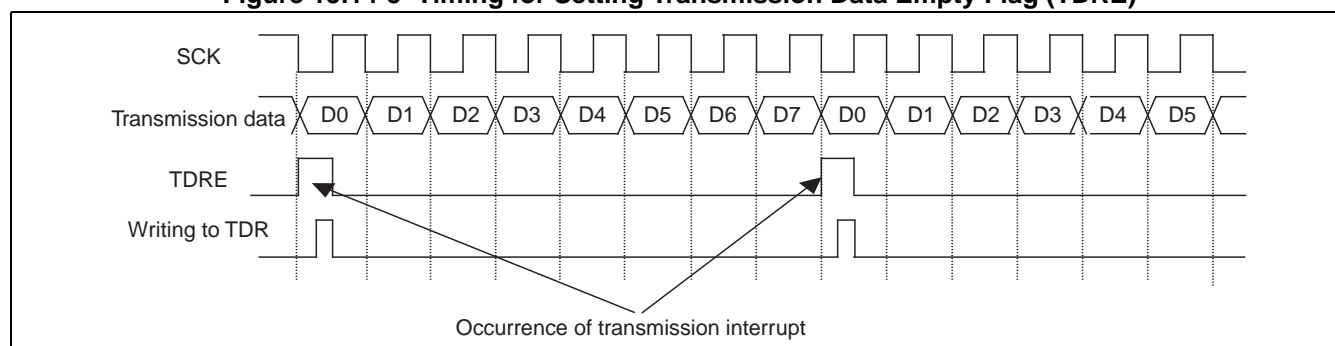
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR0/TDR1) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ Occurrence of Transmission Interrupts and Flag Set Timing

● Timing for setting the transmission data empty flag (TDRE)

It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR0/TDR1) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR0/TDR1).

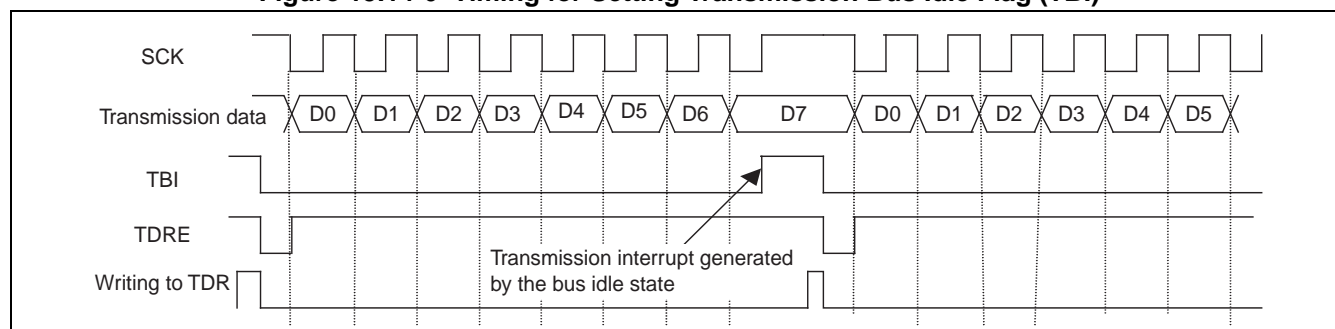
Figure 15.14-5 Timing for Setting Transmission Data Empty Flag (TDRE)



● Timing for setting the transmission bus idle flag (TBI)

The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR0/TDR1).

Figure 15.14-6 Timing for Setting Transmission Bus Idle Flag (TBI)



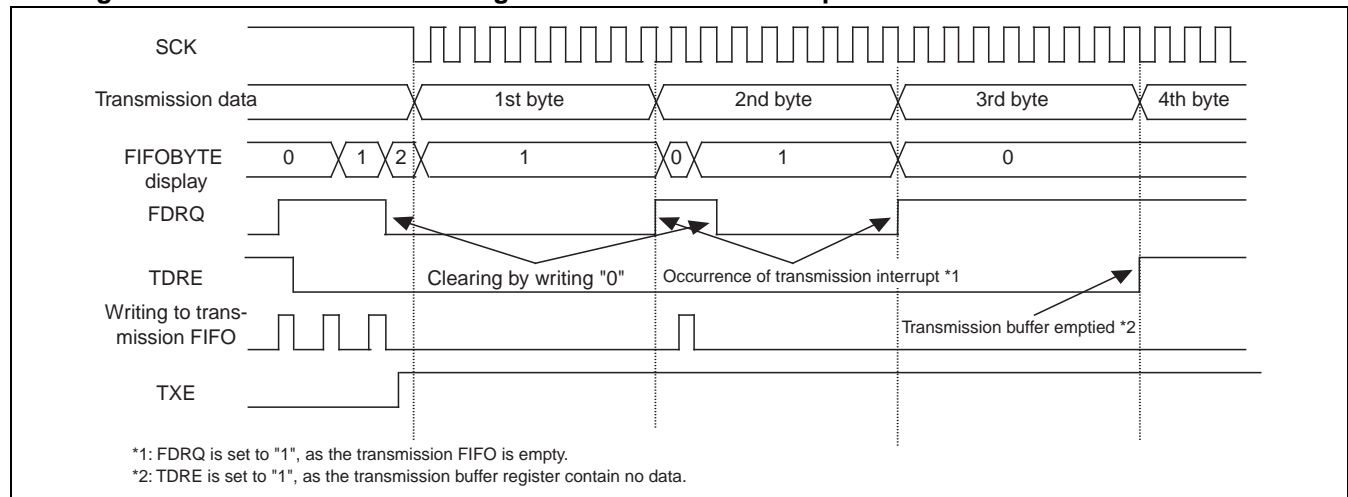
15.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

■ Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data. At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data.
FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 15.14-7 Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



15.15 Operation of CSIO (Clock Synchronous Serial Interface)

CSIO uses clock synchronization for its transfer system.

■ Operation of CSIO (Clock Synchronous Serial Interface)

■ Normal Transfer (I)

● Features

Table 15.15-1 Features of Normal Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Timing for transmission data output	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The setting values of registers required for the normal transfer (I) are shown below.

Table 15.15-2 Register Settings for Normal Transfer (I)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	0	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

Note:

The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master transmission : SCR:MS = 0, SMR:SCKE = 1, SOE = 1

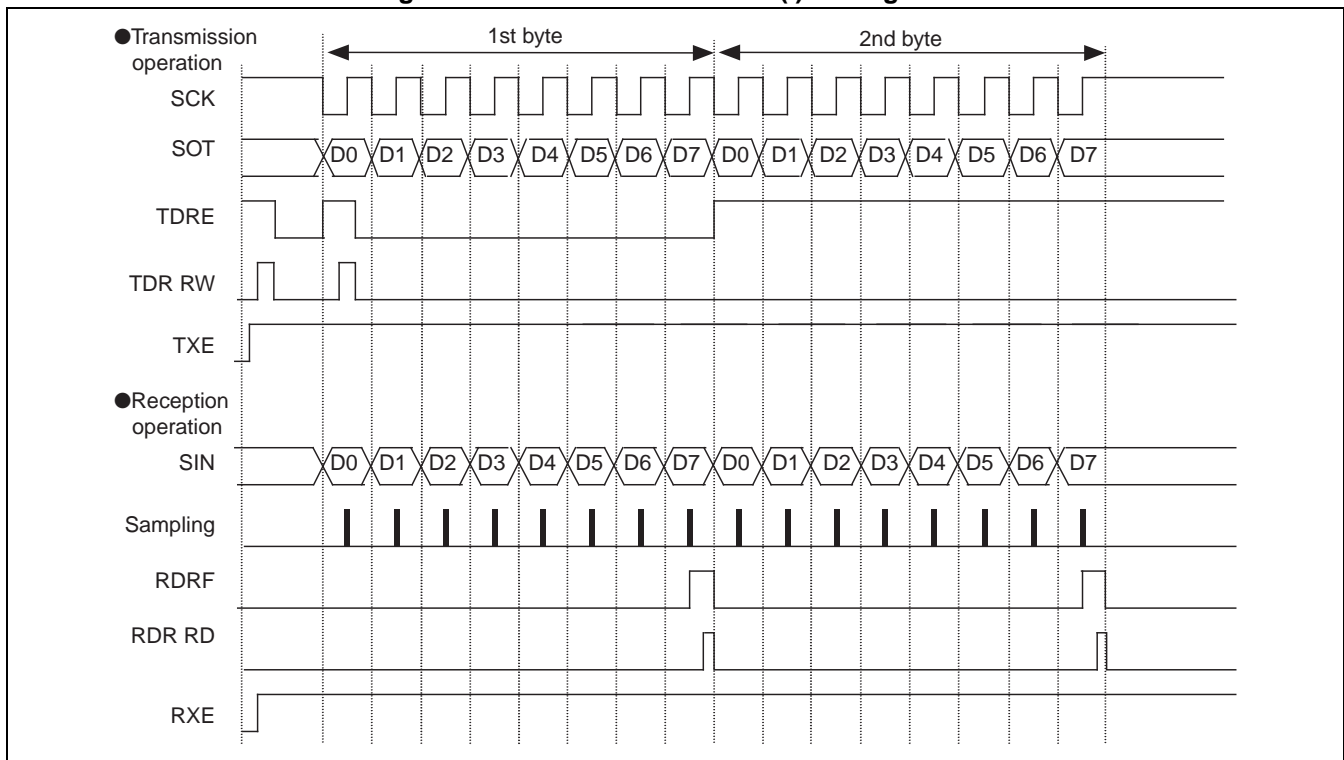
Master reception : SCR:MS = 0, SMR:SCKE = 1, SOE = 0

Slave transmission : SCR:MS = 1, SMR:SCKE = 0, SOE = 1

Slave reception : SCR:MS = 1, SMR:SCKE = 0, SOE = 0

● Normal transfer (I) timing chart

Figure 15.15-1 Normal Transfer (I) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.
- (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation
 - (1) If dummy data is written to TDR0/TDR1 when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).
 - (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

Notes:

- To only perform reception operation, write dummy data to TDR0/TDR1 to output the serial clock (SCK).
 - A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.
-

2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input.
 - (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
- Reception operation
 - (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE=1), reception data will be sampled at the rising edge of the serial clock input (SCK).
 - (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

■ Normal Transfer (II)

● Features

Table 15.15-3 Features of Normal Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The table below shows the register setting values required for the normal transfer (II).

Table 15.15-4 Register Settings for Normal Transfer (II)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	1	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	v
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

Note:

The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master transmission : SCR:MS = 0, SMR:SCKE = 1, SOE = 1

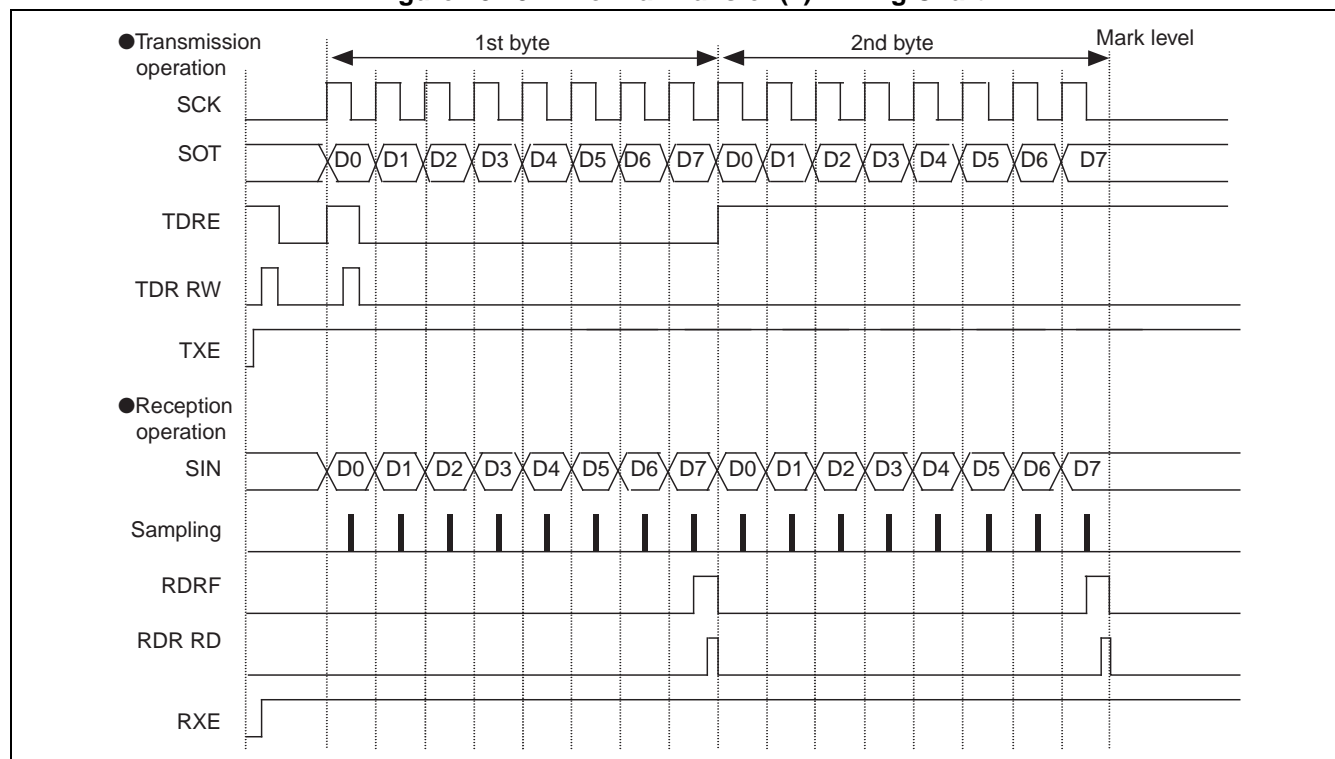
Master reception : SCR:MS = 0, SMR:SCKE = 1, SOE = 0

Slave transmission : SCR:MS = 1, SMR:SCKE = 0, SOE = 1

Slave reception : SCR:MS = 1, SMR:SCKE = 0, SOE = 0

■ Normal Transfer (II) Timing Chart

Figure 15.15-2 Normal Transfer (II) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.
- (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

• Reception operation

- (1) If dummy data is written to TDR0/TDR1 when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).
- (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
- (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

Notes:

- To only perform reception operation, write dummy data to TDR0/TDR1 to output the serial clock (SCK).
 - A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.
-

2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input.
 - (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
- Reception operation
 - (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operations is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).
 - (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

■ SPI Transfer (I)

● Features

Table 15.15-5 Features of SPI Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The table below shows the register setting values required for the SPI transfer (I).

Table 15.15-6 SPI Transfer (I) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	0	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	v
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

Note:

The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master transmission : SCR:MS = 0, SMR:SCKE = 1, SOE = 1

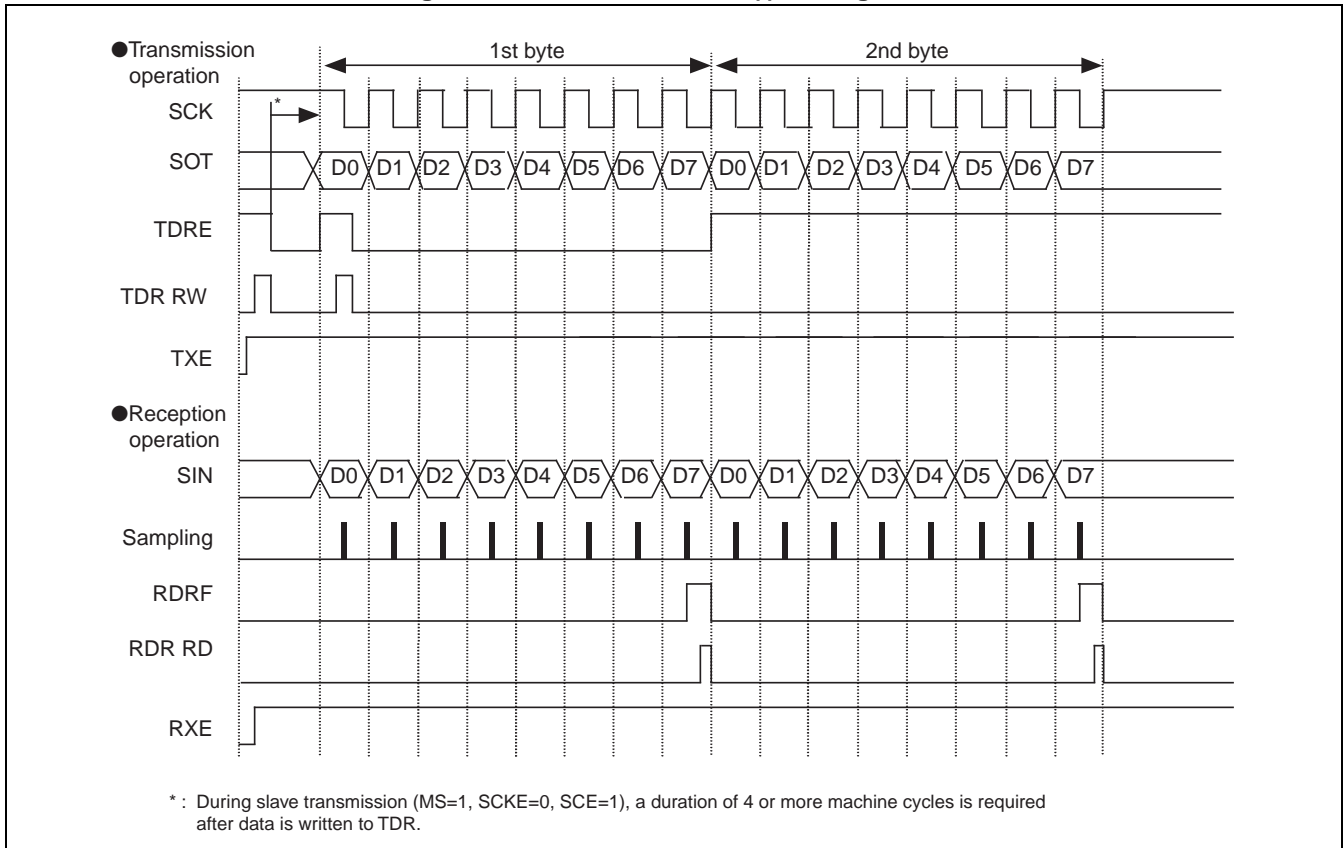
Master reception : SCR:MS = 0, SMR:SCKE = 1, SOE = 0

Slave transmission : SCR:MS = 1, SMR:SCKE = 0, SOE = 1

Slave reception : SCR:MS = 1, SMR:SCKE = 0, SOE = 0

● SPI transfer (I) timing chart

Figure 15.15-3 SPI Transfer (I) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.
- (2) SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

• Reception operation

- (1) If dummy data is written to TDR0/TDR1 when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).
- (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
- (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

Notes:

- To only perform reception operation, write dummy data to TDR0/TDR1 to output the serial clock (SCK).
 - A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.
-

2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.
 - (2) SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
- Reception operation
 - (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).
 - (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

■ SPI Transfer (II)

● Features

Table 15.15-7 Features of SPI Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 to 9 bits

● Register settings

The table below shows the register setting values required for the SPI transfer (II).

Table 15.15-8 SPI Transfer (II) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	-	-	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	-	-	*	*	*
TDR/ RDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

Note:

The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master transmission : SCR:MS = 0, SMR:SCKE = 1, SOE = 1

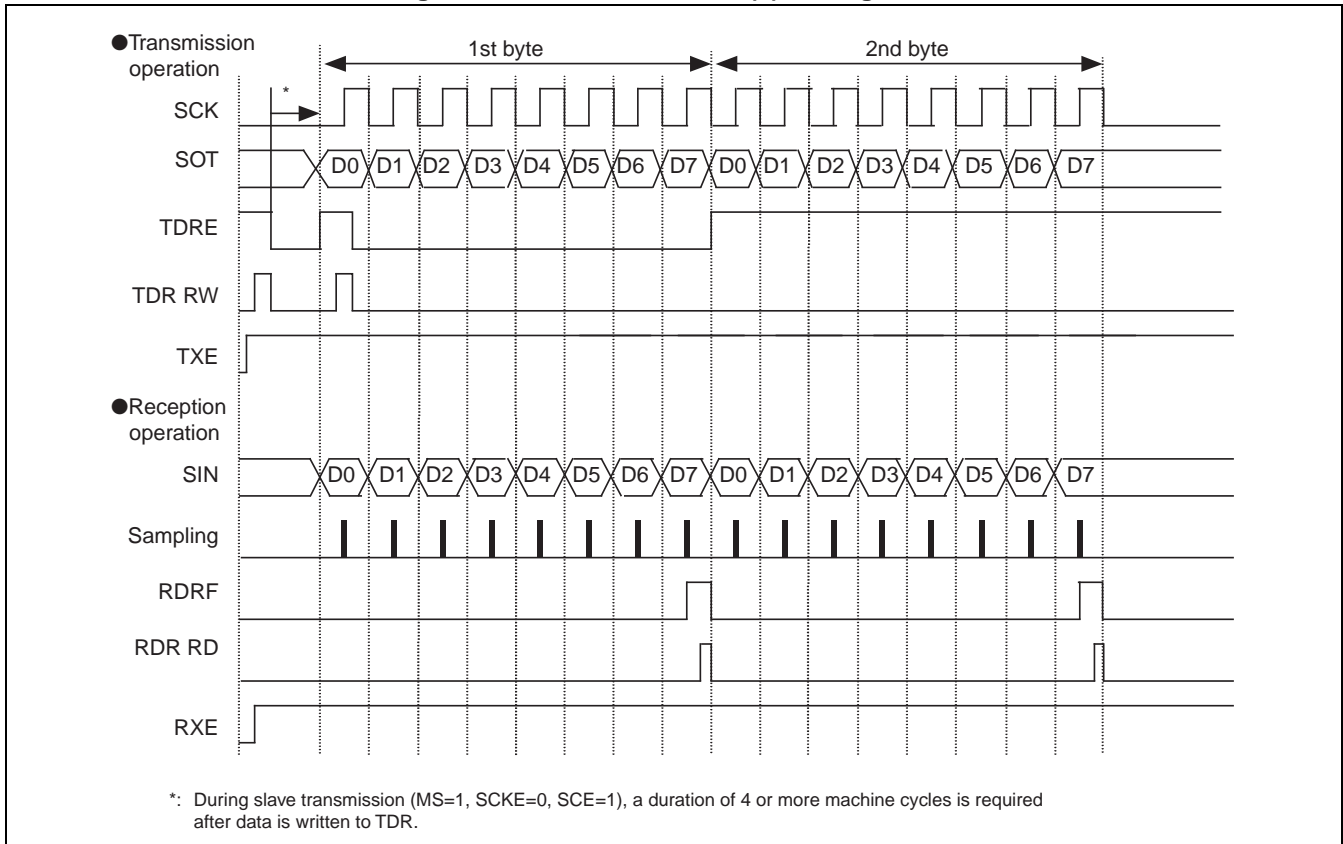
Master reception : SCR:MS = 0, SMR:SCKE = 1, SOE = 0

Slave transmission : SCR:MS = 1, SMR:SCKE = 0, SOE = 1

Slave reception : SCR:MS = 1, SMR:SCKE = 0, SOE = 0

● SPI transfer (II) timing chart

Figure 15.15-4 SPI Transfer (II) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.
- (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

• Reception operation

- (1) If dummy data is written to TDR0/TDR1 when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).
- (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
- (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

Notes:

- To only perform reception operation, write dummy data to TDR0/TDR1 to output the serial clock (SCK).
 - A specified number of frames of the serial clock (SCK) will be output if the byte number of frames to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.
-

2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) If transmission data is written to TDR0/TDR1 when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.
 - (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
- Reception operation
 - (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock input (SCK).
 - (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR0/RDR1) can be read.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR0/RDR1) is read.

15.16 Dedicated Baud Rate Generator

The dedicated baud rate generator only functions in master operation. However, set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

■ Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)

The dedicated baud rate generator settings are different between master and slave operations.

● Master operation

The baud rate is selected by dividing the internal clock using the dedicated baud rate generator.

- There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).
- The reload counter divides the internal clock, according to the set value.

● Slave operation

In slave operation (SCR:MS = 1), the dedicated baud rate generator does not function.

(The slave operation directly uses the external clock which is input from the clock input pin SCK.)

Note:

Set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

15.16.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value

b: Baud rate

ϕ : Peripheral clock (CLKP) frequency

(2) Example of calculation

If the peripheral clock (CLKP) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:

Reload value:

$$V = (16 \times 1000000) / 19200 - 1 = 832$$

So baud rate is:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

(3) Baud rate error

The following formula is used to calculate a baud rate error.

$$\text{Error (\%)} = (\text{calculated value} - \text{target value}) / \text{target value} \times 100$$

Example: peripheral clock (CLKP) = 20MHz, target baud rate = 153600bps

$$\text{Reload value} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{Baud rate (calculated value)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error (\%)} = (153846 - 153600) / 153600 \times 100 = 0.16 \text{ (\%)}$$

Notes:

- The reload counter halts when the reload value is set to "0".
 - When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the SCINV bit settings. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
 - When SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (CLKP) cycle longer.
 - When SCINV is set to "1", the "L" width of the serial clock is one peripheral clock (CLKP) cycle longer.
 - Select 3 or a larger value for the reload value.
-

MB91470/480 Series

■ Reload Values and Baud Rates for Different Peripheral Clock (CLKP) Frequencies

Table 15.16-1 Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	-	-	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	26666	<0.01	-	-	-	-	-	-	-	-	-

- Value: the value set in BGR1/BGR0 registers
- ERR: baud rate error (%)

■ Functions of Reload Counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock.

■ Starting a Count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

■ **Restart**

The reload counter restarts under the following conditions.

- For both transmission and reception reload counters

Programmable reset (SCR:UPCL bit)

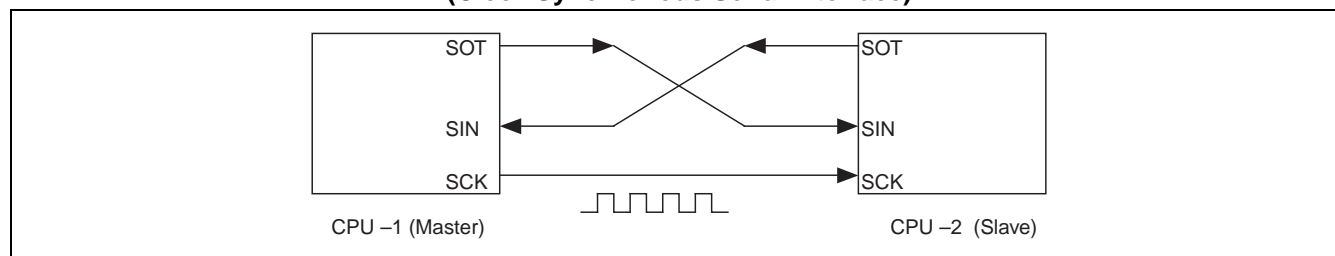
15.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

Two-way serial synchronous communication is enabled in CSIO (clock synchronous serial interface).

■ Connection between CPUs

Two-way communication should be selected for CSIO (clock synchronous serial interface). Two CPUs are connected to each other, as shown in Figure 15.17-1.

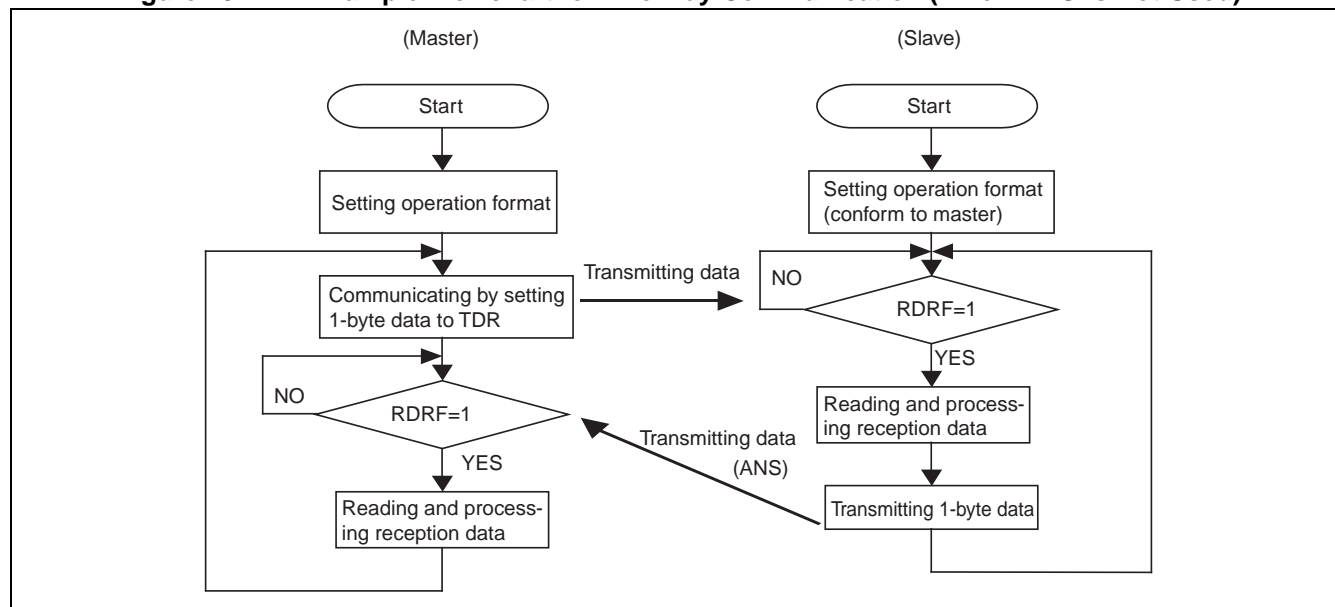
**Figure 15.17-1 Example of Two-way Communication Connection for CSIO
(Clock Synchronous Serial Interface)**



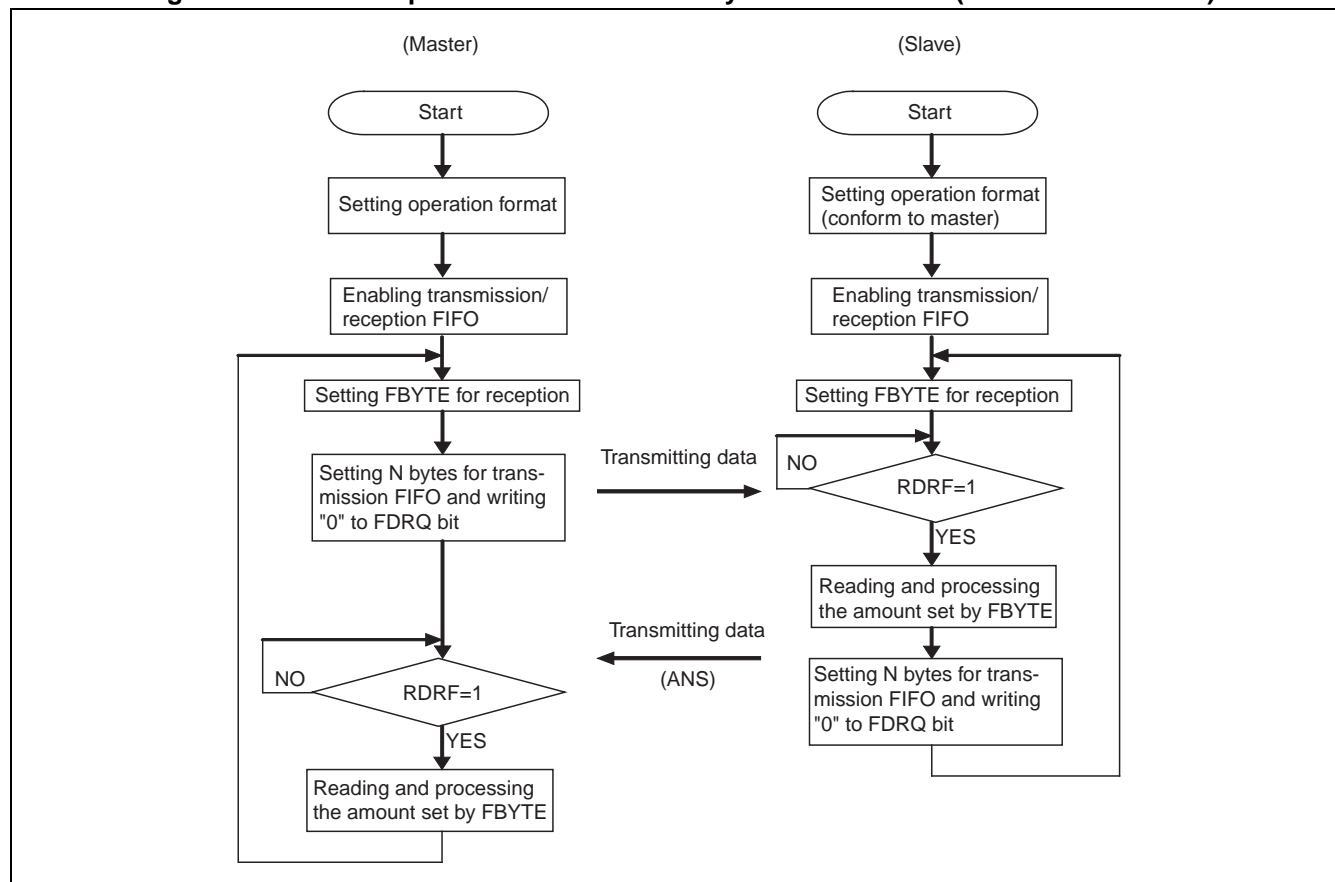
■ Flowchart

- When FIFO is not used

Figure 15.17-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



- When FIFO is used

Figure 15.17-3 Example Flowchart for Two-way Communication (When FIFO is Used)

15.18 Notes on CSIO Mode

The notes for when you use the CSIO mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.

15.19 I²C Interface

Of all the functions of the multi-function serial interface, this section describes the I²C interface that is supported in operation mode 4.

- I²C Interface
- Overview of I²C Interface
- Registers of I²C Interface
 - I²C Bus Control Register (IBCR)
 - Serial Mode Register (SMR)
 - I²C Bus Status Register (IBSR)
 - Serial Status Register (SSR)
 - Reception Data Register / Transmission Data Register (RDR/TDR)
 - 7-bit Slave Address Mask Register (ISMK)
 - 7-bit Slave Address Register (ISBA)
 - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
 - FIFO Control Register 1 (FCR1)
 - FIFO Control Register 0 (FCR0)
 - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of I²C Interface
 - Operation of I²C Interface Communication
 - Master Mode
 - Slave Mode
 - Bus Error
- Dedicated Baud Rate Generator
 - Example Flowchart for I²C Interface

15.20 Overview of I²C Interface

The I²C interface supports a bus between ICs and operates as a master/slave device on the I²C bus. This interface also comes with transmission/reception FIFO (up to 16 bytes each).

■ Functions of I²C Interface

The I²C interface has the following functions.

- Master/slave transmission & reception functionality
- Arbitration function
- Clock synchronization
- Transmission direction detection
- Generation and detection of repeated start condition
- Bus error detection
- General call addressing
- 7-bit addressing as master/slave
- Interrupts can be generated during transmission and bus errors.
- 10-bit addressing can be supported by a program.

■ Functions of FIFO

The FIFO has the following functions.

- Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)
- Transmission FIFO or reception FIFO selectable
- Transmission data can be resent.
- The interrupt timing for reception FIFO can be modified by software.
- FIFO reset is supported separately.

15.21 Registers of I²C Interface

This section lists the registers of the I²C interface.

■ List of Registers of I²C Interface

Figure 15.21-1 List of Registers of I²C Interface

Address		bit15	bit8 bit7	bit0
I ² C	000062 _H 000063 _H	IBCR (I ² C bus control register)	SMR (serial mode register)	
	000072 _H 000073 _H			
	000082 _H 000083 _H			
	000092 _H 000093 _H			
	000562 _H 000563 _H			
	000572 _H 000573 _H			
	000060 _H 000061 _H	SSR (serial status register)	IBSR (I ² C bus status register)	
	000070 _H 000071 _H			
	000080 _H 000081 _H			
	000090 _H 000091 _H			
	000560 _H 000561 _H			
	000570 _H 000571 _H			
	000066 _H 000067 _H	-	RDR/TDR (transmission/reception data register)	
	000076 _H 000077 _H			
	000086 _H 000087 _H			
	000096 _H 000097 _H			
	000566 _H 000567 _H			
	000576 _H 000577 _H			
	000064 _H 000065 _H	BGR1 (baud rate generator register 1)	BGR0 (baud rate generator register 0)	
	000074 _H 000075 _H			
	000084 _H 000085 _H			
	000094 _H 000095 _H			
	000564 _H 000565 _H			
	000574 _H 000575 _H			
	00006A _H 00006B _H	ISMK (7-bit slave address mask register)	ISBA (7-bit slave address register)	
	00007A _H 00007B _H			
	00008A _H 00008B _H			
	00009A _H 00009B _H			
	00056A _H 00056B _H			
	00057A _H 00057B _H			

(Continued)

(Continued)

Address		bit15	bit8 bit7	bit0
FIFO	00006E _H 00006F _H	FCR1 (FIFO control register 1)	FCR0 (FIFO control register 0)	
	00007E _H 00007F _H			
	00008E _H 00008F _H			
	00009E _H 00009F _H			
	00056E _H 00056F _H			
	00057E _H 00057F _H			
	00006C _H 00006D _H	FBYTE2 (FIFO2 byte register)	FBYTE1 (FIFO1 byte register)	
	00007C _H 00007D _H			
	00008C _H 00008D _H			
	00009C _H 00009D _H			
	00056C _H 00056D _H			
	00057C _H 00057D _H			

Table 15.21-1 Bit Assignment of I²C Interface

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	-	RIE	TIE	ITST1	ITST0
SSR/ IBSR	REC	TSET	-	-	ORE	RDRF	TDRE	-	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
RDR/ TDR	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

15.21.1 I²C Bus Control Register (IBCR)

The I²C bus control register (IBCR) selects master/slave mode, generates a repeated start condition, enables the acknowledge function, enables interrupts and displays an interrupt flag.

■ I²C Bus Control Register (IBCR)

Figure 15.21-2 shows the bit structure of the I²C bus control register (IBCR), and Table 15.21-2 describes the function of each bit.

Figure 15.21-2 Bit Structure of I²C Bus Control Register (IBCR)

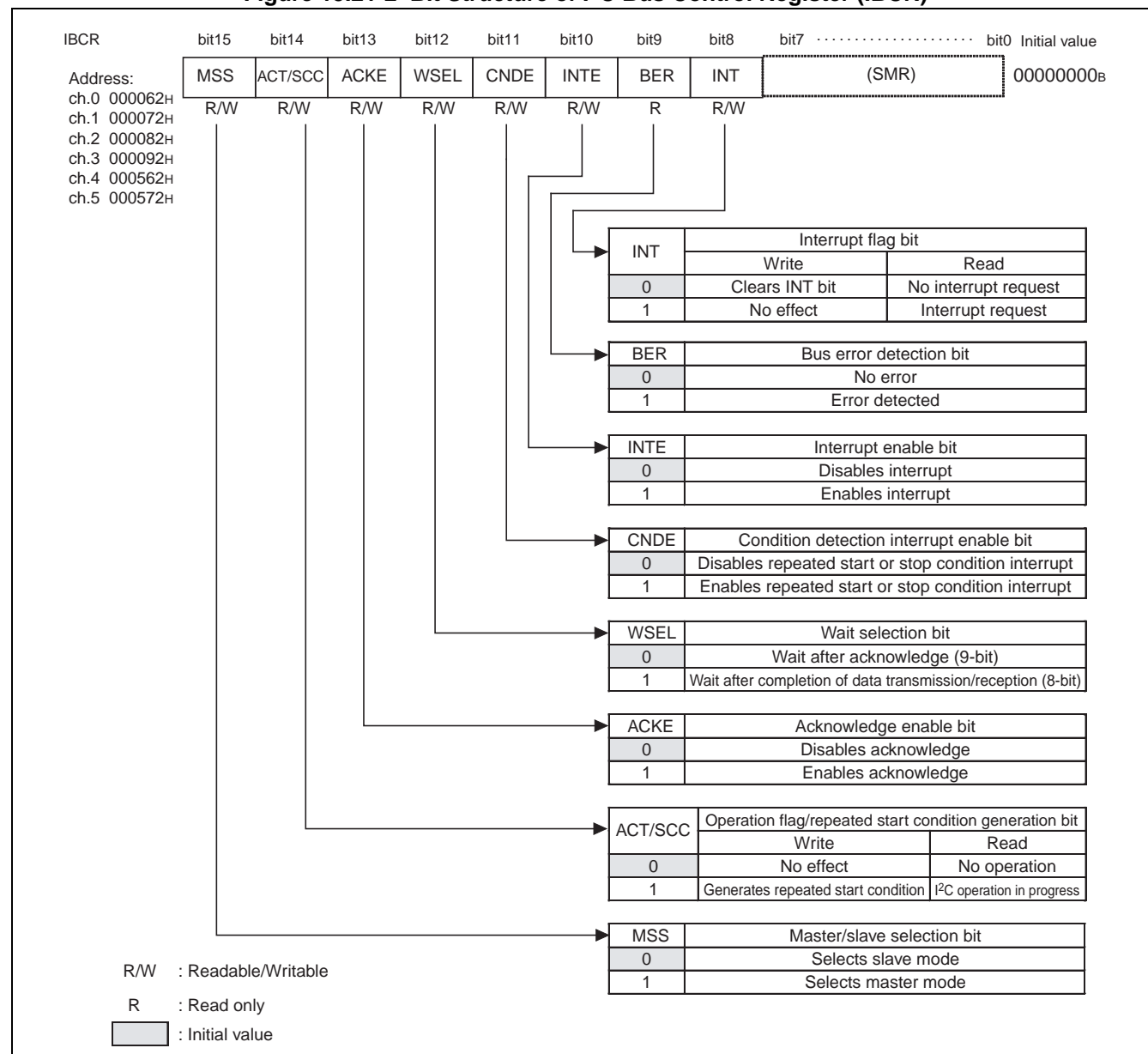


Table 15.21-2 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (1 / 4)

Bit name		Function															
bit15	MSS: Master/slave selection bit	<ul style="list-style-type: none"> Master mode will be selected if this bit is set to "1" when the I²C bus is in the idle state (EN = 1, BB = 0). If this bit is set to "1" when the BB bit in the IBSR register is set to "1", the register will wait to generate a start condition until the BB bit becomes "0". If a slave address match occurs during that wait and the device operates as a slave, this bit will be set to "0" and the AL bit in the IBSR register will be set to "1". A stop condition will be generated if "0" is written to this bit when the device is operating as the master (MSS = 1, ACT = 1) and the interrupt flag (INT) is set to "1". <p>The MSS bit is cleared under the following conditions.</p> <ul style="list-style-type: none"> The I²C interface is disabled (EN bit = 0). An arbitration lost condition occurs. A bus error is detected (BER bit = 1). "0" is written to the MSS bit when INT is "1". <p>The relationship between the MSS and ACT bits is shown below.</p> <table border="1"> <thead> <tr> <th>MSS bit</th><th>ACT bit</th><th>Status</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Idle</td></tr> <tr> <td>0</td><td>1</td><td>Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address</td></tr> <tr> <td>1</td><td>0</td><td>Master operation on standby</td></tr> <tr> <td>1</td><td>1</td><td>Master operation in progress (master mode)</td></tr> </tbody> </table> <p>*:ACK response: indicates that SDA of the I²C bus is at "L" during acknowledge.</p> <p>Note: Change the MSS bit from "1" to "0" when the INT bit and MSS bit are set to "1". If "0" is written to the MSS bit when the ACT bit is set to "1", the INT bit will also be cleared to "0". Writing "0" to the MSS bit returns "1" during master operation, as long as the ACT bit is set to "1".</p>	MSS bit	ACT bit	Status	0	0	Idle	0	1	Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address	1	0	Master operation on standby	1	1	Master operation in progress (master mode)
MSS bit	ACT bit	Status															
0	0	Idle															
0	1	Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address															
1	0	Master operation on standby															
1	1	Master operation in progress (master mode)															

Table 15.21-2 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (2 / 4)

Bit name		Function				
bit14	ACT/SCC: Operation flag / repeated start condition generation bit	This bit has different meanings between read and write. <div><table><tr><td>Read</td><td>Write</td></tr><tr><td>ACT bit</td><td>SCC bit</td></tr></table></div>	Read	Write	ACT bit	SCC bit
		Read	Write			
		ACT bit	SCC bit			
		The ACT bit indicates that the device is operating in master or slave mode.				
		Setting conditions for the ACT bit: <ul style="list-style-type: none">A start condition is output to the I²C bus (master mode).A slave address matches the address transmitted from the master (slave mode).A reserved address is detected and then an acknowledge is returned as a response (MSS = 0: slave mode).				
		Reset conditions for the ACT bit:				
		<Master mode> <ul style="list-style-type: none">A stop condition is detected.An arbitration lost condition is detected.A bus error is detected.The I²C interface is disabled (EN bit = 0).				
		<Slave mode> <ul style="list-style-type: none">A (repeated) start condition is detected.A stop condition is detected.An acknowledge is not returned although a reserved address is detected (RSA bit = 1).The I²C interface is disabled (EN bit = 0).A bus error occurs (BER bit = 1).				
		A repeated start is performed when "1" is written to this bit during master mode.				
		Writing "0" is invalid.				
Note:						
Write "1" to the SCC bit while an interrupt is occurring in master mode (MSS = 1, ACT = 1, INT = 1). The INT bit will be cleared to "0" if "1" is written to the SCC bit when the ACT bit is set to "1".						
In slave mode (MSS = 0, ACT = 1), it is prohibited to write "1" to this bit.						
The MSS bit has higher priority than the SCC bit, when "1" is written to the SCC bit and "0" is written to the MSS bit.						
The SCC bit is read when a read modify write (RMW) instruction is used.						

Table 15.21-2 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (3 / 4)

Bit name		Function
bit13	ACKE: Acknowledge enable bit	<ul style="list-style-type: none"> If this bit is set to "1", "L" will be output when an acknowledge is returned. When ACT is set to "1", this bit must be modified, if necessary, while the INT bit is set to "1". <p>This bit is invalid under the following conditions.</p> <ul style="list-style-type: none"> An acknowledge is returned to an address field other than the reserved address (automatic generation). Data transmission (RSA = 0, TRX = 1, FBT = 0) An ACK is returned whenever the reception FIFO is enabled and slave reception is selected (FE = 1, MSS = 0, ACT = 1). When the reception FIFO is enabled, WSEL is "0", and master reception is selected (FE = 1, MSS = 1, ACT = 1, WSEL = 0), setting the TDRE bit to "0" returns an ACK while setting it to "1" returns a NACK. An ACK is always returned when the reception FIFO is enabled, WSEL is "0", and slave transmission is performed through reserved address detection (RSA = 1, TRX = 1, FBT = 1). To allow a NACK to be returned, disable the reception FIFO and set ACKE to "0" during an interrupt after the reserved address detection. The reception FIFO is enabled, WSEL is "1", and the transmission data register contains data in master reception (FE = 1, MSS = 1, ACT = 1, WSEL = 1, TDRE = 0).
bit12	WSEL: Wait selection bit	<ul style="list-style-type: none"> This bit is used to determine whether an interrupt should occur (INT = 1) before or after an acknowledgement to put the I²C bus in a wait state. The WSEL bit is invalid under the following conditions. <ul style="list-style-type: none"> An interrupt occurs for the first byte^{*1} (INT = 1). A reserved address is detected (FBT = 1, RSA = 1). A NACK response^{*2} is detected during a data transfer when FIFO is used (FE = 1, RACK = 1, ACT = 1). The reception FIFO becomes full when it is used. <p>*1: First byte: indicates the data after a (repeated) start condition *2: NACK response: indicates that SDA of the I²C bus is at "H" during acknowledgement.</p>
bit11	CNDE: Condition detection interrupt enable bit	<p>This bit is used to enable the occurrence of interrupts when a stop condition or a repeated start condition is detected in master or slave mode (ACT = 1). An interrupt occurs when the RSC or SPC bit in the IBSR register is set to "1" and this bit is set to "1".</p>
bit10	INTE: Interrupt enable bit	<p>This bit is used to enable an interrupt (INT = 1) for data transmission/reception and a bus error in master or slave mode.</p>
bit9	BER: Bus error detection bit	<p>This bit indicates that an error is detected on the I²C bus.</p> <p>Setting conditions for the BER bit:</p> <ul style="list-style-type: none"> A start condition or stop condition is detected during the transfer of the first byte[*]. A (repeated) start condition or stop condition is detected at the 2nd bit - 9th (acknowledge) bit of data in the second or succeeding byte. <p>Reset conditions for the BER bit:</p> <ul style="list-style-type: none"> "0" is written to the INT bit when BER is set to "1". The I²C interface is disabled (EN = 0). <p>*: First byte: indicates the data after (repeated) start condition</p> <p>Note: Data cannot be transmitted or received properly if this bit is set to "1" when the interrupt flag (INT bit) is set to "1". In this case, take an action such as retransmitting the data.</p>

Table 15.21-2 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (4 / 4)

Bit name	Function
bit8	<p data-bbox="469 298 1474 422">This bit is set to "1" after the 8th or 9th bit (ACK) of data transmission/reception in master or slave mode, or upon the occurrence of a bus error. In cases other than the occurrence of a bus error, SCL is set to "L" when the INT bit is set to "1". When the INT bit is set to "0", SCL is released from the "L" state.</p> <p data-bbox="469 432 834 462">Setting conditions for the INT bit:</p> <p data-bbox="469 472 574 501"><8th bit></p> <ul data-bbox="505 506 1474 785" style="list-style-type: none"> • A reserved address is detected in the first byte. • WSEL is "1", and an arbitration lost condition is detected in the second or succeeding byte. • WSEL is "1", and the TDRE bit is set to "1" in the second or succeeding byte during master operation. • WSEL is "1", the reception FIFO is disabled and the TDRE bit is set to "1" in the second or succeeding byte during slave operation. • WSEL is set to "1", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. <p data-bbox="469 798 574 827"><9th bit></p> <ul data-bbox="505 831 1474 1493" style="list-style-type: none"> • An arbitration lost condition is detected in the first byte. • A NACK is received at times other than when a stop condition output is set ("0" written to the MSS bit during master operation). • The TDRE bit is set to "1" in the transmission direction (TRX = 1) of master or slave mode without the detection of a reserved address in the first byte. • The reception FIFO contains data when the reception FIFO is enabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. • The TDRE bit is set to "1" when the reception FIFO is disabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. • WSEL is set to "0", and an arbitration lost condition is detected in the second or succeeding byte. • WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during master mode operation. • WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. • WSEL is set to "0", the reception FIFO is disabled, and slave reception is selected. In slave reception, however, an interrupt does not occur in the 9th bit of the first byte in which a reserved address is detected. • The reception FIFO is enabled, and it becomes full in slave reception. <p data-bbox="469 1505 672 1535"><Other condition></p> <p data-bbox="505 1539 753 1568">A bus error is detected.</p> <p data-bbox="505 1572 850 1602">Reset conditions for the INT bit:</p> <ul data-bbox="505 1606 1377 1692" style="list-style-type: none"> • (1)"0" is written to the INT bit. • (2)"0" is written to the MSS bit when the INT bit is "1" and the ACT bit is "1". • (3)"1" is written to the SCC bit when the INT bit is "1" and the ACT bit is "1". <p data-bbox="505 1696 891 1726">Writing "1" to the INT bit is invalid.</p> <p data-bbox="469 1736 532 1766">Note:</p> <p data-bbox="505 1770 1474 1829">Setting the EN bit to "0" may set the RDRF and INT bits to "1", depending on the reception timing. In this case, read the reception data to clear the INT bit.</p> <p data-bbox="505 1833 1198 1862">"1" is read when a read modify write (RMW) instruction is used.</p> <p data-bbox="505 1866 1417 1925">The INT bit cannot be set to "1" even if the reception FIFO is full in master reception operation when the reception FIFO is enabled.</p>

MB91470/480 Series**15.21.2 Serial Mode Register (SMR)**

The serial mode register (SMR) sets the operation mode, and enables or disables transmission/reception interrupts.

Serial Mode Register (SMR)

Figure 15.21-3 shows the bit structure of the serial mode register (SMR), and Table 15.21-3 describes the function of each bit.

Figure 15.21-3 Bit Structure of Serial Mode Register (SMR)

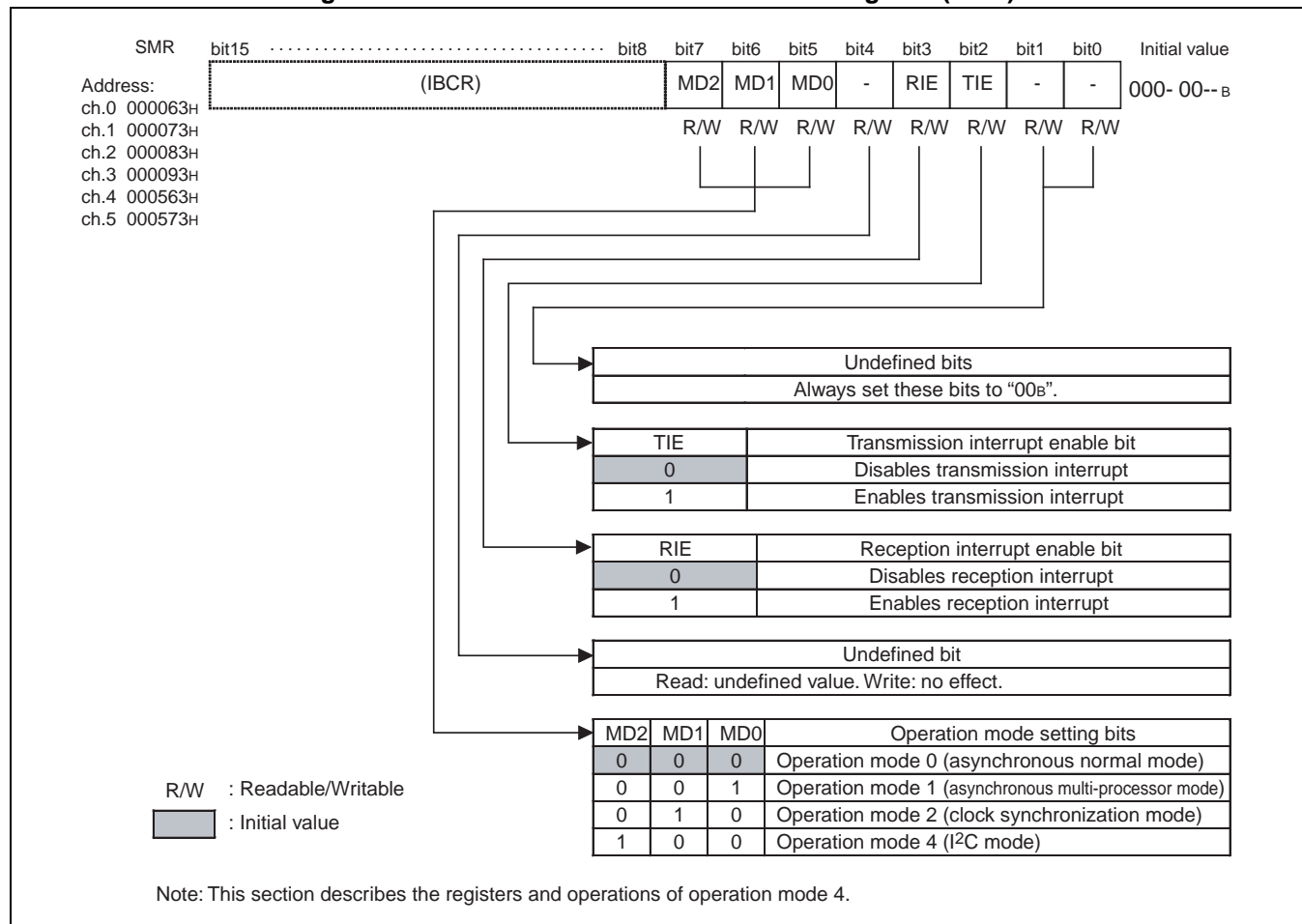


Table 15.21-3 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	<p>These bits are used to select the operation mode.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 4 (I²C mode).</p> <p>Note:</p> <p>Settings other than above are prohibited.</p> <p>To switch the operation mode, disable I²C first (ISMK:EN = 0).</p> <p>Set each register after selecting the operation mode.</p>
bit4	Undefined bit	<p>Read: undefined value</p> <p>Write: no effect</p>
bit3	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable or disable the output of reception interrupt requests to the CPU. A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or the error flag bit (ORE) is set to "1". <p>Note:</p> <p>Set this bit to "0" when receiving data using the INT bit in the I²C bus control register (IBCR).</p>
bit2	TIE: Transmission interrupt enable bit	<ul style="list-style-type: none"> This bit is used to enable or disable the output of transmission interrupt requests to the CPU. A transmission interrupt request is output when the TIE and TDRE bits are set to "1". <p>Note:</p> <p>Set this bit to "0" when transmitting data using the INT bit in the I²C bus control register (IBCR).</p>
bit1, bit0	Undefined bits	Always set these bits to "00 _B ".

Note:

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when IBCR and SMR are written simultaneously with 16-bit write access, IBCR reflects the written content.

15.21.3 I²C Bus Status Register (IBSR)

The I²C bus status register (IBSR) indicates the detection of a repeated start condition, acknowledge, data direction, arbitration lost condition, stop condition, I²C bus status and bus error.

■ I²C Bus Status Register (IBSR)

Figure 15.21-4 shows the bit structure of the I²C bus status register (IBSR) and Table 15.21-4 describes the function of each bit.

Figure 15.21-4 Bit Structure of I²C Bus Status Register (IBSR)

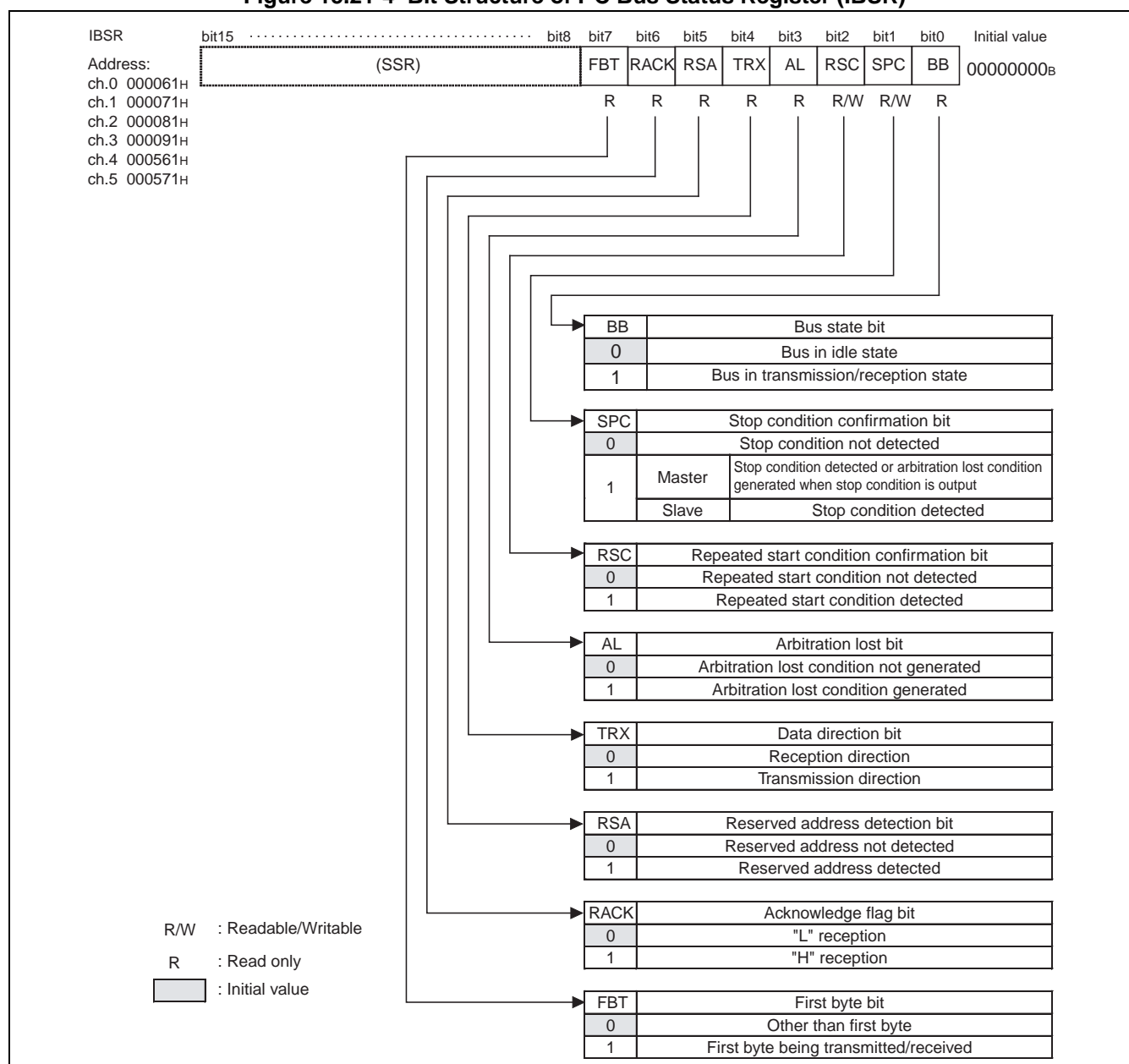


Table 15.21-4 Functional Description of Each Bit of I²C Bus Status Register (1 / 3)

Bit name		Function
bit7	FBT: First byte bit	<p>This bit indicates the first byte.</p> <p>Setting condition for the FBT bit: A (repeated) start condition is detected.</p> <p>Clearing conditions for the FBT bit:</p> <ul style="list-style-type: none"> (1) The second byte is transmitted or received. (2) A stop condition is detected. (3) The I²C interface is disabled (EN bit = 0). (4) A bus error is detected (BER bit = 1).
bit6	RACK: Acknowledge flag bit	<p>This bit is used to indicate the acknowledge received for the first byte during master or slave mode.</p> <p>Update conditions for the RACK bit</p> <ul style="list-style-type: none"> (1) Acknowledge for the first byte (2) Acknowledge for data in master or slave mode <p>Clearing conditions for the RACK bit (RACK bit = 0)</p> <ul style="list-style-type: none"> (1) A (repeated) start condition is detected. (2) The I²C interface is disabled (EN bit = 0). (3) A bus error is detected (BER bit = 1).
bit5	RSA: Reserved address detection bit	<p>This bit indicates the detection of a reserved address.</p> <p>Setting condition for the RSA bit (RSA = 1) The first byte is set to "0000XXXX_B" or "1111XXXX_B". "X" can be "0" or "1".</p> <p>Reset conditions for the RSA bit (RSA = 0)</p> <ul style="list-style-type: none"> (1) A (repeated) start condition is detected. (2) A stop condition is detected. (3) The I²C interface is disabled (EN bit = 0). (4) A bus error is detected (BER bit = 1). <p>When the RSA bit is set to "1" in the first byte, the interrupt flag (INT) is set to "1" at the falling edge of SCL in the 8th bit of the first byte to set the SCL to "L", whether the FIFO is enabled or disabled. In this case, ACKE should be set to "1" and the interrupt flag (INT) should be cleared to "0" in order to read reception data and allow the device to operate as a slave. If the TRX bit is set to "0", the device will receive data as a slave. To disable data reception in the middle of the operation, set the ACKE bit to "0". No more data will be received afterward.</p> <p>Note:</p> <p>When ACKE is set to "0" during a data transfer, it is prohibited to set ACKE to "1" until a stop condition or repeated start condition is detected.</p> <p>If slave transmission is confirmed during an interrupt by the detection of a reserved address, an ACK will be returned when the reception FIFO has been enabled. Therefore, disable the reception FIFO and set ACKE to "0".</p>

Table 15.21-4 Functional Description of Each Bit of I²C Bus Status Register (2 / 3)

Bit name		Function
bit4	TRX: Data direction bit	<p>This bit indicates the data direction.</p> <p>Setting conditions for the TRX bit:</p> <ul style="list-style-type: none"> (1) A (repeated) start condition is transmitted in master mode. (2) The 8th bit of the first byte is "1" in slave mode (transmission direction as a slave). <p>Reset conditions for the TRX bit:</p> <ul style="list-style-type: none"> (1) An arbitration lost condition is generated (AL = 1). (2) The 8th bit of the first byte is "0" in slave mode (reception direction as a slave). (3) The 8th bit of the first byte is "1" in master mode (reception direction as the master). (4) A stop condition is detected. (5) A (repeated) start condition is detected in modes other than master mode. (6) The I²C interface is disabled (EN bit = 0). (7) A bus error is detected (BER bit = 1).
bit3	AL: Arbitration lost bit	<p>This bit indicates an arbitration lost condition.</p> <p>Setting conditions for the AL bit:</p> <ul style="list-style-type: none"> (1) The output data is different from the received data in master mode. (2) The device is operating as a slave although the MSS bit has been set to "1". (3) A repeated start condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. (4) A stop condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. (5) A repeated start condition cannot be generated in master mode, despite attempts to do so. (6) A stop condition cannot be generated in master mode, despite attempts to do so. <p>Reset conditions for the AL bit:</p> <ul style="list-style-type: none"> (1) "1" is written to the MSS bit. (2) "0" is written to the INT bit. (3) "0" is written to the SPC bit when the AL and SPC bits are set to "1". (4) The I²C interface is disabled (EN bit = 0). (5) A bus error is detected (BER bit = 1).
bit2	RSC: Repeated start condition confirmation bit	<p>This bit indicates that a repeated start condition has been detected in master or slave mode.</p> <p>Setting condition for the RSC bit:</p> <p>A repeated start condition is detected after acknowledgement during slave or master mode operation.</p> <p>Reset conditions for the RSC bit:</p> <ul style="list-style-type: none"> (1) "0" is written to the RSC bit. (2) "1" is written to the MSS bit. (3) The I²C interface is disabled (EN bit = 0). <p>Writing "1" to this bit is invalid.</p> <p>Note:</p> <p>Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a repeated start condition is detected.</p> <p>"1" is read when a read modify write (RMW) instruction is used.</p>

Table 15.21-4 Functional Description of Each Bit of I²C Bus Status Register (3 / 3)

Bit name		Function
bit1	SPC: Stop condition confirmation bit	<p>This bit indicates that a stop condition has been detected in master or slave mode.</p> <p>Setting conditions for the SPC bit:</p> <ul style="list-style-type: none"> (1) A stop condition is detected during slave or master mode operation. (2) An arbitration lost condition is generated when a stop condition is generated in master mode. <p>Reset conditions for the SPC bit:</p> <ul style="list-style-type: none"> (1) "0" is written to this bit. (2) "1" is written to the MSS bit. (3) The I²C interface is disabled (EN bit = 0). <p>Writing "1" to this bit is invalid.</p> <p>Note:</p> <p>Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a stop condition is detected.</p> <p>"1" is read when a read modify write (RMW) instruction is used.</p>
bit0	BB: Bus state bit	<p>This bit indicates the bus state.</p> <p>Setting condition for the BB bit:</p> <p>"L" is detected at SDA or SCL of the I²C bus.</p> <p>Reset conditions for the BB bit:</p> <ul style="list-style-type: none"> (1) A stop condition is detected. (2) The I²C interface is disabled (EN bit = 0). (3) A bus error is detected (BER bit = 1).

MB91470/480 Series**15.21.4 Serial Status Register (SSR)**

The serial status register (SSR) checks the transmission/reception status.

■ **Serial Status Register (SSR)**

Figure 15.21-5 shows the bit structure of the serial status register (SSR) and Table 15.21-5 describes the function of each bit.

Figure 15.21-5 Bit Structure of Serial Status Register (SSR)

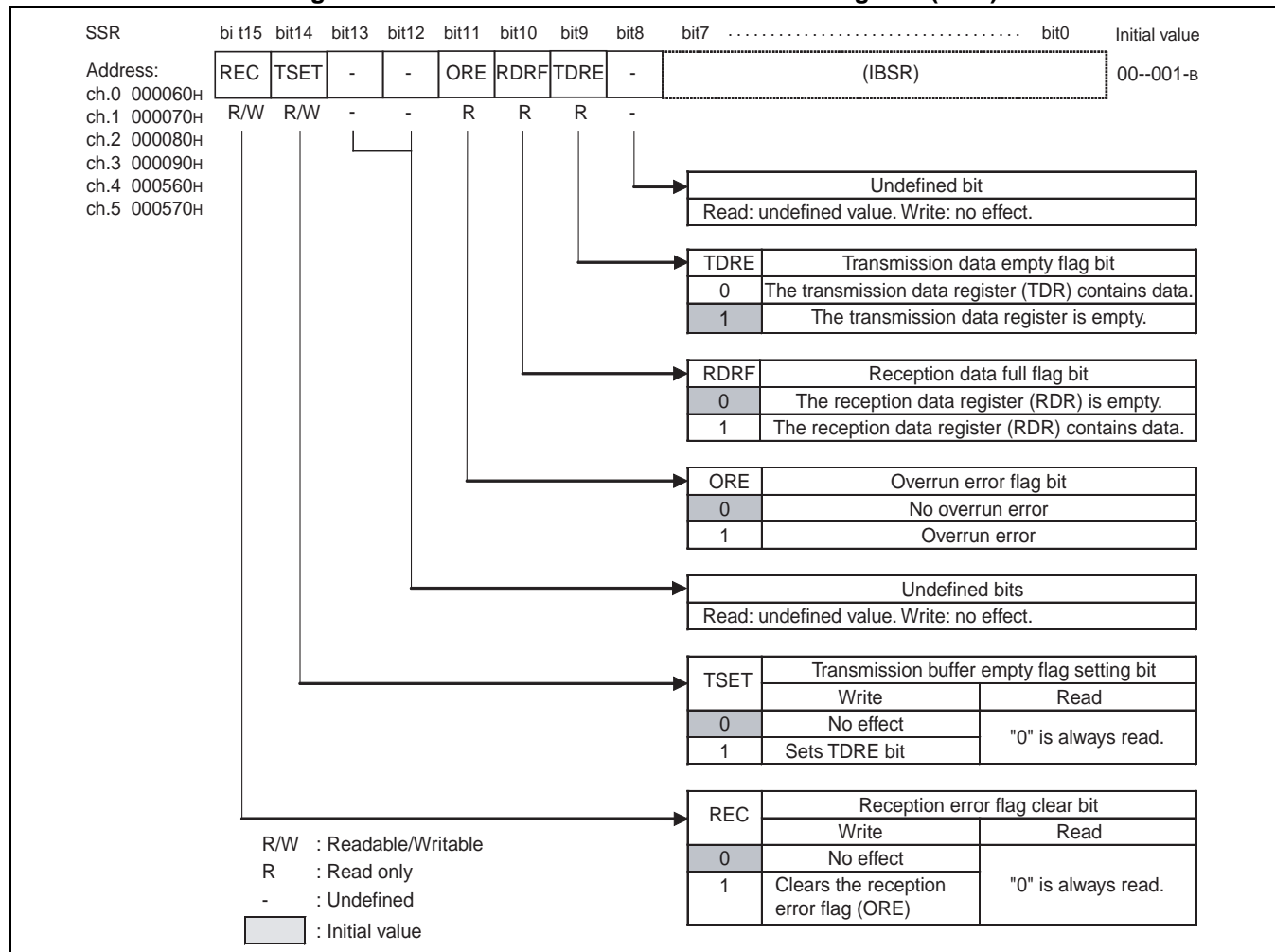


Table 15.21-5 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	<p>This bit is used to clear the ORE bit in the serial status register (SSR).</p> <ul style="list-style-type: none"> • Writing "1" clears the ORE bit. • Writing "0" has no effect. <p>Reading this bit always returns "0".</p>
bit14	TSET: Transmission buffer empty flag setting bit	<p>This bit is used to set the TDRE bit in the serial status register (SSR).</p> <ul style="list-style-type: none"> • Writing "1" sets the TDRE bit. • Writing "0" has no effect. <p>Reading this bit always returns "0".</p>
bit13, bit12	Undefined bits	<p>Read: undefined value Write: no effect</p>
bit11	ORE: Overrun error flag bit	<ul style="list-style-type: none"> • This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of the reception FIFO, the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	<ul style="list-style-type: none"> • This flag indicates the status of the reception data register (RDR). • A reception interrupt request is output when the RIE bits and the reception data flag bit (RDRF) are set to "1". • The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. • This bit is set at the falling edge of SCL in the 8th bit of data. • It is also set by a NACK response. • RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. • This bit is cleared to "0" when the reception FIFO, if used, becomes empty. • During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the reception baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO as well as the BER bit is set to "0". If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. <p>Note: NACK response: indicates that SDA of the I²C bus is at "H" during acknowledge.</p>

Table 15.21-5 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit9	TDRE: Transmission data empty flag bit	<ul style="list-style-type: none"> • This flag indicates the status of the transmission data register (TDR). • A transmission interrupt request is output when the TIE and TDRE bits are set to "1" • When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. • This bit is set when "1" is written to the TSET bit in the serial status register (SSR). This bit is used to set the TDRE bit to "1" when an arbitration lost condition or bus error is detected.
bit8	Undefined bit	Read: undefined value Write: no effect

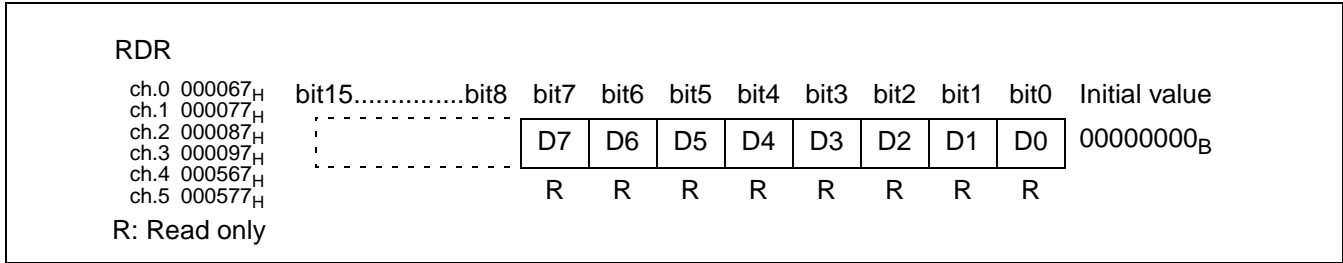
15.21.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception and transmission data registers are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

■ Reception Data Register (RDR)

Figure 15.21-6 illustrates the bit structure of the serial reception register (RDR).

Figure 15.21-6 Bit Structure of Reception Data Register (RDR)



The reception data register (RDR) is a data buffer register for serial data reception.

- A serial data signal sent to a serial data line (SDA pin) is converted through the shift register and then stored in the reception data register (RDR).
- When the first byte* is received, the least significant bit (RDR: D0) becomes the data direction bit.
- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR).
- The reception data full flag bit (SSR: RDRF) is cleared to "0" automatically, when the reception data register (RDR) is read.

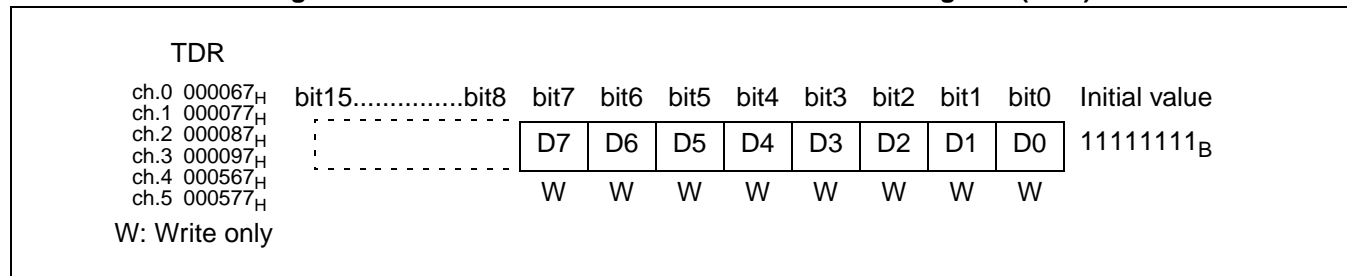
*: Indicates the data after a (repeated) start condition.

Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.

MB91470/480 Series**■ Transmission Data Register (TDR)**

Figure 15.21-7 illustrates the bit structure of the transmission data register.

Figure 15.21-7 Bit Structure of Transmission Data Register (TDR)

The transmission data register (TDR) is a data buffer register for serial data transmission.

- Data is output to the serial data line (SDA pin), based on the transmission data register (TDR) value (MSB first).
- The least significant bit (TDR: D0) becomes the data direction bit when transmitting the first byte.
- The transmission data empty flag (SSR: TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- The transmission data empty flag (SSR: TDRE) is set to "1" when transmission data is transferred to the transmission shift register.
- Write the next transmission data under the following conditions.
 - The interrupt flag (INT bit) is set to "1".
 - No bus error is occurring (BER bit = 0).
 - Acknowledge is returned as ACK response ("0" is received as acknowledgement).
- Transmission data cannot be written to the transmission data register (TDR) if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is disabled.
- Transmission data can be written up to the capacity of the transmission FIFO, even if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is used.

Note:

The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.

15.21.6 7-bit Slave Address Mask Register (ISMK)

The 7-bit slave address mask register (ISMK) determines whether each bit of a slave address should be compared.

■ 7-bit Slave Address Mask Register (ISMK)

Figure 15.21-8 shows the bit structure of the 7-bit slave address mask register (ISMK) and Table 15.21-6 describes the function of each bit.

Figure 15.21-8 Bit Structure of 7-bit Slave Address Mask Register (ISMK)

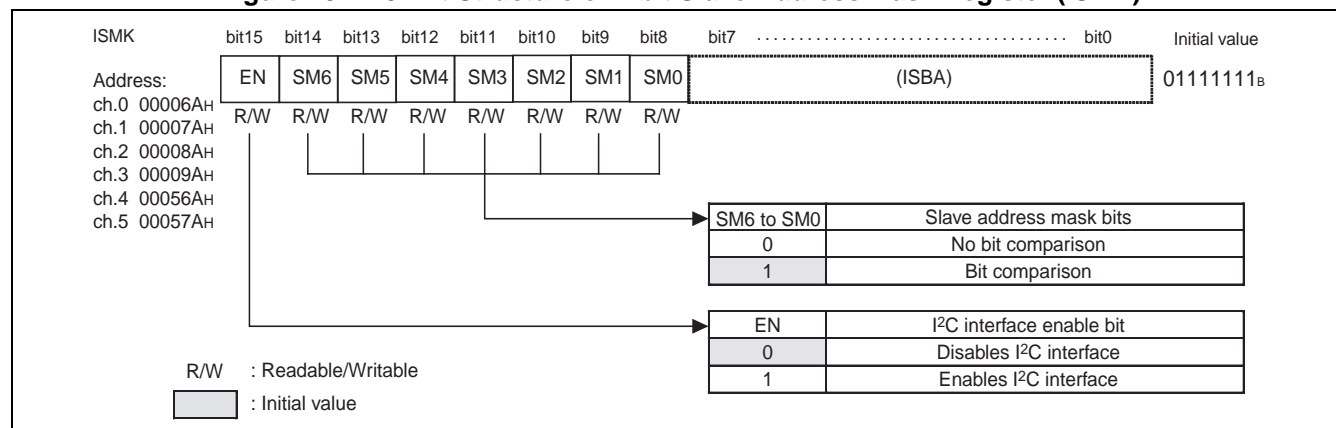


Table 15.21-6 Functional Description of Each Bit of 7-bit Slave Address Mask Register (ISMK)

Bit name		Function
bit15	EN: I ² C interface enable bit	<p>This bit is used to enable or disable the operation of the I²C interface. Setting the bit to "0" disables the operation of the I²C interface. Setting the bit to "1" enables the operation of the I²C interface.</p> <p>Note:</p> <ul style="list-style-type: none"> This bit is not cleared to "0" even when the BER bit in the IBSR register is set to "1". Set the baud rate generator when this bit is set to "0". Set a 7-bit slave address and the 7-bit slave mask register when this bit is set to "0". Setting the EN bit to "0" during transmission may generate a pulse at SDA/SCL of the I²C bus. If FIFO has been enabled, write "0" to the EN bit after disabling FIFO.
bit14 to bit8	SM6 to SM0: Slave address mask bits	<p>These bits are used to determine whether to compare the 7-bit slave address with the received address.</p> <ul style="list-style-type: none"> Bit set to "1": compared Bit set to "0": handled as matched <p>Note:</p> <ul style="list-style-type: none"> Set this register when the EN bit is "0".

MB91470/480 Series**15.21.7 7-bit Slave Address Register (ISBA)**

The 7-bit slave address register (ISBA) sets a slave address.

■ **7-bit Slave Address Register (ISBA)**

Figure 15.21-9 shows the bit structure of the 7-bit slave address register (ISBA) and Table 15.21-7 describes the function of each bit.

Figure 15.21-9 Bit Structure of 7-bit Slave Address Register (ISBA)

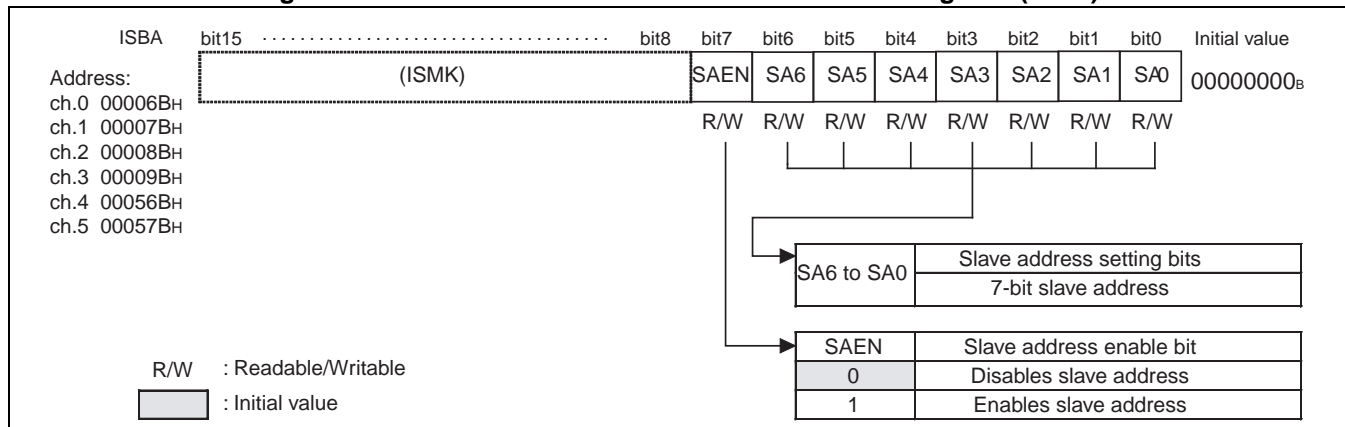


Table 15.21-7 Functional Description of Each Bit of 7-bit Slave Address Register (ISBA)

Bit name		Function
bit7	SAEN: Slave address enable bit	This bit is used to enable the detection of a slave address. Setting the bit to "0": Slave address not detected Setting the bit to "1": ISBA/ISMK setting compared with the first byte of received data
bit6 to bit0	SA6 to SA0: 7-bit slave address	If slave address detection has been enabled (SAEN = 1), the 7-bit data which is received after the detection of a (repeated) start condition will be compared with the value contained in the 7-bit slave address register (ISBA). If all the bits match, the device will operate in slave mode and output an ACK. At this point, the received slave address will be set to this register (An ACK will not be output if SAEN is set to "0"). The address bits which are set to "0" in the ISMK register are not subject to this comparison. Note: It is prohibited to set a reserved address. Set this register when the EN bit in the ISMK register is "0".

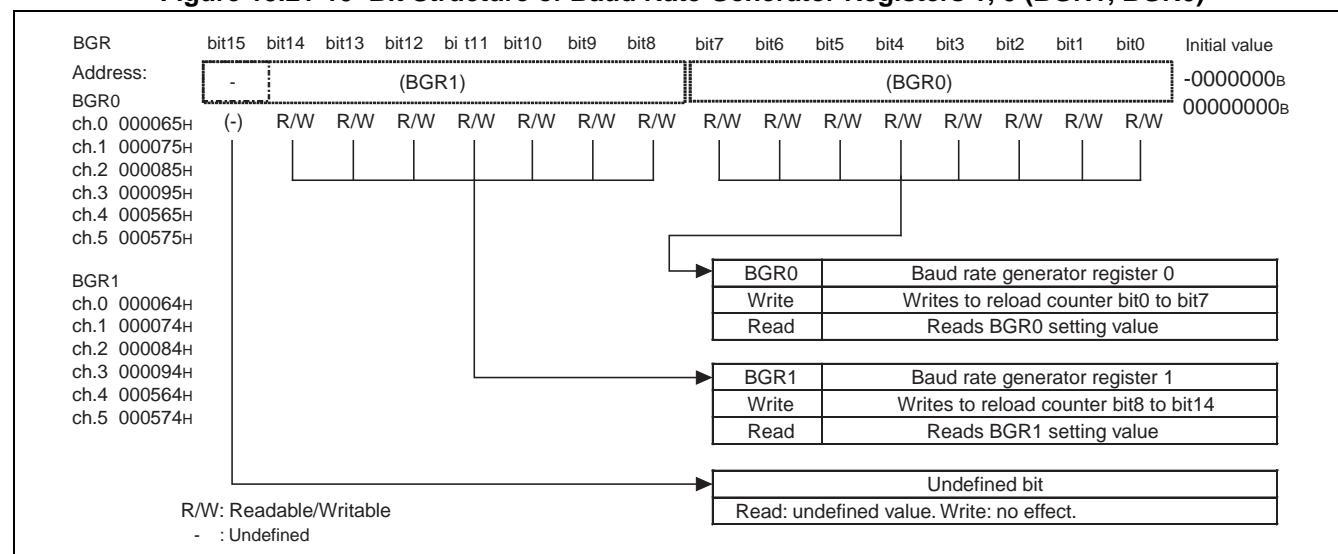
15.21.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 15.21-10 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 15.21-10 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



The baud rate generator registers are used to set a division ratio for the serial clock.

BGR0 and BGR1 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.

The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- Set the baud rate generator registers when the EN bit in the ISMK register is "0".
- Set a baud rate regardless of master or slave mode.
- Use the peripheral clock (CLKP) at 8 MHz or higher in operation mode 4 (I²C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

MB91470/480 Series**15.21.9 FIFO Control Register 1 (FCR1)**

The FIFO control register (FCR1) sets a FIFO test, selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 15.21-11 shows the bit structure of the FIFO control register 1 (FCR1) and Table 15.21-8 describes the function of each bit.

Figure 15.21-11 Bit Structure of FIFO Control Register 1 (FCR1)

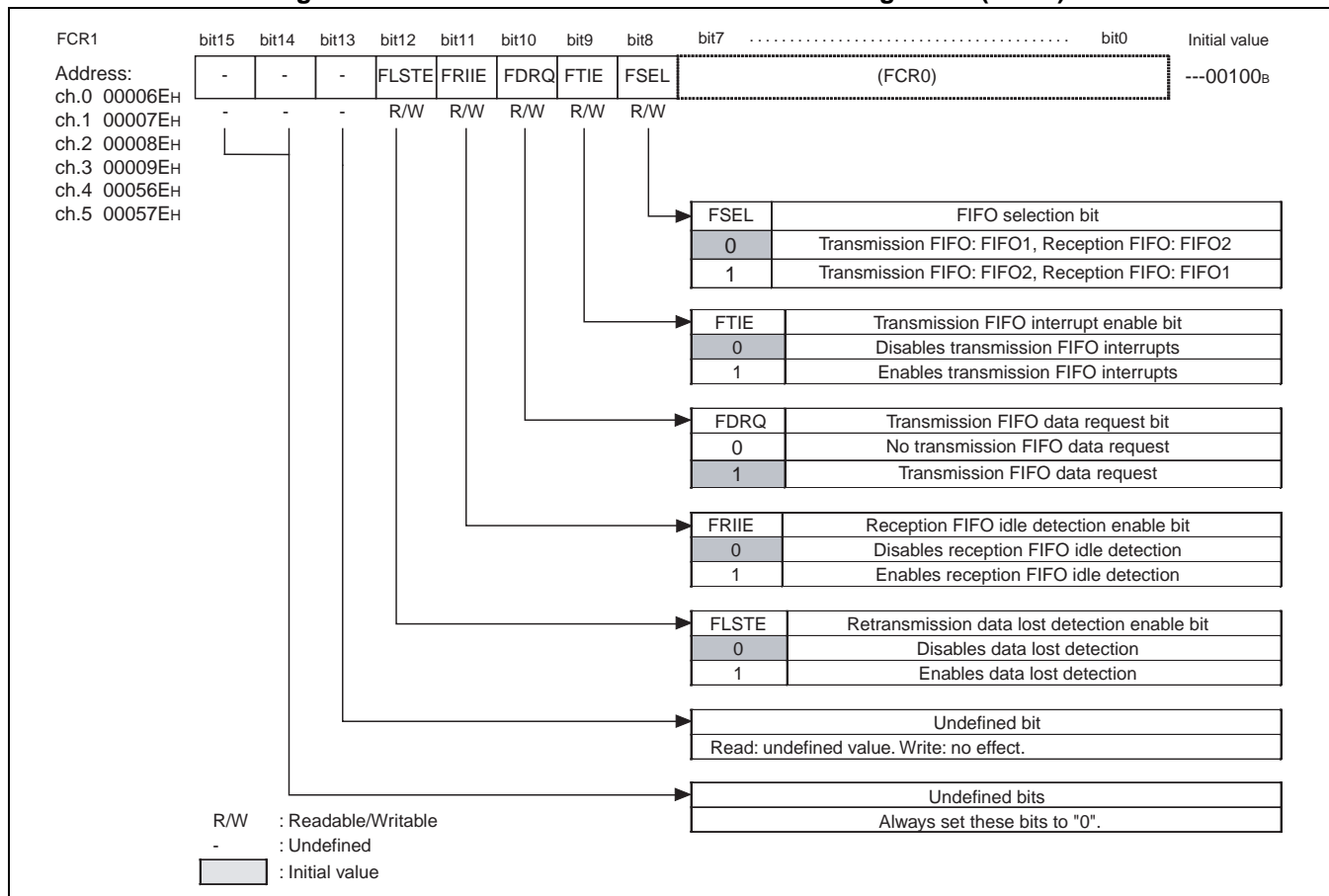


Table 15.21-8 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Undefined bits	Always set these bits to "0".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8-bit timer or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SMR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition <ul style="list-style-type: none"> • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition <ul style="list-style-type: none"> • Writing "0" to this bit • When the transmission FIFO is full. Note: It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. Note: This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FE2, FE1 = 0) and I ² C interface (ISMK:EN=0) operations beforehand.

MB91470/480 Series**15.21.10 FIFO Control Register 0 (FCR0)**

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 15.21-12 shows the bit structure of the FIFO control register 0 (FCR0) and Table 15.21-9 describes the function of each bit.

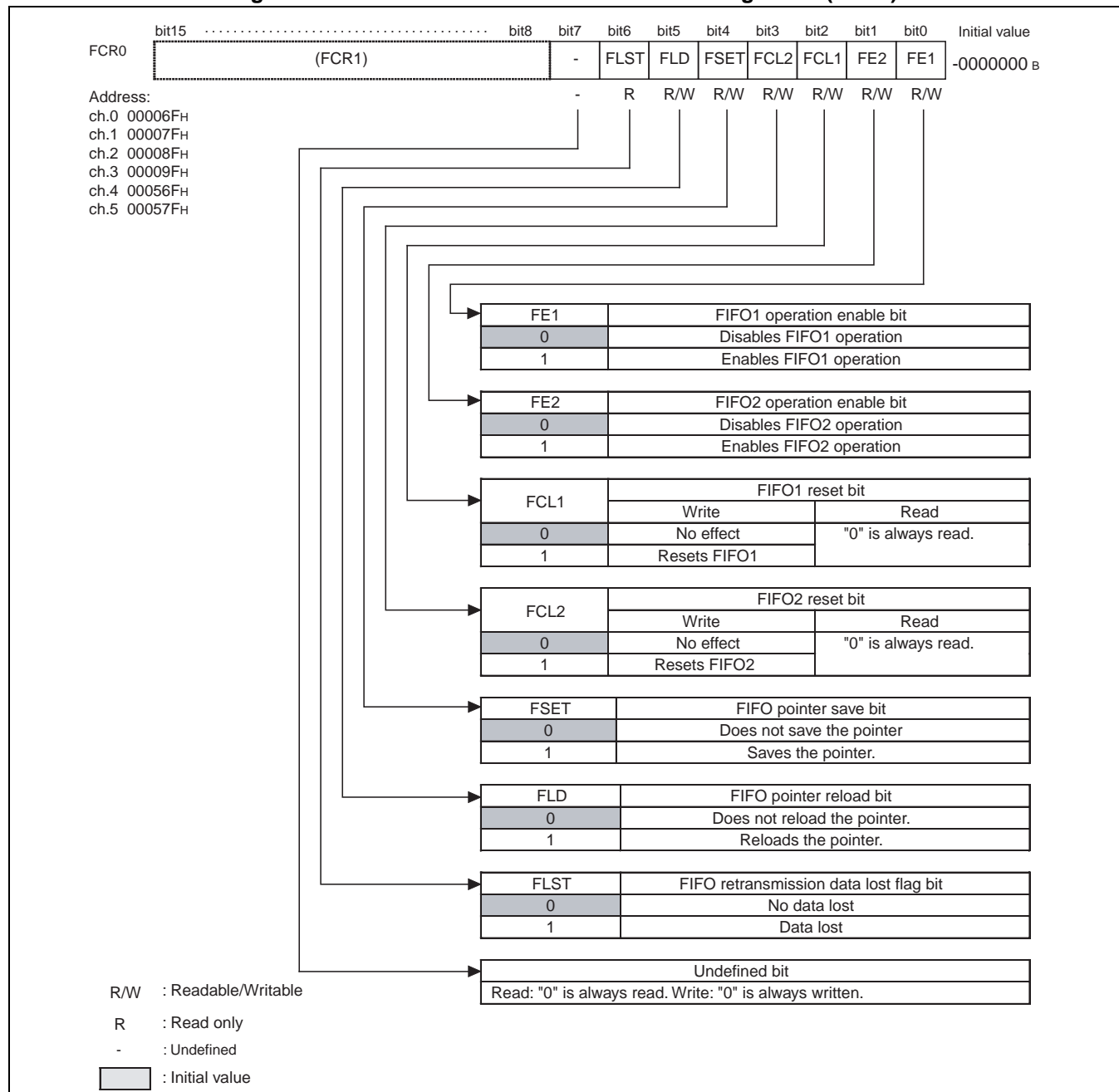
Figure 15.21-12 Bit Structure of FIFO Control Register 0 (FCR0)

Table 15.21-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 3)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	<p>This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit.</p> <p>FLST reset conditions</p> <ul style="list-style-type: none"> FIFO reset (writing "1" to FCL) Writing "1" to the FSET bit <p>Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.</p>
bit5	FLD: FIFO pointer reload bit	<p>This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs.</p> <p>The bit becomes "0" when retransmission has been set.</p> <p>Note:</p> <p>Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset.</p> <p>It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress.</p> <p>Write "1" to this bit after setting the TIE bit to "0". And then, set the TIE bit to "1" when the transmission FIFO has been enabled.</p>
bit4	FSET: FIFO pointer save bit	<p>This bit is used to save the read pointer of the transmission FIFO.</p> <p>If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs.</p> <p>Setting the bit to "1" saves the current read pointer value.</p> <p>Setting the bit to "0" has no effect.</p> <p>Note:</p> <p>Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".</p>
bit3	FCL2: FIFO2 reset bit	<p>This bit is used to reset FIFO2.</p> <p>Setting this bit to "1" initializes the internal state of FIFO2.</p> <p>Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained.</p> <p>Note:</p> <p>Disable FIFO2 before resetting it.</p> <p>Set the transmission FIFO interrupt enable bit to "0" before the reset.</p> <p>The number of valid data elements for the FBYTE2 register will become "0".</p>

Table 15.21-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 3)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable FIFO1 before resetting it. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".</p>
bit1	FE2: FIFO2 operation enable bit	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> • To use FIFO2, set this bit to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO2 is disabled, its status is retained. <p>Note: Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKC to "0". If the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". To switch this bit from "0" to "1", set the FIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO2 contains data.</p>

Table 15.21-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (3 / 3)

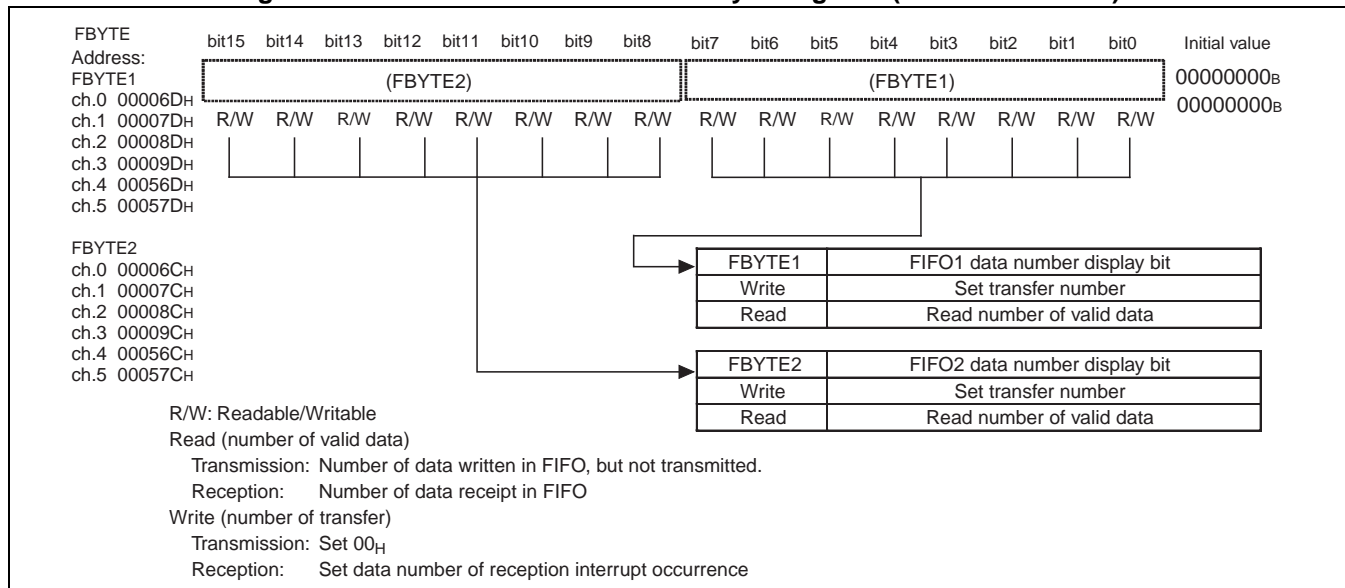
Bit name		Function
bit0	FE1: FIFO1 operation enable bit	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> • To use FIFO1, set this bit to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO1 is disabled, its status is retained. <p>Note:</p> <p>Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKC to "0".</p> <p>If the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0".</p> <p>To switch this bit from "0" to "1", set the FIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO1 contains data.</p>

MB91470/480 Series**15.21.11 FIFO Byte Register (FBYTE1/FBYTE2)**

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. This register can also be used to determine whether a reception interrupt should occur when the reception FIFO receives a specified number of data elements.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 15.21-13 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 15.21-13 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the FCR1:FSEL bit setting.

Table 15.21-10 Displaying the Number of Data Elements

FSEL	FIFO selection	Displaying the number of data elements
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is "08H".
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.

- To receive data in master operation (master reception), set the TIE bit to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. The SCL clock will be output for a specified amount of data, and then the INT bit will be set to "1". To set the TIE bit to "1", wait until the FDRQ becomes "1".

Notes:

- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
 - Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE bit is set to "0".
 - To disable the I²C interface (EN = 0) during data reception in master operation, disable transmission/reception FIFO first.
 - Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
 - Modify the register after disabling transmission/reception.
 - Read modify write (RMW) instructions cannot be used for this register.
 - Settings that will exceed the capacity of FIFO are prohibited.
-

MB91470/480 Series**15.22 Interrupts of I²C Interface**

The following sources can be used to generate interrupt requests for the I²C interface.

- After the transmission/reception of the first byte or data
- Stop condition
- Repeated start condition
- FIFO transmission data request
- Completion of FIFO reception data

■ Interrupts of I²C Interface

Table 15.22-1 shows the interrupt control bits and interrupt sources of the I²C interface.

Table 15.22-1 Interrupt Control Bits and Interrupt Sources of I²C Interface (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Reception	INT	IBCR	After transmission/reception of 1st byte ^{*1}	IBCR:INTE	Writing "0" to interrupt flag bit (IBCR:INT)
			After transmission/reception of data ^{*1}		
			Detection of bus error		
			Detection of arbitration lost condition		Writing "0" to interrupt flag bit (IBCR:INT) after reading reception data (RDR) until reception FIFO becomes empty
			Reception of the amount set by FBYTE1/FBYTE2		
			Detection of the idle state of reception for 8-bit time or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	RDRF	SSR	Detection of reserved address	SMR:RIE	Reading reception data (RDR)
			After reception of data		Reading reception data (RDR) until reception FIFO becomes empty
			Reception of the amount set by FBYTE1/FBYTE2		
	ORE	SSR	Overrun error	IBCR:CNDE	Writing "1" to reception error flag bit (SSR:REC)
	SPC	IBSR	Stop condition		Writing "0" to stop condition detection bit
	RSC	IBSR	Repeated start condition		Writing "0" to repeated start detection flag bit (IBSR:RSC)

Table 15.22-1 Interrupt Control Bits and Interrupt Sources of I²C Interface (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Transmission	TDRE	SSR	Transmission register being empty Writing "1" to transmission buffer empty flag setting bit (SSR:TSET)	SMR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission) *2
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit, or transmission FIFO being full

*1: Normal data can be transmitted or received. An interrupt does not occur when TDRE is set to "0". This function is designed to support DMA transfer.

The TDRE bit must be set to "1" before the INT flag is set, in order to generate the INT flag in data transmission/reception.

*2: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

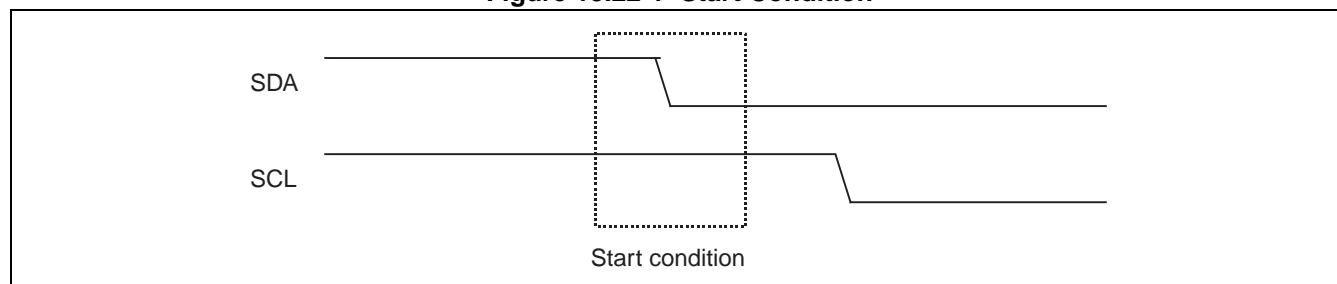
15.22.1 Operation of I²C Interface Communication

The I²C interface communication uses 2 two-way bus lines, a serial data line (SDA) and a serial clock line (SCL).

■ Start Condition for I²C Bus

The start condition for the I²C bus is shown below.

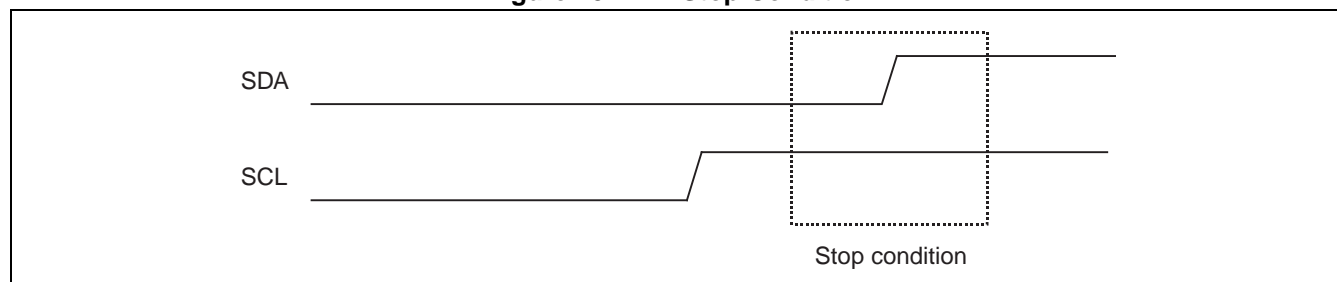
Figure 15.22-1 Start Condition



■ Stop Condition for I²C Bus

The stop condition for the I²C bus is shown below.

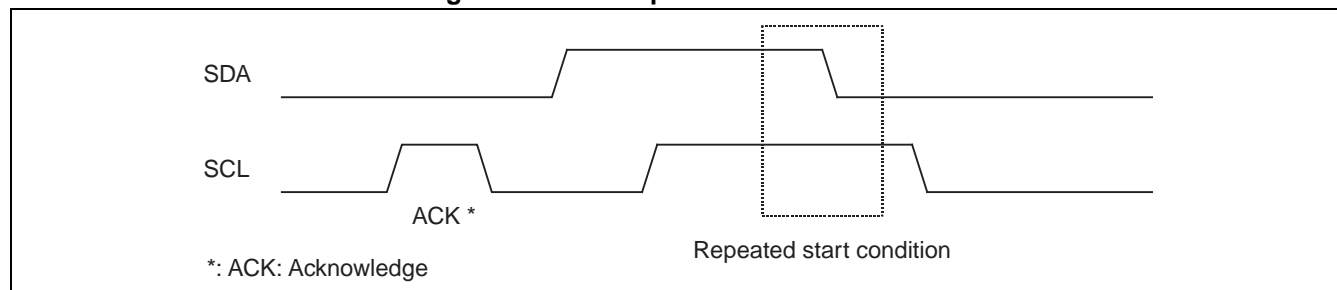
Figure 15.22-2 Stop Condition



■ Repeated Start Condition for I²C Bus

The repeated start condition for the I²C bus is shown below.

Figure 15.22-3 Repeated Start Condition



15.22.2 Master Mode

Master mode generates a start condition for the I²C bus and outputs a clock to the I²C bus. Master mode will be selected and the ACT bit in the IBCR register will be set to "1", if the MSS bit in the IBCR register is set to "1" when the I²C bus is in an idle state (SCL = "H", SDA = "H").

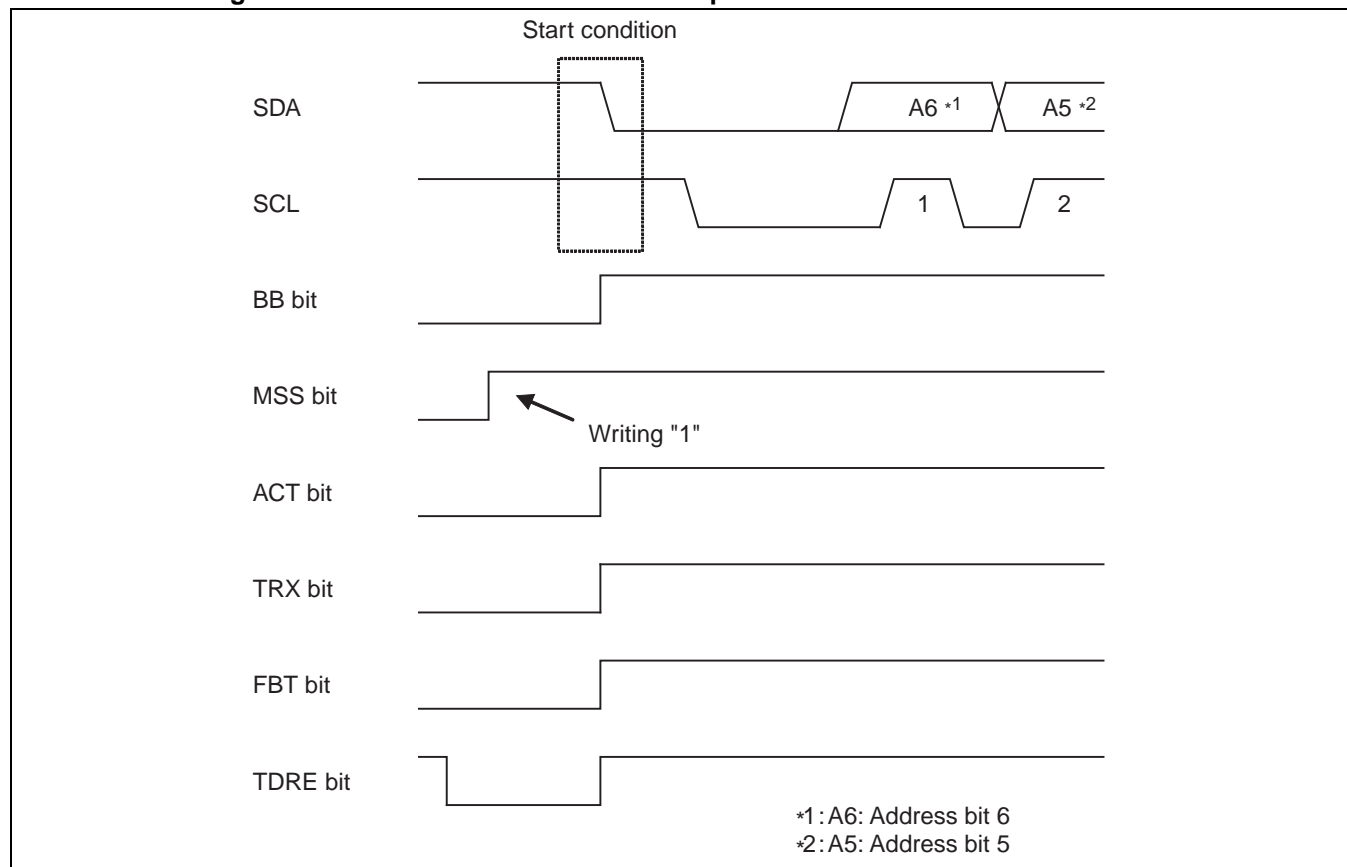
■ Generating a Start Condition

A start condition is output under the following conditions.

"1" is written to the MSS bit when SDA = "H", SCL = "H", EN = 1, and BB = 0.

Outputting a start condition to the I²C bus sets the ACT bit to "1". After that, the BB bit is set to "1", indicating that the I²C bus is in the middle of communication, once the start condition is received (see Figure 15.22-4).

Figure 15.22-4 Correlation between Output of Start Condition and Each Bit



Note:

Use the peripheral clock (CLKP) at 8 MHz or higher in operation mode 4 (I²C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

MB91470/480 Series**■ Outputting a Slave Address**

When a start condition is output, the data set in the TDR register is output from bit7 as an address. When FIFO has been enabled, the first data written in the TDR register is output. Bit0 is used as the data direction bit (R/W), and the data indicates the write direction (master →slave) when the data direction bit (R/W) is set to "0". Set an address to the TDR register before writing "1" to MSS or SCC.

Figure 15.22-5 and Figure 15.22-6 show the address and data direction output timings.

Figure 15.22-5 Address and Data Direction (When FIFO is Disabled)

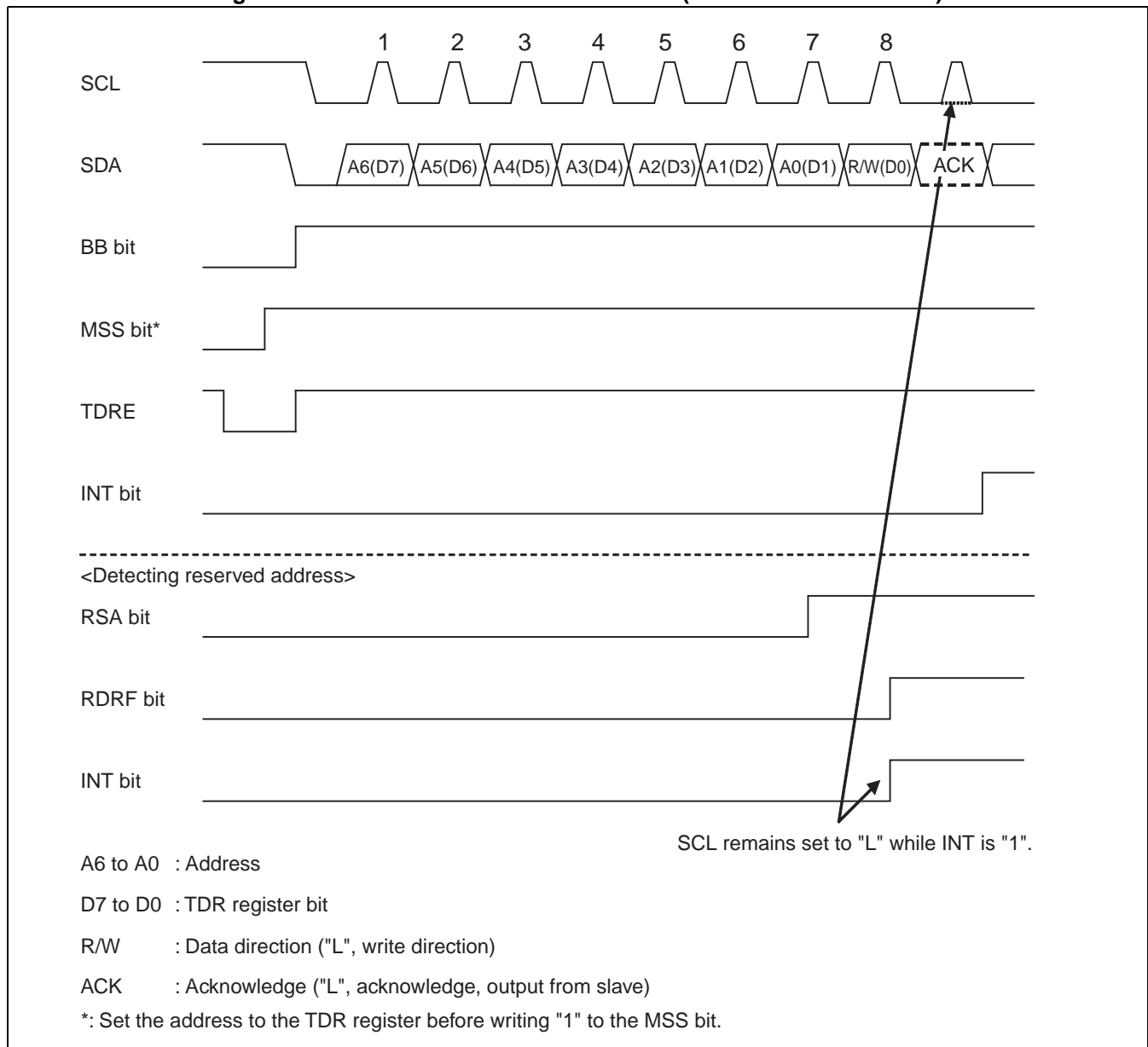
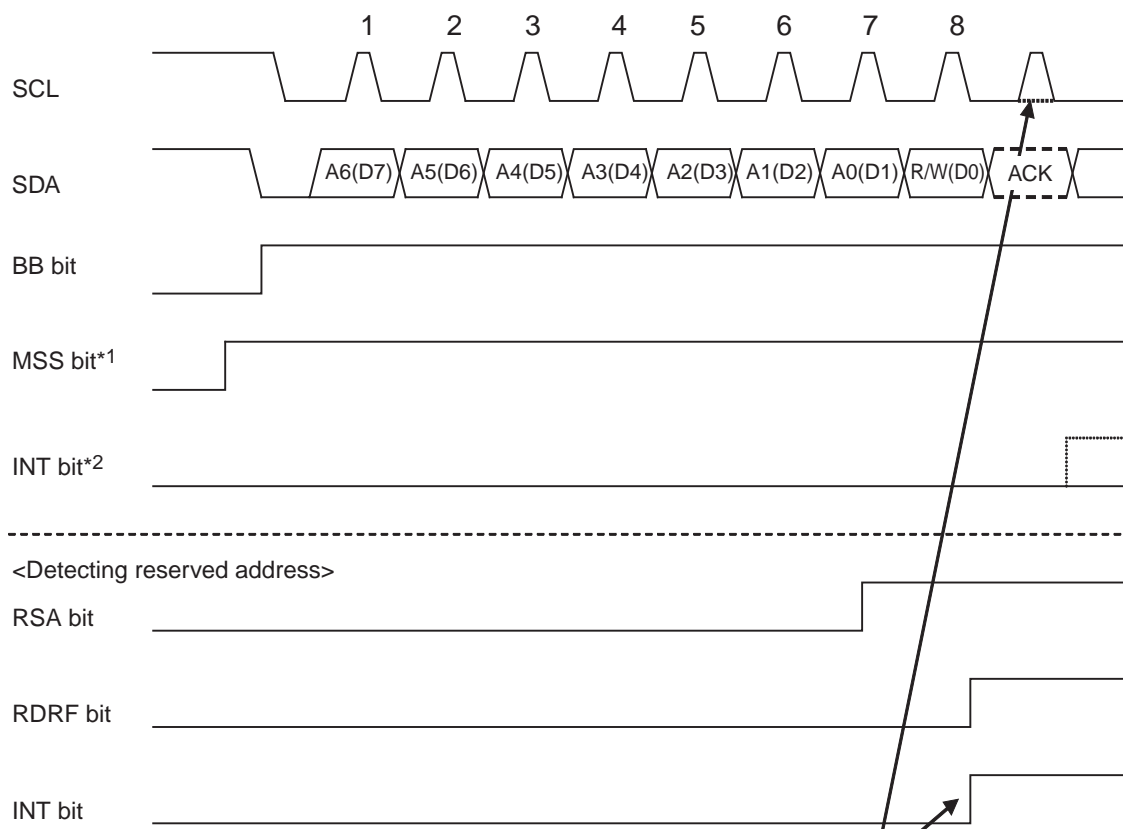


Figure 15.22-6 Address and Data Direction (When Transmission/Reception FIFO is Enabled)



A6 to A0 : Address

D7 to D0 : TDR register bit

R/W : Data direction ("L", write direction)

ACK : Acknowledge ("L", acknowledge, output from slave)

*1: Set the address to the TDR register before writing "1" to the MSS bit.

*2: The INT bit will not be set to "1" if the transmission FIFO contains data when acknowledge is "L" and R/W is "L" or if the reception FIFO contains no data when acknowledge is "L" and R/W is "H".

SCL remains set to "L" while INT is "1".

■ Receiving Acknowledge after Transmitting 1st Byte

The I²C interface receives an acknowledge from the slave when the data direction bit (R/W) is output. The following operations are performed when FIFO is enabled and disabled.

Table 15.22-2 Operations after Reception of Acknowledge (RSA Bit = 0)

Trans- mission FIFO	Reception FIFO	Trans- mission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after reception of acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait.	The INT bit is set to "1", causing a wait
				1	When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
Disabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data contained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
Enabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait.	The INT bit is set to "1", causing a wait
				1	When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
Enabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data contained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	

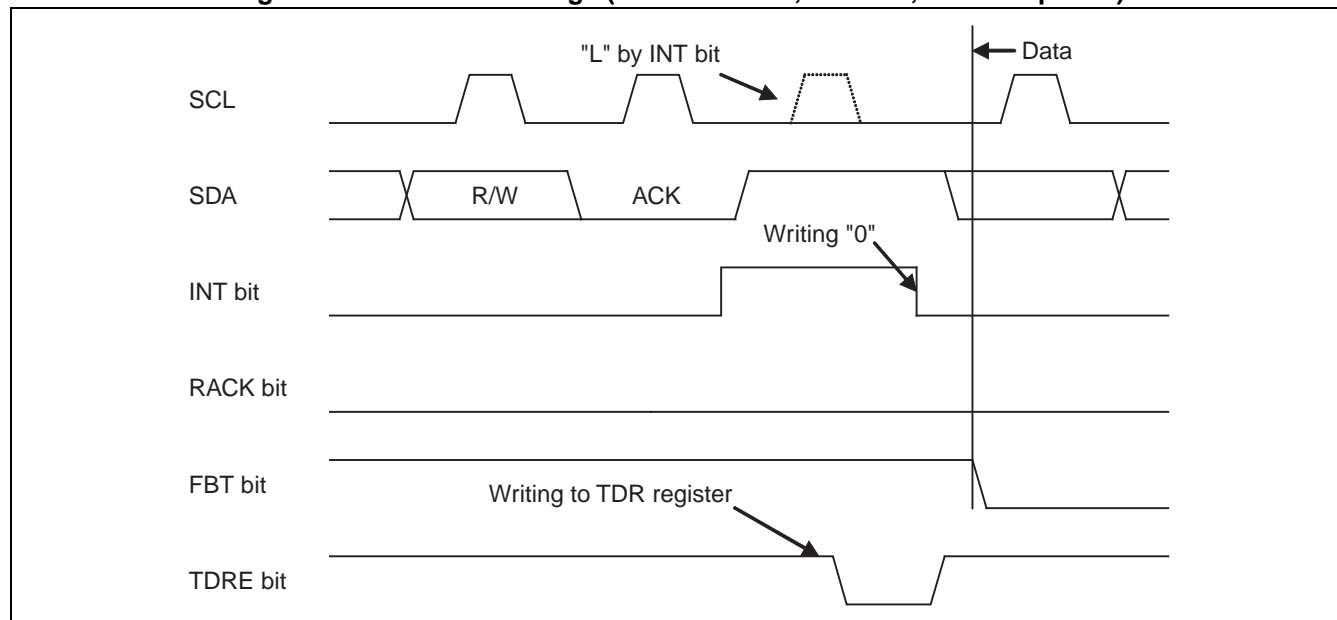
■ FIFO disabled (both transmission FIFO and reception FIFO disabled)

- The interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L", if the TDRE bit is set to "1" after the reception of an acknowledge when the RSA bit is set to "0". The wait is cancelled when "0" is written to the interrupt flag to set it to "0". If the TDRE bit has been set to "0", a clock will be generated to SCL without setting the interrupt flag to "1" when an ACK is received.
- When the RSA bit is set to "1", the interrupt flag (INT) is set to "1", causing a wait while maintaining SCL at "L", after a reserved address is received (before acknowledge). The interrupt flag will be set to "0" to cancel the wait, if the ACKE bit and transmission data are set and "0" is written to the interrupt flag after the RDR register has been read.
- The received acknowledge is set to the RACK bit. If NACK is identified when the RACK bit is checked during the wait, a stop condition or repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

■ FIFO enabled

- The following FIFO settings must be performed before the MSS bit is set to "1".
 - For transmission to the slave (data direction bit = 0), data including a slave address should be set to the transmission FIFO.
 - For reception of data from the slave (data direction bit = 1), the number of receptions should be set to the FIFO byte number register to write to the transmission data register using dummy data for the number of data elements to be received for the slave address and data direction bit.
- When the RSA bit is set to "0", the interrupt flag (INT) will not be set to "1" and data will be transmitted/received according to the data direction bit, if the received acknowledge is an ACK (no wait). If the acknowledge is a NACK, the interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L".
- The received acknowledge is stored in the RACK bit. If the acknowledge is a NACK when the RACK bit is checked during the wait, a stop condition or a repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

Figure 15.22-7 Acknowledge (FIFO Disabled, RSA = 0, ACK Response)



Address wait timings:

- RSA = 0: after receiving acknowledge
- RSA = 1: before receiving acknowledge

The above timings are not dependent on the WSEL setting.

Figure 15.22-8 Acknowledge (FIFO Disabled, RSA = 0, NACK Response)

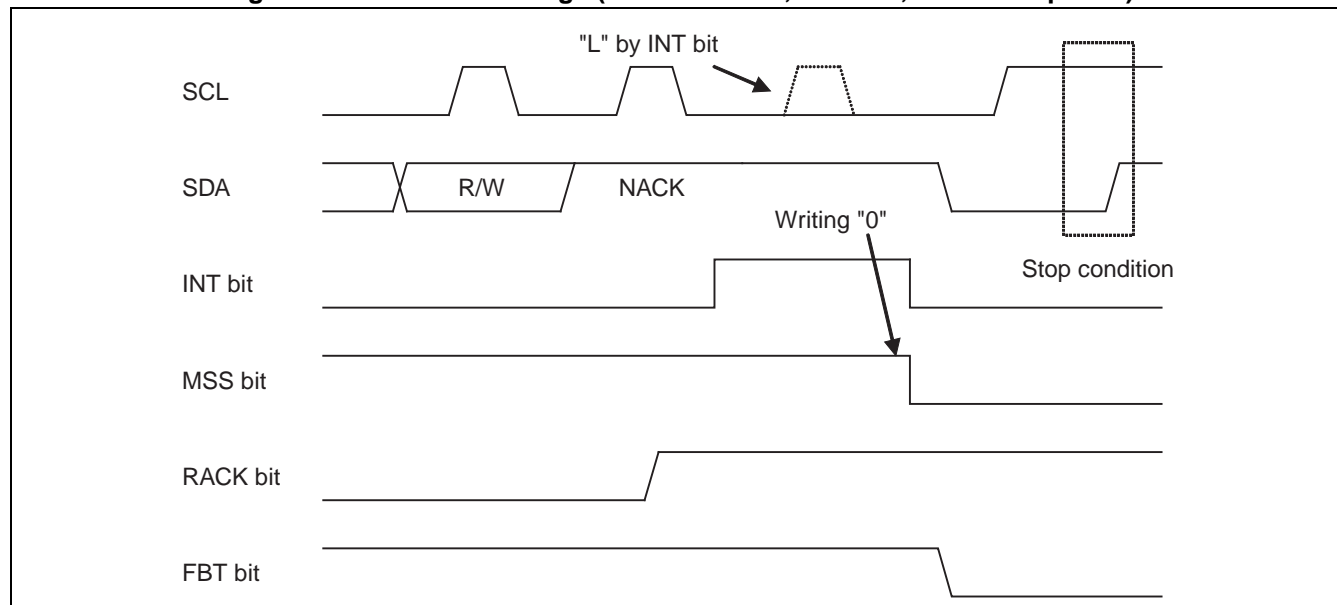


Figure 15.22-9 Acknowledge (FIFO Disabled, RSA = 1, ACK Response)

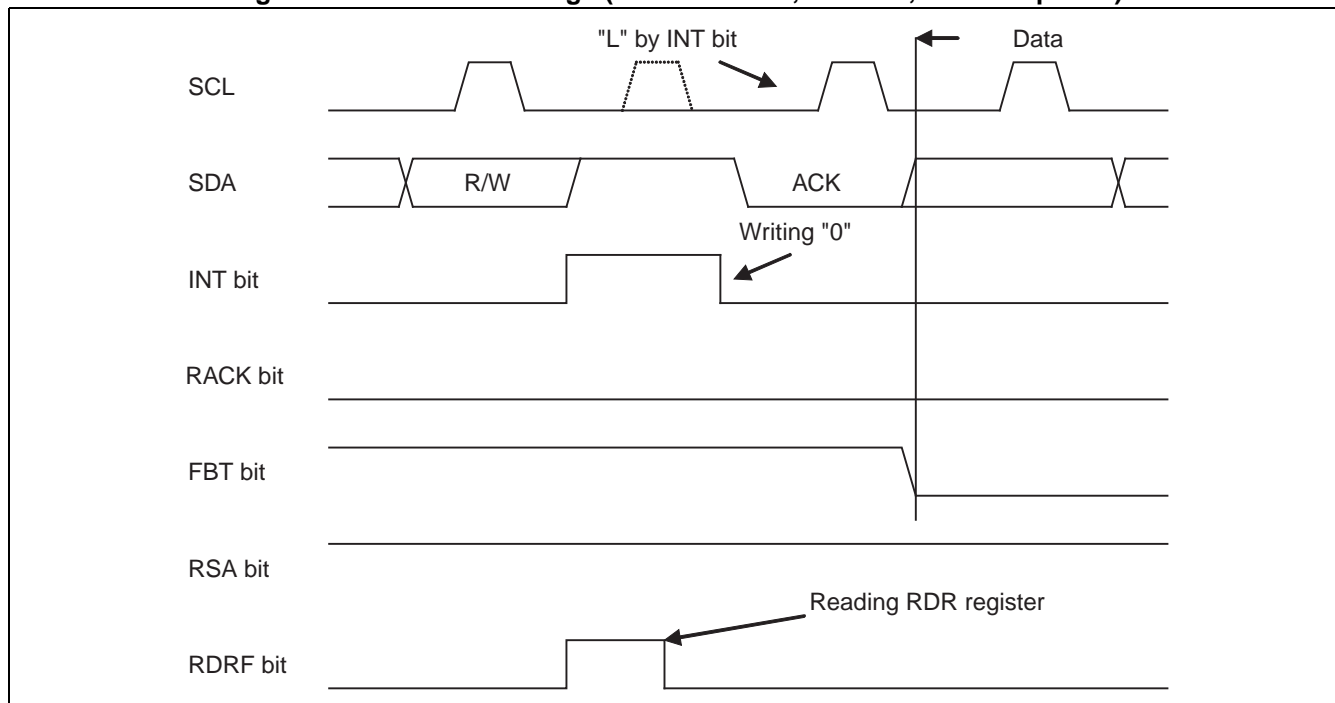


Figure 15.22-10 Acknowledge (FIFO Disabled, RSA = 1, NACK Response)

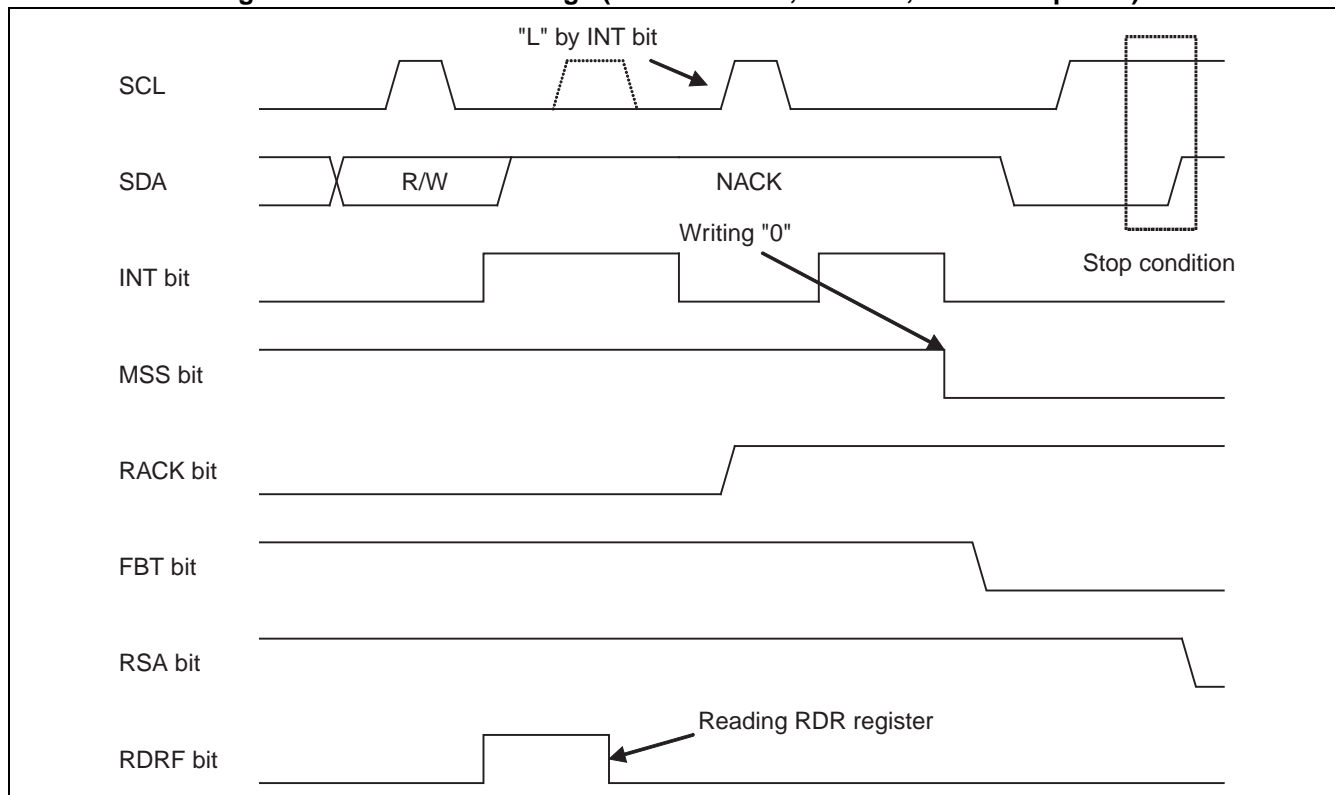
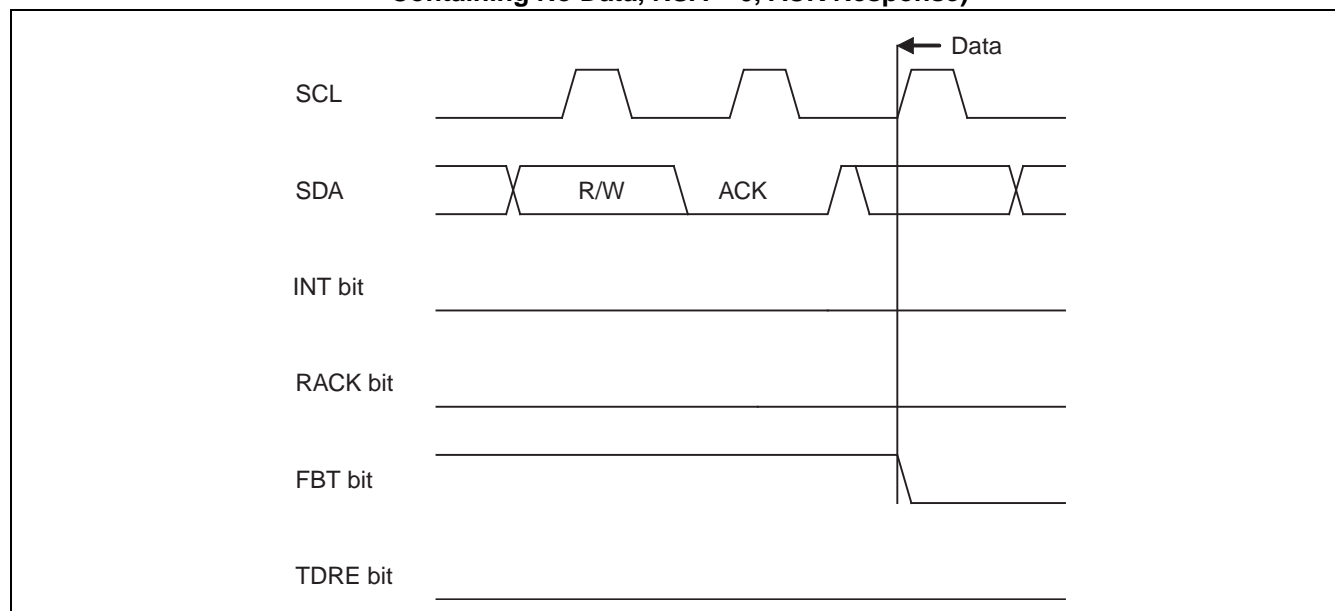


Figure 15.22-11 Acknowledge (FIFO Enabled, Transmission FIFO Containing Data, Reception FIFO Containing No Data, RSA = 0, ACK Response)



■ Master Data Transmission

Data is transmitted from the master when the data direction bit (R/W) is set to "0". The slave returns an ACK or NACK response for each byte transmitted.

The location in which a wait occurs is as follows, depending on the WSEL bit setting.

Table 15.22-3 WSEL Bit During Master Data Transmission

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after an acknowledge by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after data transmission by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).

However, if a NACK is received at times other than when a stop condition is set (MSS = 0, MAS = 1), the interrupt flag (INT) is set after an acknowledge, regardless of the WSEL setting.

An example procedure for transmitting data to the slave is shown below.

● Transmitting data to any address other than reserved address

- When transmission FIFO is disabled:
 - (1) Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
 - (2) An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
 - (3) Write the data to be transmitted to the TDR register.
 - (4) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I²C bus.
 - (5) Put the I²C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".
 - (6) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.
- When transmission FIFO is enabled:
 - (1) Write the slave address (including the data direction bit) and transmission data to the TDR register.
 - (2) Set the WSEL bit and write "1" to the MSS bit.
 - (3) If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I²C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I²C bus in a wait.
 - (4) Write "0" to the MSS bit to generate a stop condition.

● Transmitting data to reserved address

- When transmission FIFO is disabled:
 - (1) Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.
 - (2) The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.
 - (3) Read from the RDR register to check the reserved address.*
 - (4) Write the data to be transmitted to the TDR register.
 - (5) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel the wait for the I²C bus.
 - (6) Put the I²C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (4) to (6) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".
 - (7) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

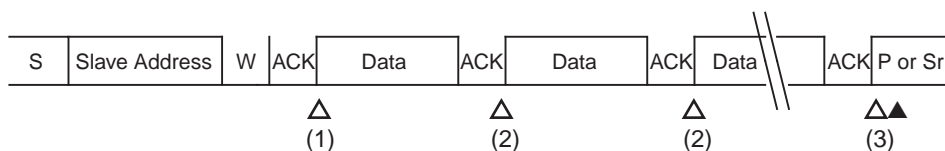
- When transmission FIFO is enabled:
 - (1) Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.
 - (2) The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.
 - (3) Read from the RDR register to check the reserved address.*
 - (4) Write all the data to be transmitted (in case that the transmission FIFO becomes full, write as much until reaching that state) to the TDR register.
 - (5) If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I²C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I²C bus in a wait.
 - (6) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

*: When the reserved address is a general call address in multi-master operation, it is necessary to confirm whether the device will operate as the master or slave for the next data by setting the ACKE and WSEL bits to "1", if the device may operate as the slave due to the generation of an arbitration lost condition.

Notes:

- To modify the IBCR register during transmission or reception, modify it when the interrupt flag (INT) is set to "1".
 - When the WSEL bit has been modified, this will be used as a condition for generating the interrupt flag (INT) for the next data.
 - If transmission data is written to the TDR register and an ACK response is detected when the TDRE is set to "1" during data transmission, the written data will be transmitted without setting the interrupt flag (INT) to "1".
 - If transmission data is written to the TDR register and an ACK is returned when the TDRE is set to "1" during data reception, only RDRF will be set to "1" without setting the interrupt flag (INT) to "1" (when the reception FIFO is enabled, and the amount set in the FBYTE1/FBYTE2 register is received).
-

Figure 15.22-12 Master Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to TDR register

(2) Interrupt generated by transmission of 1 byte + reception of acknowledge

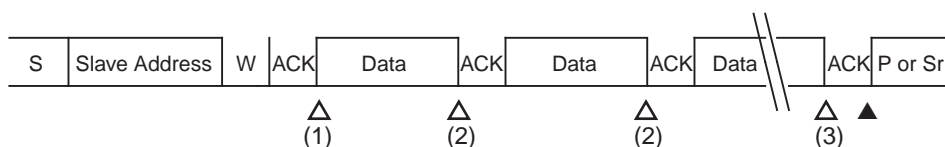
Writing "0" to INT after writing transmission data to TDR register

(3) Interrupt generated by transmission of 1 byte + reception of acknowledge

Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 15.22-13 Master Transmission Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0, ACK Response)



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to TDR register

(2) Interrupt generated by transmission of 1 byte

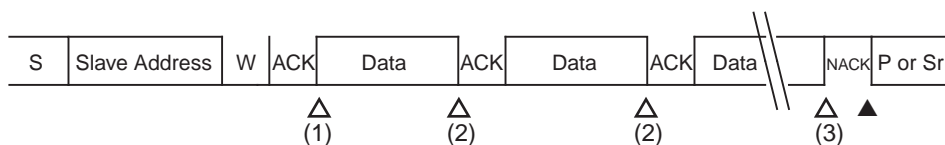
Writing "0" to INT after writing transmission data to TDR register

(3) Interrupt generated by transmission of 1 byte

Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

**Figure 15.22-14 Master Transmission Interrupt (3) - when FIFO is Disabled
(WSEL = 1, RSA = 0, NACK Response)**



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to TDR register

(2) Interrupt generated by transmission of 1 byte

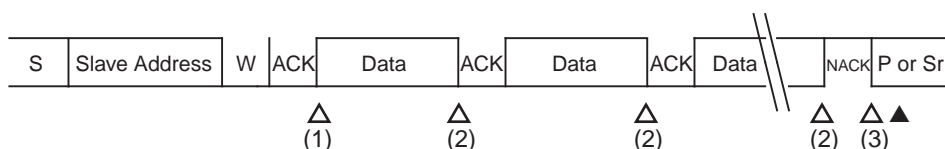
Writing "0" to INT after writing transmission data to TDR register

(3) Interrupt generated by transmission of 1 byte

Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

**Figure 15.22-15 Master Transmission Interrupt (4) - when FIFO is Disabled
(WSEL = 1, RSA = 0, NACK Response in the Middle of Operation)**



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to TDR register

(2) Interrupt generated by transmission of 1 byte

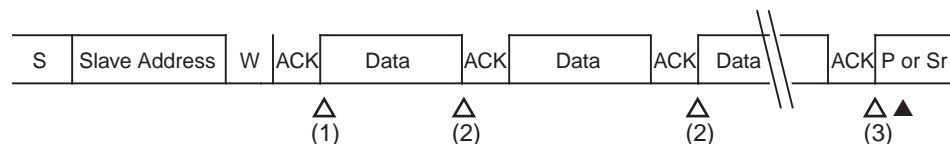
Writing "0" to INT after writing transmission data to TDR register

(3) Interrupt generated by NACK response

Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

**Figure 15.22-16 Master Transmission Interrupt (5) - when FIFO is Disabled
(WSEL = 1 → 0, RSA = 0, ACK Response)**



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to the transmission buffer

(2) Interrupt generated by transmission of 1 byte

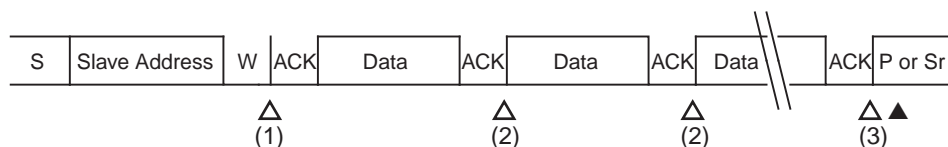
Writing "0" to WSEL and INT after writing transmission data to the transmission buffer

(3) Interrupt generated by transmission of 1 byte

Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 15.22-17 Master Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)



S : Start condition

W: Data direction bit (write direction)

P : Stop condition

Sr: Repeated start condition

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by transmission of slave address (reserved address) + transmission of direction bit + reception of acknowledge

Writing "0" to INT after writing transmission data to TDR register

(2) Interrupt generated by transmission of 1 byte + reception of acknowledge

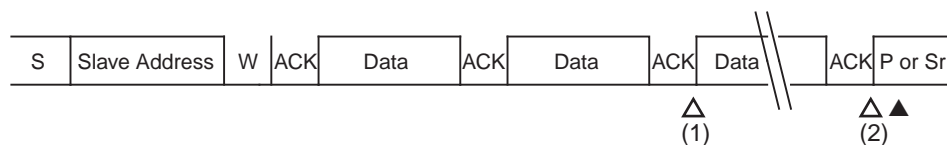
Writing "0" to INT after writing transmission data to TDR register

(3) Interrupt generated by transmission of 1 byte + reception of acknowledge

Setting MSS=0 or MSS=1, and SCC=1

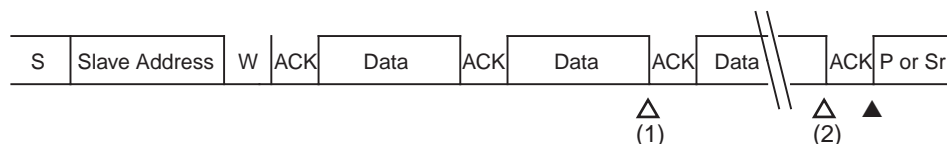
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

**Figure 15.22-18 Master Transmission Interrupt (7) - when FIFO is Enabled
(WSEL = 0, RSA = 0, ACK Response)**



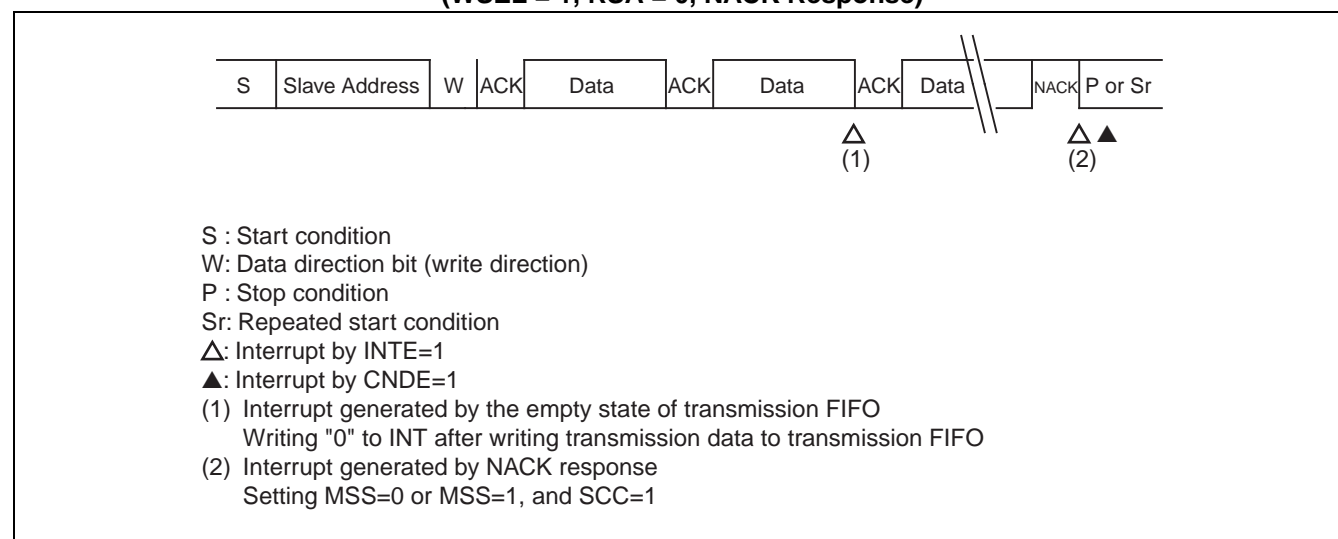
- S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
- Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty)
+ reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1

Figure 15.22-19 Master Transmission Interrupt (8) - when FIFO is Enabled (WSEL = 1, RSA = 0)



- S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty)
Setting MSS=0 or MSS=1, and SCC=1

**Figure 15.22-20 Master Transmission Interrupt (9) - when FIFO is Enabled
(WSEL = 1, RSA = 0, NACK Response)**



■ Master Data Reception

The data transmitted from the slave is received when the data direction bit (R/W) is set to "1".

When FIFO is disabled, the master will generate a wait for reception of each byte if the TDRE bit is set to "1" (INT = 1, RDRF = 1), and an ACK or NACK will be returned by the setting of the ACKE bit in the IBCR register, according to the WSEL bit. When the TDRE bit is set to "0", a wait will not be generated (INT = 0) and the next data will be received if ACK has been selected by the ACKE bit in the IBCR register, or a wait will be generated (INT = 1) if NACK has been selected.

When FIFO is enabled, the RDRF bit will be set if the same number of bytes as a specified number of bytes to be received is received. The interrupt flag is set and puts the I²C bus in a wait, when the TDRE bit is set to "1". When WSEL is set to "0", setting the TDRE bit to "1" returns a NACK and sets the interrupt flag to "1". When WSEL is set to "1", a wait is generated after the last byte has been received. Therefore, the ACKE bit should be set during that wait to clear the interrupt flag to "0", and then an ACK or NACK should be returned depending on the ACKE setting. Even when a NACK is output, it will be stored as reception data to the reception FIFO.

For interrupt-triggered waits, refer to the following section.

Table 15.22-4 WSEL Bit During Master Data Reception

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1".
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1".

An example procedure for receiving data from the slave is shown below.

- When reception FIFO is disabled:
 - (1) Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
 - (2) An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
 - (3) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I²C bus.
 - (4) Put the I²C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when WSEL is set to "0", or immediately after one byte has been received when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are received.
 - (5) Output a NACK after the reception of the last data, and set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.
- When transmission/reception FIFO is enabled:
 - (1) Set the number of receptions to the FBYTE1/FBYTE2 register.
 - (2) Write the slave address (including the data direction bit) and dummy data for the number of receptions to the TDR register.
 - (3) Write "1" to the MSS bit.
 - (4) ACK will be returned and reception will continue as long as the TDRE bit is set to "0". RDRF is set to "1" once the amount set in FBYTE1/FBYTE2 is received during that reception. When RDRF is set to "1", the RDR register is read.
 - (5) When the TDRE bit is set to "1", the interrupt flag will be set to "1" to put the I²C bus in a wait, after the output of a NACK if WSEL is set to "0", or immediately after the reception of one byte if WSEL is set to "1".
 - (6) The ACKE bit should be set to "0" when WSEL is set to "1", or the ACKE bit does not have to be set when WSEL is set to "0". Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

Notes:

- An acknowledge will be output to handle the next data according to the setting of the ACKE bit, even if an overrun error occurs when TDRE is set to "0".
 - Modify the IBCR register during transmission/reception, if necessary, when the interrupt flag (INT) is set to "1".
 - In master reception, the next data will be received with the interrupt flag (INT) still set to "0", if the TDRE bit is set to "0" when dummy data is written to the TDR register and the interrupt flag (INT) is set to "1".
 - If data is received when the reception FIFO has been enabled and WSEL is set to "0", the RDRF bit will be set to "1" upon the reception of the last bit and the interrupt flag (INT) will be set to "1" after an ACK is transmitted.
-

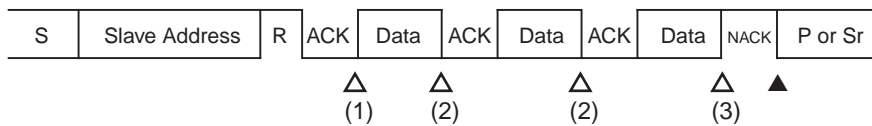
Figure 15.22-21 Master Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

- (1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
 - Interrupt cleared to "0" by writing "0" to INT
- (2) Interrupt generated by reception of 1 byte + transmission of acknowledge
 - Setting ACKE to "0" and writing "0" to INT after reading reception data
- (3) Interrupt generated by reception of 1 byte + transmission of acknowledge
 - Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

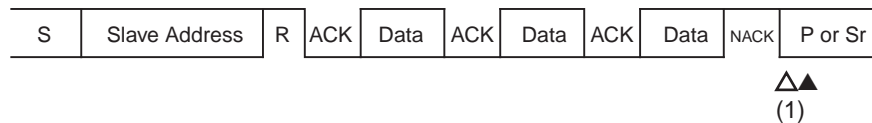
Figure 15.22-22 Master Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

- (1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
 - Interrupt cleared to "0" by writing "0" to INT
- (2) Interrupt generated by reception of 1 byte
 - Writing "0" to INT after reading reception data
- (3) Interrupt generated by reception of 1 byte
 - Setting ACKE=0, and then setting MSS=0 or MSS=1, and SCC=1 after reading reception data

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

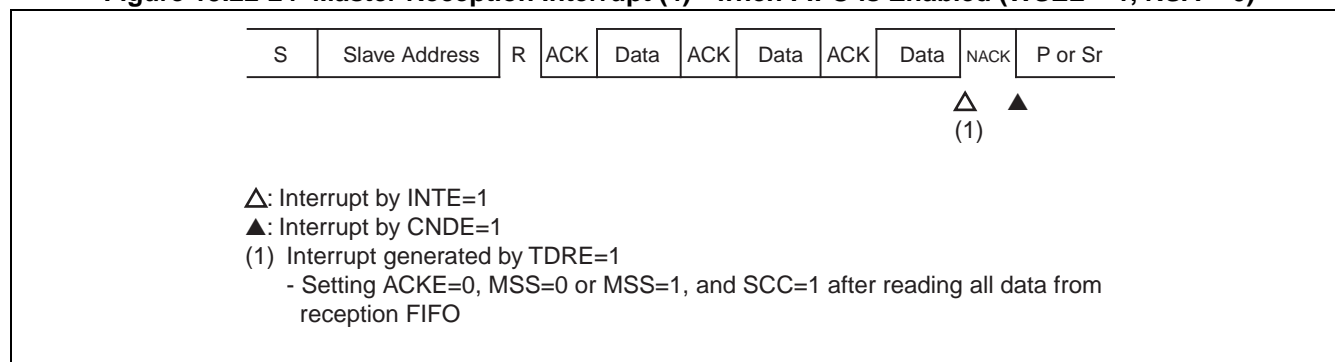
Figure 15.22-23 Master Reception Interrupt (3) - when FIFO is Enabled (WSEL = 0, ACKE = 0, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

- (1) Interrupt generated by TDRE=1
 - Setting MSS=0 or MSS=1, and SCC=1 after reading all data from reception FIFO

Figure 15.22-24 Master Reception Interrupt (4) - when FIFO is Enabled (WSEL = 1, RSA = 0)



■ Arbitration Lost Condition

When a master receives data which is different from the transmitted data due to a data collision with the data from another master, this is determined as an arbitration lost condition. Consequently, the MSS bit is set to "0" and the AL bit to "1" to allow the device to operate in slave mode.

The AL bit can be cleared to "0" under the following conditions.

- "1" is written to the MSS bit.
- "0" is written to the INT bit.
- "0" is written to the SPC bit when the AL and SPC bits are set to "1".
- The I²C interface is disabled (EN bit = 0).

When an arbitration lost condition occurs, the interrupt flag (INT) is set to "1" and the SCL of the I²C bus is set to "L", according to the setting of WSEL.

■ Wait in Master Mode

If the device is not operating in slave mode when the MSS bit is set to "1" with the BB bit set to "1", the master mode will be put in a wait as long as the BB bit remains set to "1". It will transmit a start condition once the BB bit becomes "0". The MSS and ACT bits can be used to determine whether the master mode is in a wait or not (MSS = 1, ACT = 0: wait state). To allow the device to operate in slave mode after the MSS bit is set to "1", set AL = 1, MSS = 0, and ACT = 1.

15.22.3 Slave Mode

In slave mode, the device detects a (repeated) start condition and returns an ACK when the combination of the ISBA and ISMK registers matches the received address, in order to operate in slave mode.

■ Slave Address Match Detection

When a (repeated) start condition is detected, the next 7-bit data is received as an address. Each bit of the ISBA register is compared with the corresponding bit of the received address for the bits which are set to "1" in the ISMK register. An ACK will be output if there is a match.

Table 15.22-5 Operation Immediately after Output of Acknowledge for Slave Address

Transmission FIFO	Reception FIFO	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Disabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	
Enabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Enabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	

- Reserved address detection

When the first byte matches a reserved address ("0000XXXX_B" or "1111XXXX_B"), the INT bit is set to "1" to put the I²C bus in a wait upon the reception of data from the 8th bit, whether or not the transmission/reception FIFO is enabled. At this point, ACKE is set to "1" and the INT bit is cleared when allowing the device to operate as a slave. The device will then start slave operation. When ACKE is set to "0", the device does not operate as a slave after the output of an acknowledge.

■ Data Direction Bit

The data direction bit, which determines data transmission or reception, is received after an address is received. When this bit is set to "0", this indicates transmission from the master, therefore, as a slave, the device will receive data.

■ Slave Reception

Reception is performed in slave mode when there is a slave address match and the data direction bit is set to "0". An example procedure for reception in slave mode is shown below.

- When reception FIFO is disabled:
 - (1) Set the interrupt flag (INT) to "1" to put the I²C bus in a wait after an ACK is transmitted. When the MSS, ACT and FBT bits determine that the interrupt is caused by a slave address match, set the ACKE bit to "1" and write "0" to the interrupt flag (INT) to cancel the I²C bus wait. (Refer to Table 15.22-5.)
 - (2) After 1-byte data is received, set the interrupt flag (INT) to "1" according to the WSEL setting to put the I²C bus in a wait.
 - (3) Read the data received from the RDR register, set the ACKE bit and then write "0" to the interrupt flag (INT) to cancel the I²C bus wait.
 - (4) Repeat (2) and (3) until a stop condition or a repeated start condition is detected.
- When reception FIFO is enabled:
 - (1) The interrupt flag (INT) is set to "1" to put the I²C bus in a wait when a NACK is detected or the reception FIFO becomes full. When a stop condition or a repeated start condition is detected, the SPC and RSC bits are set to "1" but not the interrupt flag (INT) (no I²C bus wait). The reception FIFO sets the RDRF bit to "1" when the value set in the FBYTE1/FBYTE2 register matches the number of data elements received. At this point, a reception interrupt will occur if the RIE bit has been set to "1".
 - (2) When the interrupt flag (INT) is set to "1", read the data received from the RDR register. After reading all the data, write "0" to the interrupt flag to cancel the I²C bus wait. Read all the data received from the RDR register and clear the SPC or RSC bit to "0", if a stop condition or a repeated start condition is detected.

Figure 15.22-25 Slave Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by ACK output due to slave address match

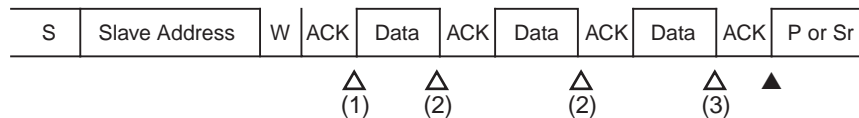
- Writing "1" to ACKE and "0" to INT

(2) Interrupt generated by reception of 1 byte + ACK response

- Writing "0" to INT after reception data is read from reception buffer

(3) Interrupt generated by reception of 1 byte + NACK response

- Writing "0" to INT after reception data is read from reception buffer

Figure 15.22-26 Slave Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by ACK output due to slave address match

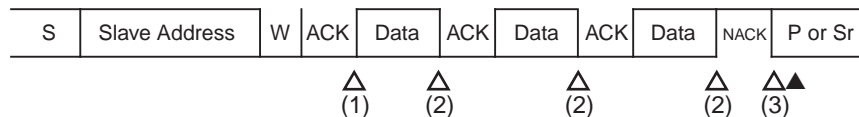
- Writing "1" to ACKE and "0" to INT

(2) Interrupt generated by reception of 1 byte

- Writing "0" to INT after reception data is read from reception buffer

(3) Interrupt generated by reception of 1 byte

- Writing "0" to INT after reception data is read from reception buffer

Figure 15.22-27 Slave Reception Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0)

△: Interrupt by INTE=1

▲: Interrupt by CNDE=1

(1) Interrupt generated by ACK output due to slave address match

- Writing "1" to ACKE and "0" to INT

(2) Interrupt generated by reception of 1 byte

- Writing "0" to INT after reception data is read from reception buffer

(3) Interrupt generated by NACK response

- Writing "0" to INT

Figure 15.22-28 Slave Reception Interrupt (4) - when Reception FIFO is Enabled (RSA = 0)

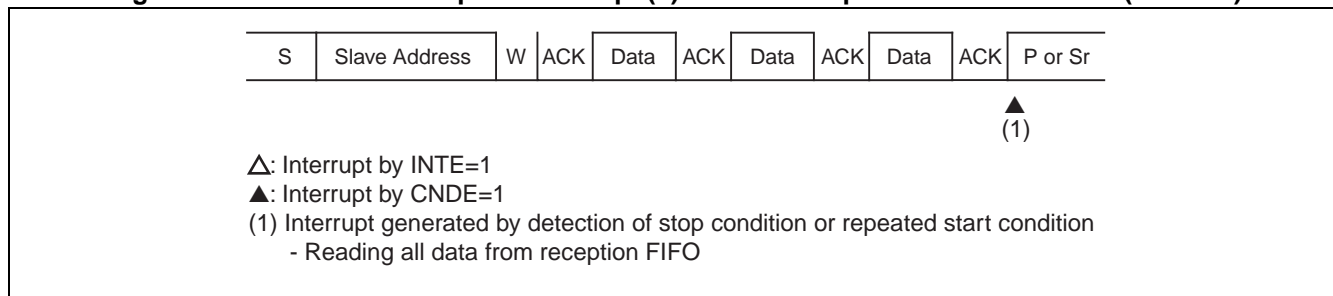


Figure 15.22-29 Slave Reception Interrupt (5) - when Reception FIFO is Enabled (RSA = 0)

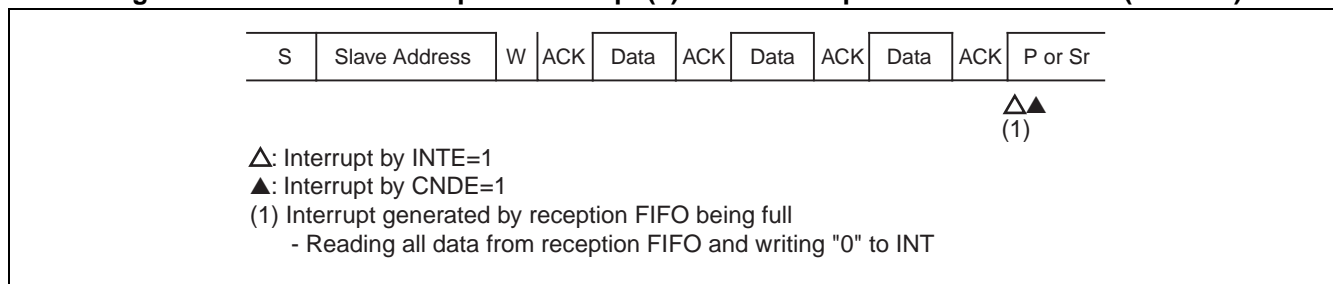
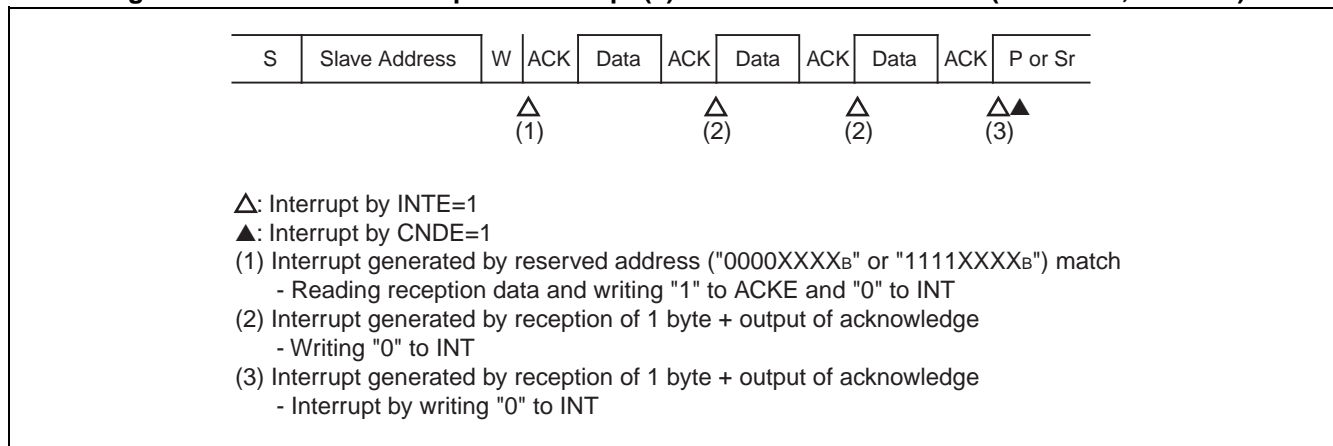


Figure 15.22-30 Slave Reception Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)



■ Slave Transmission

Transmission is performed in slave mode when there is a slave address match and the data direction bit is set to "1". When FIFO is disabled, a wait is generated by setting the interrupt flag (INT) to "1" after transmitting one byte or after returning an acknowledge, depending on the WSEL setting (see Table 15.22-5).

The RACK bit can be used to confirm the acknowledge output from the master. It indicates the end of the data reception, determining whether or not the master succeeded in the reception at a time of NACK response. An interrupt will occur to generate a wait if a NACK is detected when WSEL is set to "1".

15.22.4 Bus Error

A case where a stop condition or a (repeated) start condition is detected during data transmission/reception on the I²C bus is handled as a bus error.

■ Conditions for the Occurrence of Bus Errors

A bus error sets the BER bit to "1" under the following conditions.

- A (repeated) start condition or a stop condition is detected during the transfer of the first byte.
- A (repeated) start condition or a stop condition is detected in the 2nd bit - 9th (acknowledge) bit of data.

■ Bus Error Operation

Check the BER bit when transmission/reception sets the interrupt flag (INT) to "1". If the BER bit is set to "1", the error must be treated. The BER bit is cleared when "0" is written to the INT bit.

Although a bus error sets the INT bit to "1", the I²C bus does not enter a wait state with SCL set to "L".

15.23 Dedicated Baud Rate Generator

The dedicated baud rate generator sets a serial clock frequency.

■ Baud Rate Selection

- Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock, according to the set value.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value b: Baud rate ϕ : Peripheral clock (CLKP) frequency

Note that the set baud rate may not be generated depending on the SCL rising time of the I²C bus. In that case, the reload value must be adjusted.

(2) Example of calculation:

If the peripheral clock (CLKP) is 16MHz and the baud rate is 400kbps, the reload value will be:

Reload value:

$$V = (16 \times 1000000) / 400000 - 1 = 39$$

As a result, the baud rate is:

$$b = (16 \times 1000000) / (39 + 1) = 400 \text{ kbps}$$

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
 - Set the baud rate generator registers when the EN bit in the ISMK register is "0".
 - Use the peripheral clock (CLKP) at 8 MHz or higher in operation mode 4 (I²C mode). It is prohibited to set the baud rate generator to higher than 400kbps.
 - The reload counter stops when the reload value is set to "0".
-

■ Reload Values and Baud Rates for Different Peripheral Clock (CLKP) Frequencies

Table 15.23-1 Reload Values and Baud Rates

Baud rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32MHz
	Reload value	Reload value	Reload value	Reload value	Reload value	Reload value
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

These numerical values are based on the SCL rising time of I²C bus set to "0". If the rising is slower, the actual baud rates should also be slower than the numerical values above.

■ Functions of Reload Counters

Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock. In addition, the count value of the transmission reload counter can be read from the baud rate generator registers 1, 0 (BGR1, BGR0).

■ Starting a Count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

15.23.1 Example of I²C Flowcharts

Below are some example flowcharts for I²C communication.

■ I²C Master Reception/ Slave Transmission FIFO Communication Flow

Figure 15.23-1 Master Reception Main Settings

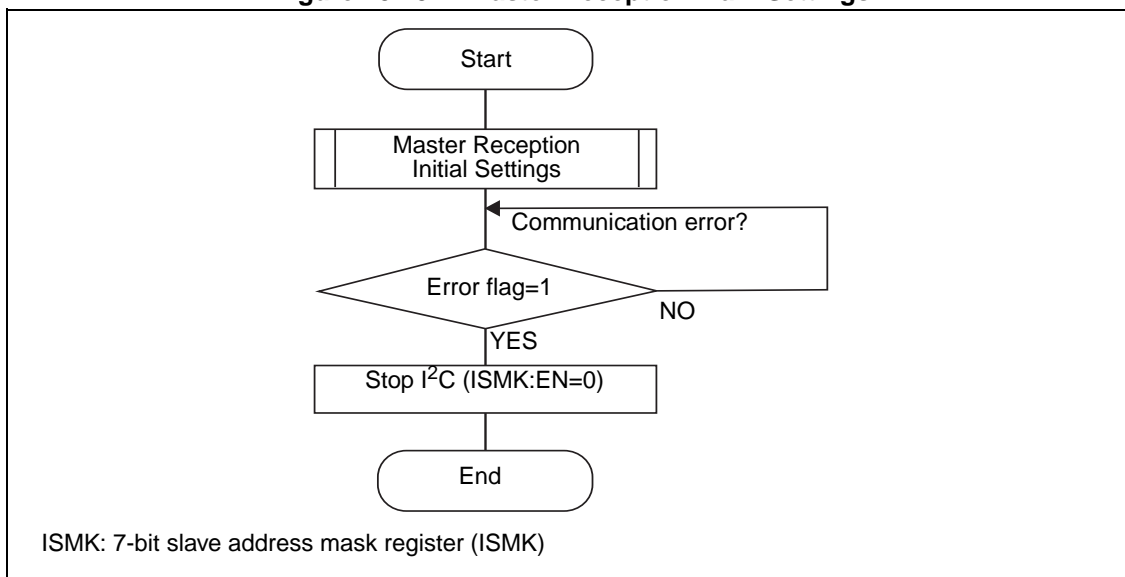


Figure 15.23-2 Master Reception Initial Settings

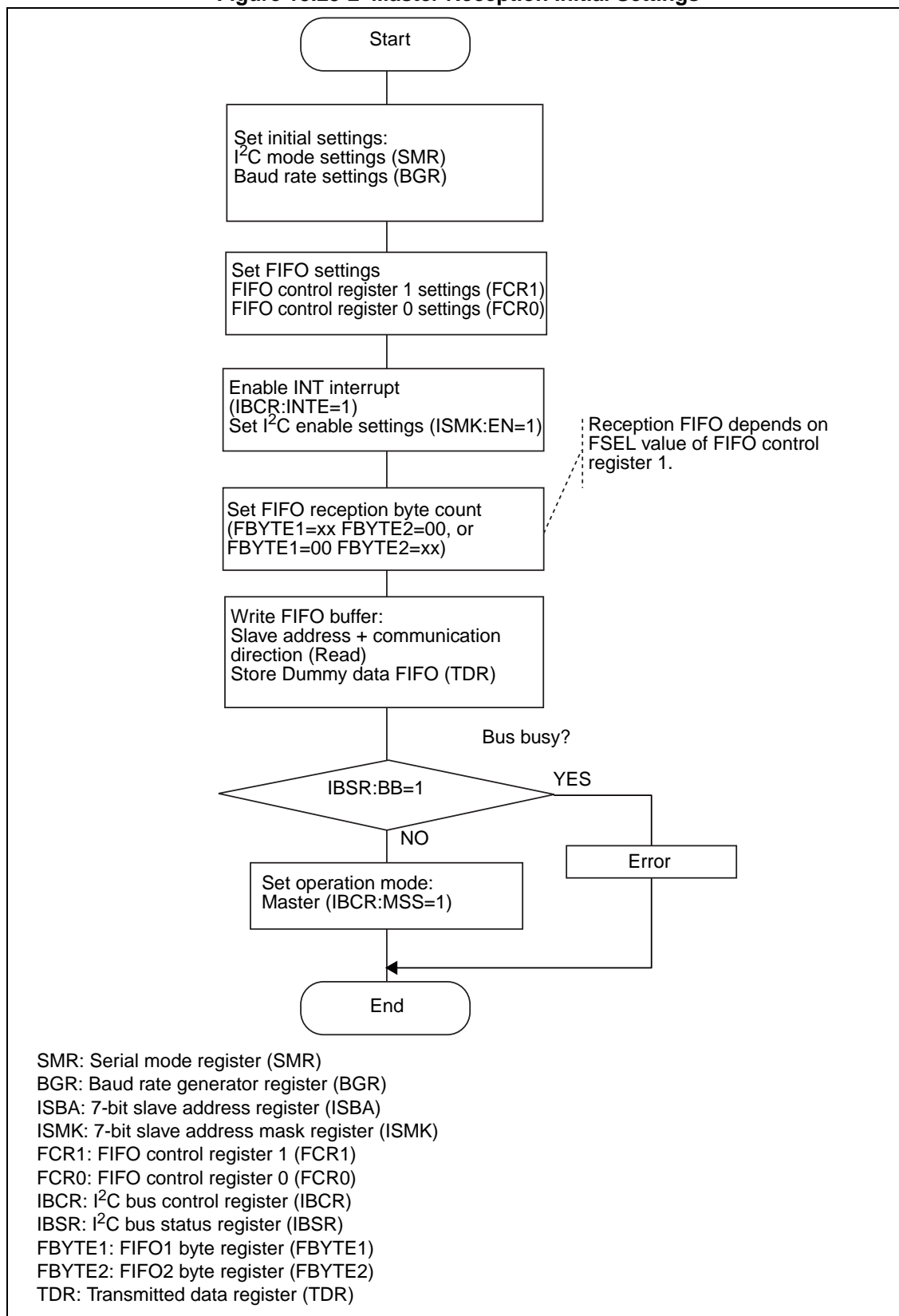


Figure 15.23-3 Master Reception Interrupt Process

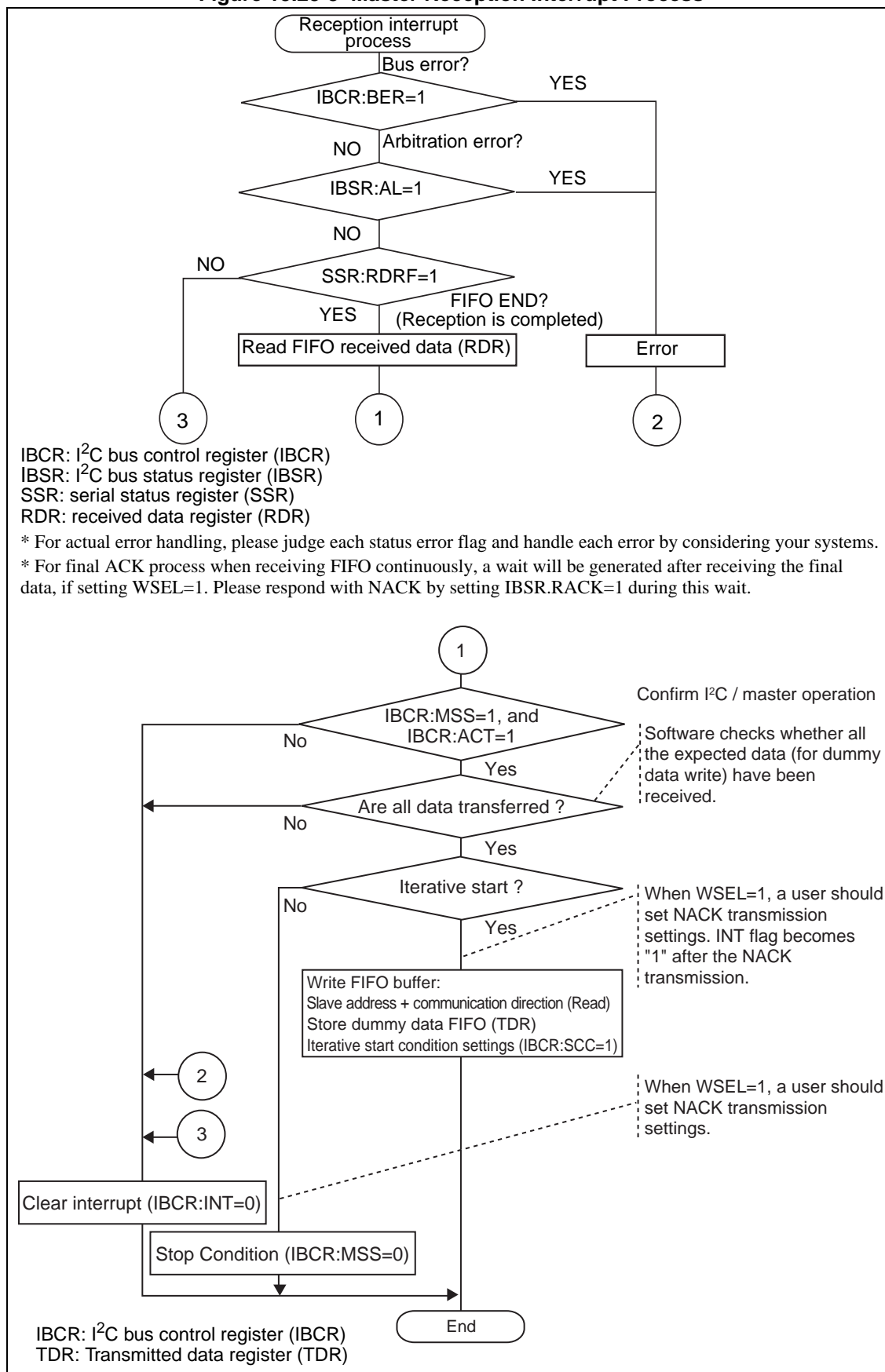


Figure 15.23-4 Slave Transmission Main Settings

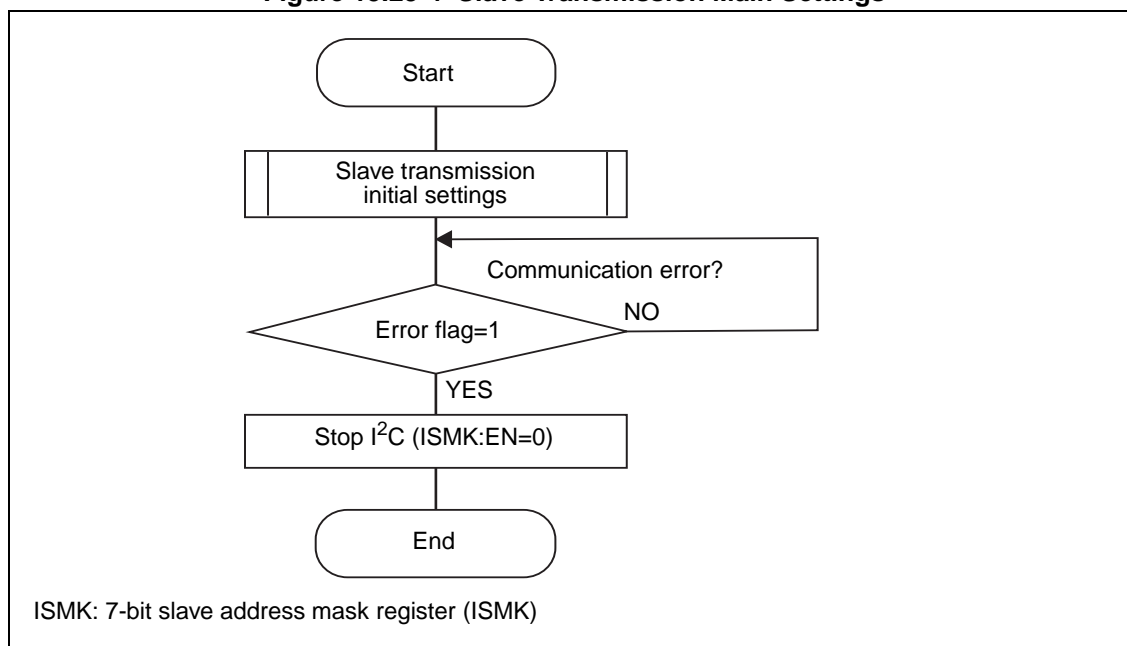


Figure 15.23-5 Slave Transmission Initial Settings

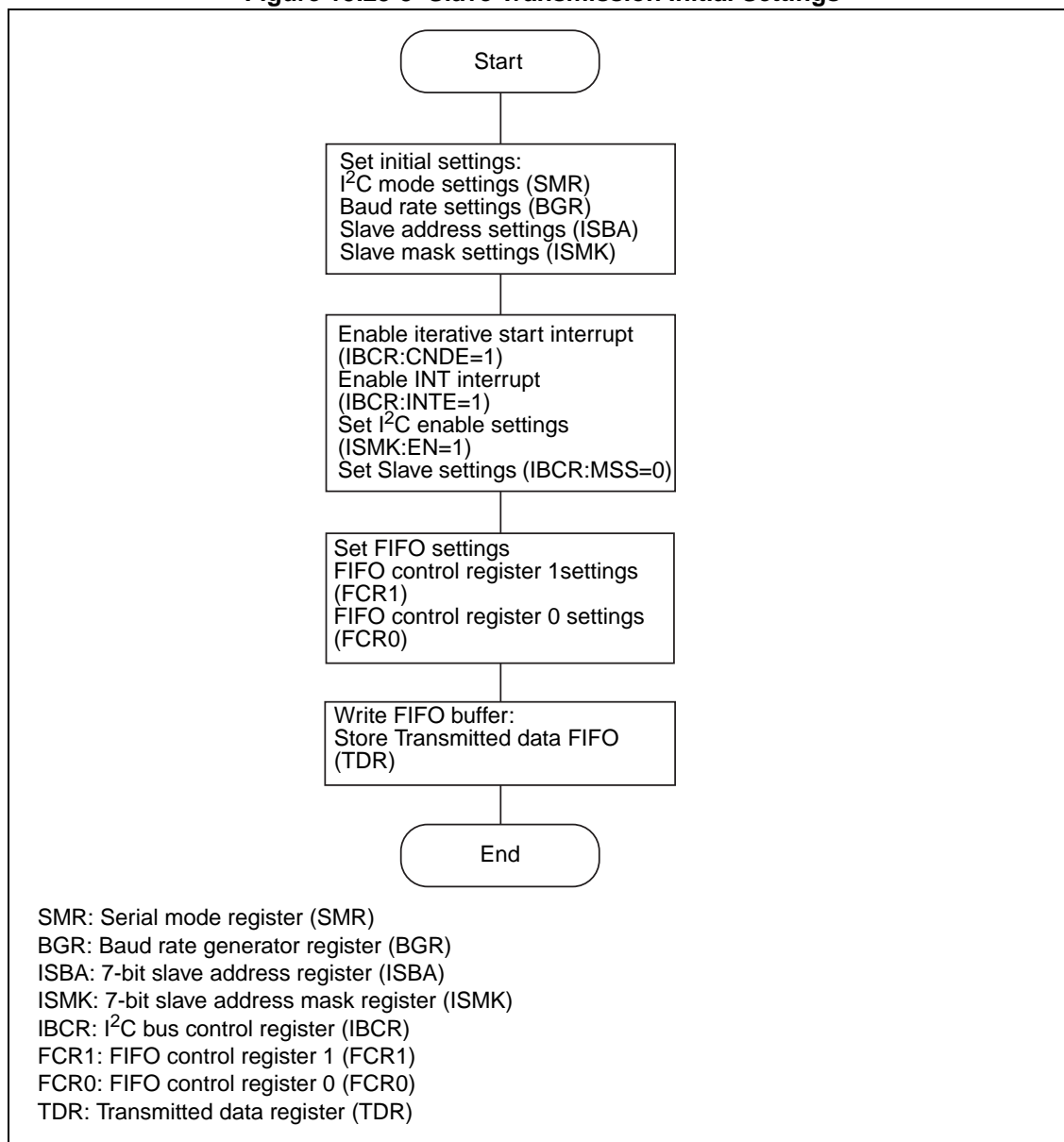
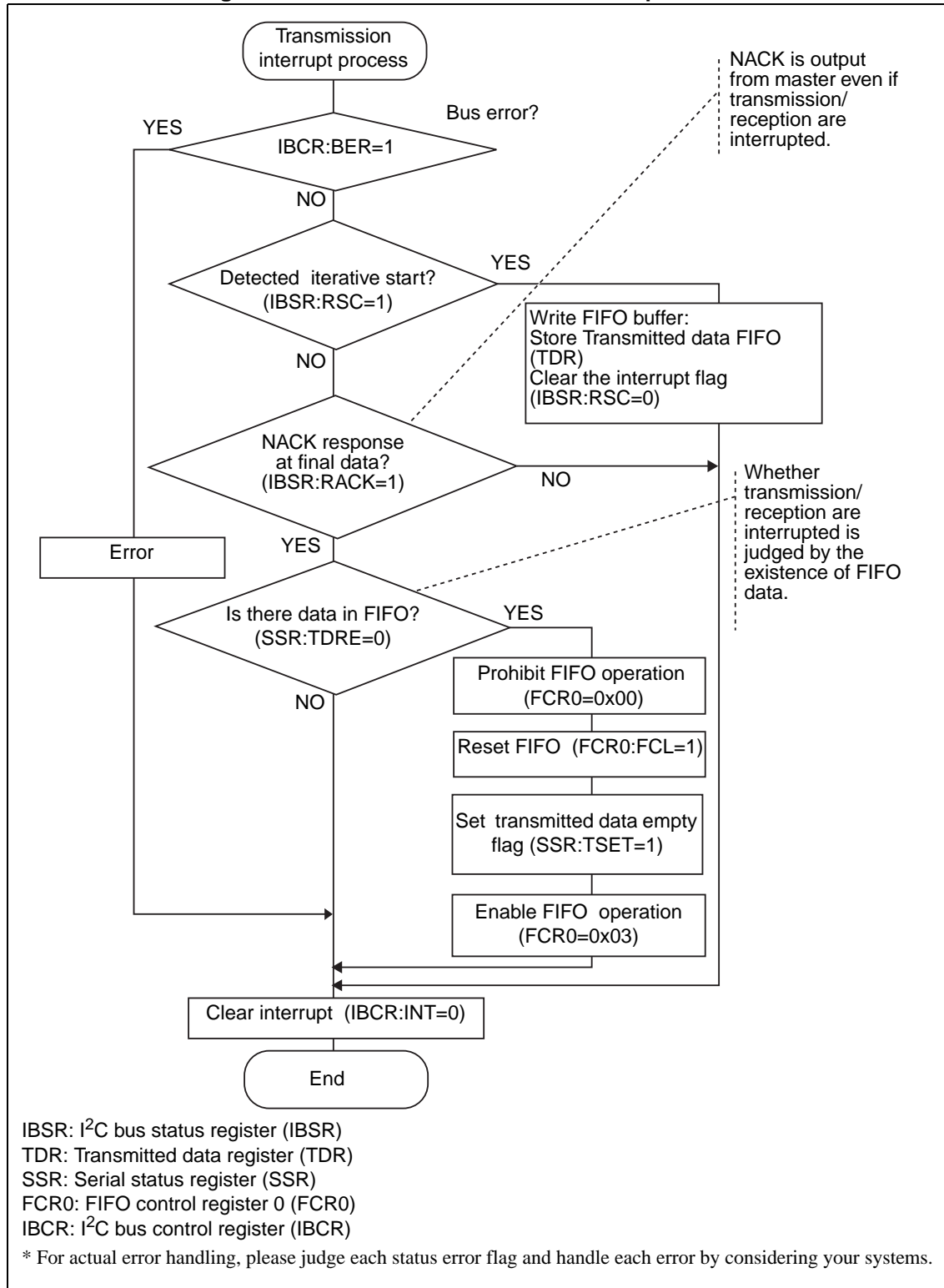


Figure 15.23-6 Slave Transmission Interrupt Process



■ I²C Master Transmission/ Slave Reception FIFO Communication Flow

Figure 15.23-7 Master Transmission Main Settings

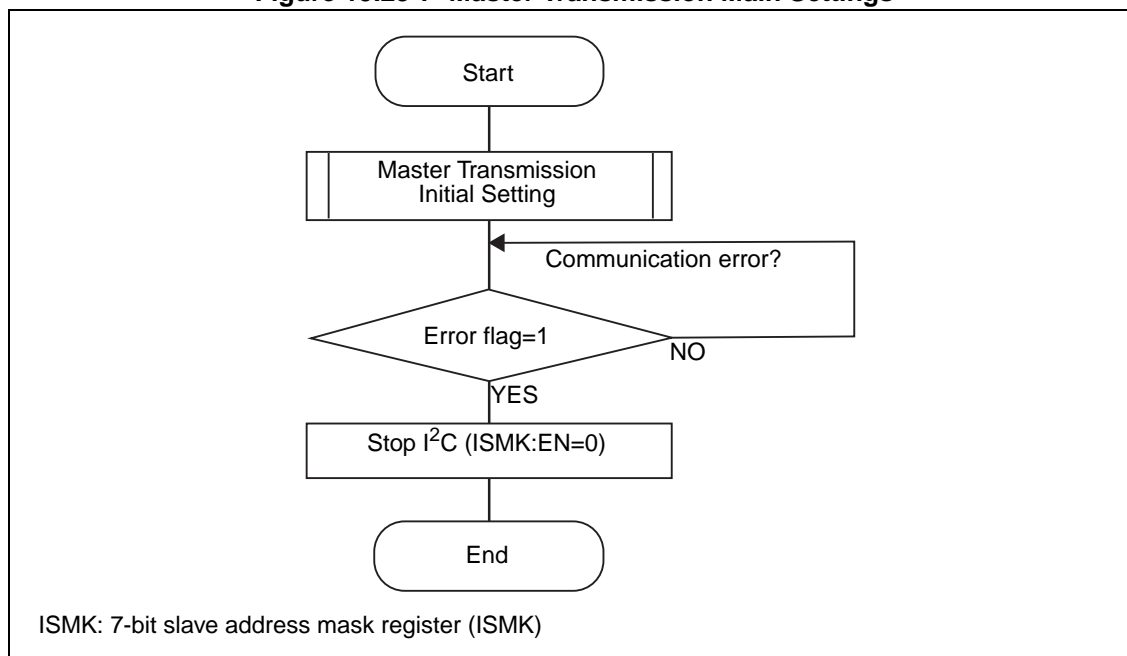


Figure 15.23-8 Master Transmission Initial Settings

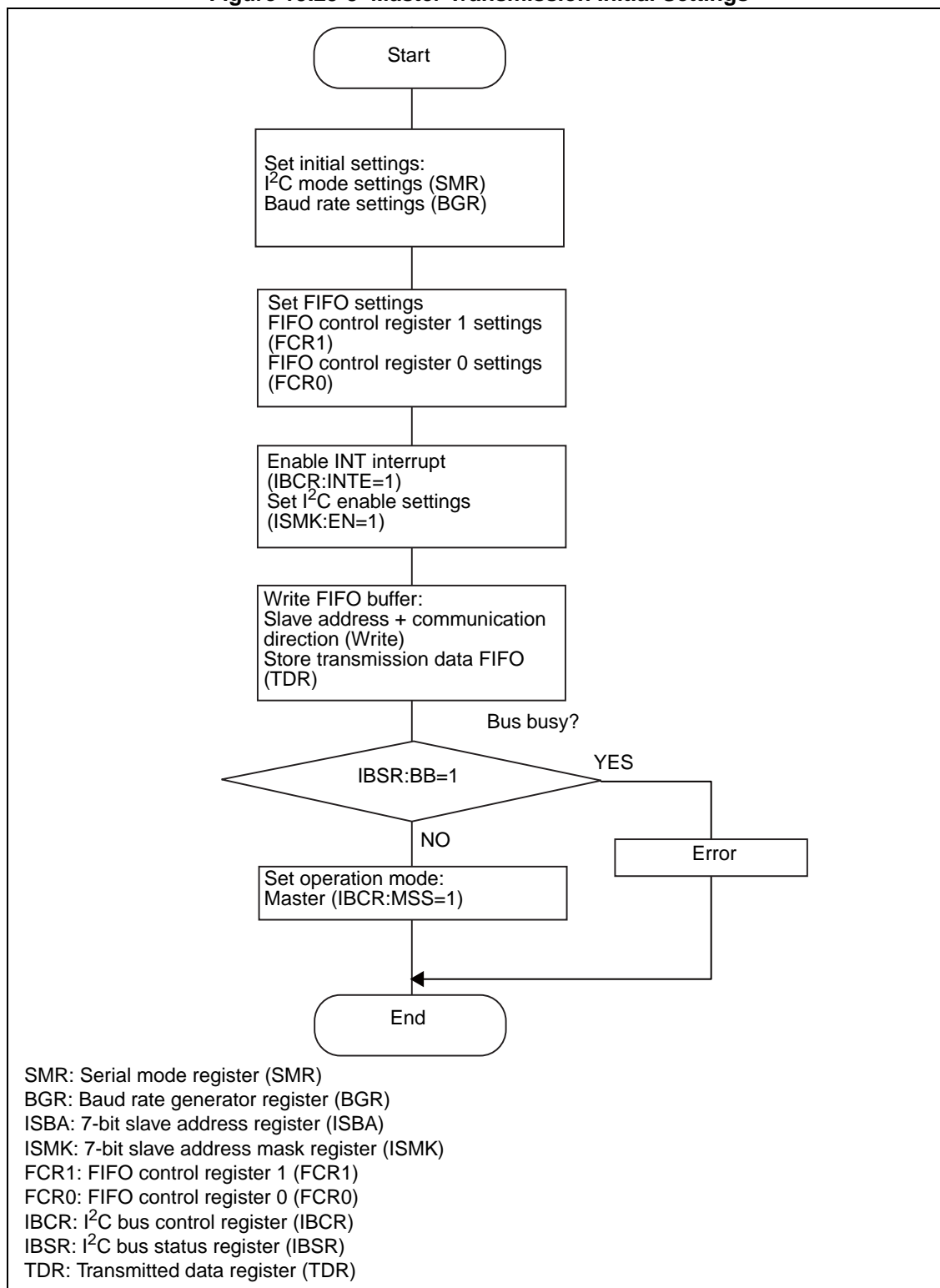


Figure 15.23-9 Master Transmission Interrupt Process

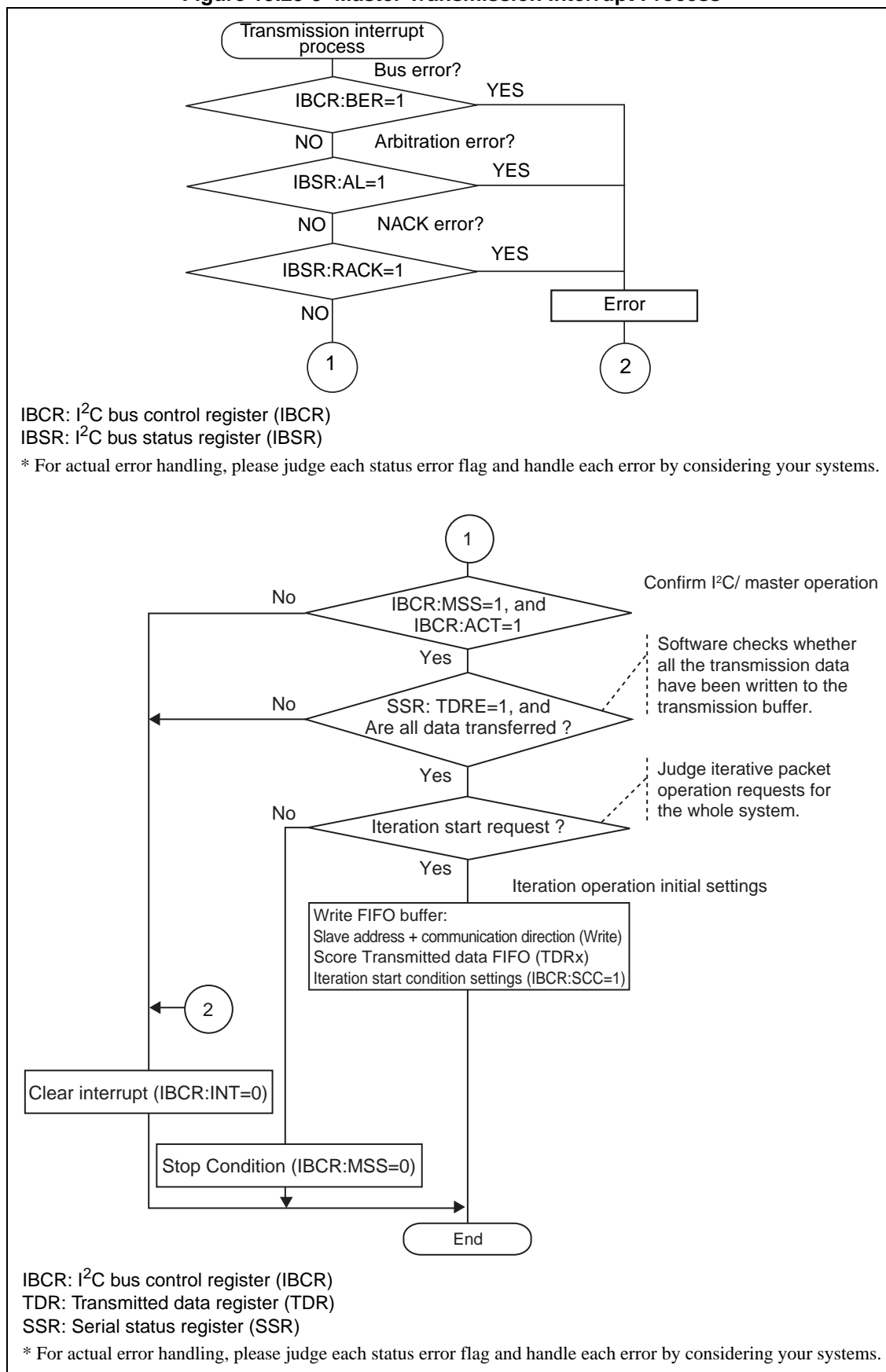


Figure 15.23-10 Slave Reception Main Settings

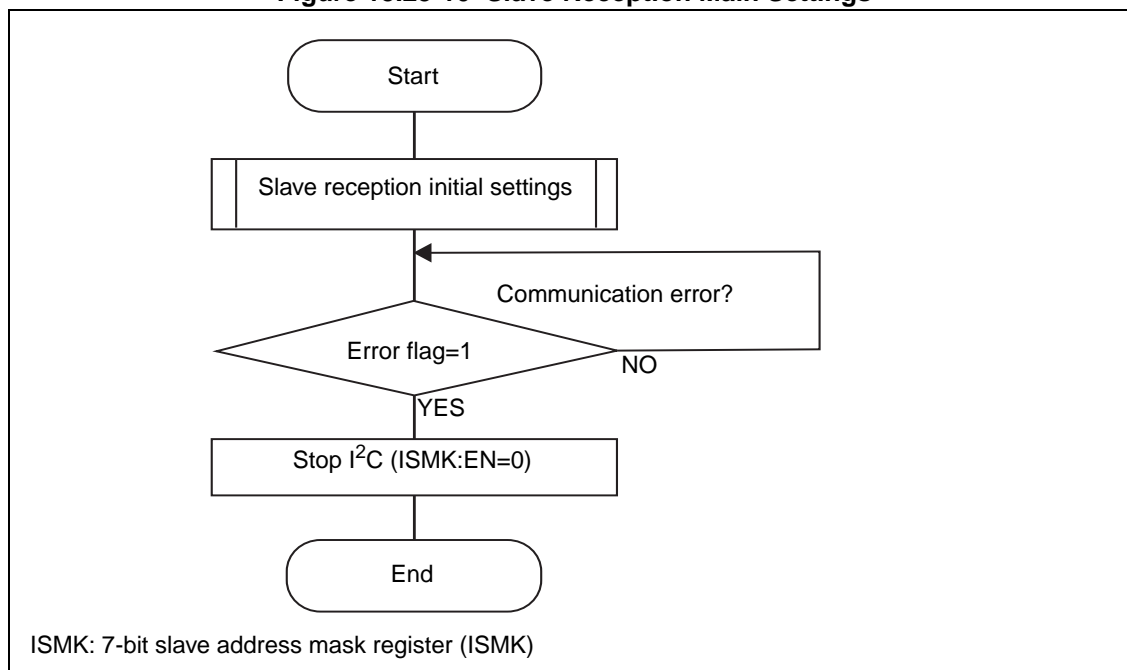


Figure 15.23-11 Slave Reception Initial Settings

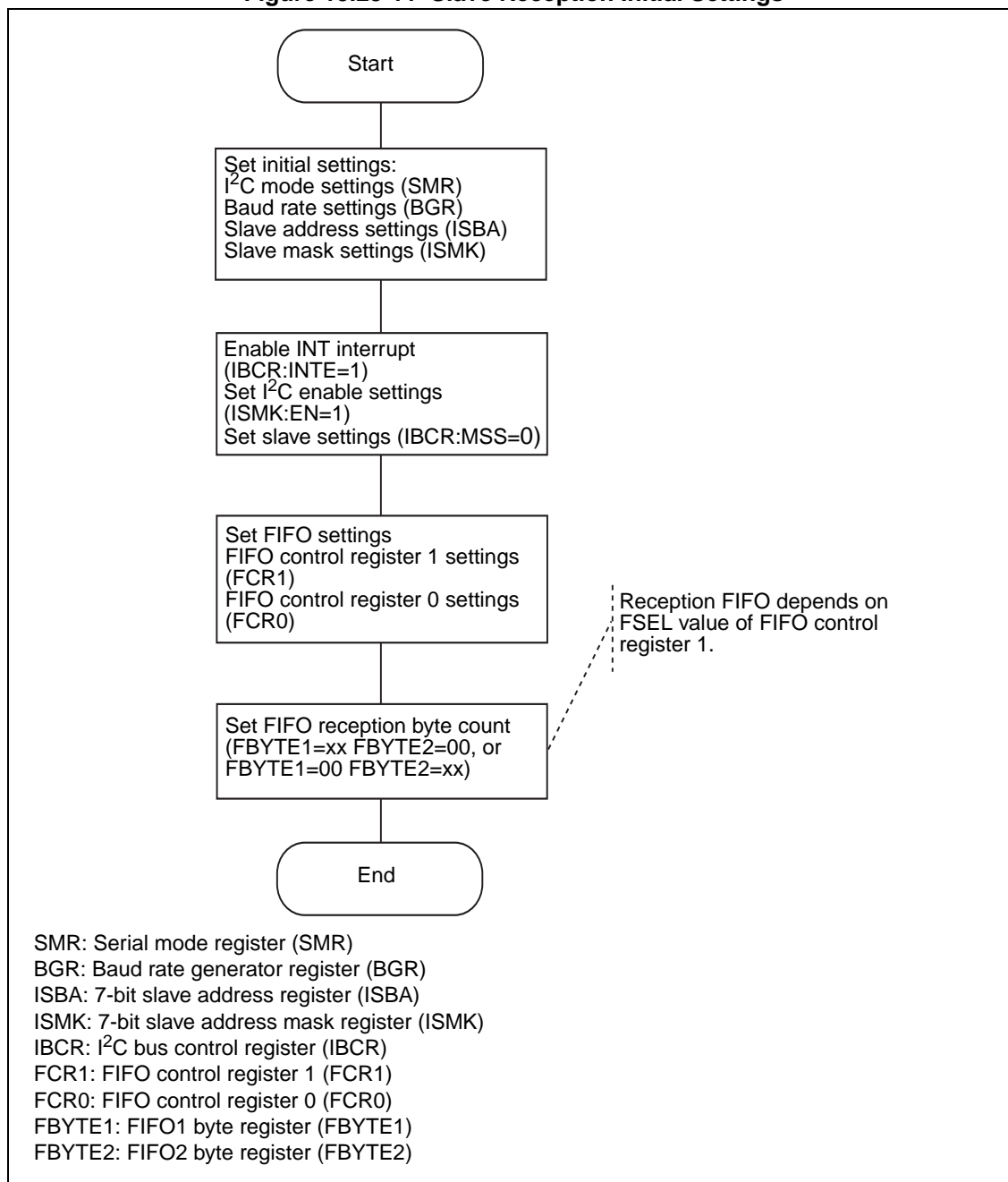
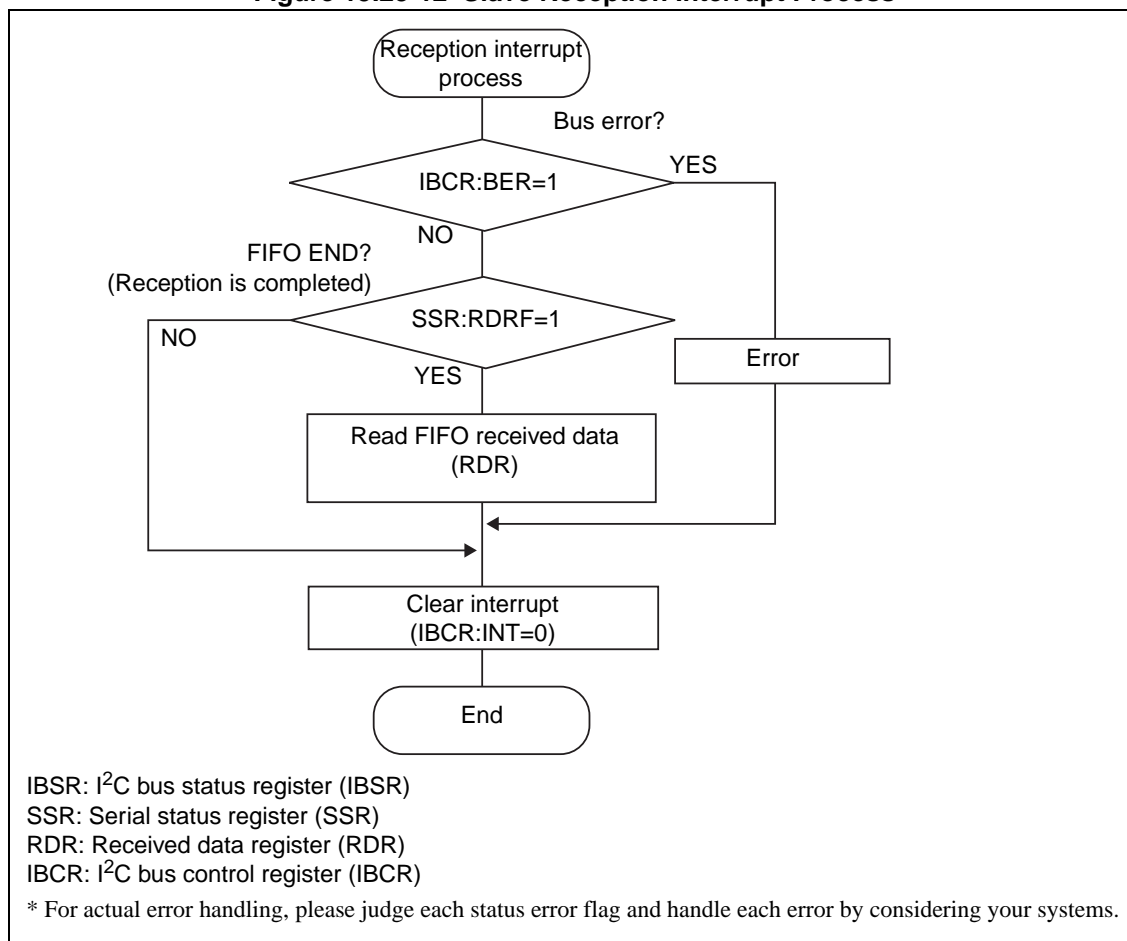


Figure 15.23-12 Slave Reception Interrupt Process

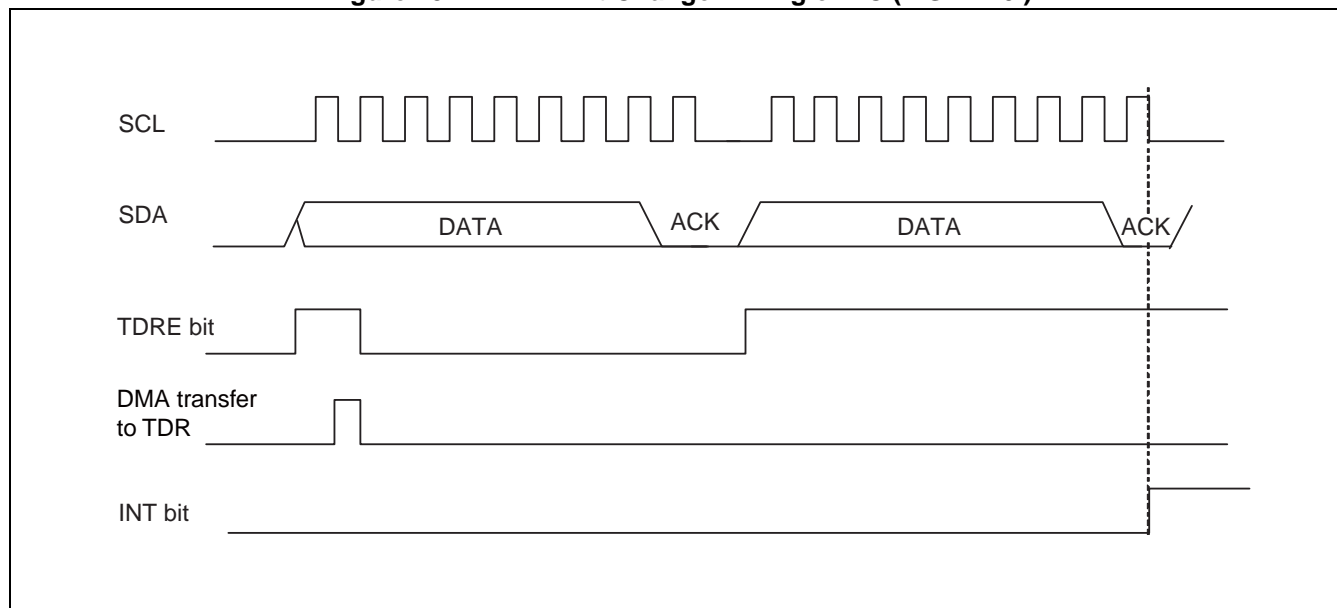


15.24 Notes on I²C Mode

The notes for when you use the I²C mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.
- In I²C mode, if there is no valid data in transmission register (TDR), and transmission data empty flag bit (TDRE) is "1", the interrupt flag (INT) becomes "1" as shown in Figure 15.24-1 when the data on I²C bus for 9 bits (WSEL=0) or for 8 bits (WSEL=1) is transmitted. When the interrupt flag (INT) becomes "1" during DMA transfer, DMA transfer cannot be continued unless clearing the bit to "0" by software. (Common to master transmission, slave transmission, master reception, and slave reception.)

Figure 15.24-1 INT Bit Change Timing of I²C (WSEL= 0)



To perform DMA transfer in I²C mode, since the specification is as shown above, such operations listed below are required for performing DMA transfer to TDR before the interrupt flag (INT) becomes "1".

Below operations are possible to perform to prioritize DMA transfer of I²C.

- Use DMA which has a higher priority (channel number is small). It is enabled to use by fixing the priority setting bit (AT=0).
- Set the value of DMA-halt by interrupt level bit as small as possible (LVL4-LVL0 bit in DILVR register).

- In case of writing the transmission data to transmission data register (TDR) by DMA transfer after transmission data empty flag (SSR:TDRE) becomes "1", or writing the data by software confirming the transmission data empty flag (SSR:TDRE), transmission data empty flag (SSR:TDRE) may not become "0". Therefore, the transmission data should be written before SCL in ACK field falls. There are no restrictions on writing the transmission data by software after the interrupt flag (IBCR:INT) becomes "1".

When performing DMA transfer or sending the data by software confirming the transmission data empty flag (SSR:TDRE), please follow below procedures if the data cannot be written before SCL in ACK field falls.

- Setting

Set the timing of interrupt flag (IBCR:INT) becoming "1" to the 8th bit (WSEL=1).

- Procedures

To transmit or receive data by master, the following procedures are required. To transmit or receive data by slave, it is not required to perform the following.

1. Write the first byte (slave address) to the transmission data register by software.
2. Set to 8-bit for wait selection (IBCR:WSEL="1" write) at the same time that master is started (IBCR:MSS="1" write).
3. After sending the first byte, the interrupt flag (IBCR:INT) becomes "1". Write the second byte to transmission data register (TDR) by software after confirming ACK response (IBSR:RACK="0"). Set the DMAC, and activate DMA transfer, then write "0" to interrupt flag (IBCR:INT).
4. After transmission and reception are completed, terminate the master (IBCR:MSS="0" write) or reboot (IBCR:SCC="1" write).

CHAPTER 16

8/10-BIT A/D CONVERTER

This chapter describes the overview of the 8/10-bit A/D converter, the configuration and functions of registers, and the operation of the 8/10-bit A/D converter.

- 16.1 Overview of the 8/10-bit A/D Converter
- 16.2 Configuration of the 8/10-bit A/D Converter
- 16.3 Pin of the 8/10-bit A/D Converter
- 16.4 Registers of the 8/10-bit A/D Converter
- 16.5 Interrupt of the 8/10-bit A/D Converter
- 16.6 Operation Explanation of the 8/10-bit A/D Converter
- 16.7 A/D conversion Data Protection Function of the 8/10-bit A/D Converter
- 16.8 Using Memorandum of the 8/10-bit A/D Converter
- 16.9 Notes on Using the 8/10-bit A/D Converter

16.1 Overview of the 8/10-bit A/D Converter

The 8/10-bit A/D converter has a feature to convert analog input voltage to a 8/10-bit digital value, using the RC successive comparison/conversion method. The input signal can be selected from different channels of analog input pin, and three types of conversions can be activated: software, internal timer, and external pin trigger.

■ Function of 8/10-bit A/D Converter

There is an A/D conversion feature to convert analog voltage (input voltage) input to the analog input pins into digital values.

- The conversion time is a minimum of 1.2 μ s (including the sampling time at 33 MHz peripheral clock (CLKP)).
- The conversion method used is the RC successive comparison conversion with sample hold circuit.
- The resolution of the conversion result can be selected to be 8 bits or 10 bits
- The analog input pin can be selected using program.
- The A/D data register is provided for each analog input channel.
- The A/D data register has an error flag bit and a error status bit, it can know the A/D conversion data state according to these values.
- DMAC can be started by the A/D conversion end interrupt.
- One of the following conversion activation causes can be selected: software, 16-bit reload timer 1, multi-functional timer (rising edge), or external pin triggers (falling edge).
- There is register bit to select 1 out of 2 sets of A/D converter function:

Function 1:

- Each analog input channel has an A/D data register.
- Generate interrupt request after all selected channels finished A/D conversion
- No conversion data protection

Function 2:

- Only 1 A/D data register available. All channels use the same data register.
- Generate interrupt request after all selected channels finished A/D conversion
- Data is not missed even if the continuous conversion is done, because the conversion data protection function is operating in the interrupt enable status.

There are three conversion modes.

Table 16.1-1 Conversion Modes of 8/10-bit A/D Converter

Conversion Mode	Single Conversion Operation	Scanning Conversion Operation
Single conversion mode	The specified channel (one channel only) is converted for one time and terminated.	Continuous multiple channels (more than 1 channel can be specified) are converted for one time and terminated.
Continuous conversion mode	Repeatedly convert specified channel (1 channel only).	Repeatedly convert multiple channels in succession (more 1 channel can be specified).
Pause-conversion mode	The specified channel (one channel only) is converted for one time and suspended until the next start.	Continuous multiple channels (more than 1 channel can be specified) are converted. However, one channel is converted and suspended until the next start.

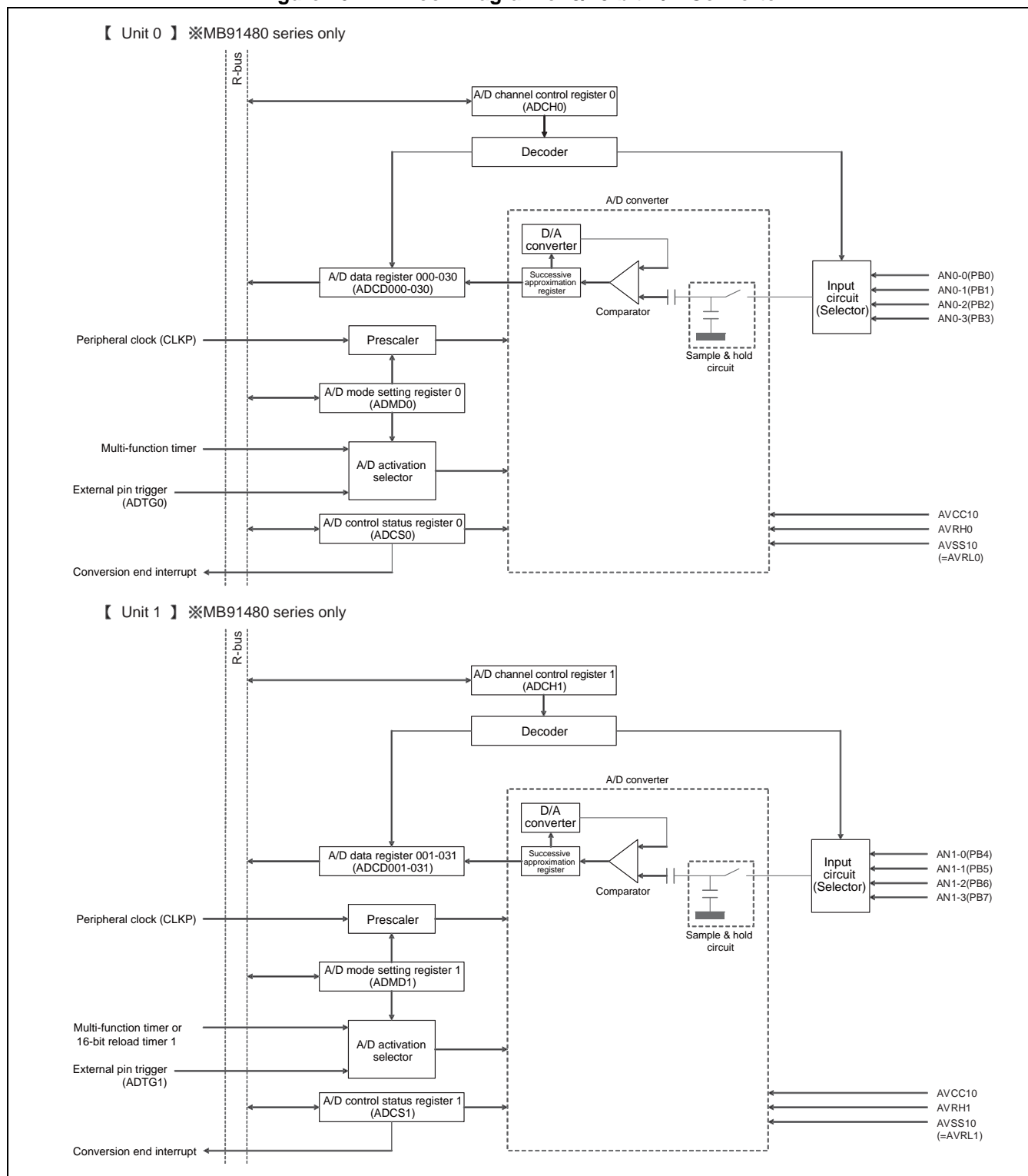
16.2 Configuration of the 8/10-bit A/D Converter

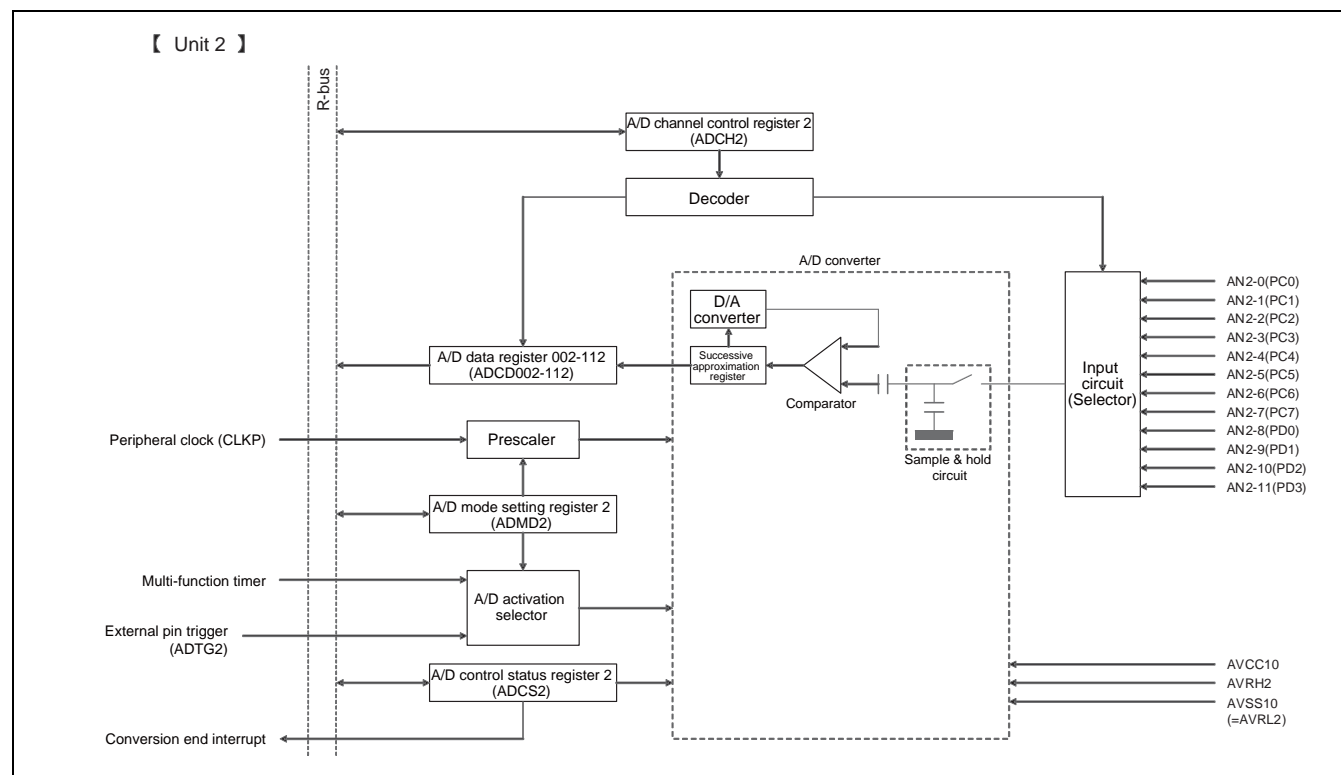
The 8/10-bit A/D converter is made up of the following 11 blocks.

- A/D control status registers (ADCS)
 - A/D channel control register (ADCH)
 - A/D mode setting register (ADMD)
 - A/D data register (ADCD)
 - Clock selector (input clock selector for activation of A/D conversion)
 - Decoder
 - Analog channel selector
 - Sample hold circuit
 - D/A converter
 - Comparator
 - Control circuit
-

■ Block Diagram of 8/10-bit A/D Converter

Figure 16.2-1 Block Diagram of 8/10-bit A/D Converter





In MB91470 series, unit 2: AN0 to AN11

In MB91480 series, unit 0: AN0 to AN3

unit 1: AN0 to AN3

unit 2: AN0 to AN9

● A/D control status registers (ADCS)

Features are available to suspend and confirm conversion, enable/disable interrupt requests, confirm the status of interrupt requests and select the A/D conversion resolution and the conversion function (function 1/ function 2).

● A/D channel control register (ADCH)

There is a feature to select the A/D channel.

● A/D mode setting register (ADMD)

There is a feature to select a conversion mode and to set the A/D conversion compare time and sampling time.

● A/D data register (ADCD)

This register stores A/D conversion results. Flag bits to indicate the status of data in data register.

● Clock selector (Input clock selector for activation of A/D conversion)

This is an A/D conversion activation clock selector. 16-bit reload timer channel 1 output, multi-functional timer and external pin trigger can be selected as the activation clock.

- Decoder

The A/D channel control register (ADCH) ANE0 to ANE3 and ANS0 to ANS3 bit settings are a circuit to select the analog input pin to use.

- Analog channel selector

This circuit selects the pin to be used from different analog input pins.

- Sample hold circuit

This circuit holds the input voltage selected by the analog channel selector. The input voltage can be converted without affected by the input voltage fluctuation in the A/D conversion (in the comparison) by holding the sample of input voltage immediately after starting the A/D conversion.

- D/A converter

The reference voltage is generated to compare with the held sample of input voltage.

- Comparator

This compares the input voltage for which sample hold is performed, with the output voltage of the D/A converter to determine which is the greater of the two.

- Control circuit

The signal from the comparator (higher or lower) determines the A/D conversion value. When the A/D conversion is terminated, this result is stored in the A/D data register (ADCD) and the interrupt request is generated.

16.3 Pin of the 8/10-bit A/D Converter

Pins of 8/10-bit A/D converter and block diagram of pin are shown.

■ Pins of 8/10-bit A/D Converter

The A/D converter pin serves dual use as a general-purpose port. Table 16.3-1 shows pin functions, I/O type, and settings when using the 8/10-bit A/D converter.

Table 16.3-1 Pins of 8/10-bit A/D Converter

Function	Pin Name	Pin Function	I/O Type	Pull-up Setting	Stand-by Control	I/O Port Setting for Using Pin
Unit 0 ch.0 to ch.3	PB0/AN0-0	Port B I/O/ analog input	CMOS output/ CMOS hysteresis input of analog input	Yes (the pull-up function does not work when analog input is enabled.)	Yes	Input setting of port B (DDRB: bit0 to bit3=0) Set to analog input (AICR0 bit0 to bit3=1)
	PB1/AN0-1					
	PB2/AN0-2					
	PB3/AN0-3					
Unit 1 ch.0 to ch.3	PB4/AN1-0					
	PB5/AN1-1					
	PB6/AN1-2					
	PB7/AN1-3					
Unit 2 ch.0 to ch.11	PC0/AN2-0	Port C I/O/ analog input				
	PC1/AN2-1					
	PC2/AN2-2					
	PC3/AN2-3					
	PC4/AN2-4					
	PC5/AN2-5					
	PC6/AN2-6					
	PC7/AN2-7					
	PD0/AN2-8	Port D I/O/ analog input	Input setting of port B (DDRC: bit0 to bit7=0) Set to analog input (AICR2: bit0 to bit7=1)			
	PD1/AN2-9					
	PD2/AN2-10					
	PD3/AN2-11					
External trigger input ADTG0 to ADTG2	PA0/ADTG0	Port A I/O/ external trigger input	CMOS output/ CMOS hysteresis input	Yes	Input setting of port A (DDRA: bit0 to bit2=0)	
	PA1/ADTG1					
	PA2/ADTG2					

16.4 Registers of the 8/10-bit A/D Converter

Register list of 8/10-bit A/D converter is shown.

■ Register List of 8/10-bit A/D Converter

Figure 16.4-1 Register List of 8/10-bit A/D Converter

AICR2

Analog input control register (upper): unit 2

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000170 _H	-	-	-	-	AN11E	AN10E	AN9E	AN8E	----1111 _B
	-	-	-	-	R/W	R/W	R/W	R/W	

Analog input control register (lower): unit 2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000171 _H	AN7E	AN6E	AN5E	AN4E	AN3E	AN2E	AN1E	AN0E	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

AICR0/AICR1

Analog input control register (Lower): unit 0/1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000501 _H	-	-	-	-	AN3E	AN2E	AN1E	AN0E	----1111 _B
000511 _H	-	-	-	-	R/W	R/W	R/W	R/W	

ADCS0/ADCS1/ADCS2

A/D control status register: unit 0/1/2

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000504 _H	BUSY	INT	INTE	PAUS	S10	FuncSet	START	-	0000000- _B
000514 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	
000174 _H									

ADCH0/ADCH1/ADCH2

A/D channel control register: unit 0/1/2

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000506 _H	ANS3	ANS2	ANS1	ANS0	ANE3	ANE2	ANE1	ANE0	00000000 _B
000516 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
000176 _H									

ADMD0/ADMD1/ADMD2

A/D mode setting register: unit 0/1/2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000507 _H	MD1	MD0	STS1	STS0	CT1	CT0	ST1	ST0	00001111 _B
000517 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
000177 _H									

(Continued)

(Continued)

ADCD000 to ADCD030/ADCD001 to ADCD031/ADCD002 to ADCD112

A/D data register (upper): unit 0/1/2

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000508 _H to 00050E _H	ERR	ERRST	-	-	-	-	D9	D8	10- - - - XX _B
000518 _H to 00051E _H	R	R	-	-	-	-	R	R	
000178 _H to 00018E _H									

A/D data register (lower): unit 0/1/2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000508 _H to 00050E _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
000518 _H to 00051E _H	R	R	R	R	R	R	R	R	
000178 _H to 00018E _H									

R/W: Readable/writable
R: Read only

16.4.1 A/D Channel Control Register (ADCH)

The A/D channel control register has a feature to select the A/D conversion channel.

■ A/D Channel Control Register (ADCH: ADCH0 to ADCH2)

Address									Initial value
000506H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
000516H	ANS3	ANS2	ANS1	ANS0	ANE3	ANE2	ANE1	ANE0	0000 0000B
000176H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>									

Table 16.4-1 Functions of Each Bit in A/D Channel Control Register (ADCH)

Bit Name		Function
bit15 to bit12	ANS3 to ANS0: A/D conversion start channel selection bits	<ul style="list-style-type: none"> These bits set the start channel of the A/D conversion and indicate the channel numbers under A/D conversion during conversion operation. When A/D conversion is activated, A/D conversion starts from the channels written to these bits. Channel numbers under the conversion can be read during the A/D conversion. The channel number converted immediately before can be read during the suspension in the pause conversion mode. <p>Notes:</p> <ul style="list-style-type: none"> Be sure to set the smaller number than the number of input channel to ANS bit. Example: ADMD0; "ANS[3:0] ≤0011_B" because MB91480 series has up to 4 channels for unit 0 Only rewrite these bits before conversion begins, with the A/D operation stopped. Do not set this register bit by the read-modify-write instruction after setting the start channel to A/D conversion start channel select bits (ANS3 to ANS0). Because the last conversion channel is read from the ANS3 to ANS0 bits until starting the A/D conversion operation, when this register bit is set by the read-modify-write instruction after setting the start channel to the ANS3 to ANS0 bits, the value of the ANE3 to ANE0 bits should be re-written.
bit11 to bit8	ANE3 to ANE0: A/D conversion end channel selection bits	<ul style="list-style-type: none"> These bits set the end channel of the A/D conversion. A/D conversion is performed up to the specified channel written in these bits. When the same channels with ANS3 to ANS0 are set, only those channels are converted. If continuous conversion mode or stop conversion mode is set, when the conversion up to the channels specified in these bits is completed, conversion returns to the start channel set in ANS3 to ANS0. <p>Notes:</p> <ul style="list-style-type: none"> Never set ANE bit larger than the max. available channel in the product series. Example: ADMD0; "ANS[3:0] ≤0011_B" because MB91480 series has up to 4 channels for unit 0 Be sure to set become "ANS ≤ANE". Only rewrite these bits before conversion begins, with the A/D operation stopped.

16.4.2 A/D Mode Setting Register (ADMD)

The A/D mode setting register has a feature to select a conversion mode and to set the A/D conversion compare time and sampling time.

■ A/D Mode Setting Register (ADMD: ADMD0 to ADMD2)

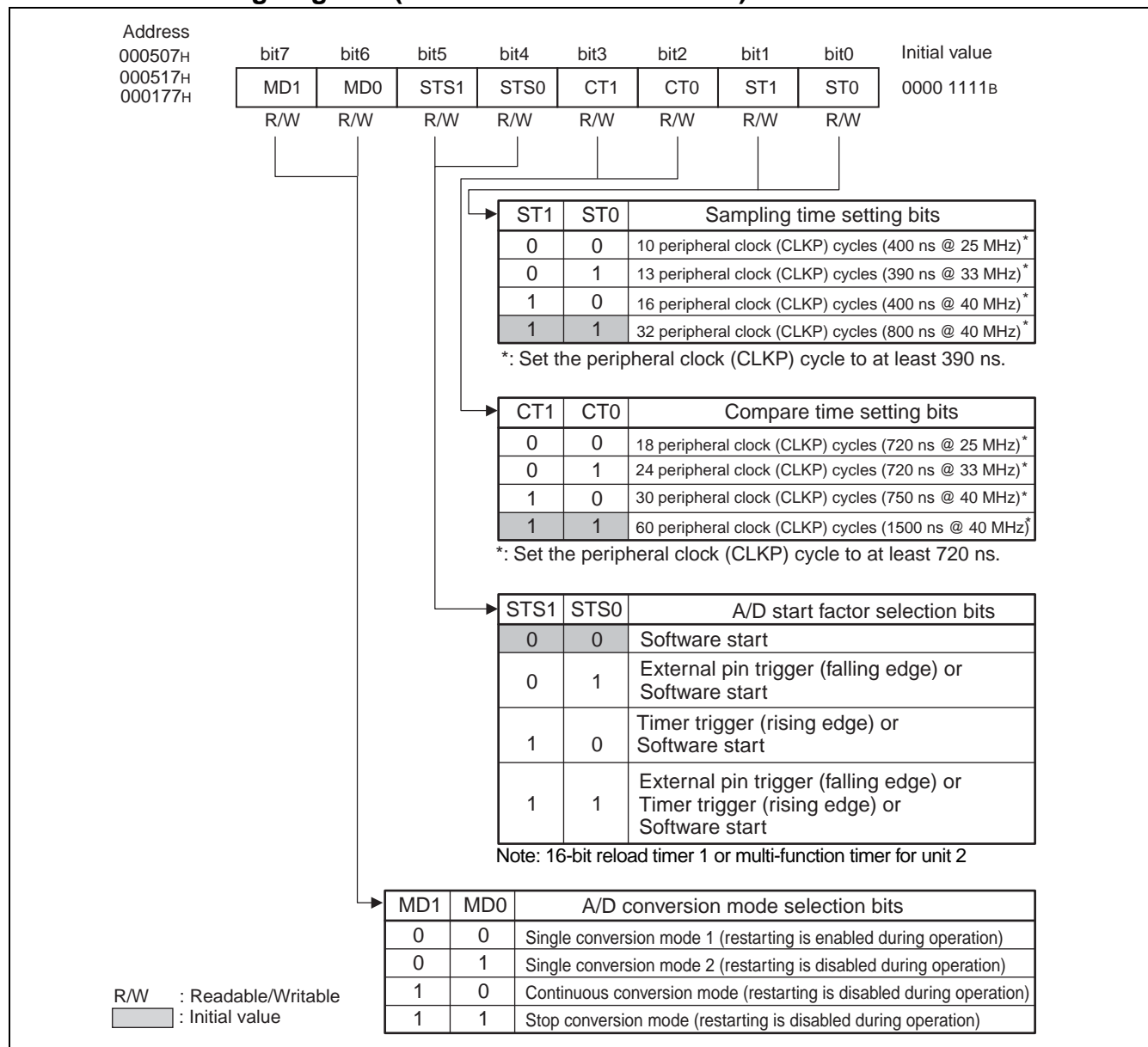


Table 16.4-2 Functions of Each Bit in A/D Mode Setting Register (ADMD: ADMD0 to ADMD2) (1 / 2)

Bit Name		Function
bit7, bit6	MD1, MD0: A/D conversion mode selection bits	<ul style="list-style-type: none"> These bits are used to select the conversion mode during the A/D conversion operation. Two bit values of MD1 and MD0 allow the selection of either the single conversion mode 1, the single conversion mode 2, the continuous conversion mode, or the stop conversion mode. The meaning of each mode is as follows. <ul style="list-style-type: none"> Single conversion mode 1: The continuous A/D conversion from the setting channels of ANS3 to ANS0 to the setting channels of ANE3 to ANE0 is performed only once. It is possible to reactivate while operating. Single conversion mode 2: The continuous A/D conversion from the setting channels of ANS3 to ANS0 to the setting channels of ANE3 to ANE0 is performed only once. It is not possible to reactivate while operating. Continuous conversion mode: Sequentially perform A/D conversion from the channel set in ANS3 to ANS0 to the channel set in ANE3 to ANE0, and repeat until forcibly stopped by means of the BUSY bit. It is not possible to reactivate while operating. Stop conversion mode: Perform A/D conversion from the channel set in ANS3 to ANS0 to the channel set in ANE3 to ANE0 one channel at a time, pausing between each, and repeat until forcibly stopped by means of the BUSY bit. It is not possible to reactivate while operating. The suspended conversion is restarted by the start factor occurrence selected by the STS1 and STS0 bits. <p>Notes:</p> <ul style="list-style-type: none"> Single, continuous, and stop conversion modes that cannot be re-started can be used to activate all timers, external triggers, and software. Only rewrite these bits before conversion begins, with the A/D operation stopped. When A/D conversion mode selection bits (MD1, MD0) are set to "00_B", it is possible to reactivate in the A/D conversion. Only software activating (STS1, STS0=00_B) can be set in this mode. Reactivate according to the following procedure; <ol style="list-style-type: none"> (1) Clear the INT bit to "0". (2) Write "1" to the START bit and write "0" to the INT bit at the same time.
bit5, bit4	STS1, STS0: A/D start factor selection bits	<ul style="list-style-type: none"> The start factor of A/D conversion is selected. When the start factor is in the common use, the first start factor generation starts the operation. <p>Notes:</p> <ul style="list-style-type: none"> The activation trigger changes as soon as the bits are rewritten, so if you wish to rewrite them while A/D conversion is ongoing, switch to a state where your target activation trigger does not exist. If the STS1, STS0 bits are 11_B, the timer cannot start then the external trigger input is "L". Also, the external trigger cannot be started when the timer is "H".

Table 16.4-2 Functions of Each Bit in A/D Mode Setting Register (ADMD: ADMD0 to ADMD2) (2 / 2)

Bit Name		Function
bit3, bit2	CT1, CT0: Compare time setting bits	<p>These bits are used to select the comparison time at the A/D conversion.</p> <ul style="list-style-type: none"> After analog input is loaded (after the sampling time has elapsed), then after the time specified in these bits has passed, the conversion results are checked, and stored in the A/D data register (ADCD). <p>Notes:</p> <ul style="list-style-type: none"> Set for the compare time to become 720 ns or more. If the compare time is not set to 720 ns or more, the normal analog conversion value might not be obtained. Only rewrite these bits before conversion begins, with the A/D operation stopped.
bit1, bit0	ST1, ST0: Sampling time setting bits	<p>These bits are used to select the sampling time at the A/D conversion.</p> <p>When A/D is activated, analog input is retrieved for the time set in these bits.</p> <p>Notes:</p> <ul style="list-style-type: none"> Set for the sampling time to become 390 ns or more. If the sampling time is not set to 390 ns or more, the normal analog conversion value might not be obtained. Only rewrite these bits before conversion begins, with the A/D operation stopped.

16.4.3 A/D Control Status Register (ADCS)

The A/D control status register has features to suspend and confirm conversion, enable/disable interrupt requests, confirm the status of interrupt requests, and select the A/D conversion resolution and the conversion function (function 1/function 2).

■ A/D Control Status Register (ADCS: ADCS1, ADCS2)

Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000504H 000514H 000174H	BUSY	INT	INTE	PAUS	S10	FuncSet	START	-	0000 000-B
	R/W	R/W	R/W	R/W	R/W	R/W	W	-	

START	A/D conversion start bit (Only enable at software starting)	
0	A/D conversion function is not starting.	
1	A/D conversion function is starting.	

FuncSet	A/D conversion function selection bit	
0	Set 1: Each analog input channel has 1 A/D data register. Generate an interrupt request after all selected channels finished A/D conversion. No conversion data protection function operates.	
1	Set 2: All analog input channel share 1 A/D data register. Generate an interrupt request after all selected channels finished A/D conversion. Data is not missed even if the continuous conversion is done, because the conversion data protection function is operating in the interrupt enable status.	

S10	A/D conversion resolution selection bit	
0	10-bit resolution (D0 to D9)	
1	8-bit resolution (D0 to D7)	

PAUS	Temporary stop flag bit	
0	Generate no halting of A/D conversion operation	
1	Halt on A/D conversion operation	

INTE	Interrupt request enable bit	
0	Interrupt request output is disabled.	
1	Interrupt request output is enabled.	

INT	Interrupt request flag bit	
	At read	At write
	0	No A/D conversion ends Bit clear
1	A/D conversion ends No change, no impact to others.	

BUSY	A/D converting bit	
	At read	At write
	0	Now stop A/D conversion A/D conversion forcibly stops
1	Now A/D converting No change, no impact to others	

R/W : Readable/Writable
 W : Write only
 : Initial value

Table 16.4-3 Function of Each Bit in A/D Control Status Register (ADCS) (1 / 2)

Bit Name		Function
bit15	BUSY: A/D converting bit	<ul style="list-style-type: none"> Operational display bit of A/D converter When reading, if this bit is "0", it indicates that A/D conversion is stopped. If it is "1", it indicates that A/D conversion is ongoing. When writing, writing "0" to this bit forcibly stops A/D conversion. When "1" is written, the conversion is not changed and no others are affected. "1" is always read during read modify write (RMW) instruction. <p>Note:</p> <p>Do not perform a forced stop and software activation (BUSY = 0, START = 1) at the same time.</p>
bit14	INT: Interrupt request flag bit	<ul style="list-style-type: none"> This INT bit is set to 1 if data is set in A/D data register through A/D conversion. If this bit and the interrupt request enable bit (ADCS: INTE) are "1", an interrupt request is generated. When writing, "0" clears this bit, and with "1" no change is made, and there are no other effects. "1" is always read during read modify write (RMW) instruction. <p>Note:</p> <p>Clear this bit by writing "0" while the A/D is stopping.</p>
bit13	INTE: Interrupt request enable bit	<ul style="list-style-type: none"> This bit is used to enable and disable the interrupt output to CPU. If this bit and the interrupt request flag bit (ADCS: INT) are "1", an interrupt request is generated.
bit12	PAUS: Temporary stop flag bit	<ul style="list-style-type: none"> It is set to 1 when A/D conversion is suspended. This bit is set to 1 automatically when A/D enters conversion data protection function. During this time, A/D conversion will halt and will not store any new coming data Clear this flag only by writing "0" to this register. 1 is always read during read modify write (RMW) instruction. Refer to Section "16.7 A/D conversion Data Protection Function of the 8/10-bit A/D Converter" for detail operation.
bit11	S10: A/D conversion resolution selection bit	<ul style="list-style-type: none"> This bit is used to select the A/D conversion resolution. Write 0 to this bit to select 10-bit resolution. Write 1 to this bit select 8-bit resolution <p>Notes:</p> <ul style="list-style-type: none"> The data bit used are different depending on the resolution. Only rewrite this bit before conversion begins, with the A/D operation stopped.

Table 16.4-3 Function of Each Bit in A/D Control Status Register (ADCS) (2 / 2)

Bit Name		Function
bit10	Func Set: A/D conversion function selection bit	<ul style="list-style-type: none"> This bit is used to select the function set for A/D conversion. Write 0 to this bit select function set 1: <ul style="list-style-type: none"> Each input channel has 1 data register Generate interrupt after all selected channel finished conversion No data protection Write 1 to this bit select function set 2. <ul style="list-style-type: none"> All input channel share 1 data register Generate interrupt after each channel finished conversion Data protection when interrupt request enabled <p>Note: Only rewrite this bit before conversion begins, with A/D operation stopped</p>
bit9	START: A/D conversion start bit	<ul style="list-style-type: none"> This bit is used to start the A/D conversion operation by the software. Write "1" to this bit to activate A/D conversion. Restart by this bit cannot be used at the stop conversion mode. "0" is always read during read modify write (RMW) instruction. <p>Note: Do not perform a forced stop and software activation (BUSY = 0, START = 1) at the same time.</p>
bit8	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect to operation.

16.4.4 A/D Data Register (ADCD)

The A/D data register stores A/D conversion results.

■ A/D Data Register
(ADCD: ADCD000 to ADCD030, ADCD001 to ADCD031, ADCD002 to ADCD112)

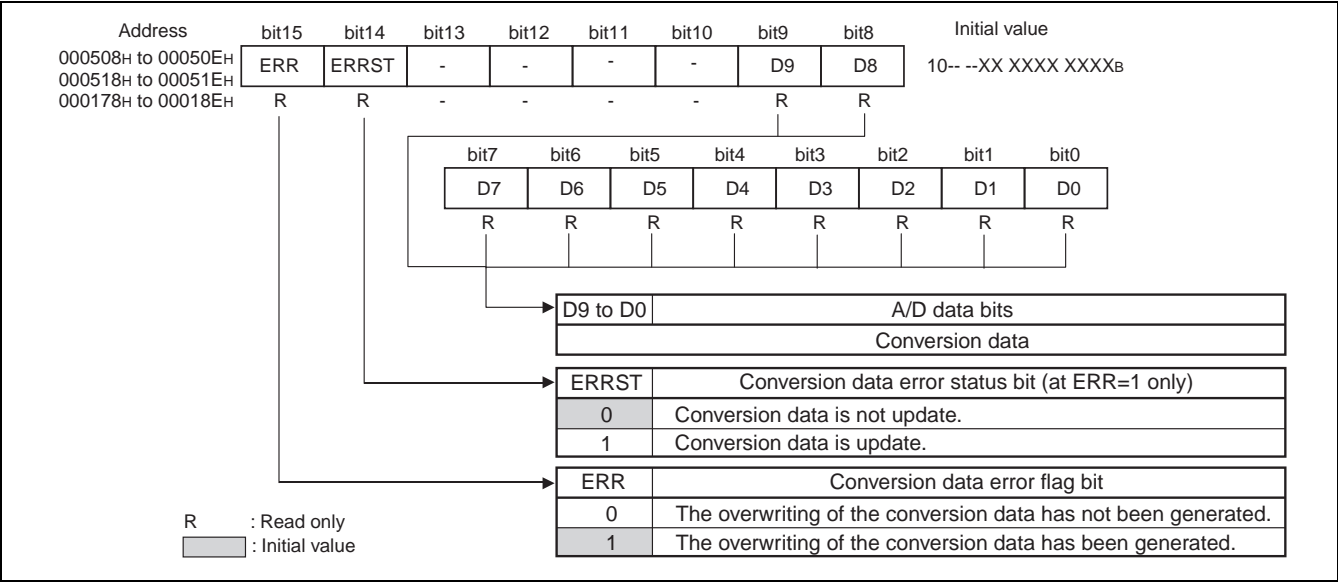


Table 16.4-4 Function of Each Bit in A/D Data Register (ADCD)

Bit Name		Function
bit15	ERR: Conversion data error flag bit	<ul style="list-style-type: none"> When this bit is "1", the content of the error can be known by the value of the ERRST bit in the bit that shows that there was an error in the A/D conversion data. This bit is set to "1" when reading it. When a new conversion result is written in this register, it is cleared to "0". <p>Note:</p> <ul style="list-style-type: none"> When conversion data protection function is used FuncSet = 1 and INTE = 1, this bit is always "0".
bit14	ERRST: Conversion data error status bit	<ul style="list-style-type: none"> It is a flag that shows the content of the error of the A/D conversion data at ERR bit = 1. It is shown that the conversion result by CPU reading is old at ERR bit = 1 and this bit = 0. The conversion result by CPU reading shows that the old conversion data was lost from the overwriting of a new conversion result without completing reading the old conversion result with CPU at ERR bit = 1 and this bit = 1. When the old conversion data is lost from the overwriting of a new conversion result without completing reading the old conversion result with CPU, it is set in "1". This bit is cleared to "0" when reading it. <p>Note:</p> <p>When conversion data protection function is used (FuncSet = 1 and INTE = 1), this bit is always "0".</p>
bit13 to bit10	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect to operation.
bit9 to bit0	D9 to D0: A/D data bits	<ul style="list-style-type: none"> The result of the A/D conversion is stored, and the register is rewritten at each conversion end. The final conversion value is stored usually. The initial value of this register is undefined. <p>Notes:</p> <ul style="list-style-type: none"> The conversion data protection function is provided. Do not write data to these bits while A/D conversion is ongoing. When 8-bit resolution is selected, bit8 & bit9 read 0.

16.4.5 Analog Input Control Register (AICR)

The analog input control register controls analog input.

■ Analog Input Control Register (AICR: AICR0 to AICR2)

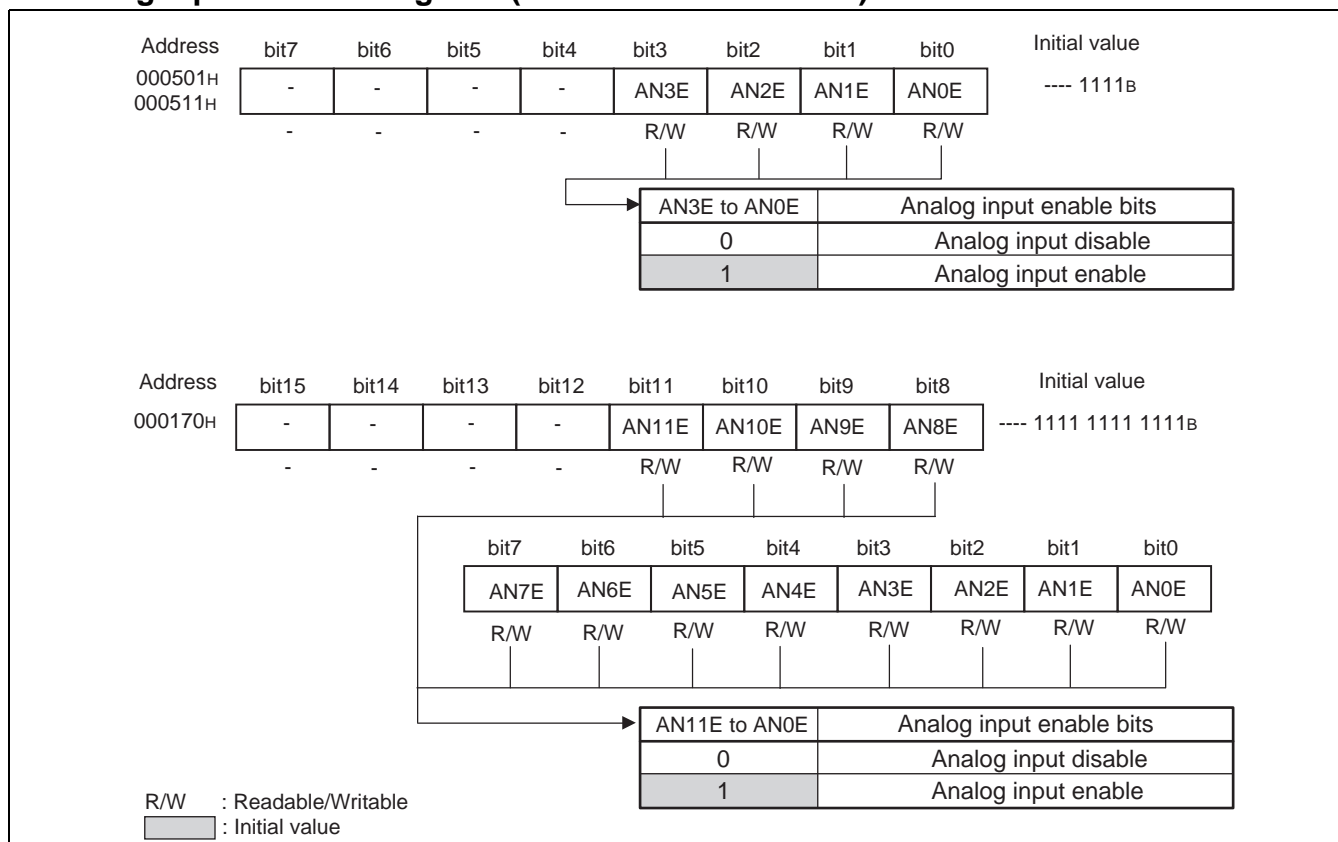


Table 16.4-5 Functions of Each Bit in Analog Input Control Register (AICR)

Bit Name		Function
(AICR0, AICR1) bit7 to bit4 (AICR2) bit15 to bit12	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect to operation.
(AICR0, AICR1) bit3 to bit0 (AICR2) bit11 to bit0	AN3E to AN0E, AN11E to AN0E, Analog input enable bits	<ul style="list-style-type: none"> When these bits are "0", analog input is disabled. When these bits are "1", analog input is enabled. Set the AICR register bit corresponding to the pin to be used as the analog input pin to "1". When this is done, the value "0" will be read from the PDR register.

16.5 Interrupt of the 8/10-bit A/D Converter

The 8/10-bit A/D converter can generate interrupt requests during A/D conversion by setting data in the A/D data register.

■ Interrupt of 8/10-bit A/D Converter

See Table 16.5-1 for the interrupt control bits and interrupt cause of the 8/10-bit A/D converter.

Table 16.5-1 Interrupt Control Bits and Interrupt Cause of 8/10-bit A/D Converter

	8/10-bit A/D converter
Interrupt request flag bit	ADCS: INT
A/D conversion function selection bit	ADCS: FuncSet
Interrupt request enable bit	ADCS: INTE
Interrupt cause	Writing of A/D conversion result to A/D data register

When FuncSet = 0, interrupt could be generated after all selected channel conversion completed. The INT bit of the A/D control status register (ADCS) is set to "1" when all the conversion end and A/D conversion results are set in the A/D data register (ADCD). At this time, an interrupt request is generated to the interrupt controller if interrupt request is enabled (ADCS:INTE=1).

When FuncSet = 1, interrupt could be generated after each channel conversion completed. The INT bit of the A/D control status register (ADCS) is set to "1" when each conversion ends and A/D conversion results are set in the A/D data register (ADCD). At this time, an interrupt request is generated to the interrupt controller if interrupt request is enabled (ADCS:INTE=1).

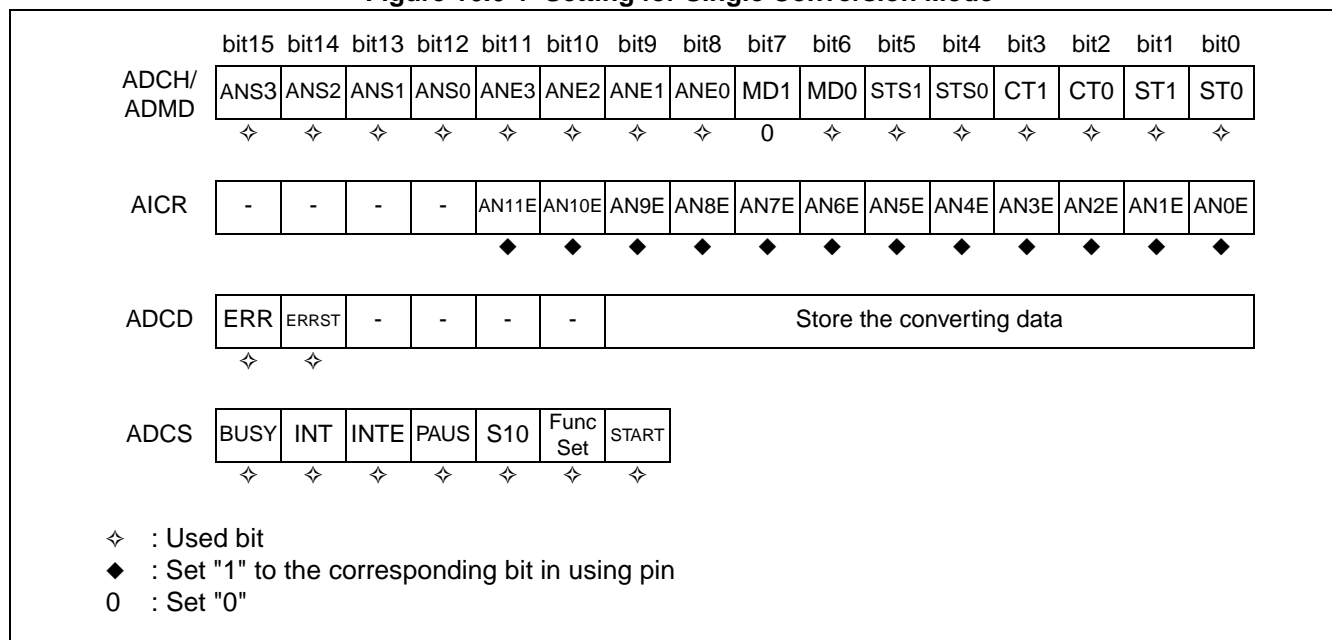
16.6 Operation Explanation of the 8/10-bit A/D Converter

Three mode types, the single conversion, continuous conversion, and stop conversion modes are available for the 8/10-bit A/D converter. The operation explanation in each mode is done.

■ Operation of Single Conversion Mode

In single conversion mode, it sequentially converts the analog input which has been set by the ANS bit and ANE bit, and when it reaches to the end channel set in ANE bit, it stops the A/D conversion. If the start channel and end channel are the same (ANS=ANE), only one channel specified in the ANS bit will be converted. The settings in Figure 16.6-1 are required in order to operate in single-conversion mode.

Figure 16.6-1 Setting for Single Conversion Mode



Reference:

The example of conversion order in single conversion mode is shown in following.

When ANS=000_B, ANE=011_B: AN0 → AN1 → AN2 → AN3 → end

(FuncSet = 0) ADCD00 → ADCD01 → ADCD02 → ADCD03 → end

(FuncSet = 1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → end

When ANS=011_B, ANE=011_B: AN3 → end

(FuncSet = 0) ADCD03 → end

(FuncSet = 1) ADCD03 → end

Note:

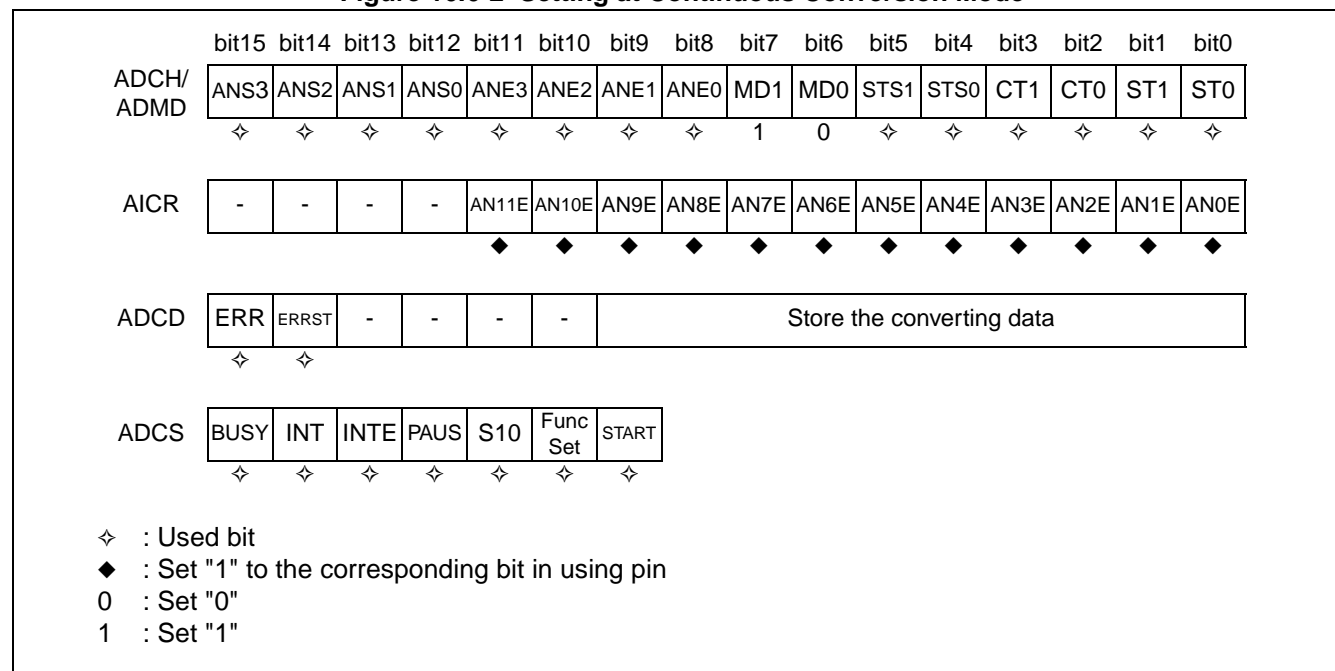
When A/D conversion mode selection bits (MD1, MD0) are set to "00_B", it is possible to reactivate in the A/D conversion. Only software activating (STS1, STS0=00_B) can be set in this mode. Reactivate according to the following procedure.

1. Clear the INT bit to "0".
 2. Write "1" to the START bit and write "0" to the INT bit at the same time.
-

■ Operation of Continuous Conversion Mode

In the continuous conversion mode, the analog inputs set by the ANS and ANE bits are sequentially converted, the analog input set by the ANS bit is resumed at the end of conversion of the end channel set by the ANE bit, and the A/D conversion operation is continued. If the start channel and end channel are identical (ANS=ANE), conversion loops on the channel specified by ANS only. The settings shown in Figure 16.6-2 are required in order to operate in continuous conversion mode.

Figure 16.6-2 Setting at Continuous Conversion Mode



Reference:

The example of conversion order in continuous conversion mode is shown in following.

When ANS=000_B, ANE=011_B: AN0 → AN1 → AN2 → AN3 → Repeat

(FuncSet =0) ADCD00 → ADCD01 → ADCD02 → ADCD03 → Repeat

(FuncSet =1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → Repeat

When ANS=011_B, ANE=011_B: AN3 → AN3 → AN3 → AN3 → Repeat

(FuncSet =0) ADCD03 → ADCD04 → ADCD05 → ADCD06 →

ADCD07 → ADCD08 → ADCD09 → ADCD10 →

ADCD11* → ADCD00 → ADCD01 → ADCD02 → Repeat

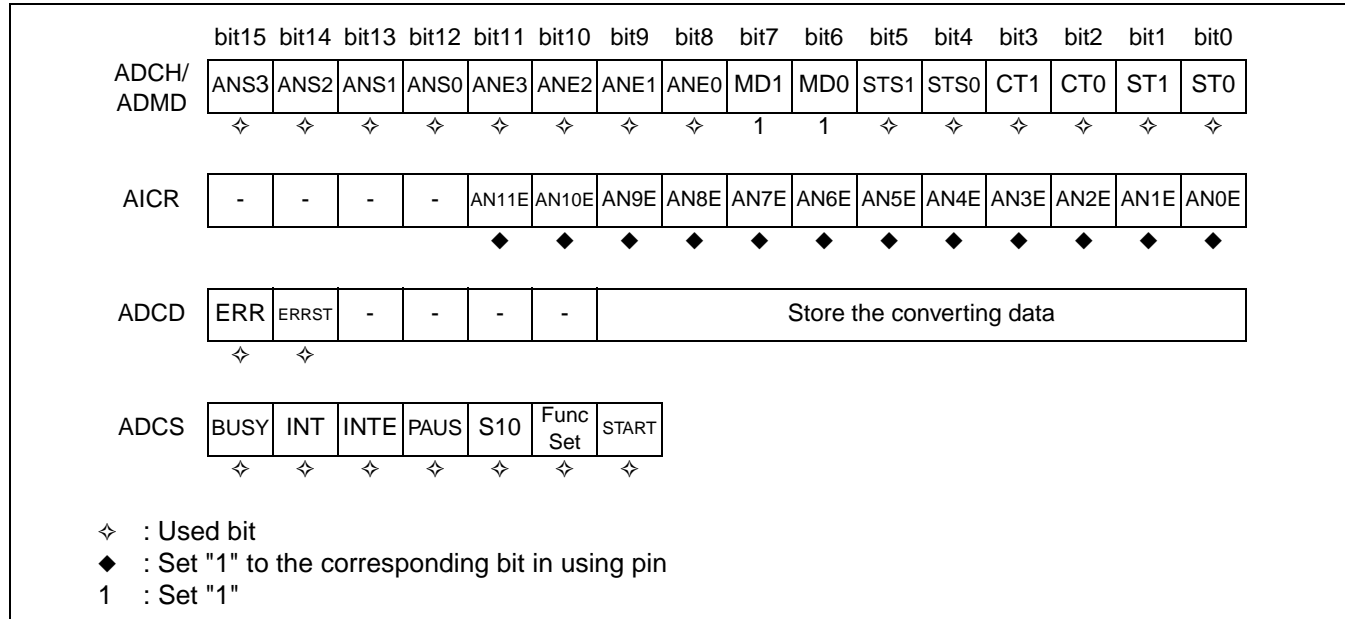
(FuncSet =1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → Repeat

*: The number of data register to be used depend on the series

■ Operation of Pause-Conversion Mode

In the stop conversion mode, the analog input set by the ANS and ANE bits is converted by being suspended for every channel, the analog input set by the ANS bit is resumed at the end of conversion of the end channel set by the ANE bit, and the operation of A/D conversion and suspension is continued. If the start channel and end channel are identical (ANS=ANE), conversion loops on the channel specified by the ANS bits only. When the conversion is restarted during the suspension, the start factor specified by the STS1 and STS0 bits is generated. The settings in Figure 16.6-3 are required in order to operate in stop conversion mode.

Figure 16.6-3 Setting at Pause-conversion Mode



Reference:

The example of conversion order in stop conversion mode is shown in following.

- When ANS=000_B, ANE=011_B:
 AN0 → Pause → AN1 → Pause → AN2 → Pause → AN3 → Repeat
 FuncSet =0
 ADCD00 → Pause → ADCD01 → Pause → ADCD02 → Pause → ADCD03 → Repeat
 FuncSet =1
 ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Repeat
 - When ANS=011_B, ANE=011_B:
 AN3 → Pause → AN3 → Pause → AN3 → Pause → AN3 → Repeat
 FuncSet =0
 ADCD03 → Pause → ADCD04 → Pause → ADCD05 → Pause → ADCD06 → Pause
 ADCD07 → Pause → ADCD08 → Pause → ADCD09 → Pause → ADCD10 → Pause
 ADCD11* → Pause → ADCD00 → Pause → ADCD01 → Pause → ADCD02 → Repeat
 FuncSet =1
 ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Repeat
- *: The number of data register to be used depend on the series

16.7 A/D conversion Data Protection Function of the 8/10-bit A/D Converter

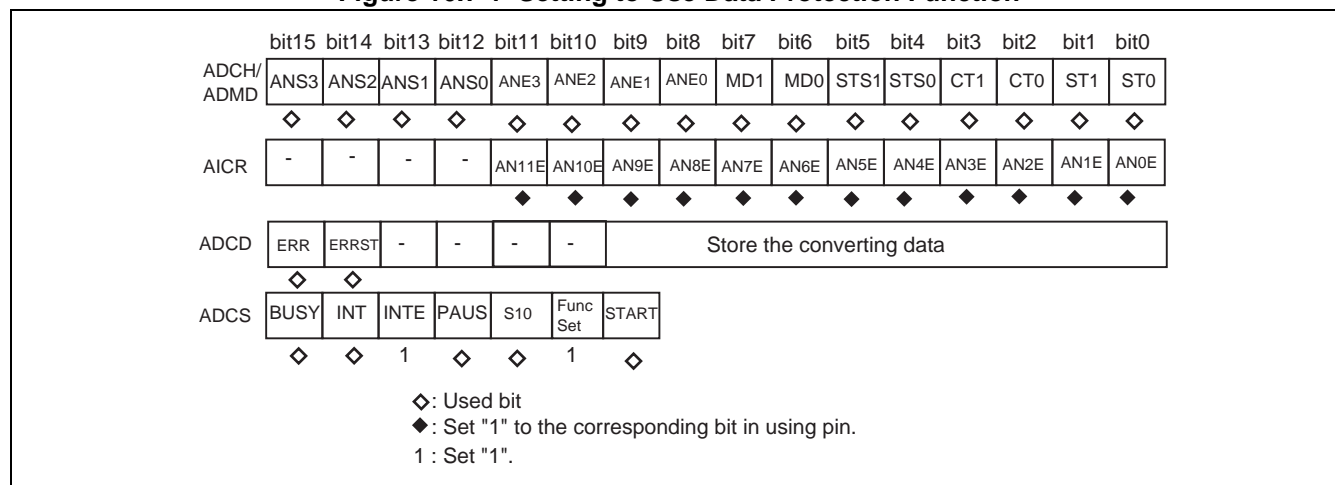
When the A/D conversion operates in the interrupt enabled state, the conversion data protection function works.

■ A/D Conversion Data Protection Function

When selected ADCS: FuncSet = 1 (conversion function 2), there is only 1 data register for storing conversion result. For this reason, when performing A/D conversion, the data stored in data register is rewritten after conversion is completed. Therefore, part of previous data may be lost when the converted data transfer to memory is delayed. To get around this, when interrupt is enabled (ADCS:INTE=1), data protection feature works as described below.

When conversion data is stored in the A/D data register (ADCD), the ADCS:INT bit set to "1". While the ADCS:INT bit is "1", conversion data will not be stored to ADCD after the next conversion ends. The ADCS:PAUS bit is set and A/D conversion becomes suspended. While suspended, the value immediately prior is retained. In order to cancel the suspend, clear ADCS:INT bit. After the suspended status is cleared, the conversion data that had been maintained is stored in ADCD and the next operation is performed.

Figure 16.7-1 Setting to Use Data Protection Function



Notes:

- Conversion data protection function only operates in conversion function 2 (ADCS:FuncSet=1) and interrupt enabled (ADCS:INTE=1)
- When the conversion is restarted during suspension, the waiting data is destroyed.

16.8 Using Memorandum of the 8/10-bit A/D Converter

This section is a memorandum when 8/10-bit A/D converter is used.

■ ADMD Register Setting

The A/D converter sampling time and compare time based on three types of frequencies (25MHz, 33MHz, 40MHz) can be set by the ADMD register. The minimum conversion time corresponding to each frequency can be set. Set ADMD by the following two methods when a set frequency is different from three above-mentioned types of values.

- ST[1:0]/CT[1:0] bit (bit3, bit2/bit1, bit0) of the ADMD register is set so that neither the time of the sample nor the compare time may become below recommended value of 8/10-bit A/D converter.
- Set P3-to-P0 bit of DIVR0 register (bit3-to-bit0) so that the frequency of the peripheral clock (CLKP) may become one of the above-mentioned.

● Example:

- When the peripheral clock (CLKP) frequency is 16MHz: Method 1

Cycle: 62.5ns

Sample time: ST[1:0] = 00_B

→ 10 peripheral clock (CLKP) cycles

→ $10 \times 62.5\text{ns} = 625\text{ns} > 390\text{ns}$ (Minimum value)

Compare time: CT[1:0] = 00_B

→ 18 peripheral clock (CLKP) cycles

→ $18 \times 62.5\text{ns} = 1125\text{ns} > 720\text{ns}$ (Minimum value)

∴ Total conversion time = 1750ns

- When the peripheral clock (CLKP) frequency is 16MHz: Method 2

Source oscillation frequency: 10MHz

PLL multiplication rate: $\times 5$ -multiplication

DIVR0: P3 to P0 = 0001_B

→ $\text{CLKP} = 10 \times 5 / 2 = 25\text{MHz}$

ST[1:0] = 00_B, CT[1:0] = 00_B

∴ Total conversion time = 1120ns

16.9 Notes on Using the 8/10-bit A/D Converter

This section describes notes on using 8/10-bit A/D converter.

■ Notes on Using 8/10-bit A/D Converter

● Analog input pin

The A/D input pin does double duty as a port I/O pin. The port-direction register (DDR) and analog input enable register (AICR) are switched and used. For the pins used as analog input, set the bits corresponding to DDR to 0 and port to input, then set AICR register analog input mode (AICR x = 1). Lock the input gate on the port side. When the intermediate level signal is inputted in the port input mode (AICRx = 0), the input leak current flows through the gate.

● Cautions for use of internal timers

To activate the A/D converter by an internal timer, set the STS1 and STS0 bits of the A/D control status register (ADMD). When doing so, set the internal timer input value to the inactive side (for internal timer, this is "L"). If you set it to the active side, the timer may start to operate as soon as you write to the ADMD register.

● Turning-on sequence of power supply to A/D converter and analog inputs

Make sure to apply to the A/D converter power source (AVCC10, AVRH 0 to AVRH 2, and AVSS10) and apply analog input (AN0-0 to AN0-3, AN1-0 to AN1-3, AN2-0 to AN2-11) after or at the same time as applying digital power source (VCC). When cutting off the power, cut off the digital power source (VCC) after or at the same time as cutting off the A/D converter power source and analog input.

● Power voltage of A/D converter

In order to prevent latch-ups, make sure that the A/D converter power source (AVCC10) does not exceed the voltage of the digital power source (VCC).

● Setting of ADCH register

Set to become $ANS \leq ANE$.

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Be sure to set "0" to the ANS3-2, ANE3-2 bits of the ADCH0/ADCH1 register. The ANS and ANE bits of the ADCH2 register should be set to "1011_B" or lower.

● Setting of ADMD register

Set ST[1:0]/CT[1:0] bit (bit3, bit2/bit1 bit0) so that neither the time of the sample nor the compare time may become below recommended value of 8/10-bit A/D converter.

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Refer to "16.8 Using Memorandum of the 8/10-bit A/D Converter" for a detailed explanation.

● **Setting of ADCS register**

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Do neither A/D conversion starting setting (START=1) with software nor stop setting (BUSY=1) at the same time.

Refer to Section "16.4 Registers of the 8/10-bit A/D Converter" for a detailed explanation.

● **Notice in A/D conversion data protection function**

Conversion data protection function is only available in conversion function 2 (ADCS:FuncSet=1) and interrupt enabled (ADCS:INTE=1).

● **Flag bit in A/D data register**

Even if lower 8 bits of the A/D data register are read by byte access, neither the ERRST nor ERR bits are changed.

Moreover, when the conversion data protection function is used, the ERRST and ERR bits are always "0".

● **External trigger terminal**

Return the input level of the external trigger terminal to former level by the external trigger terminal after the activating of the A/D converter.

● **Reactivation of the A/D conversion**

When A/D conversion mode selection bits (MD1, MD0) are set to "00_B", it is possible to reactivate in the A/D conversion. Only software activating (STS1, STS0=00_B) can be set in this mode. Reactivate according to the following procedure.

1. Clear the INT bit to "0".
2. Write "1" to the START bit and write "0" to the INT bit at the same time.

CHAPTER 17

CLOCK MONITOR

This chapter explains the function and the operation of the clock monitor.

17.1 Overview of the Clock Monitor

17.2 Clock output enable register of the Clock Monitor

17.1 Overview of the Clock Monitor

When the output enable bit of the clock output enable register is set to "1", the clock is output from clock monitor terminal (CLKPOUT). The frequency of the output clock is set by the output frequency selection bit of the clock output enable register.

■ Output Frequency from Clock Monitor

Table 17.1-1 shows the frequency of the clock output by the clock monitor function.

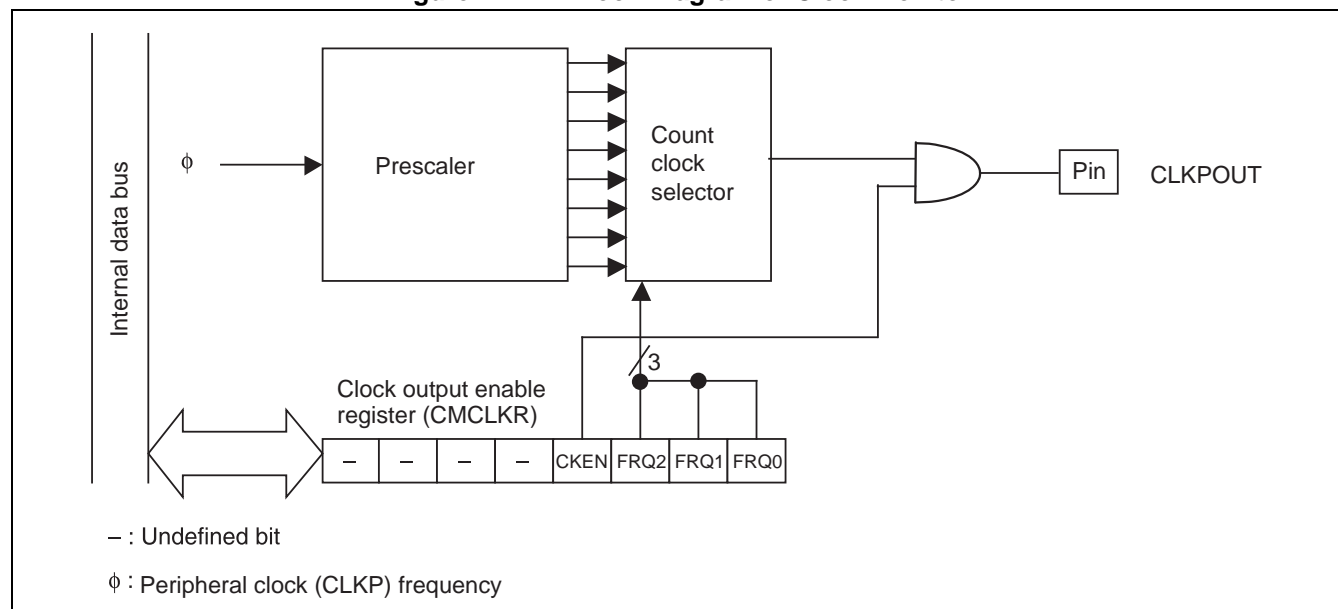
Table 17.1-1 Output Frequency of Clock Monitor Function

FRQ2 to FRQ0	Clock output frequency	$\phi=40$ MHz		$\phi=20$ MHz		$\phi=10$ MHz	
		Cycle	Frequency	Cycle	Frequency	Cycle	Frequency
000 _B	$\phi/2^1$	50 ns	20 MHz	100 ns	10 MHz	200 ns	5 MHz
001 _B	$\phi/2^2$	100 ns	10 MHz	200 ns	5 MHz	400 ns	2.5 MHz
010 _B	$\phi/2^3$	200 ns	5 MHz	400 ns	2.5 MHz	800 ns	1.25 MHz
011 _B	$\phi/2^4$	400 ns	2.5 MHz	800 ns	1.25 MHz	1.6 μ s	625 kHz
100 _B	$\phi/2^5$	800 ns	1.25 MHz	1.6 μ s	625 kHz	3.2 μ s	312.5 kHz
101 _B	$\phi/2^6$	1.6 μ s	625 kHz	3.2 μ s	312.5 kHz	6.4 μ s	156.25 kHz
110 _B	$\phi/2^7$	3.2 μ s	312.5 kHz	6.4 μ s	156.25 kHz	12.8 μ s	78.1 kHz
111 _B	$\phi/2^8$	6.4 μ s	156.25 kHz	12.8 μ s	78.1 kHz	25.6 μ s	39.1 kHz

ϕ : Peripheral clock (CLKP) frequency

■ Block Diagram of Clock Monitor

Figure 17.1-1 Block Diagram of Clock Monitor



17.2 Clock output enable register of the Clock Monitor

In the clock output enable register, the clock output is set.

■ Bit Configuration of Clock Output Enable Register

Figure 17.2-1 Bit Configuration of Clock Output Enable Register

CMCLKR									Initial value ---- 0000 _B
Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0000015F _H	-	-	-	-	CKEN	FRQ2	FRQ1	FRQ0	
	-	-	-	-	R/W	R/W	R/W	R/W	

R/W : Readable/Writable

[bit7 to bit4] Reserved: Reserved bits

These are reserved bits. "1111_B" is read in reading.

Writing in these bits are invalid.

[bit3] CKEN: Output enable bit

The output of clock monitor terminal (CLKPOUT) is permitted.

CKEN	Clock output enable
0	Clock output disable [initial value]
1	Clock output enable

[bit2 to bit0] FRQ2 to FRQ0: Output frequency selection bits

The frequency of the output clock is set.

The ratio of dividing frequency of peripheral clock (CLKP) is selected from eight kinds and set.

FRQ2	FRQ1	FRQ0	Divide frequency ratio
0	0	0	Divided-by-2 [initial value]
0	0	1	Divided-by-4
0	1	0	Divided-by-8
0	1	1	Divided-by-16
1	0	0	Divided-by-32
1	0	1	Divided-by-64
1	1	0	Divided-by-128
1	1	1	Divided-by-256

CHAPTER 18

12-BIT A/D CONVERTER

This chapter describes the overview of the 12-bit A/D converter, the configuration and functions of registers, and the operation of the 12-bit A/D converter.

- 18.1 Overview of the 12-bit A/D Converter
- 18.2 Configuration the 12-bit A/D Converter
- 18.3 Pin the 12-bit A/D Converter
- 18.4 Registers the 12-bit A/D Converter
- 18.5 Interrupt the 12-bit A/D Converter
- 18.6 Operation Explanation the 12-bit A/D Converter
- 18.7 A/D conversion Data Protection Function the 12-bit A/D Converter
- 18.8 Differential input mode the 12-bit A/D Converter
- 18.9 Using Memorandum of the 12-bit A/D Converter
- 18.10 Notes on Using the 12-bit A/D Converter

18.1 Overview of the 12-bit A/D Converter

The 12-bit A/D converter has a feature to convert analog input voltage to a 12-bit digital value, using the RC successive comparison/conversion method. The input signal can be selected from analog input pins, and conversion can be activated: by software, internal timer, or external pin trigger.

■ Function of 12-bit A/D Converter

There is an A/D conversion feature to convert analog voltage (input voltage) input to the analog input pins into digital values.

- The conversion time is a minimum of 2.0 μ s (including the sampling time at 33 MHz peripheral clock (CLKP)).
- The conversion method used is the RC successive comparison conversion with sample hold circuit.
- The analog input pin can be selected from more than 1 channels using the program.
- The A/D data register is provided for analog input channel.
- In each A/D data register, the error flag bit and the error status bit exist. The state of the A/D conversion data is known by these values.
- DMAC can be started by the A/D conversion end interrupt.
- The analog input mode can select either of the normal input or the difference input by the program.
- One of the following conversion activation causes can be selected: software, 16-bit reload timer 1, multi-functional timer (rising edge), or external pin triggers (falling edge).
- The following 2 modes can be set by the A/D conversion function selection bit.

[Function 1]

- Each analog input channel has 1 A/D data register.
- Generate interrupt request after all selected channels finished A/D conversion
- No conversion data protection function

[Function 2]

- Only 1 A/D data register available. All channels use the same data register.
- Generate interrupt request after all selected channel finished conversion
- Data is not missed even if the continuous conversion is done, because the conversion data protection function is operating in the interrupt enable status.

There are three conversion modes.

Table 18.1-1 Conversion Modes of 12-bit A/D Converter

Conversion Mode	Single Conversion Operation	Scanning Conversion Operation
Single conversion mode	The specified channel (one channel only) is converted for one time and terminated.	Continuous multiple channels (more than 1 channel can be specified) are converted for one time and terminated.
Continuous conversion mode	Repeatedly convert specified channel (one channel only).	Repeatedly convert multiple channels (more than 1 channel can be specified) in succession.
Pause-conversion mode	The specified channel (one channel only) is converted for one time and suspended until the next start.	Continuous multiple channels (more than 1 channel can be specified) are converted. However, one channel is converted and suspended until the next start.

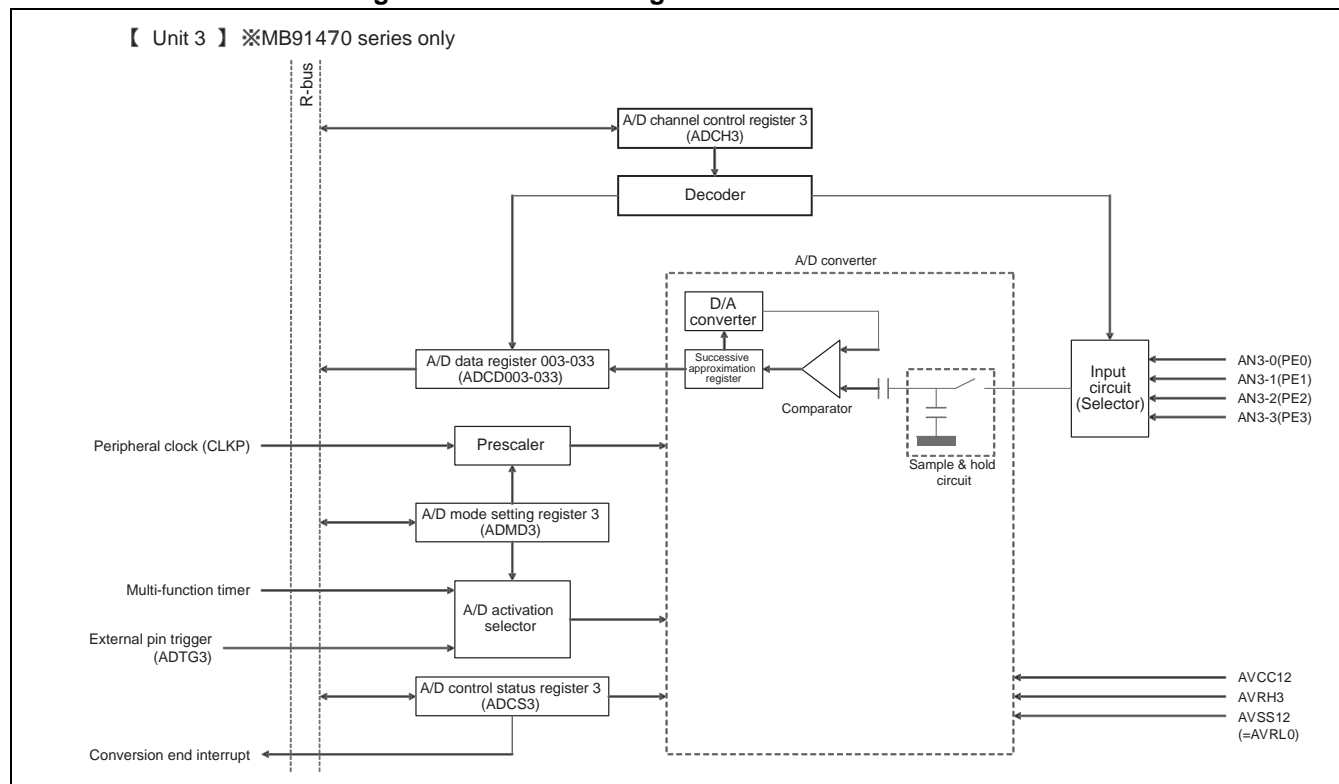
18.2 Configuration the 12-bit A/D Converter

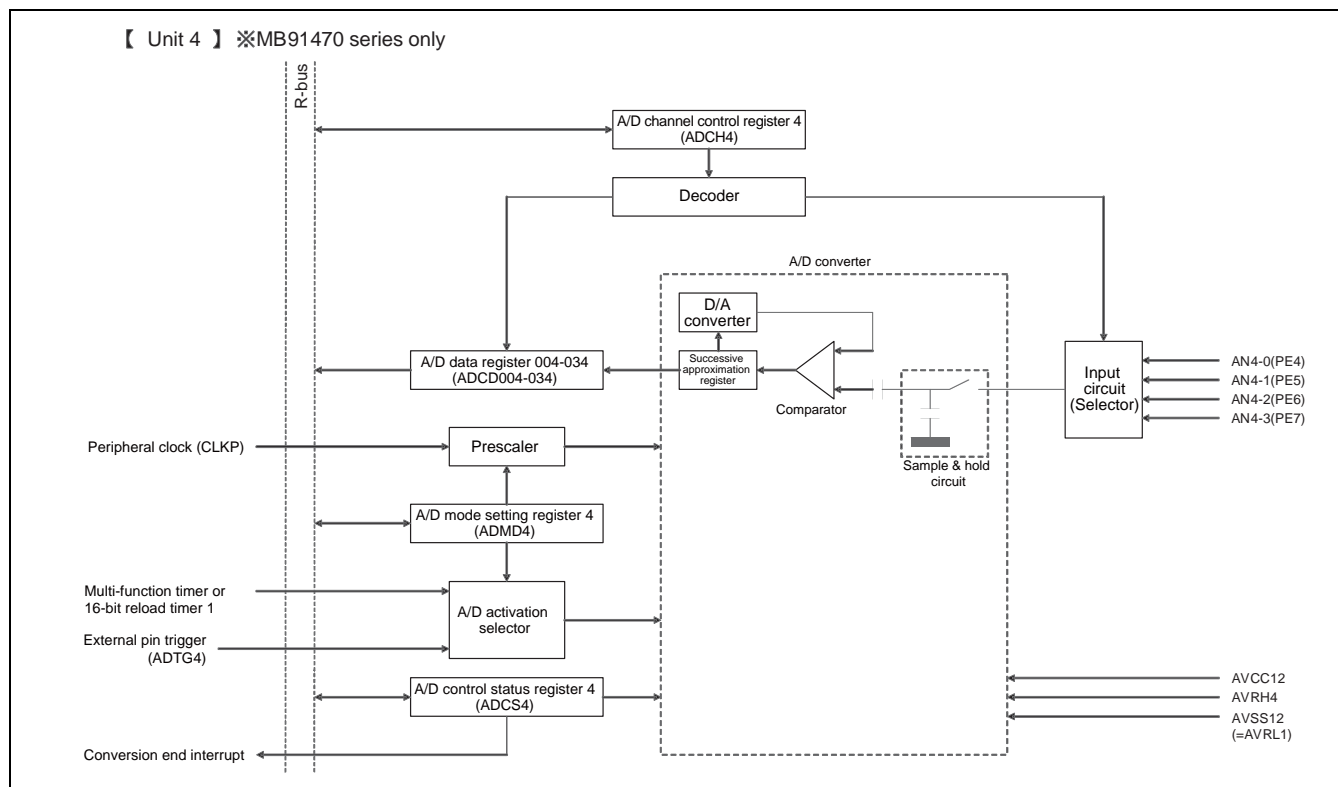
The 12-bit A/D converter is made up of the following 11 blocks.

- A/D control status registers (ADCS)
- A/D channel control register (ADCH)
- A/D mode setting register (ADMD)
- A/D data register (ADCD)
- Clock selector (input clock selector for activation of A/D conversion)
- Decoder
- Analog channel selector
- Sample hold circuit
- D/A converter
- Comparator
- Control circuit

■ Block Diagram of 12-bit A/D Converter

Figure 18.2-1 Block Diagram of 12-bit A/D Converter





References:

- MB91470 series:
Unit 3, AN3-0 to AN3-3 (normal)/AN3-0 to AN3-1 (differential)
Unit 4, AN4-0 to AN4-3 (normal)/AN4-0, AN4-1 (differential)
- MB91480 series: Non-installing

● A/D control status registers (ADCS)

Features are available to suspend and confirm conversion, enable/disable interrupt requests, select analog input mode (normal/differential), confirm the status of interrupt requests and select the conversion function (function 1/function 2).

● A/D channel control register (ADCH)

There is a feature to select the A/D channel for conversion.

● A/D mode setting register (ADMD)

There is a feature to select a conversion mode and to set the A/D conversion compare time and sampling time.

● A/D data register (ADCD)

This register stores A/D conversion results. There are flag bits to indicate the status of conversion data (new data/overwrite).

● Clock selector (Input clock selector for activation of A/D conversion)

This is an A/D conversion activation clock selector. 16-bit reload timer ch.1 output, multi-functional timer and external pin trigger (ADTGn) can be selected as the activation clock.

● Decoder

The A/D channel control register (ADCH) ANE0 to ANE1 and ANS0 to ANS1 bit settings are a circuit to select the analog input pin to use.

● Analog channel selector

This circuit selects the pin to be used from the different analog input pins.

● Sample hold circuit

This circuit holds the input voltage selected by the analog channel selector. The input voltage can be converted without affected by the input voltage fluctuation in the A/D conversion (in the comparison) by holding the sample of input voltage immediately after starting the A/D conversion.

● D/A converter

The reference voltage is generated to compare with the held sample of input voltage.

● Comparator

This compares the input voltage for which sample hold is performed, with the output voltage of the D/A converter to determine which is the greater of the two.

● Control circuit

The signal from the comparator (higher or lower) determines the A/D conversion value. When the A/D conversion is terminated, this result is stored in the A/D data register (ADCD) and the interrupt request is generated.

18.3 Pin the 12-bit A/D Converter

Pins of 12-bit A/D converter and block diagram of pin are shown.

■ Pins of 12-bit A/D Converter

The A/D converter pin serves dual use as a general-purpose port. Table 18.3-1 shows pin functions, I/O type, and settings when using the 12-bit A/D converter.

Table 18.3-1 Pins of 12 Bit A/D Converter

Function		Pin Name	Pin Function	I/O Type	Pull-up Setting	Stand-by Control	I/O Port Setting for Using Pin
Normal input	Differential input						
Unit 3 ch.0	Unit 3 ch.0	PE0/AN3-0 (AN3-0P)	Port E I/O/ analog input	CMOS output/ CMOS hysteresis input or analog input	Yes (However, the pull-up function cannot be used at the analog input enable.)	Yes	Input setting of port E (DDRE: bit0 to bit3=0) Set to analog input (AICR3: bit0 to bit3=1) Set to normal input (ADCS3: bit11=0) Set to differential input (ADCS3: bit11=1) Input setting of port E (DDRE: bit4 to bit7=0) Set to analog input (AICR4: bit0 to bit3=1) Set to normal input (ADCS4: bit11=0) Set to differential input (ADCS4: bit11=0)
Unit 3 ch.1	Unit 3 ch.0	PE1/AN3-1 (AN3-0N)					
Unit 3 ch.2	Unit 3 ch.1	PE2/AN3-2 (AN3-1P)					
Unit 3 ch.3	Unit 3 ch.1	PE3/AN3-3 (AN3-1N)					
Unit 4 ch.0	Unit 4 ch.0	PE4/AN4-0 (AN4-0P)					
Unit 4 ch.1	Unit 4 ch.0	PE5/AN4-1 (AN4-0N)					
Unit 4 ch.2	Unit 4 ch.1	PE6/AN4-2 (AN4-1P)					
Unit 4 ch.3	Unit 4 ch.1	PE7/AN4-3 (AN4-1N)					
External trigger input ADTG3, ADTG4		PA3/ADTG3 PA4/ADTG4	Port A I/O/ External trigger input	CMOS output/ CMOS hysteresis input	Yes		Input setting of port A (DDRA:bit3, bit4=0)

18.4 Registers the 12-bit A/D Converter

Register list of 12-bit A/D converter is shown.

■ Register List of 12-bit A/D Converter

Figure 18.4-1 Register List of 12-bit A/D Converter

AICR3/AICR4

Analog input control register: unit 3/4

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000521 _H	-	-	-	-	AN3E	AN2E	AN1E	AN0E	----1111 _B
000531 _H	-	-	-	-	R/W	R/W	R/W	R/W	

ADCS3/ADCS4

A/D control status register: unit 3/4

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000524 _H	BUSY	INT	INTE	PAUS	PINMD	FuncSet	START	-	0000000- _B
000534 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	

ADCH3/ADCH4

A/D channel control register: unit 3/4

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000526 _H	-	-	ANS1	ANS0	-	-	ANE1	ANE0	--00--00 _B
000536 _H	-	-	R/W	R/W	-	-	R/W	R/W	

ADMD3/ADMD4

A/D mode setting register: unit 3/4

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000527 _H	MD1	MD0	STS1	STS0	CT1	CT0	ST1	ST0	00001111 _B
000537 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ADCD003 to ADCD033/ADCD004 to ADCD034

A/D data register (upper): unit 3/4

Address:	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
000528 _H to 00052E _H	ERR	ERRST	-	-	D11	D10	D9	D8	10-XXXX _B
000538 _H to 00053E _H	R	R	-	-	R	R	R	R	

A/D data register (lower): unit 3/4

Address:	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000528 _H to 00052E _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
000538 _H to 00053E _H	R	R	R	R	R	R	R	R	

R/W : Readable/Writable

R : Read only

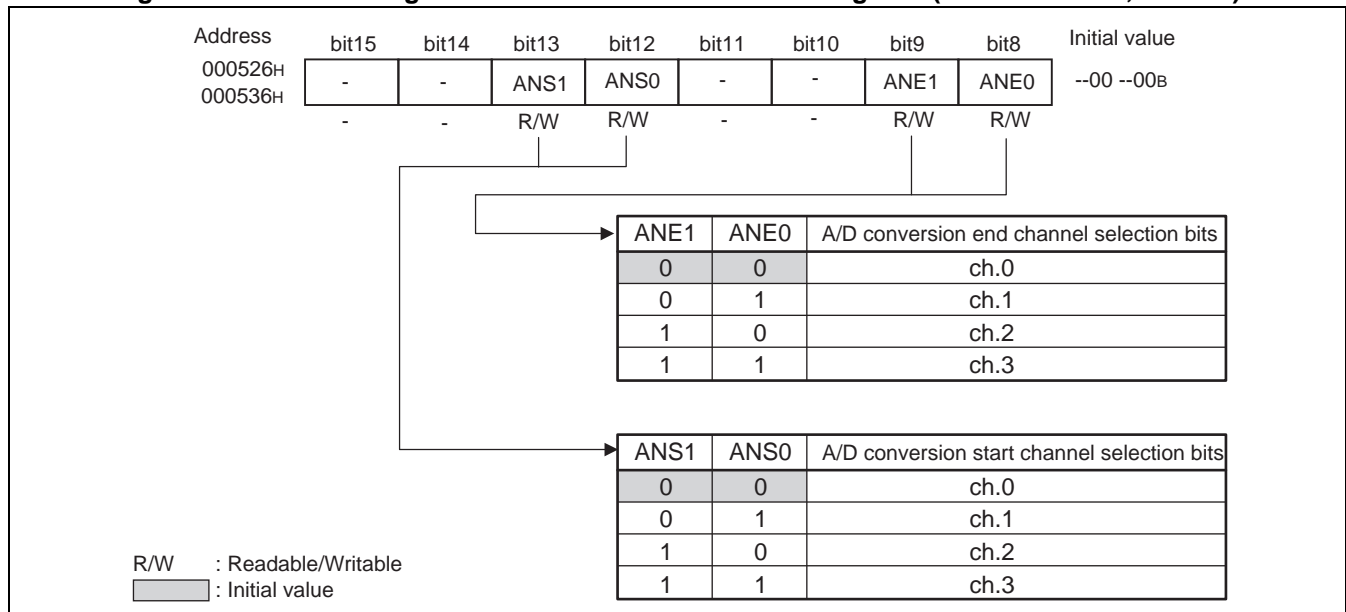
MB91470/480 Series

18.4.1 A/D Channel Control Register (ADCH)

The A/D channel control register has a feature to select the A/D conversion channel.

■ A/D Channel Control Register (ADCH: ADCH3, ADCH4)

Figure 18.4-2 Bit Configuration of A/D Channel Control Register (ADCH: ADCH3, ADCH4)



Notes:

- When ADCS:PINMD (bit11)=0,
 - Unit 3 is ch.0 to ch.3 = AN3-0 to AN3-3
 - Unit 4 is ch.0 to ch.3 = AN4-0 to AN4-3.
- When ADCS:PINMD (bit11)=1, ANS[0] & ANE[0] are ignored.
 - Unit 3 is ch.0/ch.1 = AN3-0P, AN3-0N; ch.2/ch.3 = AN3-1P, AN3-1N
 - Unit 4 is ch.0/ch.1 = AN4-0P, AN4-0N; ch.2/ch.3 = AN4-1P, AN4-1N
- Be sure to set to become "ANS ≤ ANE".

Table 18.4-1 Functions of Each Bit in A/D Channel Control Register (ADCH)

Bit Name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect to operation.
bit14	Undefined bit	<ul style="list-style-type: none"> Be sure to write "0".
bit13, bit12	ANS1, ANS0: A/D conversion start channel selection bits	<ul style="list-style-type: none"> These bits set the start channel of the A/D conversion and indicate the channel numbers under A/D conversion during conversion operation. When A/D conversion is activated, A/D conversion starts from the channels written to these bits. Channel numbers under the conversion can be read during the A/D conversion. The channel number converted immediately before can be read during the suspension in the pause conversion mode. <p>Notes:</p> <ul style="list-style-type: none"> When using differential input mode (ADCS:PINMD=1), the LSB (i.e. ANS0) will be ignored. For example, set ANS[1:0]=1X_B can use AN3-1 & AN3-1 as start channel. Only rewrite these bits before conversion begins, with the A/D operation stopped.
bit11	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect to operation.
bit10	Undefined bit	Be sure to write "0".
bit9, bit8	ANE2, ANE1, ANE0: A/D conversion end channel selection bits	<ul style="list-style-type: none"> These bits set the end channel of the A/D conversion. A/D conversion is performed up to the specified channel written in these bits. When the same channels with ANS1 and ANS0 are set, only those channels are converted. <p>If continuous conversion mode or stop conversion mode is set, when the conversion up to the channels specified in these bits is completed, conversion returns to the start channel set in ANS1 and ANS0.</p> <p>Notes:</p> <ul style="list-style-type: none"> When using differential input mode (ADCS:PINMD=1), the LSB (i.e. ANE0) will be ignored. For example, set ANE[1:0] = 1X_B when use AN3-1 & AN3-1 as end channel. Be sure to set to become "ANS ≤ ANE". Only rewrite these bits before conversion begins, with the A/D operation stopped.

MB91470/480 Series

18.4.2 A/D Mode Setting Register (ADMD)

The A/D mode setting register has a feature to select a conversion mode and to set the A/D conversion compare time and sampling time.

■ A/D Mode Setting Register (ADMD: ADMD3, ADMD4)

Figure 18.4-3 Bit Configuration of A/D Mode Setting Register (ADMD: ADMD3, ADMD4)

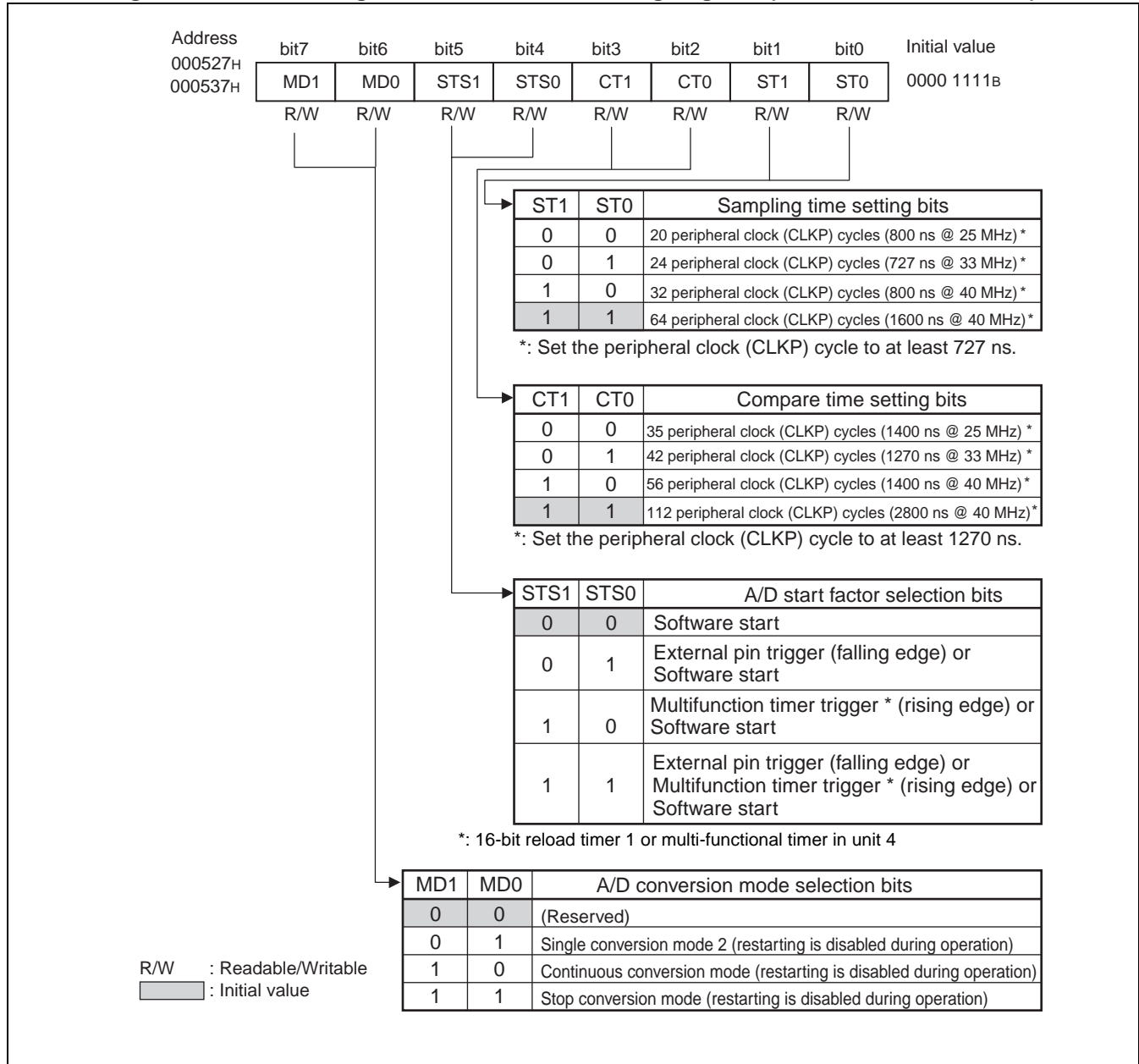


Table 18.4-2 Functions of Each Bit in A/D Mode Setting Register (ADMD) (1 / 2)

Bit Name	Function
bit7, bit6 MD1, MD0: A/D conversion mode selection bits	<ul style="list-style-type: none"> These bits are used to select the conversion mode during the A/D conversion operation. Two bit values of MD1 and MD0 allow the selection of either the single conversion mode 2, the continuous conversion mode, or the stop conversion mode. The meaning of each mode is as follows. <p>Single conversion mode 2: The continuous A/D conversion from the setting channels of ANS1 to ANS0 to the setting channels of ANE1 to ANE0 is performed only once. It is not possible to reactivate while operating.</p> <p>Continuous conversion mode: Sequentially perform A/D conversion from the channel set in ANS1 to ANS0 to the channel set in ANE1 to ANE0, and repeat until forcibly stopped by means of the BUSY bit. It is not possible to reactivate while operating.</p> <p>Stop conversion mode: Perform A/D conversion from the channel set in ANS1 to ANS0 to the channel set in ANE1 to ANE0 one channel at a time, pausing between each, and repeat until forcibly stopped by means of the BUSY bit. It is not possible to reactivate while operating. The suspended conversion is restarted by the start factor occurrence selected by the STS1 and STS0 bits.</p> <p>Notes:</p> <ul style="list-style-type: none"> Single, continuous, and stop conversion modes that cannot be re-started can be used to activate all timers, external triggers, and software. Only rewrite these bits before conversion begins, with the A/D operation stopped. It is not possible to reactivate while converting this 12 bits ADC.
bit5, bit4 STS1, STS0: A/D start factor selection bits	<ul style="list-style-type: none"> The start factor of A/D conversion is selected. When the start factor is in the common use, the first start factor generation starts the operation. <p>Notes:</p> <ul style="list-style-type: none"> The activation trigger changes as soon as the bits are rewritten, so if you wish to rewrite them while A/D conversion is ongoing, switch to a state where your target activation trigger does not exist. The timer cannot start when it is STS1, STS0=11_B, and the external trigger input is "L". And, an external trigger cannot start when the timer is "H".
bit3, bit2 CT1, CT0: Compare time setting bits	<p>These bits are used to select the comparison time at the A/D conversion.</p> <ul style="list-style-type: none"> After analog input is loaded (after the sampling time has elapsed), then after the time specified in these bits has passed, the conversion results are checked, and stored in the A/D data register (ADCD). <p>Notes:</p> <ul style="list-style-type: none"> Set for the compare time to become 1270ns or more. The normal analog conversion value might not be obtained with 1270ns or less. Only rewrite these bits before conversion begins, with the A/D operation stopped.

Table 18.4-2 Functions of Each Bit in A/D Mode Setting Register (ADMD) (2 / 2)

Bit Name		Function
bit1, bit0	ST1, ST0: Sampling time setting bits	<p>These bits are used to select the sampling time at the A/D conversion. When A/D is activated, analog input is retrieved for the time set in these bits.</p> <p>Notes:</p> <ul style="list-style-type: none"> • Set for the sampling time to become 727ns or more. The normal analog conversion value might not be obtained with 727ns or less. • Only rewrite these bits before conversion begins, with the A/D operation stopped.

18.4.3 A/D Control Status Register (ADCS)

The A/D control status register has features to suspend and confirm conversion, enable/disable interrupt requests, confirm the status of interrupt requests, and select the analog input mode (normal/differential) and the conversion function (function 1/function 2).

■ A/D Control Status Register (ADCS: ADCS3, ADCS4)

Figure 18.4-4 Bit Configuration of A/D Control Status Register (ADCS)

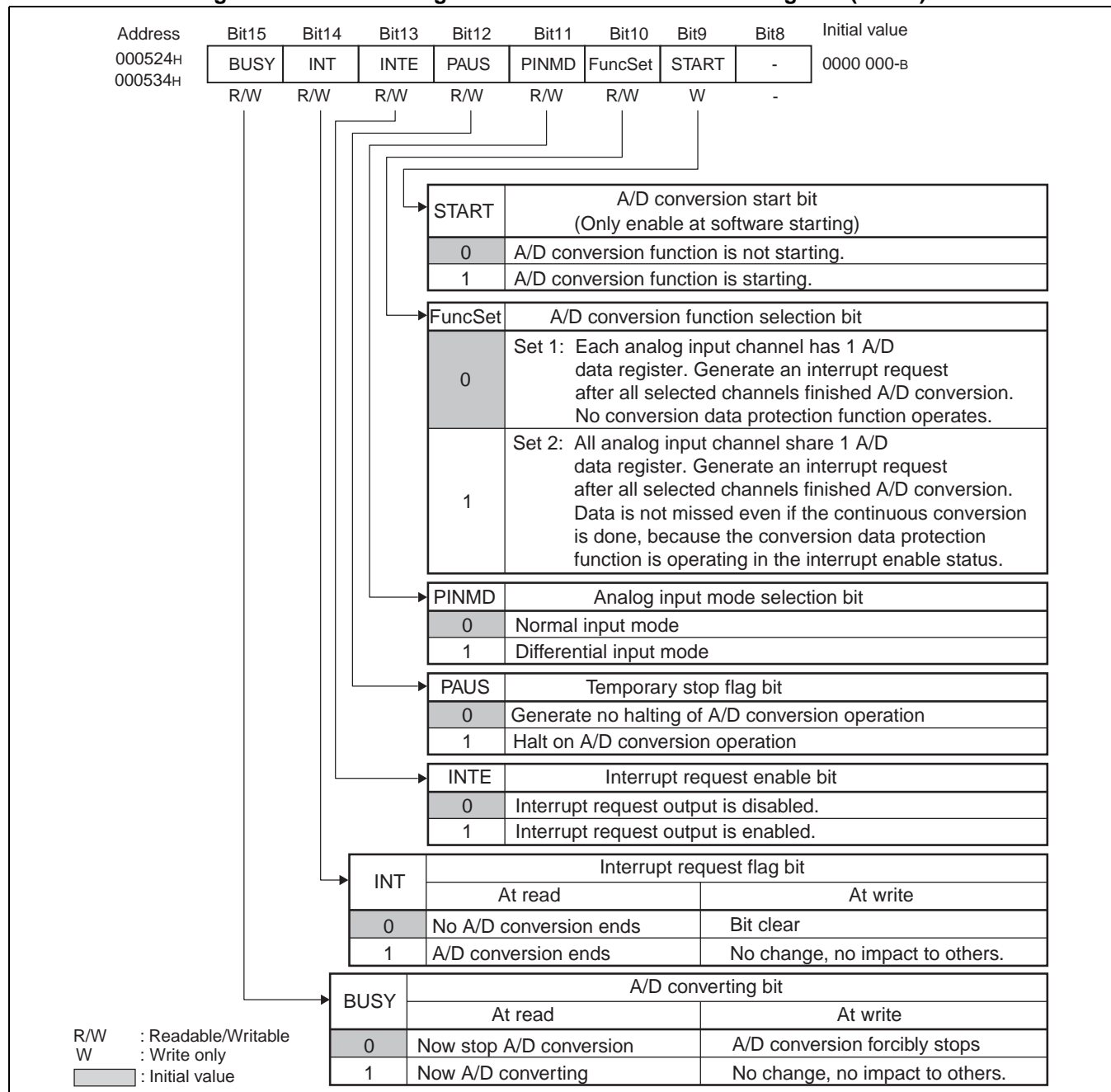


Table 18.4-3 Function of Each Bit in A/D Control Status Register (ADCS) (1 / 2)

Bit Name		Function
bit15	BUSY: A/D converting bit	<ul style="list-style-type: none"> Operational display bit of A/D converter When reading, if this bit is "0", it indicates that A/D conversion is stopped. If it is "1", it indicates that A/D conversion is ongoing. When writing, writing "0" to this bit forcibly stops A/D conversion. When "1" is written, the conversion is not changed and no others are affected. "1" is always read during read modify write (RMW) instruction. <p>Note:</p> <p>Do not perform a forced stop and software activation (BUSY = 0, START = 1) at the same time.</p> <p>Execute after waiting for 2μs or more without fail when you reactivate the A/D conversion again after the A/D conversion operation compulsion stops.</p>
bit14	INT: Interrupt request flag bit	<ul style="list-style-type: none"> This INT bit is set to "1" if data is set in A/D data register through A/D conversion. If this bit and the interrupt request enable bit (ADCS: INTE) are "1", an interrupt request is generated. When writing, "0" clears this bit, and with "1" no change is made, and there are no other effects. "1" is always read during read modify write (RMW) instruction. <p>Note:</p> <p>Clear this bit by writing "0" with the A/D operation has stopped.</p>
bit13	INTE: Interrupt request enable bit	<ul style="list-style-type: none"> This bit is used to enable and disable the interrupt output to CPU. If this bit and the interrupt request flag bit (ADCS: INT) are "1", an interrupt request is generated.
bit12	PAUS: Temporary stop flag bit	<ul style="list-style-type: none"> It is set to 1 when A/D conversion is suspended. This bit is set to "1" automatically when A/D enters conversion data protection function. During this time, A/D conversion will halt and A/D data register will not be overwritten by any new coming data. Clear this flag only by writing "0" to this register. "1" is always read during read modify write (RMW) instruction. Refer to section "18.7 A/D conversion Data Protection Function the 12-bit A/D Converter" for detail operation.
bit11	PINMD: Analog input mode selection bit	<ul style="list-style-type: none"> This bit is used to select the mode of analog input to be normal or differential input. 0: All analog inputs are input to (+) side of the A/D conversion part at the normal input mode, and (-) side is fixed to the AVSS12 side internally. 1: AN3-0/AN4-0, AN3-1/AN4-1, AN3-2/AN4-2, and AN3-3/AN4-3 become pairs at a differential input mode and it is input to (+)(-) side respectively. <p>Note:</p> <p>Only rewrite this bit before conversion begins, with A/D operation stopped</p>

Table 18.4-3 Function of Each Bit in A/D Control Status Register (ADCS) (2 / 2)

Bit Name		Function
bit10	FuncSet: A/D conversion function selection bit	<ul style="list-style-type: none"> This bit is used to select the function set for A/D conversion. Write "0" to this bit select function set 1: <ul style="list-style-type: none"> Each input channel have 1 data register Generate interrupt after all selected channel finished conversion No data protection Write "1" to this bit select function set 2: <ul style="list-style-type: none"> All input channel share 1 data register Generate interrupt after each channel finished conversion Data protection when interrupt request enabled <p>Note: Only rewrite this bit before conversion begins, with A/D operation stopped</p>
bit9	START: A/D conversion start bit (Effective only that software starts)	<ul style="list-style-type: none"> This bit is used to start the A/D conversion operation by the software. Write "1" to this bit to activate A/D conversion. Restart by this bit cannot be used at the stop conversion mode. "0" is always read in read modify write (RMW) instruction. <p>Note: Do not perform a forced stop and software activation (BUSY = 0, START = 1) at the same time.</p>
bit8	Undefined bit	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect to operation.

18.4.4 A/D Data Register (ADCD)

The A/D data register stores A/D conversion results.

■ A/D Data Register (ADCD: ADCD003 to ADCD033, ADCD004 to ADCD034)

Figure 18.4-5 Bit Configuration of A/D Data Register (ADCD: ADCD003 to ADCD033, ADCD004 to ADCD034)

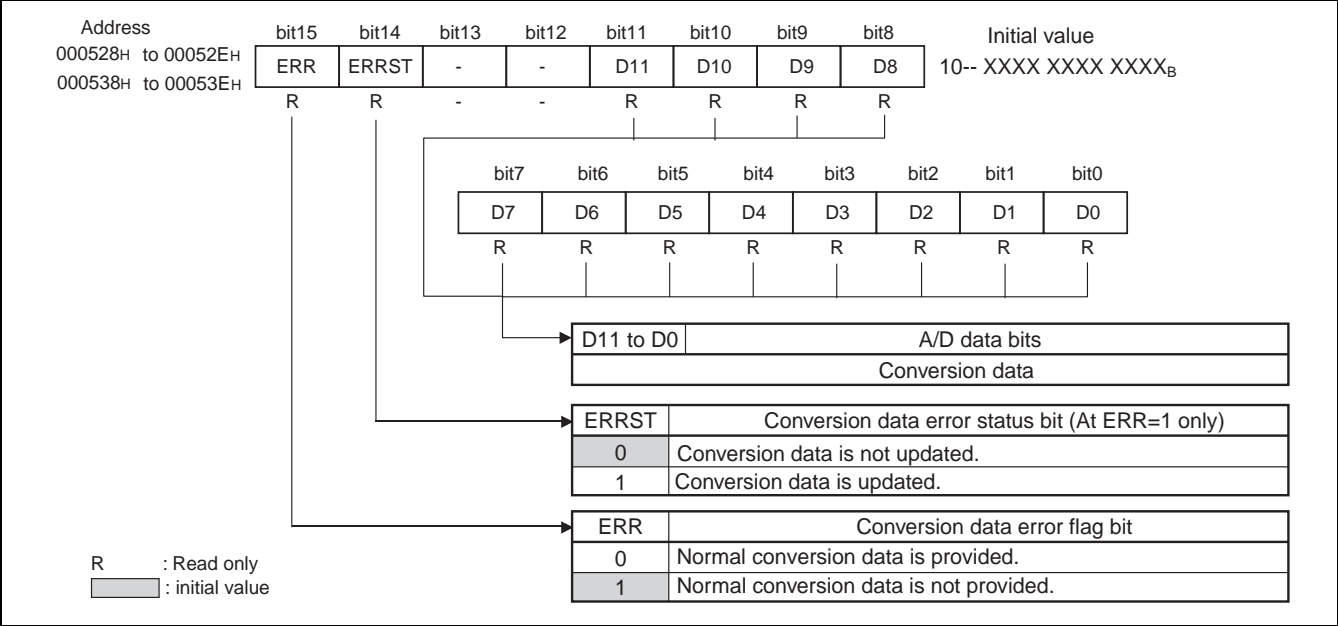


Table 18.4-4 Function of Each Bit in A/D Data Register (ADCD)

Bit Name		Function
bit15	ERR: Conversion data error flag bit	<ul style="list-style-type: none"> When this bit is "1", the content of the error can be known of by the value of the ERRST bit in the bit that shows that there was an error in the A/D conversion data. This bit is set to "1" when reading it. When a new conversion result is written in this register, it is cleared to "0". <p>Note: When conversion data protection function is used (FuncSet=1 and INTE=1), this bit is always "0. "</p>
bit14	ERRST Conversion data error status bit	<ul style="list-style-type: none"> It is a flag that shows the content of the error of the A/D conversion data at ERR bit =1. It is shown that the conversion result by CPU reading is old at ERR bit =1 and this bit =0. The conversion result by CPU reading shows that the old conversion data was lost from the overwriting of a new conversion result without completing reading the old conversion result with CPU at ERR bit =1 and this bit =1. When the old conversion data is lost from the overwriting of a new conversion result without completing reading the old conversion result with CPU, it is set in "1". This bit is cleared to "0" when reading it. <p>Note: When conversion data protection function is used (FuncSet=1 and INTE=1), this bit is always "0. "</p>
bit13, bit12	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect to operation.
bit11 to bit0	D11 to D0: A/D Data bits	<ul style="list-style-type: none"> The result of the A/D conversion is stored, and the register is rewritten each time conversion is completed. The final conversion value is stored usually. The initial value of this register is undefined. <p>Notes:</p> <ul style="list-style-type: none"> The conversion data protection function is provided. Do not write data to these bits while A/D conversion is ongoing.

MB91470/480 Series

18.4.5 Analog Input Control Register (AICR)

The analog input control register controls analog input.

■ Analog Input Control Register (AICR: AICR3, AICR4)

Figure 18.4-6 Bit Configuration of Analog Input Control Register (AICR: AICR3, AICR4)

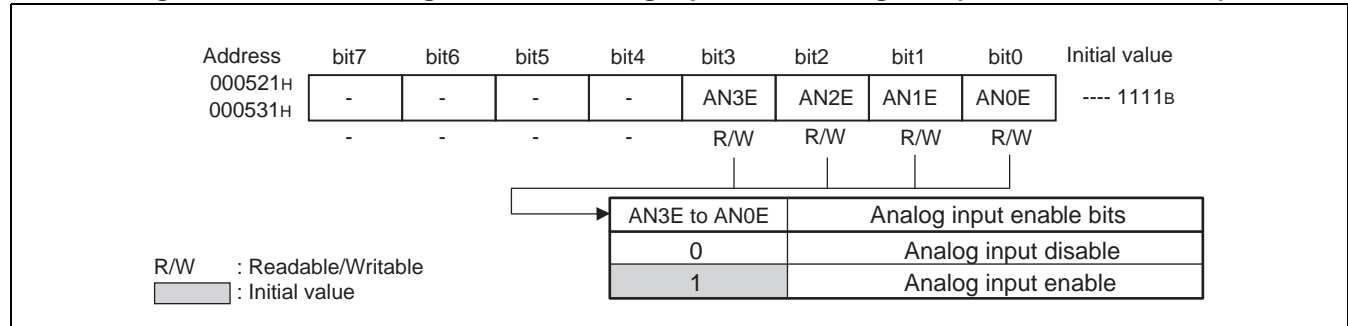


Table 18.4-5 Functions of Each Bit in Analog Input Control Register (AICR)

Bit Name		Function
bit7 to bit4	Undefined bits	<ul style="list-style-type: none"> The read value is undefined. Writing to these bits has no effect to operation.
bit3 to bit0	AN3E to AN0E: Analog input enable bits	<ul style="list-style-type: none"> When these bits are "0", analog input is disabled. When these bits are "1", analog input is enabled. Set the AICR register bit corresponding to the pin to be used as the analog input pin to "1". When this is done, the value "0" will be read from the PDR register.

18.5 Interrupt the 12-bit A/D Converter

The 12-bit A/D converter can generate interrupt requests during A/D conversion by setting data in the A/D data register.

■ Interrupt of 12-bit A/D Converter

See Table 18.5-1 for the interrupt control bits and interrupt cause of the 12-bit A/D converter.

Table 18.5-1 Interrupt Control Bits and Interrupt Cause of 12-bit A/D Converter

	12-bit A/D converter
Interrupt request flag bit	ADCS: INT
Conversion function selection bit	ADCS: FuncSet
Interrupt request enable bit	ADCS: INTE
Interrupt cause	Writing of A/D conversion result to A/D data register

When FuncSet = 0, interrupt could be generated after all selected channel conversion completed. The INT bit of the A/D control status register (ADCS) is set to "1" when all the conversion end and A/D conversion results are set in the A/D data register (ADCD). At this time, an interrupt request is generated to the interrupt controller if interrupt request is enabled (ADCS:INTE=1).

When FuncSet = 1, interrupt could be generated after each channel conversion completed. The INT bit of the A/D control status register (ADCS) is set to "1" when each conversion ends and A/D conversion results are set in the A/D data register (ADCD). At this time, an interrupt request is generated to the interrupt controller if interrupt request is enabled (ADCS:INTE=1).

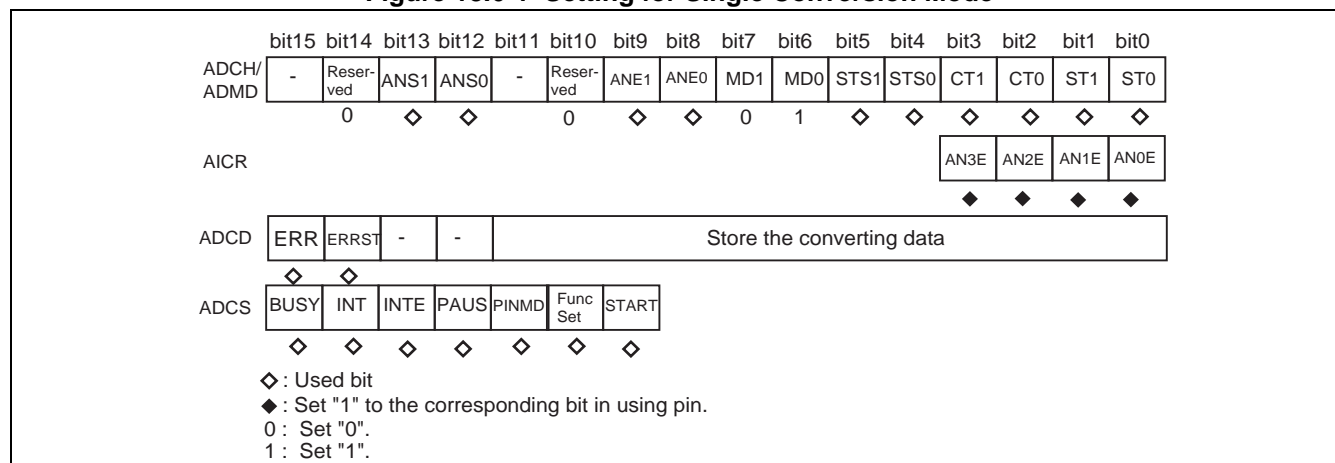
18.6 Operation Explanation the 12-bit A/D Converter

Three mode types, the single conversion, continuous conversion, and stop conversion modes are available for the 12-bit A/D converter. The operation explanation in each mode is done.

■ Operation of Single Conversion Mode

In single conversion mode, it sequentially converts the analog input which has been set by the ANS bit and ANE bit, and when it reaches to the end channel set in ANE bit, it stops the A/D conversion. If the start channel and end channel are the same (ANS=ANE), only one channel specified in the ANS bit will be converted. The settings in Figure 18.6-1 are required in order to operate in single-conversion mode.

Figure 18.6-1 Setting for Single Conversion Mode



Reference:

The example of conversion order in single conversion mode is shown in following.

When ANS=000_B, ANE=011_B: AN0 → AN1 → AN2 → AN3 → end

(FuncSet = 0) ADCD00 → ADCD01 → ADCD02 → ADCD03 → end

(FuncSet = 1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → end

When ANS=011_B, ANE=011_B: AN3 → end

(FuncSet = 0) ADCD03 → end

(FuncSet = 1) ADCD03 → end

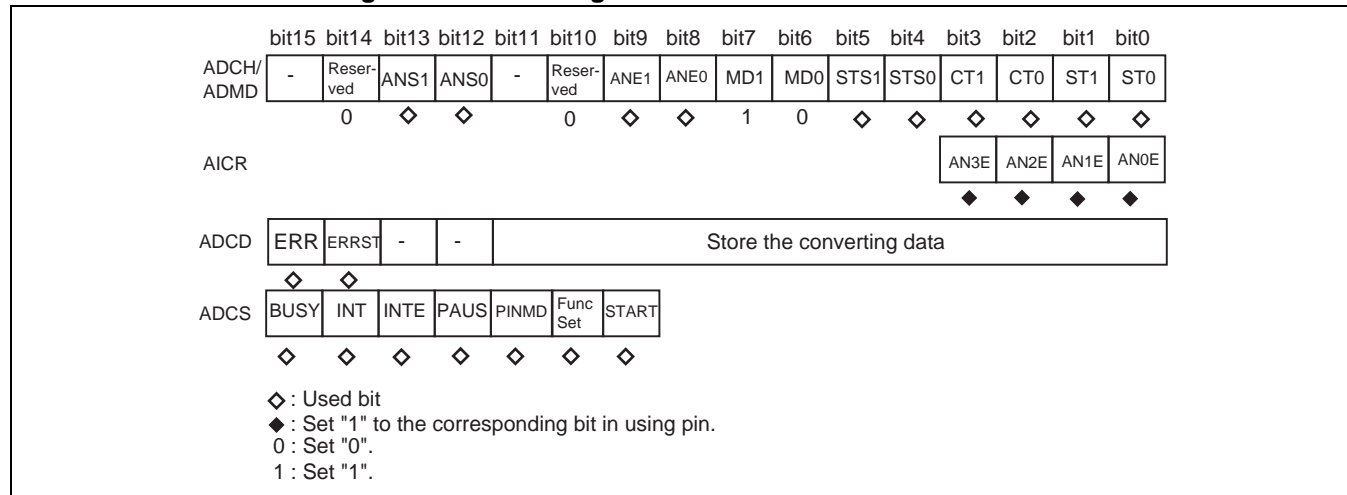
Note:

Be sure to set to become "ANS ≤ ANE".

■ Operation of Continuous Conversion Mode

In the continuous conversion mode, the analog inputs set by the ANS and ANE bits are sequentially converted, the analog input set by the ANS bit is resumed at the end of conversion of the end channel set by the ANE bit, and the A/D conversion operation is continued. If the start channel and end channel are identical (ANS=ANE), conversion loops on the channel specified by ANS only. The settings shown in Figure 18.6-2 are required in order to operate in continuous conversion mode.

Figure 18.6-2 Setting at Continuous Conversion Mode



Reference:

The example of conversion order in continuous conversion mode is shown in following.

When ANS=000_B, ANE=011_B: AN0 → AN1 → AN2 → AN3 → Repeat

(FuncSet =0) ADCD00 → ADCD01 → ADCD02 → ADCD03 → Repeat

(FuncSet =1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → Repeat

When ANS=011_B, ANE=011_B: AN3 → AN3 → AN3 → AN3 → Repeat

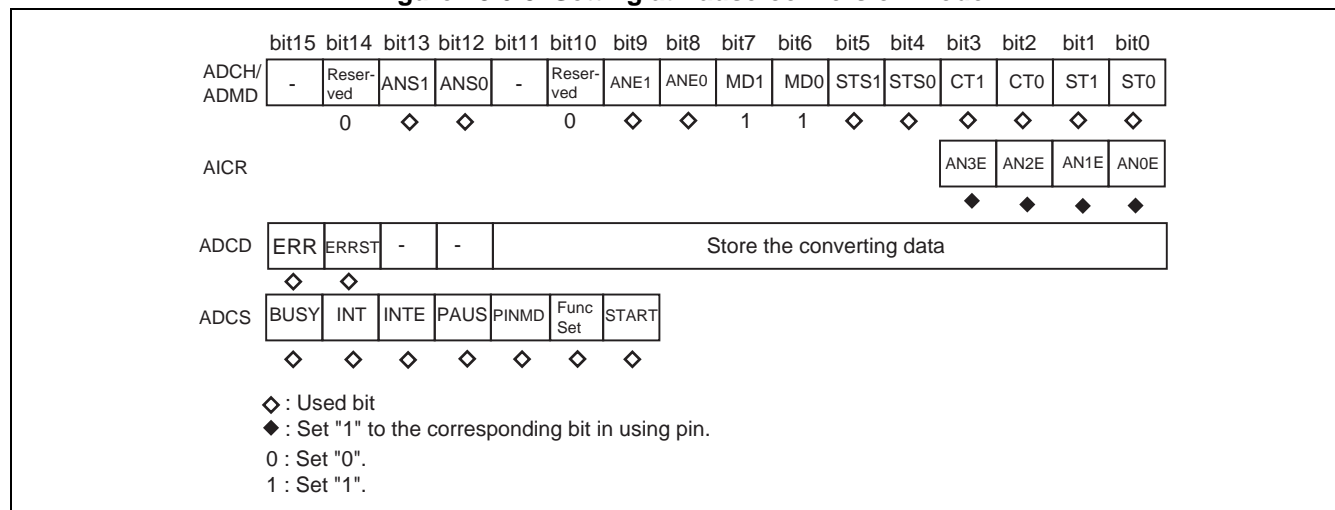
(FuncSet =0) ADCD03 → ADCD00 → ADCD01 → ADCD02 → Repeat

(FuncSet =1) ADCD00 → ADCD00 → ADCD00 → ADCD00 → Repeat

■ Operation of Pause-Conversion Mode

In the stop conversion mode, the analog input set by the ANS and ANE bits is converted by being suspended for every channel, the analog input set by the ANS bit is resumed at the end of conversion of the end channel set by the ANE bit, and the operation of A/D conversion and suspension is continued. If the start channel and end channel are identical (ANS=ANE), conversion loops on the channel specified by the ANS bits only. When the conversion is restarted during the suspension, the start factor specified by the STS1 and STS0 bits is generated. The settings in Figure 18.6-3 are required in order to operate in stop conversion mode.

Figure 18.6-3 Setting at Pause-conversion Mode



Reference:

The example of conversion order in stop conversion mode is shown in following.

- When ANS=000_B, ANE=011_B:

AN0 → Pause → AN1 → Pause → AN2 → Pause → AN3 → Repeat

FuncSet = 0

ADCD00 → Pause → ADCD01 → Pause → ADCD02 → Pause → ADCD03 → Repeat

FuncSet = 1

ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Repeat

- When ANS=011_B, ANE=011_B:

AN3 → Pause → AN3 → Pause → AN3 → Pause → AN3 → Repeat

FuncSet = 0

ADCD03 → Pause → ADCD00 → Pause → ADCD01 → Pause → ADCD02 → Repeat

FuncSet = 1

ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Pause → ADCD00 → Repeat

18.7 A/D conversion Data Protection Function the 12-bit A/D Converter

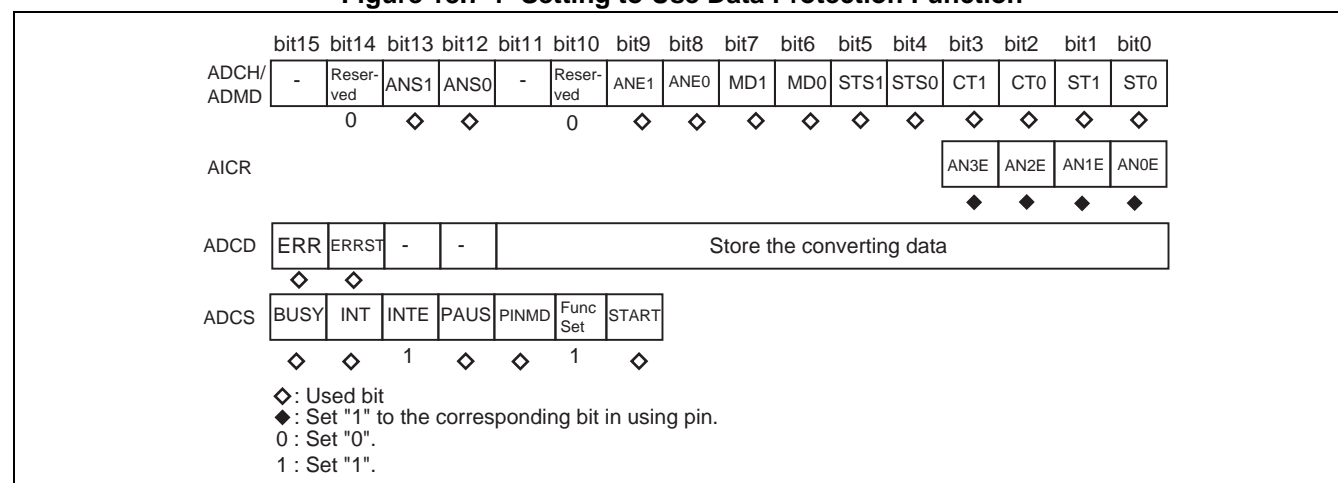
When the A/D conversion operates in the interrupt enabled state, the conversion data protection function works.

■ A/D Conversion Data Protection Function

When selected ADCS:FuncSet=1 (conversion function 2), there is only 1 data register for storing conversion result. For this reason, when performing A/D conversion, the data stored in data register is rewritten after conversion is completed. Therefore, part of previous data may be lost when the converted data transfer to memory is delayed. To get around this, when interrupt is enabled (ADCS:INTE=1), data protection feature works as described below.

When conversion data is stored in the A/D data register (ADCD), the ADCS:INT bit set to 1. While the ADCS:INT bit is 1, conversion data will not be stored to ADCD after the next conversion ends. The ADCS:PAUS bit is set and A/D conversion becomes suspended. While suspended, the value immediately prior is retained. In order to cancel the suspend, clear ADCS:INT bit. After the suspended status is cleared, the conversion data that had been maintained is stored in ADCD and the next operation is performed.

Figure 18.7-1 Setting to Use Data Protection Function



Notes:

- Conversion data protection function only operates in conversion function 2 (ADCS:FuncSet=1) and interrupt enabled (ADCS:INTE=1).
- When the conversion is restarted during suspension, the waiting data is destroyed.

18.8 Differential input mode the 12-bit A/D Converter

There are 2 analog input modes in 12-bit A/D converter. Set ADCS register to determine normal input mode or differential input mode. This section describes the usage of differential input mode.

■ Analog Input Pin in Differential Input Mode

When 12-bit A/D converter operates in differential input mode, 2 analog input pins could become 1 pair of analog input to the A/D converter. The number of analog input channels becomes half.

[At Normal input mode]		[At differential input mode]	
AN3-0	→	AN3-0P	(+) side input
AN3-1	→	AN3-0N	(-) side input
AN3-2	→	AN3-1P	(+) side input
AN3-3	→	AN3-1N	(-) side input

Please notice the max voltage for (-) side analog input should be $AVCC12/2$.

■ Register Setting in Differential Input Mode

When ADCS:PINMD is set to "1", the 12-bit A/D converter operates in differential input mode. Setting in other registers include: ADCH, AICR and ADCD should notice as well.

- Register ADCH

In differential input mode, number of input channel reduces from 4 to 2. So ANS[0] & ANE[1] are ignored in differential input mode. Use ANS[1] & ANE[1] to indicate the start channel and end channel.

- Register AICR

Be sure to set "1" to the analog input enable bit for (+) side input and (-) side input.

- Register ADCD

Conversion data is stored in the A/D data register corresponding to the (+) side analog input channel. The A/D data register corresponding to (-) side analog input channel is skipped.

18.9 Using Memorandum of the 12-bit A/D Converter

This section describes a memorandum when 12-bit A/D converter is used.

■ Register ADMD Setting

The A/D converter sampling time and compare time based on three types of frequencies (25MHz, 33MHz, 40MHz) can be set by the ADMD register. The minimum conversion time corresponding to each frequency can be set. Set ADMD by the following two methods when a set frequency is different from three above-mentioned types of values.

- ST[1:0]/CT[1:0] bit (bit3, bit2/bit1, bit0) of the ADMD register is set so that neither the sampling time nor the compare time may become below recommended value of 12-bit A/D converter.
- Set P3-to-P0 bit of DIVR0 register (bit3-to-bit0) so that the frequency of the peripheral clock (CLKP) may become one of the above-mentioned.

■ Examples

- When the peripheral clock (CLKP) frequency is 16MHz: Method 1

Cycle = 62.5ns

Sampling time: ST[1:0] = 00_B,

→ 20 peripheral clock (CLKP) cycles

→ $62.5\text{ns} \times 20 = 1250\text{ns} > 727\text{ns}$ (minimum)

Compare time: CT[1:0] = 00_B,

→ 35 peripheral clock (CLKP) cycles

→ $62.5\text{ns} \times 35 = 2190\text{ns} > 1270\text{ns}$ (minimum)

∴ total conversion time = 3500ns

- When the peripheral clock (CLKP) frequency is 25MHz: Method 2

Source oscillation frequency: 10MHz

PLL multiplication rate: $\times 5$ -multiplier

DIVR0: P3 to P0 = 0001_B

→ $\text{CLKP} = 10 \times 5 / 2 = 25\text{MHz}$

ST[1:0] = 00_B, CT[1:0] = 00_B

∴ Total conversion time = 2200ns

18.10 Notes on Using the 12-bit A/D Converter

This section describes notes on using 12-bit A/D converter.

■ Notes on Using 12-bit A/D Converter

● Analog input pin

The A/D input pin does double duty as a port I/O pin. The port-direction register (DDR) and analog input enable register (AICR) are switched and used. For the pins used as analog input, set the bits corresponding to DDR to 0 and port to input, then set AICR register analog input mode (AICRx = 1). Lock the input gate on the port side. When the intermediate level signal is inputted in the port input mode (AICRx = 0), the input leak current flows through the gate.

● Cautions for use of internal timers

To activate the A/D converter by an internal timer, set the STS1 and STS0 bits of the A/D control status register (ADMD). When doing so, set the internal timer input value to the inactive side (for internal timer, this is "L"). If you set it to the active side, the timer may start to operate as soon as you write to the ADMD register.

● Turning-on sequence of power supply to A/D converter and analog inputs

Make sure to apply to the A/D converter power source (AVCC12, AVRH 3, AVRH 4, and AVSS12) and apply analog input (AN3-0 to AN3-3, AN4-0 to AN4-3) after or at the same time as applying digital power source (VCC). When cutting off the power, cut off the digital power source (VCC) after or at the same time as cutting off the A/D converter power source and analog input.

● Power voltage of A/D converter

In order to prevent latch-ups, make sure that the A/D converter power source (AVCC12) does not exceed the voltage of the digital power source (VCC).

● Setting of ADCH register

Set to become $ANS \leq ANE$.

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Set "0" to the reservation bit.

● Setting of ADMD register

Set ST[1:0]/CT[1:0] bit (bit3, bit2/bit1, bit0) so that neither the time of the sample nor the compare time may become below recommended value of 8/10-bit A/D converter.

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Refer to Section "18.9 Using Memorandum of the 12-bit A/D Converter" for a detailed explanation.

● Setting of ADCS register

Rewrite the bit with the A/D operation before conversion operates without fail has stopped.

Do neither A/D conversion starting setting (START=1) with software nor stop setting (BUSY=1) at the same time.

Refer to Section "18.4 Registers the 12-bit A/D Converter" for a detailed explanation.

● Notice in A/D conversion data protection function

Conversion data protection function is only available in conversion function 2 (ADCS:FuncSet=1) and interrupt enabled (ADCS:INTE=1).

● Flag bit in A/D data register

Even if eight bits of the A/D data register on the subordinate position side are read by byte access, neither the ERRST nor ERR bits are changed.

Moreover, when the conversion data protection function is used, the ERRST and ERR bits are always "0".

● External trigger terminal

Return the input level of the external trigger terminal to former level by the external trigger terminal after the activating of the A/D converter.

● Analog input voltage level in differential input mode

Never set the voltage in negative analog input channel larger than $AVCC12/2$. Voltage in negative analog input should be less than positive analog input. Otherwise, there is error in A/D converter and may occur very large current flow.

● Differential input mode register setting

When 12-bit A/D converter operates in differential input mode, make sure the setting in each registers are correct. Please refer to Section "18.8 Differential input mode the 12-bit A/D Converter" for details

● Return from state of standby

12-bit A/D converter automatically changes to the standby mode when the device changes to the STOP mode. In this case, the oscillation stabilization wait time of 5 μ s is necessary by the return from the STOP mode, and entering the state that 12-bit A/D converter can work. So, use 12-bit A/D converter to return from the STOP mode after the oscillation stabilization wait time of 5 μ s or more passes without fail.

● Reactivation after A/D conversion operation stops

Execute after waiting for 2 μ s or more without fail when you reactivate the A/D conversion again after the A/D conversion operation compulsion stops.

CHAPTER 19

MULTIPLICATION AND ADDITION CALCULATOR

This chapter outlines the multiplication and addition calculator and describes its register configuration/function and operations.

- 19.1 Overview of the Multiplication and Addition Calculator
- 19.2 Block Diagram of the Multiplication and Addition Calculator
- 19.3 Instruction Definitions of the Multiplication and Addition Calculator
- 19.4 Register List of the Multiplication and Addition Calculator
- 19.5 Register Description of the Multiplication and Addition Calculator
- 19.6 Principles of Operation of the Multiplication and Addition Calculator
- 19.7 Details on Instructions of the Multiplication and Addition Calculator
- 19.8 Notes on Using the Sum-of-Products Circuit

19.1 Overview of the Multiplication and Addition Calculator

This section provides an overview of the multiplication and addition calculator.

■ Overview

- High-speed multiplication and addition calculation (7-stage pipelining)
- Data format : 32-bit fixed-point ($32 \times 32 + 72$ bits)
- Instruction area : I-RAM 256×16 bits
- Data area : X-RAM 64×32 bits
Y-RAM 64×32 bits
- Capable of rounding
- Saturation supported
- Number of terms to be added : Up to 64
- Instructions : MAC, STR, JMP, and NOP instructions
- Delayed processing : Free transfer within 64 words
- Fixed-point format : Selectable from among Q0 and Q26 to Q32
- Program execution control : Capable of selecting eight different instructions
- Variable monitor : Capable of monitoring up to 8 words of calculation results without halting the program
- Arithmetic result storage function : Built-in registers for storing arithmetic results (72 accumulator bits) when an operation is halted. Readable from the CPU.

MB91470/480 Series

19.2 Block Diagram of the Multiplication and Addition Calculator

This section describes a block diagram of the multiplication and addition calculator.

■ Block Diagram

Figure 19.2-1 Block Diagram of Multiplication and Addition Calculator

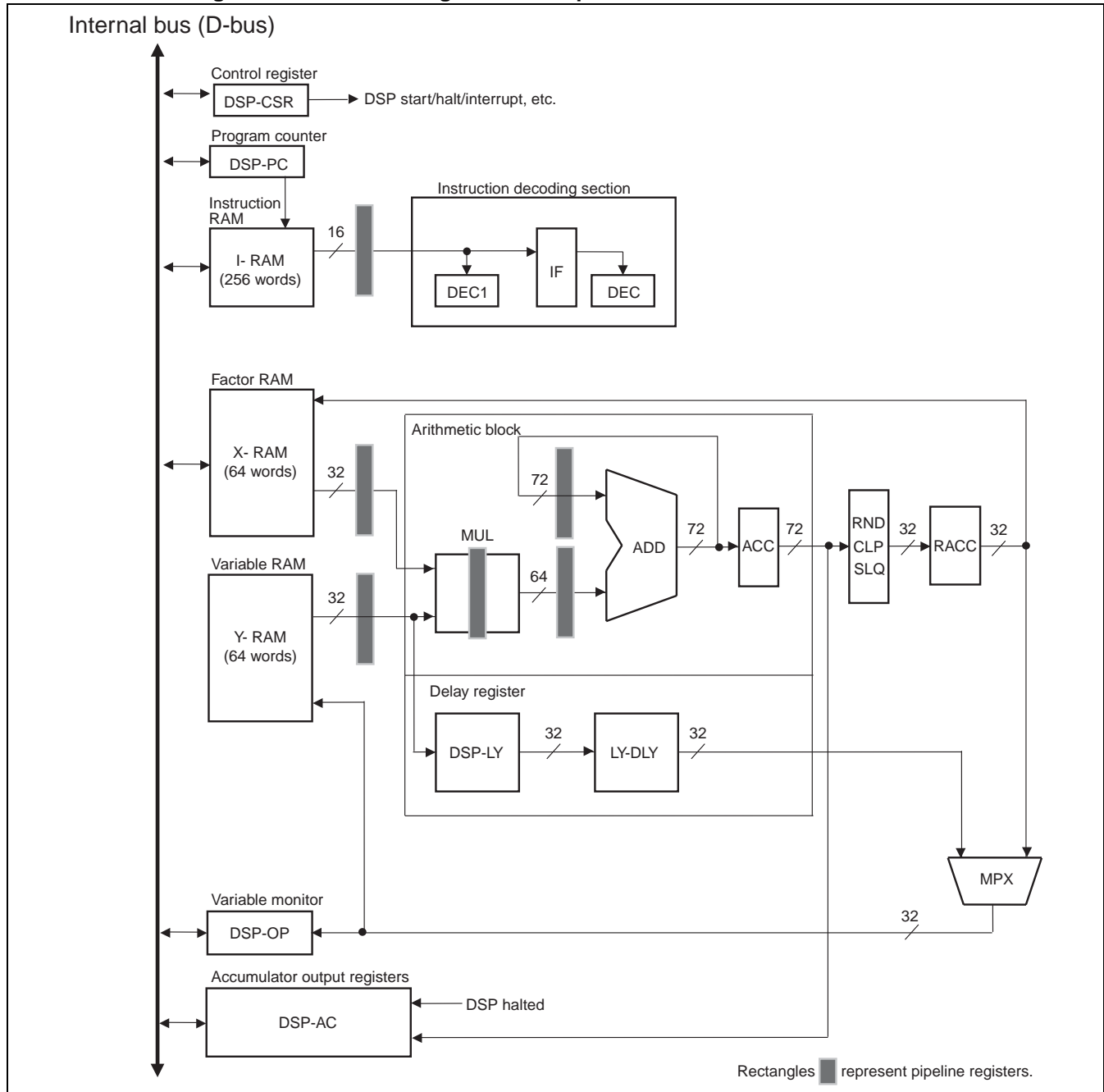


Table 19.2-1 Block Functions of Multiplication and Addition Calculator

Block	Register	Function
Operation control	DSP-CSR	Operation control register for the multiplication and addition macro, used to control the following operations from the CPU: <ul style="list-style-type: none"> • Instruction to start/end calculation • Interrupt control • Program flow control (used for conditional branch instructions of the multiplication and addition macro)
Instruction control	DSP-PC	Program counter Starts program execution at the start address set from the CPU.
	I-RAM	256 by 16 bits of instruction RAM. Readable/writable from the CPU while the multiplication and addition macro is not performing calculation. During multiplication and addition operation, I-RAM is accessed in 16 bits. From the CPU, it is accessible in bytes, halfwords, or words. Before starting calculation, load instruction code from the CPU.
	IF	Instruction fetch register
	DEC1* DEC*	Instruction decoders
Arithmetic block	X-RAM	64 by 32 bits of data RAM. Readable/writable from the CPU while the multiplication and addition macro is not performing calculation. During multiplication and addition operation, X-RAM is accessed in 32 bits. From the CPU, it is accessible in bytes, halfwords, or words. Before starting calculation, load coefficients from the CPU.
	Y-RAM	64 by 32 bits of data RAM. Readable/writable from the CPU while the multiplication and addition macro is not performing calculation. During multiplication and addition operation, Y-RAM is accessed in 32 bits. From the CPU, it is accessible in bytes, halfwords, or words. Before starting calculation, load variables from the CPU.
	MUL*	32x32 →64-bit multiplier.
	ADD*	64+72 →72-bit adder.
	ACC*	72-bit accumulator.
	CLP*	Saturates out-of-range 32-bit data to the maximum value during 72-to-32-bit transfer.
	RND* SLQ*	Rounds least significant bits during 72-to-32-bit transfer. Selects the transfer bit for 72-to-32-bit transfer.
Delay registers	DSP-LY LY-DLY*	Delay registers. Hold variables during multiplication and addition operation so that they can be written back to Y-RAM.
Variable monitor outputs	DSP-OT0 to DSP-OT7	Variable monitor output registers. Hold the same values as those at addresses 0 to 7 in Y-RAM. Capable of monitoring the values at Y-RAM addresses 0 to 7 during calculation (with Y-RAM access-barred).
Accumulator output registers	DSP-AC0 to DSP-AC2	Accumulator output registers. These registers contain 72 bits of accumulator value when multiplication and addition operation is halted. The register values can be read from the CPU.

*: Inaccessible from the CPU

19.3 Instruction Definitions of the Multiplication and Addition Calculator

The multiplication and addition macro has four major types of instructions: MAC, STR, JMP, and NOP instructions.

■ Instruction Definitions

Instructions other than the four types of instructions of the multiplication and addition macro are also used for notational purposes. Those instructions are classified hierarchically as shown below:

- MAC instructions
 - _____ Multiplication and addition instruction (CLAC bit = 0)
 - _____ Multiply instruction (CLAC bit = 1)
- STR instruction
- JMP instructions
 - _____ Unconditional branch instruction (COND bit = 0)
 - _____ Conditional branch instruction (COND bit = 1)
- NOP instructions with repeat function
 - _____ HLT instruction (HLT bit = 1)
 - _____ INT instruction (SIRQ bit = 1)
 - _____ Repeat instruction

19.4 Register List of the Multiplication and Addition Calculator

This section lists the registers of the multiplication and addition calculator.

■ Register List of Multiplication and Addition Calculator

Figure 19.4-1 Register List of Multiplication and Addition Calculator

	bit31 to bit24	bit23 to bit16	bit15 to bit8	bit7 to bit0	
Address: 0003A0 _H	DSP-PC	DSP-CSR	-		R/W
Address: 0003A4 _H	DSP-LY (Delay register)				R/W
Address: 0003A8 _H	DSP-OT0 (Output queue 0)				R
Address: 0003AC _H	DSP-OT1 (Output queue 1)				R
Address: 0003B0 _H	DSP-OT2 (Output queue 2)				R
Address: 0003B4 _H	DSP-OT3 (Output queue 3)				R
Address: 0003B8 _H	DSP-OT4 (Output queue 4)				R
Address: 0003BC _H	DSP-OT5 (Output queue 5)				R
Address: 0003C0 _H	DSP-OT6 (Output queue 6)				R
Address: 0003C4 _H	DSP-OT7 (Output queue 7)				R
Address: 0003C8 _H	DSP-AC0 (Accumulator output) bit71 to bit64				R
Address: 0003CC _H	DSP-AC1 (Accumulator output) bit63 to bit32				R
Address: 0003D0 _H	DSP-AC2 (Accumulator output) bit31 to bit0				R

Address:		multiplication and addition macro	Address
00C000 _H	X-RAM (Coefficient RAM)...64 × 32 bits	00 _H	R/W
:		:	
00C0FF _H		3F _H	
Address:		multiplication and addition macro	Address
00C100 _H	Y-RAM (Variable RAM)...64 × 32 bits	00 _H	R/W
:		:	
00C1FF _H		3F _H	
Address:		multiplication and addition macro	Address
00C200 _H	I-RAM (Instruction RAM)...256 × 16 bits	00 _H	R/W
:		:	
00C3FF _H		FF _H *	

*: As the instruction in I-RAM is executed in 16 bits during multiplication and addition operation, these addresses are 16 bits long. X-RAM/Y-RAM addresses are 32 bits long.

MB91470/480 Series

19.5 Register Description of the Multiplication and Addition Calculator

This section describes the configuration and functions of the registers used for the multiplication and addition calculator.

■ DSP Control/Status Register (DSP-CSR)

The control/status register is an 8-bit register containing the multiplication and addition macro state transition/CPU interrupt control bits and the multiplication and addition macro status display flags. This register is also used to set conditions for conditional branch instructions of the multiplication and addition macro.

8-bit register always readable/writable from external resources.

● Control functions

- State (calculation start) transition (GoDSP) of the multiplication and addition macro
- CPU interrupt mask (IeDSP)
- Setting conditions for conditional branch instructions of the multiplication and addition macro (USR2, USR1, USR0)

● Status functions

- Flag acquiring the current state of the multiplication and addition macro (RunDSP)
- Interrupt request flag (IrqDSP)
- Saturation flag (SatDSP)

Figure 19.5-1 DSP Control/Status Register (DSP-CSR)

DSP-CSR									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0003A1 _H	-	USR2	USR1	USR0	IrqDSP	IeDSP	-	GoDSP	000000-0 _B
	SatDSP	USR2	USR1	USR0	IrqDSP	IeDSP	-	RunDSP	
	R	R/W	R/W	R/W	R/W	R/W	W	R/W	

R/W: Readable/writable
R: Read only
W: Write only

[bit7] SatDSP (Saturation flag)

- This status flag indicates that saturation has been performed part of the way through calculation.
- The flag is set when saturation specified by the STR instruction (CLP = 1) is actually performed. Once set during a calculation, the flag maintains the value until the next calculation is started.
- This bit is cleared as calculation is started.
Set factor : The bit is set when saturation is performed in response to the STR instruction part of the way through calculation.
Clear factor : The bit is cleared as calculation is started. [Initial value]
- When reset : The bit is initialized to "0" (with no saturation involved).
- The bit is only readable. Writing to the bit leaves its value unchanged.

[bit6 to bit4] USR2, USR1, USR0 (Jump condition setting bits)

- These bits are referenced by the multiplication and addition macro's conditional branch instruction (COND bit = 1). A jump takes place when the value of these bits matches the UBP flag of the conditional branch instruction (when the condition is true). That is, eight types of calculation routines can be switched from the CPU by combining the conditional branch instruction with calculation instructions.
- When reset: The bits are initialized to "000_B".
- The bits are readable and writable.

Note:

Due to the instruction pipeline, the actual branch occurs after the two instructions that follow the branch instruction are executed. Between the two instructions, therefore, do not place any instruction other than NOP (REP = 001_B, HLT = 0, SIRQ = 0).

[bit3] IrqDSP (Interrupt request flag)

- This flag indicates that a software interrupt request has been generated by the multiplication and addition macro. When this bit is set with interrupt requests enabled (IeDSP = 1), an interrupt request to the CPU is issued.
- An interrupt request of the multiplication and addition macro is generated by means of software by setting the SIRQ bit of the NOP instruction to "1".
Set factor : The bit is set when a software interrupt (NOP instruction) of the multiplication and addition macro is generated.
Clear factor : The bit is cleared when "0" is written. [Initial value]
- When reset : The bit is initialized to "0" (with no interrupt request).
- The bit is readable and writable. However, only "0" can be written. An attempt to write "1" does not change the bit value.
- When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

[bit2] IeDSP (Interrupt request enable bit)

This bit controls interrupt requests (IrqDSP = 1) to the CPU as follows:

- "0": Disables output of interrupt requests (preventing an interrupt from occurring even when the IrqDSP bit is set). [Initial value]
- "1": Enables output of interrupt requests (allowing an interrupt to occur when the IrqDSP is set).
- When reset: The bit is initialized to "0" (disabling interrupt requests).
- The bit is readable and writable.

[bit1] Reserved bit

Be sure to set "0".

[bit0] GoDSP (Calculation start)

RunDSP (Calculation running flag)

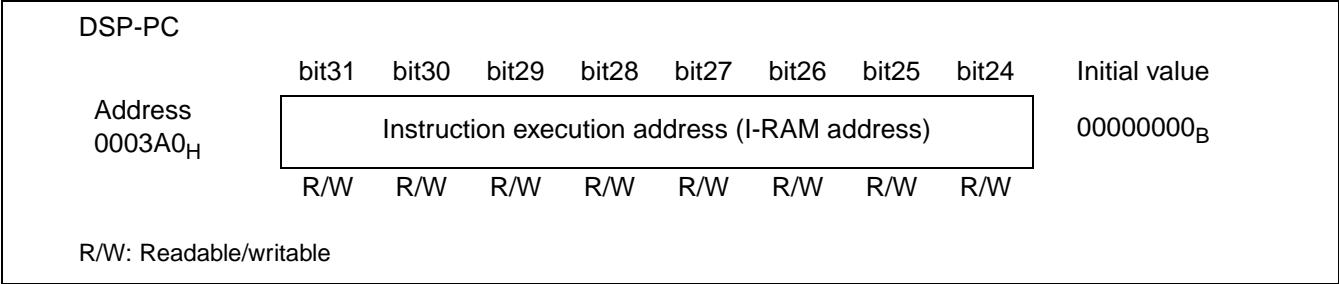
- Writing "1" to the GoDSP bit instructs the starting of calculation. If calculation has been off (RunDSP = 0), it starts calculation and sets the RunDSP flag. It has no effect if calculation is currently being executed (RunDSP = 1).
- The RunDSP flag indicates that calculation is currently being executed. The flag is set when calculation is started. The flag is cleared when the HLT instruction of the multiplication and addition macro is executed.
- During execution of calculation (RunDSP = 1), the CPU cannot access any of DSP-PC, DSP-LY, X-RAM, Y-RAM, and I-RAM. Only DSP-CSR, DSP-OT0 to DSP-OT7, and DSP-AC0 to DSP-AC2 can be monitored.
- To start calculation, write the start address of the calculation routine to the DSP-PC upon startup or before that.
- Write-time function (GoDSP: Calculation start)
 - "0": The bit has no function and neither effect on the operation.
 - "1": Starts calculation if it has been off.
 - Has no effect when calculation is currently being executed.
- Read-time function (RunDSP: Calculation running flag)
 - "0": Indicates that calculation is off. [Initial value]
 - Clear factor →The flag is cleared when the HLT instruction is executed.
 - "1": Indicates that calculation is currently being executed.
 - Set factor →The flag is set the moment calculation is started.
- When reset: The bit is initialized to "0" (with calculation halted).
- The bit is readable and writable. Note, however, that the bit carries different meanings as above between write and read accesses.
- When read by a read modify write (RMW) instruction, the bit always returns "0" irrespective of the current bit value.

■ Program Counter (DSP-PC)

The program counter is an 8-bit counter that contains the memory (I-RAM) address at which the instruction code to be executed by the multiplication and addition macro is located. Although the program counter is refreshed automatically as each instruction is executed, it can be updated with JMP of the multiplication and addition macro. The program counter is accessible (R/W) from the CPU only when calculation is off. The start address of the calculation routine must be written to the DSP-PC the moment or before calculation is started.

After the HLT instruction is executed or after "1" is written to the HltDSP bit of DSP-CSR, the DSP-PC contains the address of the instruction that follows the instruction halted, so that the program can resume execution by setting the GoDSP bit again.

Figure 19.5-2 Program Counter (DSP-PC)



- When reset: The counter is initialized to "00000000_H".
- Although the program counter is readable and writable, it is accessible only when the multiplication and addition macro is not performing calculation (DSP-CSR: RunDSP = 0).
- During execution of calculation (DSP-CSR: RunDSP = 1), the program counter cannot be accessed from the CPU as it is disconnected from the bus.

■ DSP Delay Register (DSP-LY)

The DSP-LY register is a 32-bit register used when the delayed write bit (LDLY) of the MAC instruction of the multiplication and addition macro is "1". The register cannot be access during calculation (DSP-CSR: RunDSP = 1).

- When the LDLY bit of the MAC instruction is "1", the following two operations are executed sequentially:
 - (1) The contents of the DSP-LY register are transferred to the LY-DLY register.
 - (2) Y-RAM read data selected by the MAC instruction is written to the DSP-LY register.
- When the STLY bit of the MAC instruction is "1", the value in the LY-DLY register is written to the Y-RAM address selected by the MAC instruction after the MAC instruction is executed.

Figure 19.5-3 DSP Delay Register (DSP-LY)

DSP-LY						
	bit31	to	bit16 bit15	to	bit0	Initial value
Address	Y-RAM data		Y-RAM data		XXXXXXXX XXXXXXXX	
0003A4 _H	(bit31 to bit16)		(bit15 to bit0)		XXXXXXXX XXXXXXXX _B	
	R/W		R/W			
R/W: Readable/writable						

- When reset: The register has an undefined value.
- Although the DSP-LY register is readable and writable, it is accessible only when it is not involved in calculation (DSP-CSR: RunDSP = 0). During execution of calculation (DSP-CSR: RunDSP = 1), the register cannot be accessed from the CPU as it is disconnected from the bus.

Notes:

- The register can be accessed only in words.
- As the instruction pipeline is used, an actual write is performed as the sixth instruction after the execution of the delayed write. Do not access the relevant address until five instructions are executed after the delayed write instruction.

■ DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7)

There are eight 32-bit registers as variable monitor registers. Except in the initial state immediately after the power is turned on, the registers hold the same contents as Y-RAM addresses 0 to 7. The registers are always only accessible from the CPU so that the contents of Y-RAM addresses 0 to 7 can be monitored event during calculation.

However, when the byte is written in Y-RAM, the writing data to these registers is not processed.

Figure 19.5-4 DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7)

DSP-OT0 to DSP-OT7			
Address	bit31	to	bit0
DSP-OT0 0003A8 _H	<div>Y-RAM data (bit31 to bit0)</div> <div>R</div>		
DSP-OT1 0003AC _H			
DSP-OT2 0003B0 _H			
DSP-OT3 0003B4 _H			
DSP-OT4 0003B8 _H			
DSP-OT5 0003BC _H			
DSP-OT6 0003C0 _H			
DSP-OT7 0003C4 _H			
R: Read only	Initial value: XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B		

- When reset: The registers have undefined values.
- The registers are always only readable, even while the multiplication and addition macro is executing a program.

■ Accumulator Output Registers (DSP-AC0 to DSP-AC2)

DSP-AC0 to DSP-AC2 are 32-bit registers.

A 72-bit accumulator value (ACC) is written to these registers as soon as the multiplication and addition operation is halted.

During the execution of calculation and when it has been off, the registers hold the accumulator value written the moment the last calculation was halted.

The registers are updated when the calculated started next is halted.

Figure 19.5-5 Accumulator Output Registers (DSP-AC0 to DSP-AC2)

DSP-AC0 to DSP-AC2			
	bit31	to	bit8 bit7 to bit0
Address			
DSP-AC0 0003C8 _H	-		ACC[71:64] DSP-AC0
DSP-AC1 0003CC _H	ACC[63:32]		DSP-AC1
DSP-AC2 0003D0 _H	ACC[31:0]		DSP-AC2
	R		
R: Read only	Initial value: 00000000 00000000 00000000 00000000 _B		

- When reset: The registers are initialized to 00.... .
- The registers are always only readable, even while the multiplication and addition macro is executing a program.
- Bit31 to bit8 in DSP-AC0 are vacant bits. When read, they always return "0".

19.6 Principles of Operation of the Multiplication and Addition Calculator

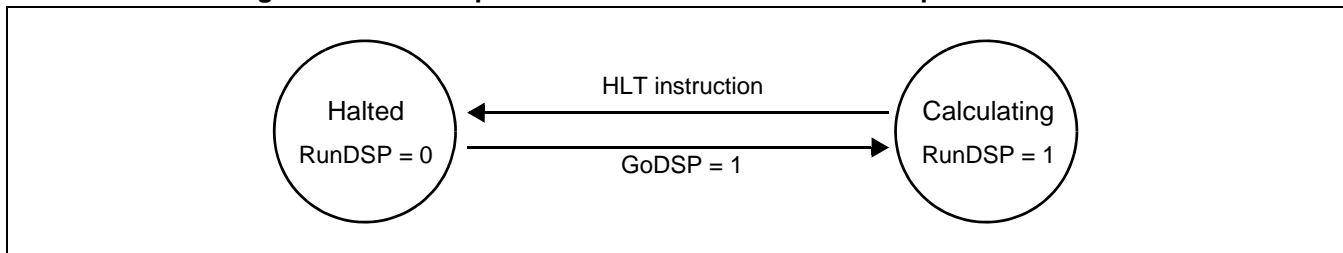
This section describes the operations and functions of the multiplication and addition calculator.

■ Operation Modes

The operation of the multiplication and addition macro is controlled by manipulating the DSP-CSR register.

The multiplication and addition macro is in either of the two states illustrated below. The multiplication and addition macro in the halt state starts program execution when "1" is written to the GoDSP bit. Note also that the registers and memory areas accessible from the CPU are different between in the halt state and during calculation.

Figure 19.6-1 Multiplication and Addition Calculator Operation Modes



Each state is as follows:

- **Halted** : The multiplication and addition macro is not working.
In this state, the CPU can access instruction RAM (I-RAM), data RAM (X-RAM and Y-RAM), and all the registers of the multiplication and addition macro.
Executing the HLT instruction causes the multiplication and addition macro to enter this state.
The multiplication and addition macro is initialized into this state by a system reset.
- **Calculating** : The multiplication and addition macro is calculating.
When "1" is written to the GoDSP bit with the multiplication and addition macro in the halt state, it enters this state to start program execution from the current address in the DSP-PC (program counter).
Executing the HLT instruction causes the multiplication and addition macro to enter the halt state to stop program execution.
In this state, the CPU can access only the DSP-CSR, DSP-OT0 to DSP-OT7, and DSP-AC0 to DSP-AC2 registers (the other registers and RAM are access-barred*).

*: Attempts to read/write these access-barred resources yield the following results:

Write → No effect (Not written)
Read → Undefined

■ Instruction Operations

Writing "1" to the GoDSP bit in the DSP-CSR register causes the multiplication and addition macro to start instruction execution from the current address in the DSP-PC (program counter) (run in parallel with the CPU).

Prior to execution, set the I-RAM and DSP-PC values (the DSP-CSR and DSP-PC can be set at the same time).

When the multiplication and addition macro starts instruction execution, the operation is controlled as follows:

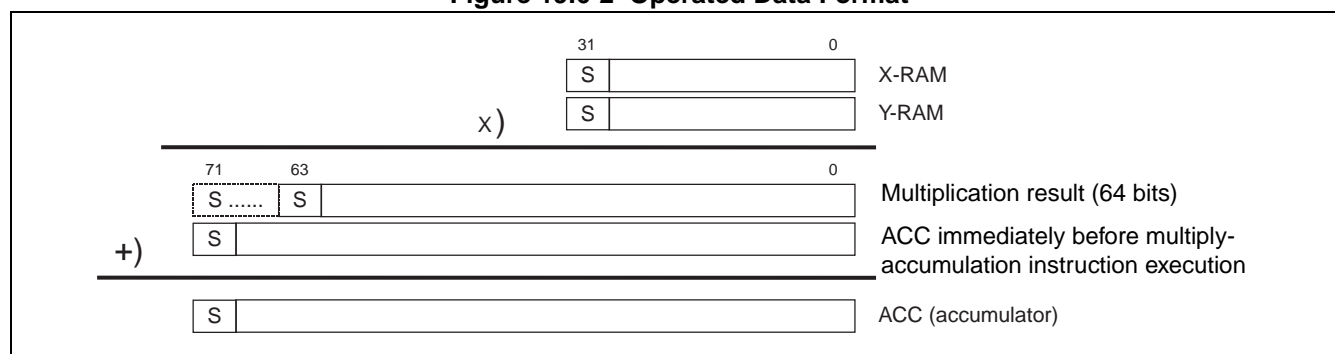
- If the multiplication and addition macro executes the HLT instruction*, it enters the halt state upon completion of the execution of that instruction.
At this time, the DSP-PC stops while holding the address of the instruction that follows the HLT instruction. In addition, the accumulator value is written to DSP-AC0 to DSP-AC2 at this time.
- By executing a NOP instruction, the multiplication and addition macro can issue an interrupt request to the CPU (maskable interrupt supported).
- Program flow can be changed by using a conditional branch instruction that references the USR0 to USR2 bits in the DSP-CSR.

*: The HLT instruction is a NOP instruction whose HLT bit contains "1".

■ Arithmetic Functions

The multiplication and addition macro has a pair of 32-bit data RAM (X-RAM and Y-RAM). When executing a multiplication and addition (or multiply) instruction, the macro loads each piece of RAM data to perform a signed multiplication and addition (or multiply) operation and stores the result in the 72-bit accumulator. The data format is illustrated below:

Figure 19.6-2 Operated Data Format



Notes:

- When a multiply instruction is executed, the multiplication result sign-extended to a 72-bit value is stored in the accumulator (the preceding contents of the accumulator is cleared to zero).
- "S" represents a sign bit.
- Repeating the multiplication and addition instruction so many times that the accumulator overflows may produce unpredictable results.
- Do not execute the multiplication and addition instruction continuously more than 512 times.

■ Delayed Write Function

When executing a multiplication and addition (or multiply) instruction, the multiplication and addition macro can perform the following transfer operations as well. The combination of the transfer and arithmetic operations allows digital filters to easily provide delayed processing of data.

- Storing the value read from Y-RAM to the DSP-LY register.
- Delayed-writing the DSP-LY register value existing prior to instruction execution to the Y-RAM read address via the LY-DLY register.

Note:

As the instruction pipeline is used, an actual write is performed as the sixth instruction after the execution of the delayed write. Do not access the relevant address until five instructions are executed after the delayed write instruction.

■ Transfer of Arithmetic Result

When the calculation result held in the accumulator is transferred to X-RAM and Y-RAM, data is transferred in 32 bits while being scaled as follows:

- Selecting the output bit width

The 72-bit accumulator allows the output bit width to be selected from among the following options:

bit63 to bit32 (Q32 format)
bit62 to bit31 (Q31 format)
bit61 to bit30 (Q30 format)
bit60 to bit29 (Q29 format)
bit59 to bit28 (Q28 format)
bit58 to bit27 (Q27 format)
bit57 to bit26 (Q26 format)
bit31 to bit0 (Q0 format)

- Rounding

The multiplication and addition macro rounds off the accumulator output by counting "1" and cutting away "0" at the bit position immediately following the LSB in the selected output bits.

- Saturation

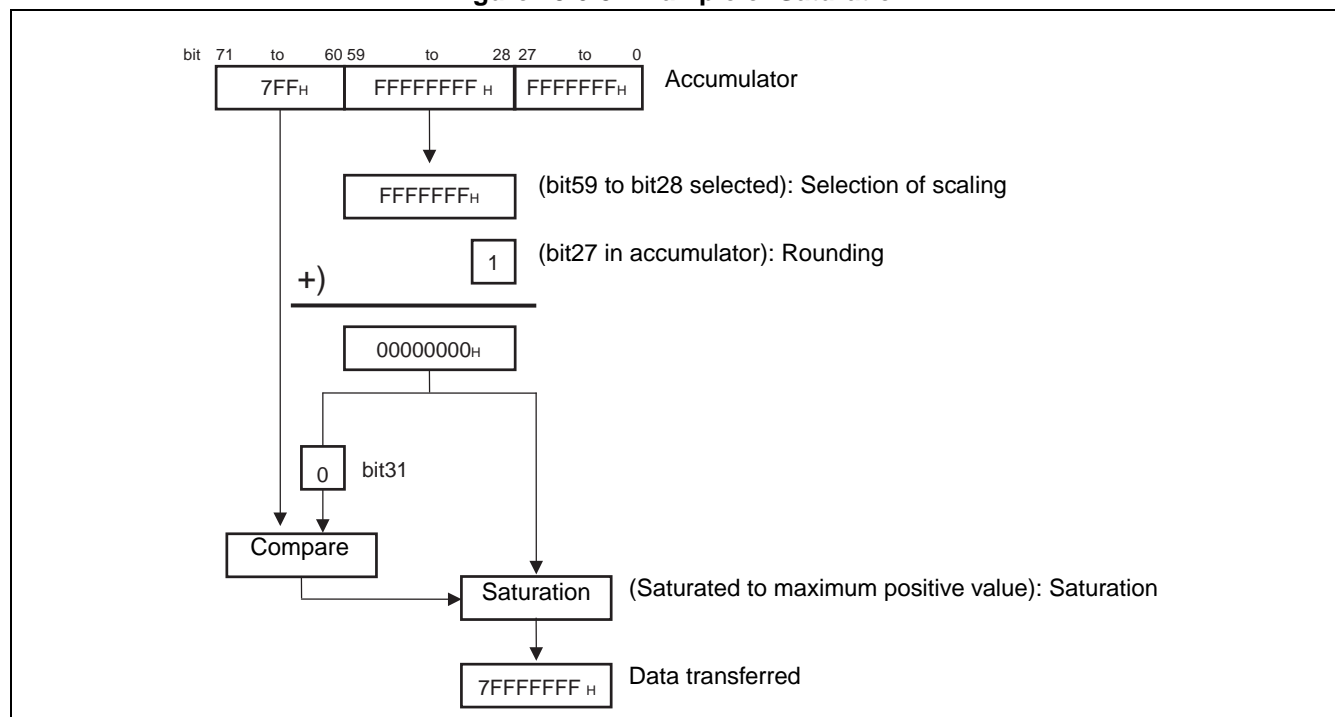
The multiplication and addition macro compares the sign bit (MSB) in rounded 32-bit data with the upper bits in the accumulator and perform saturation if any bit is different in sign. The saturation result becomes the following value depending on sign bit (MSB) in the accumulator:

The accumulator's sign bit is "0". → Saturated to a maximum positive value of "7FFFFFFF_H".

The accumulator's sign bit is "1". → Saturated to a maximum negative value of "80000000_H".

Figure 19.6-3 shows the example of saturation.

Figure 19.6-3 Example of Saturation



Note:

As the instruction pipeline is used, an actual write is performed as the sixth instruction after the execution of the STR instruction. Do not access the relevant address until five instructions are executed after the STR instruction.

■ Variable Monitor Output

The multiplication and addition macro has registers (DSP-OT0 to DSP-OT7) that always hold the contents of Y-RAM addresses 0 to 7. When data is written to Y-RAM addresses 0 to 7 (by the CPU, STR instruction, or delayed write), the same value is written to the DSP-OT0 to DSP-OT7 registers.

Although the CPU is barred access to Y-RAM during calculation, the CPU can always reference the desired calculation result by writing it to Y-RAM addresses 0 to 7 using the STR instruction.

■ Accumulator Output

The multiplication and addition calculator has registers (DSP-AC0 to DSP-AC2) that hold the 72-bit accumulator value acquired upon completion of calculation. The initial value is undefined. The performs a calculation when "1" is written to the GoDSP bit. It halts the calculation when the NOP HLT instruction is executed, when the current accumulator value is stored by the DSP-AC0 to DSP-AC2 registers. Once stored, the accumulator value remains held in the registers until the calculation performed next is halted.

The registers are initialized to be undefined by a reset.

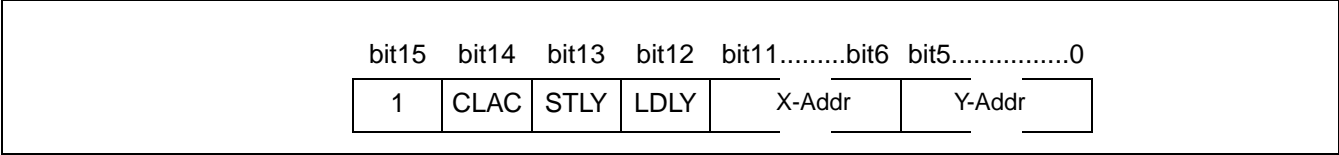
19.7 Details on Instructions of the Multiplication and Addition Calculator

This section details the three types of instructions used for the multiplication and addition calculator.

■ MAC Instruction

Operation	: $ACC \leftarrow ACC + X \text{ data} \times Y \text{ data}$ LY-DLY \leftarrow DSP-LY DSP-LY \leftarrow Y data (LDLY = 1) Y-RAM \leftarrow LY-DLY (STLY = 1) DSP-AC0 to DSP-AC2 \leftarrow ACC
Description	: Add the product of X data from X-RAM and Y-data from Y-RAM to the accumulator. Transfer the content of the DSP-LY register to the LY-DLY register at the same time.
Number of words	: 1 word (16 bits)
Number of cycles	: 1 system clock cycle
Operation code	:

Figure 19.7-1 MAC Instruction Operation Code



[bit14] CLAC (Clear ACC)

- Setting this bit causes the code to serve as a multiply instruction.
- "0": $ACC \leftarrow ACC + X \text{ data} \times Y \text{ data}$ [multiplication and addition instruction]
- "1": $ACC \leftarrow 0 + X \text{ data} \times Y \text{ data}$ [multiply instruction]

[bit13] STLY (Store LY)

- When this bit is "1", the following operation is performed. If it is "0", only the arithmetic operation is performed.
- The multiplication and addition calculator writes the content of the LY-DLY register to the Y-Addr address in Y-RAM after performing the arithmetic operation.

[bit12] LDLY (Load LY)

- When this bit is "1", the following operation is performed. If it is "0", only the arithmetic operation is performed.
- The multiplication and addition calculator writes the content of the Y-Addr address in Y-RAM to the DSP-LY register as well when performing the arithmetic operation.

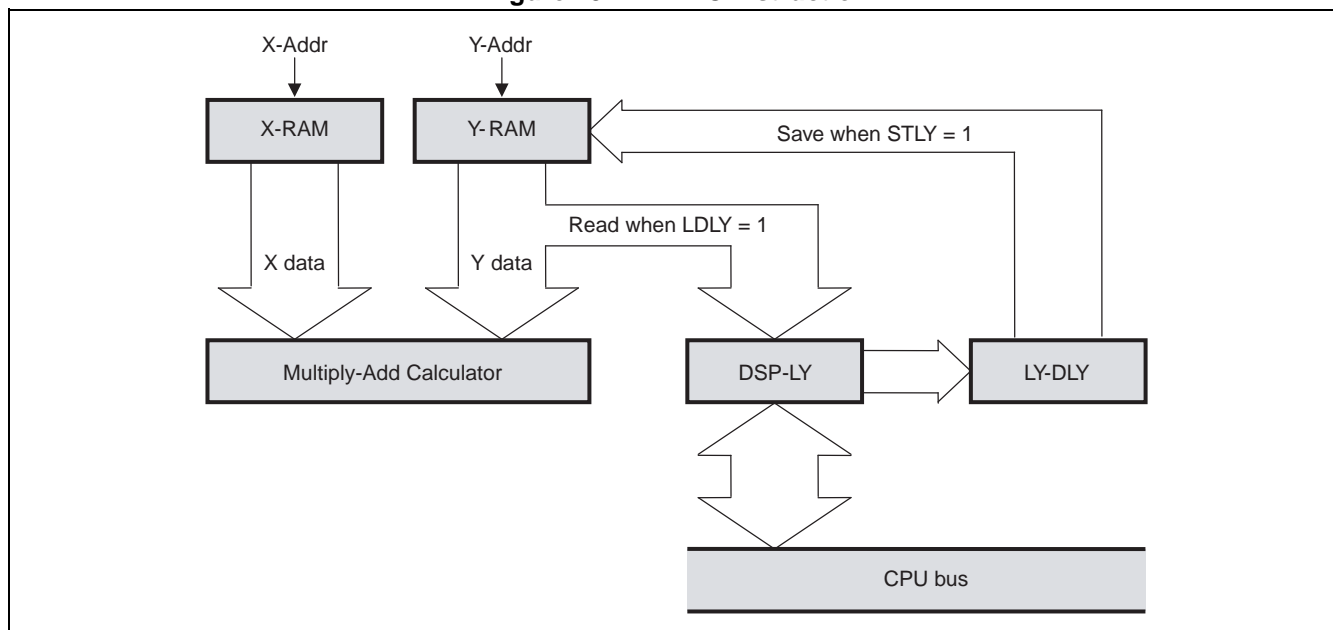
[bit11 to bit6] X-Addr (X-RAM Address)

These bits contain the address to locate X data in X-RAM.

[bit5 to bit0] Y-Addr (Y-RAM Address)

- These bits contain the address to locate Y data in Y-RAM.

Figure 19.7-2 MAC Instruction



Note:

As an actual write is performed as the sixth instruction after the execution of the delayed write, do not access the relevant address until five instructions are executed after the delayed write instruction.

■ STR Instruction (Transfer Instruction)

Operation : Data RAM ← Accumulator
 Description : Convert 72-bit data in the accumulator to 32-bit data according to the RND/CLP/SLQ flags and write the result to data RAM according to the SLY flag and X/Y-Addr bits.
 Number of words : 1 word (16 bits)
 Number of cycles : 1 system clock cycle
 Operation code :

Figure 19.7-3 STR Instruction Operation Code

bit15	bit14	bit13	bit12	bit11	bit10	bit9.....bit7	bit6	bit5.....bit0
0	1	1	0	RND	CLP	SLQ	SLY	X/Y-Addr

[bit11] RND (Rounding)

This bit specifies rounding to be performed for 32-bit data specified by the SLQ bits.

Rounding counts "0" and cuts away "1" at the bit position following the LSB in 32-bit data.

[bit10] CLP (Clipping)

This bit specifies saturation to be performed for 32-bit data specified by the SLQ bits if the accumulator calculation result is a value which overflows from the 32-bit data.

In practice, saturation is performed if the MSB (specified by SLQ) in 32-bit data is different from the MSB (bit71) in the accumulator. When rounding is specified, comparison applies to the result of rounding.

When the accumulator value before rounding is positive, the maximum positive value (7FFFFFFF_H) is transferred. When it is negative, the maximum negative value (80000000_H) is transferred.

The sign in the accumulator is retained without being inverted through rounding or saturation.

[bit9 to bit7] SLQ

These bits specify the bit positions for transfer from the accumulator to data RAM.

Table 19.7-1 Specification of the Bit Position for Transfer from Accumulator to Data RAM

SLQ bits	Overflow evaluation bits	32-bit data to be transferred	Rounded bit	Fixed-point format	Assembler code
000	bit71 to bit59	bit59 to bit28	bit27	Q28	Q12
001	bit71 to bit60	bit60 to bit29	bit28	Q29	Q13
010	bit71 to bit61	bit61 to bit30	bit29	Q30	Q14
011	bit71 to bit62	bit62 to bit31	bit30	Q31	Q15
100	bit71 to bit63	bit63 to bit32	bit31	Q32	Q8
101	bit71 to bit31	bit31 to bit0	(None)	Q0	Q9
110	bit71 to bit57	bit57 to bit26	bit25	Q26	Q10
111	bit71 to bit58	bit58 to bit27	bit26	Q27	Q11

[bit6] SLY

This bit specifies the transfer destination.

"0": X-RAM

"1": Y-RAM

[bit5 to bit0] X/Y-Addr (RAM Address)

These bits specify the direct address in data RAM.

Note:

As an actual write is performed as the sixth instruction after the execution of the STR instruction, do not access the relevant address until five instructions are executed after the STR instruction.

■ NOP Instruction

Operation : IF REP=001 DSP-PC \leftarrow DSP-PC + 1
ELSE DSP-PC \leftarrow DSP-PC

Description : Cause a branch when the condition is true or do nothing otherwise.

Number of words : 1 word (16 bits)

Number of cycles : 1 system clock cycle

Operation code :

Figure 19.7-5 NOP Instruction Operation Code

bit	15 to 5	4 to 2	1	0
	0	REP	HLT	SIRQ

[bit4 to bit2] REP (Repetition count setting)

These bits can be used to set the number of repetitions to 1 to 8. REP = 001_B and REP = 000_B cause the NOP instruction to be executed once and eight times, respectively.

[bit1] HLT (HLT instruction indication flag)

If this bit is set, the multiplication and addition macro halts program execution after instruction execution. The RunDSP flag in the DSP-CSR register is cleared. The DSP-PC stops after two instructions are executed.

Notes:

- To enable the setting of this bit, be sure to set the REP bit to "001_B".
- Due to the instruction pipeline, the actual branch to HLT occurs after the two instructions that follow the HLT instruction are executed. Between the two instructions, therefore, do not execute any instruction other than NOP (REP = 001_B, HLT = 0, SIRQ = 0).

[bit0] SIRQ (INT instruction indication flag)

Setting this bit causes an interrupt request to the CPU to be generated and the IrqDSP flag in the DSP-CSR register to be set after instruction execution.

Note:

To enable the setting of this bit, be sure to set the REP bit to "001_B".

19.8 Notes on Using the Sum-of-Products Circuit

This section describes the notes on using the sum-of-products circuit.

■ Notes on Using the Sum-of-Products Circuit

- **Undefined Instructions**

Do not attempt to execute the undefined instructions because the device operation is not guaranteed.

- **Clearing the interrupts**

If you clear the interrupts in the interrupt routine after the interrupts have occurred, make sure the IrqDSP bit is turned to "0" after you wrote "0" into DSP-CSR register IrqDSP bit (bit3).

- **Initializing the Accumulator**

Since the accumulator (ACC) in the sum-of-products circuit is not initialized, it is unstable after the power-on. So, starting with executing the sum-of-products instruction (MAC instruction CLAC=0) would yield undefined calculation results. To avoid this, execute the multiplication instruction (MAC instruction CLAC=1) to initialize the accumulator (ACC).

CHAPTER 20

DMA CONTROLLER (DMAC)

This chapter describes the DMAC, the configuration and functions of registers, and DMAC operation.

- 20.1 Overview of the DMAC
- 20.2 Detailed Explanation of the DMAC Registers
- 20.3 Explanation of the DMAC Operation
- 20.4 Operation Flowcharts of the DMAC
- 20.5 Data Bus of the DMAC

20.1 Overview of the DMAC

This module implements DMA (Direct Memory Access) transfer on FR family devices. When this module is used to control DMA transfer, various data transfer operations can be executed at high speed by bypassing the CPU, enhancing system performance.

■ Hardware Configuration of the DMAC

This module mainly consists of the following blocks:

- Five independent DMA channels
- 5 channel independent access control circuit
- 32-bit address registers (reload specifiable, two registers for each channel)
- 16-bit transfer count register (reload specifiable, one register for each channel)
- 4-bit block count register (one for each channel)
- 2-cycle transfer

■ Main DMAC Functions

Data transfer using this module mainly consists of the following functions:

- Data can be transferred independently over multiple channels (5 channels)
 - Priority (ch.0→ ch.1→ ch.2→ ch.3→ ch.4)
 - The priority can be rotated between ch.0 and ch.1.
 - DMAC start sources
 - Built-in peripheral requests (shared interrupt requests, including external interrupts)
 - Software request (register write)
 - Transfer mode
 - Burst transfer, step transfer, and block transfer
 - Addressing mode: 32-bit full addressing (increment/decrement/fixed)
(The address increment/decrement range is from -255 to + 255.)
 - Data types: Byte, halfword, and word length
 - Single shot/reload selectable

■ Register List of the DMAC Registers

Figure 20.1-1 provides an register list of the DMAC.

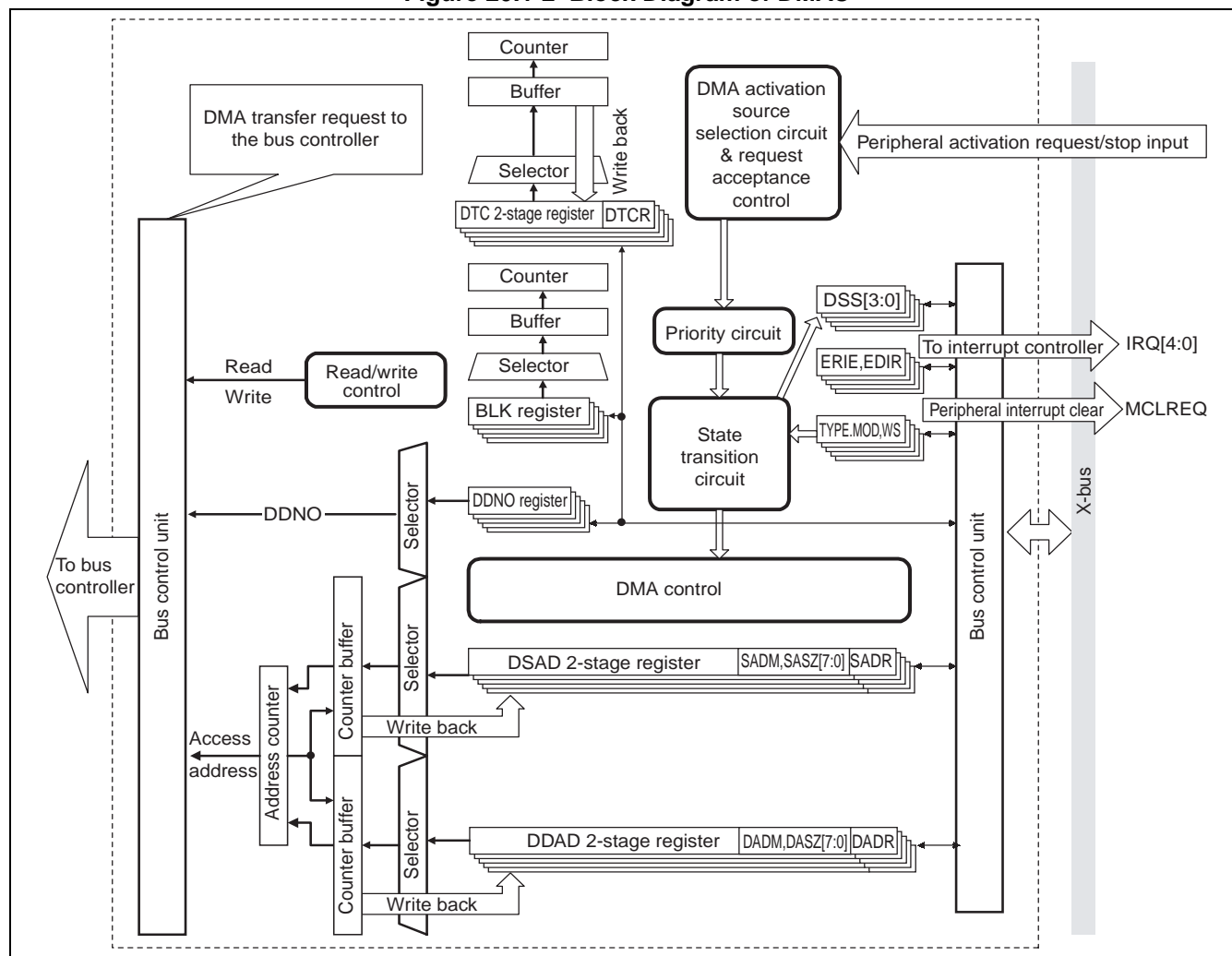
Figure 20.1-1 Register List of the DMAC

		bit 31	bit 00
ch.0 control/status register A	DMACA0 00000200 _H	<input type="text"/>	<input type="text"/>
ch.0 control/status register B	DMACB0 00000204 _H	<input type="text"/>	<input type="text"/>
ch.1 control/status register A	DMACA1 00000208 _H	<input type="text"/>	<input type="text"/>
ch.1 control/status register B	DMACB1 0000020C _H	<input type="text"/>	<input type="text"/>
ch.2 control/status register A	DMACA2 00000210 _H	<input type="text"/>	<input type="text"/>
ch.2 control/status register B	DMACB2 00000214 _H	<input type="text"/>	<input type="text"/>
ch.3 control/status register A	DMACA3 00000218 _H	<input type="text"/>	<input type="text"/>
ch.3 control/status register B	DMACB3 0000021C _H	<input type="text"/>	<input type="text"/>
ch.4 control/status register A	DMACA4 00000220 _H	<input type="text"/>	<input type="text"/>
ch.4 control/status register B	DMACB4 00000224 _H	<input type="text"/>	<input type="text"/>
All-channel control register	DMACR 00000240 _H	<input type="text"/>	<input type="text"/>
ch.0 transfer source address setting register	DMASA0 00001000 _H	<input type="text"/>	<input type="text"/>
ch.0 transfer destination address setting register	DMADA0 00001004 _H	<input type="text"/>	<input type="text"/>
ch.1 transfer source address setting register	DMASA1 00001008 _H	<input type="text"/>	<input type="text"/>
ch.1 transfer destination address setting register	DMADA1 0000100C _H	<input type="text"/>	<input type="text"/>
ch.2 transfer source address setting register	DMASA2 00001010 _H	<input type="text"/>	<input type="text"/>
ch.2 transfer destination address setting register	DMADA2 00001014 _H	<input type="text"/>	<input type="text"/>
ch.3 transfer source address setting register	DMASA3 00001018 _H	<input type="text"/>	<input type="text"/>
ch.3 transfer destination address setting register	DMADA3 0000101C _H	<input type="text"/>	<input type="text"/>
ch.4 transfer source address setting register	DMASA4 00001020 _H	<input type="text"/>	<input type="text"/>
ch.4 transfer destination address setting register	DMADA4 00001024 _H	<input type="text"/>	<input type="text"/>

■ Block Diagram of DMAC

Figure 20.1-2 is a block diagram of DMAC.

Figure 20.1-2 Block Diagram of DMAC



20.2 Detailed Explanation of the DMAC Registers

This section describes the DMAC registers in detail.

■ Notes on Setting Registers

When the DMA controller (DMAC) is set, some bits need to be set while DMA is stopped. If they are set while DMA is in progress (during transfer), correct operation cannot be guaranteed.

" * " marks indicates that the bit affects operation if it is set during DMAC transfer. Rewrite this bit while DMAC transfer is stopped (start is disabled or temporarily stopped).

The setting of this bit that is made while DMA transfer start is disabled (when the DMACR:DMAE=0 or the DMACA:DENB=0) becomes effective when DMA transfer start is enabled.

The setting of this bit that is made while DMA transfer is temporarily stopped (when the DMAH3 to DMAH0 bits of DMACR are not 0000_B or the PAUS bit of DMACA is "1") becomes effective when temporary stop is canceled.

20.2.1 DMAC ch.0,ch.1,ch.2,ch.3,ch.4 Control/Status Registers A

The [DMACA0 to DMACA4] registers control the operation of the DMAC channels.
A separate register is provided for each channel.

■ Functions of the [DMACA0 to DMACA4] Bits

The functions of the [DMACA0 to DMACA4] bits are shown below:

Figure 20.2-1 DMAC Control/Status Register A

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0: 000200 _H	DENB	PAUS	STRG	IS [4 : 0]				-				BLK [3 : 0]				
ch.1: 000208 _H	R/W	R/W	R/W	R/W				R/W				R/W				
ch.2: 000210 _H																
ch.3: 000218 _H																
ch.4: 000220 _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTC [15 : 0]															
	R/W															
	(Initial value: 00000000----XXXXXXXXXXXXXXXXXXXXX _B)															

R/W: Readable/writable

[bit31] DENB (Dma ENaBle): DMA operation enable bit

This bit, which corresponds to a transfer channel, is used to enable and disable DMA transfer.

The activated channel starts DMA transfer when a transfer request is generated and accepted.

All transfer requests that are generated for a deactivated channel are disabled.

When the transfer on an activated channel reaches the specified count, this bit is set to "0" and transfer stops.

The transfer can be forced to stop by writing "0" to this bit. Be sure to stop a transfer forcibly ("0" write) only after temporarily stopping DMA using the PAUS bit [bit30 of DMACA]. If the transfer is forced to stop without first temporarily stopping DMA, DMA stops but the transferred data cannot be guaranteed. Check whether DMA is stopped using the DSS[2:0] bits [bit18 to bit16 of DMACB].

DENB	Function
0	Disables operation of DMA on the corresponding channel (initial value).
1	Enables operation of DMA on the corresponding channel.

- If a stop request is accepted during reset: Initialized to "0".
- This bit is readable and writable.

If the operation of all channels is disabled by bit15 (DMAE bit) of the DMAC all-channel control register (DMACR), writing "1" to this bit is disabled and the stopped state is maintained. If the operation is disabled by the bit15 (DMAE bit) while it is enabled by this bit, "0" is written to this bit and the transfer is stopped (forced stop).

[bit30] PAUS (PAUSE): Temporary stop instruction

This bit temporarily stops DMA transfer on the corresponding channel. If this bit is set, DMA transfer is not performed before this bit is cleared (While DMA is stopped, the DSS bits are 1XX_B).

If this bit is set before starting, DMA transfer continues to be temporarily stopped.

New transfer requests that occur while this bit is set are accepted, but no transfer starts before this bit is cleared (See Section "20.3.10 Transfer Request Acceptance and Transfer").

PAUS	Function
0	Enables DMA operation of the corresponding channel (initial value)
1	Temporarily stops DMA on the corresponding channel.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit29] STRG (Software TRiGger): Transfer request

This bit generates a DMA transfer request for the corresponding channel. If "1" is written to this bit, a transfer request is generated when write operation to the register is completed and transfer on the corresponding channel is started. However, if the corresponding channel is not activated, operations on this bit are disabled.

Reference:

If starting by a write operation to the DMAE bit and a transfer request occurring due to this bit are simultaneous, the transfer request is enabled and transfer is started. If writing of "1" to the PAUS bit and a transfer request occurring due to this bit are simultaneous, the transfer request is enabled, but DMA transfer is not started before "0" is written to the PAUS bit.

STRG	Function
0	Disabled
1	DMA starting request

- When reset: Initialized to "0".
- The read value is always "0".
- Only a write value of "1" is valid. If "0" is written, operation is not affected.

[bit28 to bit24] IS4 to IS0 (Input Select)*: Transfer Source Selection

These bits select the source of the transfer request as listed in the table below. Note that the software transfer request by the STRG bit function is always valid regardless of the settings of these bits.

IS	Function	Transfer stop request
00000 _B	Only the software transfer request	No
00001 _B ↓ 01111 _B	Setting disabled ↓ Setting disabled	
10000 _B	Multi-function serial interface0 (receiving complete)	
10001 _B	Multi-function serial interface1 (receiving complete)	Yes
10010 _B	Multi-function serial interface2 (receiving complete)	
10011 _B	Multi-function serial interface0 (sending complete)	
10100 _B	Multi-function serial interface1 (sending complete)	No
10101 _B	Multi-function serial interface2 (sending complete)	
10110 _B	External interrupt 0	
10111 _B	External interrupt 1	
11000 _B	8/10-bit AD2	
11001 _B	8/10-bit AD0 ^{*2} / 12-bit AD3 ^{*1}	
11010 _B	8/10-bit AD1 ^{*2} / 12-bit AD4 ^{*1}	
11011 _B	Multiplication and addition calculator	
11100 _B [*]	PPG0	
11101 _B	PPG4	
11110 _B	PPG8 ^{*2}	
11111 _B	Base timer 0 (Source 0)	

*1 : MB91470 series

*2 : MB91480 series

- When reset: Initialized to 00000_B.
- These bits are readable and writable.

If DMA start resulting from an interrupt from a peripheral function is set (IS=1XXXX_B), disable interrupts from the selected peripheral function with the ICR register.

[bit23 to bit20] Reserved: Reserved bits

Be sure to set "0000_B".

[bit19 to bit16] BLK3 to BLK0 (BLoCK size): Block size specification

These bits specify the block size for block transfer on the corresponding channel. The value specified by these bits becomes the number of words in one transfer unit (more exactly, the repetition count of the data width setting). If block transfer will not be performed, set "01_H" (size 1). (This register value is ignored during demand transfer. The size becomes "1".)

BLK	Function
XXXX _B	Block size of the corresponding channel

- When reset: Not initialized.
- These bits are readable and writable.
- If "0" is specified for all bits, the block size becomes 16 words.
During reading, the block size is always read (reload value).

[bit15 to bit0] DTC15 to DTC0 (Dma Terminal Count register)*: Transfer count register

These bits compose a register for storing the transfer count. Each register has 16-bit length.

All registers have a dedicated reload register. When the register is used for a channel that is enabled to reload the transfer count register, the initial value is automatically written back to the register when the transfer is completed.

DTC	Function
XXXX _H	Transfer count for the corresponding channel

When DMA transfer is started, data in this register is stored in the counter buffer of the DMA-dedicated transfer count counter and is decremented by "1" (subtraction) after each transfer unit. When DMA transfer is completed, the contents of the counter buffer are written back to this register and then DMA ends. Thus, the transfer count value during DMA operation cannot be read.

- When reset: Not initialized.
- These bits are readable and writable. Always access DTC using halfword length or word length.
- During reading, the count value is read. The reload value cannot be read.
- When reset: Not initialized.

20.2.2 DMAC ch.1,ch.2,ch.3,ch.4 Control/Status Registers B

The DMACB0 to DMACB4 registers control the operation of the DMAC channels.
A separate register is provided for each channel.

■ Functions of the DMACB0 to DMACB4 Bits

The functions of the [DMACB0 to DMACB4] are shown below:

Figure 20.2-2 DMAC Control/Status Register B

Figure 10-12-2 Data Control Status Register 2

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0: 000204 _H	TYPE [1 : 0]		MOD [1 : 0]		WS [1 : 0]		SADM	DADM	DTCR	SADR	DADR	ERIE	EDIE	DSS[2 : 0]		
ch.1: 00020C _H	R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ch.2: 000214 _H																
ch.3: 00021C _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ch.4: 000224 _H	SASZ [7 : 0]								DASZ [7 : 0]							
	R/W								R/W							
(Initial value: 0000000000000000XXXXXXXXXXXXXXXXX _B)																
R/W: Readable/writable																

[bit31, bit30] TYPE1, TYPE0 (TYPE)*: Transfer type setting

These bits specify the operation type of the corresponding channel as described below.

2-cycle transfer mode:

In this mode, the transfer source address (DMASA) and transfer destination address (DMADA) are set and transfer is performed by repeating the read operation and write operation for the number of times specified by the transfer count. All areas can be specified as a transfer source or transfer destination (32-bit address).

TYPE	Function
00 _B	2-cycle transfer (initial value)
01 _B	Setting disabled
10 _B	Setting disabled
11 _B	Setting disabled

- When reset: Initialized to "00".
- These bits are readable and writable.

[bit29, bit28] MOD0, MOD1 (MODE)*: Transfer mode setting

These bits set the operating mode of the corresponding channel as listed in the table below:

MOD	Function
00 _B	Block/step transfer mode (initial value)
01 _B	Burst transfer mode
10 _B	Setting disabled
11 _B	Setting disabled

- When reset: Initialized to "00".
- These bits are readable and writable.

[bit27, bit26] WS1, WS0 (Word Size): Transfer data width selection

These bits are used to select the transfer data width of the corresponding channel. Transfer operations are repeated in units of the data width specified in this register for as many times as the specified count.

WS	Function
00 _B	Byte-width transfer (initial value)
01 _B	Halfword-width transfer
10 _B	Word-width transfer
11 _B	Setting disabled

- When reset: Initialized to "00".
- These bits are readable and writable.

[bit25] SADM (Source-ADdr. Count-Mode select)*: Transfer source address count mode specification

This bit specifies the address processing of the transfer source address of the corresponding channel for each transfer operation.

An address increment is added or an address decrement is subtracted after each transfer operation according to the specified transfer source address count width (SASZ). When the transfer is completed, the next access address is written to the corresponding address register (DMASA). As a result, the transfer source address register is not updated until DMA transfer is completed.

To make the address always the same, specify "0" or "1" for this bit and set the address count width (SASZ and DASZ) to "0".

SADM	Function
0	Increments the transfer source address. (initial value)
1	Decrements the transfer source address.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit24] DADM (Destination-Addr. Count-Mode select)*: Transfer destination address count mode specification

This bit specifies the address processing for the transfer destination address of the corresponding channel in each transfer operation.

An address increment is added or an address decrement is subtracted after each transfer operation according to the specified transfer destination address count width (DASZ). When the transfer is completed, the next access address is written to the corresponding address register (DMADA). As a result, the transfer destination address register is not updated until the DMA transfer is completed.

To make the address always the same, specify "0" or "1" for this bit and set the address count width (SASZ and DASZ) to "0".

DADM	Function
0	Increments the transfer destination address. (initial value)
1	Decrements the transfer destination address.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit23] DTCR (DTC-reg. Reload)*: Transfer count register reload specification

This bit controls reloading of the transfer count register for the corresponding channel.

If reloading of the counter is enabled by this bit, the count register value is restored to its initial value after transfer is completed, then DMAC stops and starts waiting for a new transfer request (an activation request by STRG or IS setting). (If this bit is "1", the DENB bit is not cleared.)

DENB=0 or DMAE=0 must be set to stop the transfer. In either case, the transfer is forcibly stopped.

If reloading of the counter is disabled, a single shot operation occurs. In single shot operation, operation stops after the transfer is completed even if reload is specified in the address register. The DENB bit is also cleared in this case.

DTCR	Function
0	Disables transfer count register reloading (initial value)
1	Enables transfer count register reloading.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit22] SADR (Source-Addr.-reg. Reload)*: Transfer source address register reload specification

This bit controls reloading of the transfer source address register for the corresponding channel.

If this bit enables the reload operation, the transfer source address register value is restored to its initial value after the transfer is completed.

If reloading of the counter is disabled, a single shot operation occurs. In single shot operation, operation stops after the transfer is completed even if reload is specified in the address register. The address register value also stops in this case while the initial value is being reloaded.

If this bit disables the reload operation, the address register value when the transfer is completed is the address to be accessed next to the final address. (When address increment is specified, the next address is an incremented address.)

SADR	Function
0	Disables transfer source address register reloading. (initial value)
1	Enables transfer source address register reloading.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit21] DADR (Dest.-Addr.-reg. Reload)*: Transfer destination address register reload specification

This bit controls reloading of the transfer destination address register for the corresponding channel.

If this bit enables reloading, the transfer destination address register value is restored to its initial value after the transfer is completed.

The details of other functions are the same as those described for bit22 (SADR).

DADR	Function
0	Disables transfer destination address register reloading. (initial value)
1	Enables transfer destination address register reloading.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit20] ERIE (Error Interrupt Enable)*: Error interrupt output enable

This bit controls the occurrence of an interrupt for termination after an error occurs. The nature of the error that occurred is indicated by DSS2 to DSS0. Note that an interrupt occurs only for specific termination causes and not for all termination causes. (Refer to bits DSS2-DSS0.)

ERIE	Function
0	Disables error interrupt request output. (initial value)
1	Enables error interrupt request output.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit19] EDIE (EnD Interrupt Enable)*: End interrupt output enable

This bit controls the occurrence of an interrupt for normal termination.

EDIE	Function
0	Disables end interrupt request output. (initial value)
1	Enables end interrupt request output.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit18 to bit16] DSS2 to DSS0 (DMA Stop Status)*: Transfer stop source indication

These bits indicate a code (end code) of 3 bits that indicates the source of stopping or termination of DMA transfer on the corresponding channel. The table below lists the end codes:

DSS	Function	Interrupt
000 _B	Initial value	None
X01 _B	Address error (underflow/overflow)	Error
X10 _B	Transfer stop request	Error
X11 _B	Normal end	End
1XX _B	DMA stopped temporarily (due, for example, to DMAH, PAUS bit, and an interrupt)	None

The code indicating a transfer stop request is set only if the request is received from a peripheral circuit. The Interrupt column indicates the type of interrupts that can occur.

- When reset: Initialized to "000_B".
- These bits can be cleared by writing "000_B" to them.
- These bits are readable and writable. Note, however, that the only valid written value is "000_B".

[bit15 to bit8] SASZ7 to SASZ0 (Source Addr count SiZe)*: Transfer source address count size specification

These bits specify the increment or decrement width for the transfer source address (DMASA) of the corresponding channel for each transfer operation. The value set by these bits becomes the address increment/decrement width for each transfer unit. The address increment/decrement width conforms to the instruction in the transfer source address count mode (SADM).

SASZ	Function
XX _H	Specify the increment/decrement width of the transfer source address. "0" to "255"

- When reset: Not initialized
- These bits are readable and writable.

[bit7 to bit0] DASZ7 to DASZ0 (Des Addr count SiZe)*: Transfer destination address count size specification

These bits specify the increment or decrement width for the transfer destination address (DMADA) of the corresponding channel for each transfer operation. The value set by these bits becomes the address increment/decrement width for each transfer unit. The address increment/decrement width conforms to the instruction in the transfer destination address count mode (DADM).

DASZ	Function
XX _H	Specify the increment/decrement width of the transfer destination address. "0" to "255"

- When reset: Not initialized
- These bits are readable and writable.

20.2.3 DMAC ch.1,ch.2,ch.3,ch.4 Transfer Source/Transfer Destination Address Setting Registers

The DMASA0 to DMASA4 registers and DMADA0 to DMADA4 registers control the operation of the DMAC channels. A separate register is provided for each channel.

■ Functions of the DMASA0 to DMASA4 and DMADA0 to DMADA4 Bits

The functions of the DMASA0 to DMASA4 and DMADA0 to DMADA4 bits are shown below:

Figure 20.2-3 DMAC Transfer Source/Transfer Destination Address Setting Registers

Figure 20-12-6 DMAS Transfer Source Register Destination Address Setting Register

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0: 001000 _H	DMASA [31 : 16]															
ch.1: 001008 _H	R/W															
ch.2: 001010 _H																
ch.3: 001018 _H																
ch.4: 001020 _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMASA [15 : 0]															
	R/W															
	(Initial value: XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B)															

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0: 001004 _H	DMADA [31 : 16]															
ch.1: 00100C _H	R/W															
ch.2: 001014 _H																
ch.3: 00101C _H																
ch.4: 001024 _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMADA [15 : 0]															
	R/W															
	(Initial value: XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX _B)															

R/W: Readable/writable

DMASA0 to 4 and DMADA0 to 4 are a group of registers used to store transfer source and transfer destination addresses. The length of each register is 32 bits.

[bit31 to bit0] DMASA31 to DMASA0 (DMA Source Addr)*: Transfer source address setting

These bits set the transfer source address.

[bit31 to bit0] DMADA31 to DMADA0 (DMA Destination Addr)*: Transfer destination address setting

These bits set the transfer destination address.

If DMA transfer is activated, data in this register is stored in the counter buffer of the DMA-dedicated address counter and then the address is counted according to the settings for the transfer operation. When the DMA transfer is completed, the contents of the counter buffer are written back to this register and then DMA ends. Thus, the address counter value during DMA operation cannot be read.

All registers have a dedicated reload register. When the register is used for a channel that is enabled for reloading of the transfer source/transfer destination address register, the initial value is automatically written back to the register when the transfer is completed. Other address registers are not affected.

- When reset: Not initialized.
- These bits are readable and writable. For this register, be sure to access these bits as 32-bit data.
- If these bits are read during transfer, the address before the transfer is read. If they are read after transfer, the next access address is read. Because the reload value cannot be read, it is not possible to read the transfer address in real time.

Note:

Do not set any of the DMAC's registers using this register. DMA transfer is not possible for the DMAC's registers themselves.

20.2.4 DMAC ch.1,ch.2,ch.3,ch.4 DMAC All-Channel Control Register

The [DMACR] register controls the operation of all five DMAC channels. Always use byte length to access this register.

■ Functions of the [DMACR] Bits

The functions of the [DMACR] bits are shown below:

Figure 20.2-4 Functions of DMACR Bits

Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
000204 _H	DMAE	-	-	PM01	DMAH[3:0]			-	-	-	-	-	-	-	-	-
	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(Initial value: 0--00000 -----B)

R/W: Readable/writable

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

This bit controls the operation of all DMA channels.

If DMA operation is disabled with this bit, transfer operations on all channels are disabled regardless of the start/stop settings for each channel and the operating status. Any channel carrying out transfer cancels the requests and stops transfer at a block boundary. All start operations on each channel in a disabled state are disabled.

If this bit enables DMA operation, start/stop operations are enabled for all channels. Simply enabling DMA operation with this bit does not activate each channel.

DMA operation can be forced to stop by writing "0" to this bit. However, be sure to force stopping ("0" write) only after temporarily stopping DMA using the DMAH[3:0] bits [bit27-bit24 of DMACR]. If forced stopping is carried out without first temporarily stopping DMA, DMA stops, but the transfer data cannot be guaranteed. Check whether DMA is stopped using the DSS[2:0] bits [bit18 to bit16 of DMACB].

DMAE	Function
0	Disables DMA transfer on all channels. (initial value)
1	Enables DMA transfer on all channels.

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit28] PM01 (Priority Mode ch.0, ch.1 robin): Channel priority rotation

This bit is set to alternate priority for each transfer between Channel0 and Channel1.

PM01	Function
0	Fixes the priority. (ch.0 > ch.1)(initial value)
1	Alternates priority. (ch.1 > ch.0)

- When reset: Initialized to "0".
- This bit is readable and writable.

[bit27 to bit24] DMAH3 to DMAH0 (DMA Halt): DMA temporary stop

These bits control temporary stopping of all DMA channels. If these bits are set, DMA transfer is not performed on any channel before these bits are cleared again.

When DMA transfer is activated after these bits are set, all channels remain temporarily stopped.

Transfer requests that occur on channels for which DMA transfer is enabled (DENB=1) while these bits are set are all enabled. The transfer can be started by clearing all these bits.

DMAH	Function
0000 _B	Enables the DMA operation on all channels. (initial value)
Other than 0000 _B	Temporarily stops DMA operation on all channels.

- When reset: Initialized to "0".
- These bits are readable and writable.

[bit30, bit29, bit23 to bit0] (Reserved): Undefined bits

- A read value is undefined.

20.3 Explanation of the DMAC Operation

This section provides an overview of DMAC operation. It also provides details of transfer request settings and transfer sequences and operational details.

■ Overview of DMAC

The DMAC block is a multi-functional DMA controller that controls high-speed data transfer without the use of CPU instructions. It is built into all FR family devices.

20.3.1 Overview of the DMAC Operation

This section provides an overview of DMAC operation.

■ Main DMAC Operations

Functions can be set for each transfer channel independently.

Once starting has been enabled, a channel starts transfer operation only after a specified transfer request has been detected.

After a transfer request is detected, a DMA transfer request is outputted to the bus controller and the bus right is acquired by the bus controller before the transfer is started. The transfer is carried out as a sequence conforming to the mode settings made independently for the channel being used.

■ Transfer Mode

Each DMA channel performs transfer according to the transfer mode set by the MOD[1:0] bits of its DMACB register.

● Block/step transfer

Only a single block transfer unit is transferred in response to one transfer request. DMA then stops requesting the bus controller for transfer until the next transfer request is received.

The block transfer unit is the specified block size: (BLK[3:0] of DMACA).

● Burst transfer

Transfer in response to one transfer request is carried out continuously for the number of times in the specified transfer count.

The specified transfer count is the block size \times transfer count: (BLK[3:0] of DMACA \times DTC[15:0] of DMACA).

■ Transfer Type

● 2-cycle transfer (normal transfer)

The DMA controller operates a read operation and a write operation as a single unit.

Data is read from an address in the transfer source register and then written to another address in the transfer destination register.

■ Transfer Address

The following types of addressing are available and can be set independently for each channel transfer source and transfer destination.

● Specifying the address for a 2-cycle transfer

The value read from a register (DMASA/DMADA) in which an address has been set in advance is used as the address for access. After receiving a transfer request, DMA stores the address from the register in the temporary storage buffer and then starts transfer.

After each transfer (access) operation, the next access address is generated (increment/decrement/fixed selectable) by the address counter and then restored to the temporary storage buffer. Because the contents of the temporary storage buffer are written back to the register (DMASA/DMADA) after each block transfer unit is completed, the address register (DMASA/DMADA) value is updated after each block transfer unit is completed, making it impossible to determine the address in real time during transfer.

■ Transfer Count and Transfer End

● Transfer count

The transfer count register is decremented (-1) after each block transfer unit is completed. When the transfer count register becomes "0", counting for the specified transfer ends, and the transfer stops with the end code displayed or is reactivated (1).

Like the address register, the transfer count register is updated only after each block transfer unit.

If transfer count register reloading is disabled, the transfer ends. If reloading is enabled, the register is initialized and then waits for transfer (DTCR of DMACB)

● Transfer end

Listed below are the sources for transfer end. When transfer ends, a source is indicated as the end code (DSS[2:0] of DMACB).

- End of the specified transfer count (DMACA:BLK[3:0] × DMACA:DTC[15:0]) => Normal end
- A transfer stop request from a peripheral circuit occurred => Error
- An address error occurred => Error
- A reset occurred => Reset

The transfer stop source is indicated (DSS) and the transfer end interrupt or error interrupt for the end source is generated.

MB91470/480 Series

20.3.2 Setting a Transfer Request

The following two types of transfer requests are provided to activate DMA transfer:

- Built-in peripheral request
- Software request

Software requests can always be used regardless of the settings for other requests.

■ Built-in Peripheral Request

A transfer request is generated by an interrupt from the built-in peripheral circuit.

For each channel, set the peripheral's interrupt by which a transfer request is generated (when the IS4 to IS0 bits of DMACA are 1XXXX_B).

Note:

Because an interrupt request used in a transfer request seems like an interrupt request to the CPU, disable interrupts from the interrupt controller (ICR register).

■ Software Request

A transfer request is generated by writing to the trigger bit of a register (STRG of DMACA).

The software request is independent of the above two types of transfer request and can always be used.

If a software request occurs concurrently with activation (transfer enable), a DMA transfer request is outputted to the bus controller immediately and transfer is started.

20.3.3 Transfer Sequence

The transfer type and the transfer mode that determine, for example, the operation sequence after DMA transfer has started can be set independently for each channel (Settings for TYPE[1:0] and MOD[1:0] of DMACB).

■ Selection of the Transfer Sequence

The following sequence can be selected with a register setting:

- Burst 2-cycle transfer
- Block/step 2-cycle transfer

■ Burst 2-Cycle Transfer

In a burst 2-cycle transfer, as many transfers as specified by the transfer count are performed continuously for one transfer source. For a 2-cycle transfer, all 32-bit areas can be specified using a transfer source/transfer destination address.

A peripheral transfer request or software transfer request can be selected as the transfer source.

Table 20.3-1 lists the specifiable transfer addresses for burst 2-cycle transfer.

Table 20.3-1 Specifiable Transfer Addresses for Burst 2-Cycle Transfer

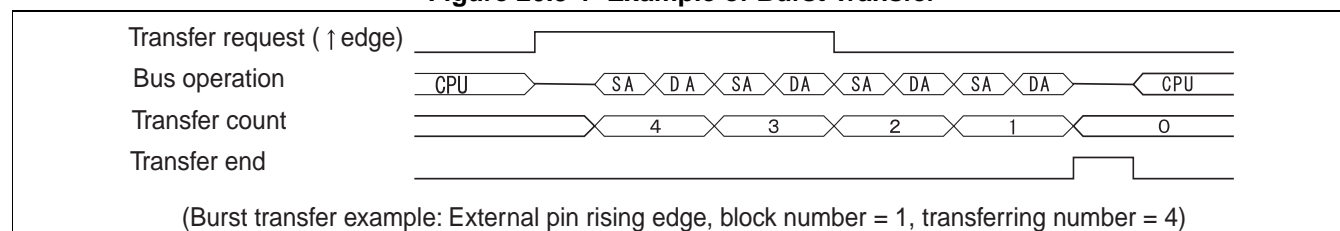
Transfer source addressing	Direction	Transfer destination addressing
All 32-bit areas specifiable	→	All 32-bit areas specifiable

[Features of a burst transfer]

- When one transfer request is received, transfer is performed continuously until the transfer count register reaches 0. The transfer count is the transfer count × block size (BLK[3:0] of DMACA × DTC[15:0] of DMACA).
- Another request occurring during transfer is ignored.
- If the reload function of the transfer count register is enabled, the next request is accepted after transfer ends.
- If a transfer request for another channel with a higher priority is received during transfer, the channel is switched at the boundary of the block transfer unit. Processing resumes only after the transfer request for the other channel is cleared.

Figure 20.3-1 shows an example of burst transfer.

Figure 20.3-1 Example of Burst Transfer



■ Step/Block Transfer 2-Cycle Transfer

For a step/block transfer (Transfer for each transfer request is performed as many times as the specified block count), all 32-bit areas can be specified as the transfer source/transfer destination address.

Table 20.3-2 lists the specifiable transfer addresses for step/block transfer 2-cycle transfer.

Table 20.3-2 Specifiable Transfer Addresses for Step/Block Transfer 2-Cycle Transfer

Transfer source addressing	Direction	Transfer destination addressing
All 32-bit areas specifiable	→	All 32-bit areas specifiable

■ Step Transferring

If "1" is set to the size of the block, it becomes a step transferring sequence.

[Feature of step transferring]

- Transferring is stopped clearing the transferring request after it transfers it once when the transferring request is accepted once (The DMA transferring request is withdrawn for the bus controller).
- When the request is generated again while transferring it, the request is ignored.
- Transferring is started continuing to switch the channel after transferring stops when the transferring request of the channel besides a high priority level is accepted by being transferring it.
Only when the transferring request is generated at the same time, the priority level in the step transferring has the meaning.

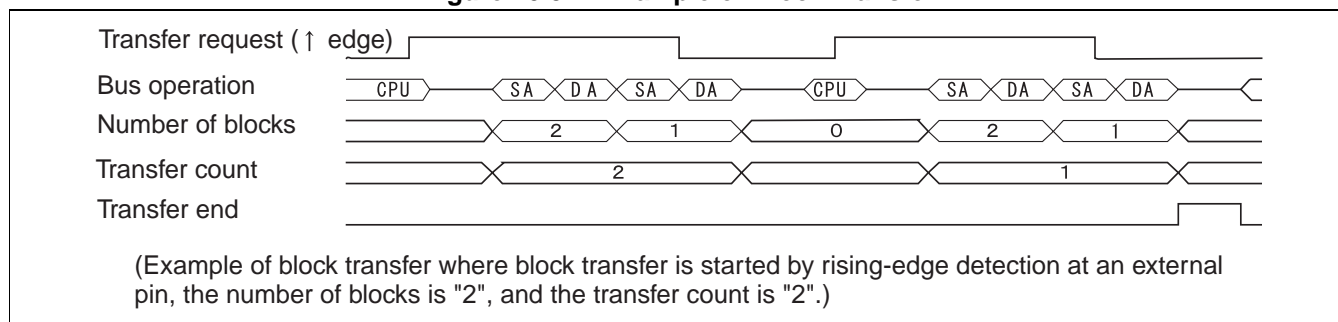
■ Block Transfer

If any value other than "1" is specified as the block size, a block transfer sequence is generated.

[Features of a block transfer]

The block transfer has the same features as those of a step transfer except that one transfer unit consists of multiple transfer cycle counts (number of blocks). Figure 20.3-2 shows an example of block transfer.

Figure 20.3-2 Example of Block Transfer



20.3.4 General Aspects of DMA Transfer

This section describes DMA transfer.

■ Block Size

The unit and increment for transfer data is a set of (the number set in the block size specification register × data width) data.

Since the amount of data transferred in one transfer cycle is determined by the value specified as the data width, one transfer unit consists of the number of transfer cycles for the specified block size.

If a transfer request with a higher priority is received during transfer or if a temporary stop request for a transfer occurs, the transfer stops only at the transfer unit boundary, whether or not the transfer is a block transfer. This arrangement makes it possible to protect data block for which division or temporary stopping is not desirable. However, if the block size is large, response time reduces.

Transfer stops immediately only when a reset occurs, in which case the data being transferred cannot be guaranteed.

■ Reload Operation

In this module, the following three types of reloading can be set for each channel:

(1) Transfer count register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer count register again and waiting for a start request starts.

Set this type of reloading when the entire transfer sequence is to be performed repeatedly.

If reload is not specified, the count register value remains "0" after the transfer is performed the specified number of times and no further transfer is performed.

(2) Transfer source address register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer source address register again.

Set this type of reloading when transfer is to be repeated from a fixed area in the transfer source address area.

If reload is not specified, the transfer source address register value after the transfer is performed the specified number of times becomes the next address. Use this type when the address area is not fixed.

(3) Transfer destination address register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer destination address register again.

Set this type of reloading when transfer is to be repeated to a fixed area in the transfer destination address area.

(The processing hereafter is the same as described in "Transfer source address register reloading" above.)

If only reloading of the transfer source/transfer destination register is enabled, restart after transfer is performed the specified number of times is not implemented and only the values of each address register are set.

[Special examples of operating mode and the reload operation]

For a transfer in burst, block, or step transfer mode, transfer stops temporarily after reload when data transfer ends. Transfer does not start until new transfer request input is detected.

20.3.5 Addressing Mode

Specify the transfer destination/transfer source address independently for each transfer channel.

This section describes the specification method. Specify the addresses based on the transfer sequence.

■ Address Register Specifications

In 2-cycle transfer mode, set the transfer source address in the transfer source address setting register (DMASA) and the transfer destination address in the transfer destination address setting register (DMADA).

[Features of the Address Register]

This register has the maximum 32-bit length. With 32-bit length, all space in the memory map can be accessed.

[Function of the Address Register]

- The address register is read in each access operation and the read value is sent to the address bus.
- At the same time, the address for the next access is calculated by the address counter and the address register is updated using the calculated address.
- For address calculation, increment or decrement is selected independently for each channel, transfer destination, and transfer source. The address increment/decrement width is specified by the address count size register (SASZ/DASZ of DMACB).
- If reloading is not enabled, the address resulting from the address calculation of the last address remains in the address register when the transfer ends.
- If reloading is enabled, the initial value of the address is reloaded.

Reference:

If an overflow or underflow occurs as a result of 32-bit length full address calculation, an address error is detected and transfer on the relevant channel is stopped.

Notes:

- Do not set the address of the register of DMAC to the address register.
 - Do not transfer DMAC to the register by DMAC.
-

MB91470/480 Series

20.3.6 Data Types

Select the data length (data width) transferred in one transfer operation from the following:

- Byte
 - Halfword
 - Word
-

■ Access Address

Since the word boundary specification is also observed in DMA transfer, different low-order bits are ignored if an address with a different data length is specified for the transfer destination/transfer source address.

- Word: The actual access address has a 4-byte length starting with "00_B" as the lowest-order 2 bits.
- Halfword: The actual access address has 2-byte length starting with "0_B" as the lowest-order bit.
- Byte: The actual access address and the addressing match.

If the lowest-order bits in the transfer source address and transfer destination address are different, the addresses as set are output on the internal address bus. However, each transfer target on the bus is accessed after the addresses are corrected according to the above rules.

20.3.7 Transfer Count Control

**Specify the transfer count within the range of the maximum 16-bit length (1 to 65536).
Set the transfer count value in the transfer count register (DTC of DMACA).**

■ Transfer Count Registers and Reload Operation

The register value is stored in the temporary storage buffer when the transfer starts and is decremented by the transfer count counter. When the counter value becomes "0", transfer end for the specified count is detected, and the transfer on the channel is stopped or waiting for a restart request starts (when reload is specified).

[Features of the group of transfer count registers:]

- Each register has 16-bit length.
- All registers have a dedicated reload register.
- If transfer is activated when the register value is "0", transfer is performed 65536 times.

[Reload operation]

- The reload operation can be used only if reloading is enabled in a register that allows reloading.
- When transfer is activated, the initial value of the count register is saved in the reload register.
- If the transfer count counter counts down to "0", end of transfer is reported and the initial value is read from the reload register and written to the count register.

20.3.8 CPU Control

When a DMA transfer request is accepted, DMA generates a transfer request to the bus controller.

The bus controller passes the right to use the internal bus to DMA at a break in bus operation and DMA transfer starts.

■ DMA Transfer and Interrupts

During DMA transfer, interrupts are generally not accepted until the transfer ends.

If a DMA transfer request occurs during interrupt processing, the transfer request is accepted and interrupt processing is stopped until the transfer is completed.

If, as an exception, an NMI request or an interrupt request with a higher level than the hold suppress level set by the interrupt controller occurs, DMAC temporarily cancels the transfer request via the bus controller at a transfer unit boundary (one block) to temporarily stop the transfer until the interrupt request is cleared. In the meantime, the transfer request is retained internally. After the interrupt request is cleared, DMAC generates a transfer request to the bus controller again to acquire the right to use the bus and then restarts DMA transfer.

■ Suppressing DMA

When an interrupt source with a higher priority occurs during DMA transfer, an FR family device interrupts the DMA transfer and branches to the relevant interrupt routine. This feature is valid as long as there are any interrupt requests. When all interrupt sources are cleared, the suppression feature no longer works and the DMA transfer is restarted by the interrupt processing routine. Thus, if you want to suppress restart of DMA transfer after clearing interrupt sources in the interrupt source processing routine at a level that interrupts DMA transfer, use the DMA suppress function. The DMA suppress function can be activated by writing any value other than "0" to the DMAH[3:0] bits of the DMA all-channel control register and can be stopped by writing "0" to these bits.

This function is mainly used in the interrupt processing routines. Before the interrupt sources in an interrupt processing routine are cleared, the DMA suppress register is incremented by "1". If this is done, then no DMA transfer is performed. After interrupt processing, decrement the DMAH[3:0] bits by "1" before returning. If multiple interrupts have occurred, DMA transfer continues to be suppressed since the DMAH[3:0] bits are not "0" yet. If a single interrupt has occurred, the DMAH[3:0] bits become "0". DMA requests are then enabled immediately.

Notes:

- Since the register has only 4 bits, this function cannot be used for multiple interrupts exceeding 15 levels.
 - Be sure to assign the priority of the DMA tasks at a level that is at least 15 levels higher than other interrupt levels.
-

20.3.9 Operation Start

Starting of DMA transfer is controlled independently for each channel, but before transfer starts, the operation of all channels needs to be enabled.

■ Enabling Operation for All Channels

Before activating each DMAC channel, operation for all channels needs to be enabled in advance with the DMA operation enable bit (DMAE of DMACR).

All start settings and transfer requests that occurred before operation is enabled are invalid.

■ Starting Transfer

The transfer operation can be started by the operation enable bit of the control register for each channel. If a transfer request to an activated channel is accepted, the DMA transfer operation is started in the specified mode.

■ Starting from a Temporary Stop

If a temporary stop occurs before starting with channel-by-channel or all-channel control, the temporary stopped state is maintained even though the transfer operation is started. If transfer requests occur in the meantime, they are accepted and retained.

When temporary stopping is released, transfer is started.

20.3.10 Transfer Request Acceptance and Transfer

This section describes transfer request acceptance and transfer.

■ Transfer Request Acceptance and Transfer

Sampling for transfer requests set for each channel starts after starting.

Since peripheral interrupts are handled as level detection, use interrupt clear by DMA to handle the interrupts.

Transfer requests are always accepted while other channel requests are being accepted and transfer performed. The channel that will be used for transfer is determined for each transfer unit after priority has been checked.

20.3.11 Clearing Peripheral Interrupts by DMA

This DMA has a function that clears peripheral interrupts. This function works when peripheral interrupt is selected as the DMA start source (when IS[4:0]=1XXXX_B). Peripheral interrupts are cleared only for the set start sources. That is, only the peripheral functions set by IS[4:0] are cleared.

■ Timing for Clearing Interrupts During DMA

The timing for clearing an interrupt depends on the transfer mode. (See Section "20.4 Operation Flowcharts of the DMAC").

[Block/step transfer]

If block transfer is selected, a clear signal is generated after one block (step) transfer.

[Burst transfer]

If burst transfer is selected, a clear signal is generated after transfer is performed the specified number of times.

20.3.12 Temporary Stopping

This section describes the temporary stopping of DMA transfer.

■ Setting of Temporary Stopping by Writing to the Control Register (Set Independently for Each Channel or All Channels Simultaneously)

If temporary stopping is set using the temporary stop bit, transfer on the corresponding channel is stopped until release of temporary stopping is set again. You can check the DSS bits for temporary stopping.

Transfer is restarted when temporary stopping is canceled.

■ NMI/Hold Suppress Level Interrupt Processing

If an NMI request or an interrupt request with a higher level than the hold suppress level occurs, all channels on which transfer is in progress are stopped at the boundary of the transfer unit and the bus right is released to give priority to NMI/interrupt processing. Transfer interrupts accepted during NMI/interrupt processing are retained, initiating a wait for completion of NMI processing.

Channels for which requests are retained restart transfer after NMI/interrupt processing is completed.

20.3.13 Operation End/Stopping

The end of DMA transfer is controlled independently for each channel. It is also possible to disable operation for all channels at once.

■ Transfer End

If reloading is disabled, transfer is stopped, "Normal end" is displayed as the end code, and all transfer requests are disabled after the transfer count register becomes "0" (Clear the DENB bit of DMACA).

If reloading is enabled, the initial value is reloaded, "Normal end" is displayed as the end code, and a wait for transfer requests starts again after the transfer count register becomes "0" (Do not clear the DENB bit of DMACA).

■ Disabling All Channels

If the operation of all channels is disabled with the DMA operation enable bit (DMAE), all DMAC operations, including operations on active channels, are stopped. Then, even if the operation of all channels is enabled again, no transfer is performed unless a channel is restarted. In this case, no interrupt whatever occurs.

20.3.14 Stopping due to an Error

In addition to normal end after transfer for the number of times specified, stopping as the result of various types of errors and the forced stopping are provided.

■ Transfer Stop Requests from Peripheral Circuits

Depending on the peripheral circuit that outputs a transfer request, a transfer stop request is issued when an error is detected (Example: Error when data is received at or sent from a communications system peripheral).

The DMAC, when it receives such a transfer stop request, displays "Transfer stop request" as the end code and stops the transfer on the corresponding channel.

■ Occurrence of an Address Error

Inappropriate addressing occurring in an addressing mode is detected as an address error. An example of inappropriate addressing is an overflow or underflow that occurs in the address counter when a 32-bit address is specified.

If an address error is detected, "An address error occurred" is displayed as the end code and transfer on the corresponding channel is stopped.

20.3.15 DMAC Interrupt Control

Independent of peripheral interrupts that become transfer requests, interrupts can also be outputted for each DMAC channel.

■ Interrupts That Enable DMAC Interrupt Control Outputs

- Transfer end interrupt:
Occurs only when operation ends normally.
- Error interrupt:
Transfer stop request due to a peripheral circuit (error due to a peripheral)
Occurrence of address error (error due to software)

All of these interrupts are outputted according to the meaning of the end code.

An interrupt request can be cleared by writing "000_B" to DSS2 to DSS0 (end code) of DMACS.

Be sure to clear the end code by writing "000_B" before restarting.

If reloading is enabled, the transfer is automatically restarted. At this point, however, the end code is not cleared and is retained until a new end code is written when the next transfer ends.

Since only one end source can be displayed in an end code, the result after considering the order of priority is displayed when multiple sources occur simultaneously. The interrupt that occurs at this point conforms to the displayed end code.

The following shows the priority for displaying end codes (in order of decreasing priority):

- Reset
- Clearing by writing "000_B"
- Peripheral stop request
- Normal end
- Stopping when address error detected
- Channel selection and control

20.3.16 DMA Transfer during Sleep

The DMAC can also operate in sleep mode.

The DMA transfer in sleep mode is described as follows.

■ Notes on DMA Transfer in Sleep Mode

If you anticipate operations during sleep mode, note the following:

- Since the CPU is stopped, DMAC registers cannot be rewritten. Make settings before sleep mode is entered.
- The sleep mode is released by an interrupt. Thus, if a peripheral interrupt is selected as the DMAC start source, interrupts must be disabled by the interrupt controller.

Similarly, if you do not want to release sleep mode with a DMAC end interrupt, disable DMAC end interrupts.

20.3.17 Channel Selection and Control

Up to five channels can be simultaneously set as transfer channels. In general, an independent function can be set for each channel.

■ Priority among Channels

Since DMA transfer is possible only on 1 channel at a time, priority must be set for the channels.

Two modes, fixed and rotation, are provided as the priority settings and can be selected for each channel group (refer to "■ Channel Group").

● Fixed mode

The order of priority is fixed by channel number, with priority ascending from ch.0 to ch.4:

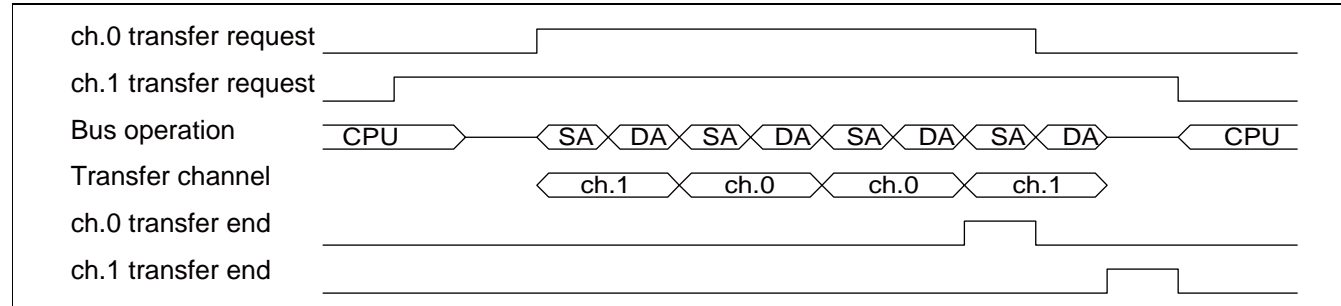
(ch.0 > ch.1 > ch.2 > ch.3 > ch.4)

If a transfer request with a higher priority is received during a transfer, the transfer channel becomes the channel with the higher priority when the transfer for the transfer unit (number set in the block size specification register × data width) ends.

When higher priority transfer is completed, transfer is restarted on the previous channel.

Figure 20.3-3 shows DMA transfer in fixed mode.

Figure 20.3-3 DMA Transfer in Fixed Mode



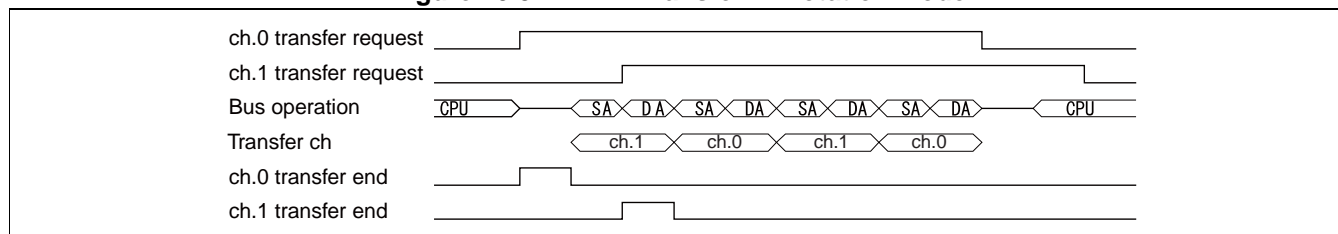
- Rotation mode (between ch.0 and ch.1 only)

When operation is enabled, the initial states have the same order that they would have in fixed mode, but at the end of each transfer operation, the priority of the channels is reversed. Thus, if more than one transfer request is outputted at the same time, the channel is switched after each transfer unit.

This mode is effective when continuous or burst transfer is set.

Figure 20.3-4 shows DMA transfer in rotation mode.

Figure 20.3-4 DMA Transfer in Rotation Mode



■ Channel Group

Set the selection priority as explained in the table below.

Table 20.3-3 lists the settings for DMA selection priority.

Table 20.3-3 Setting DMA Selection Priority

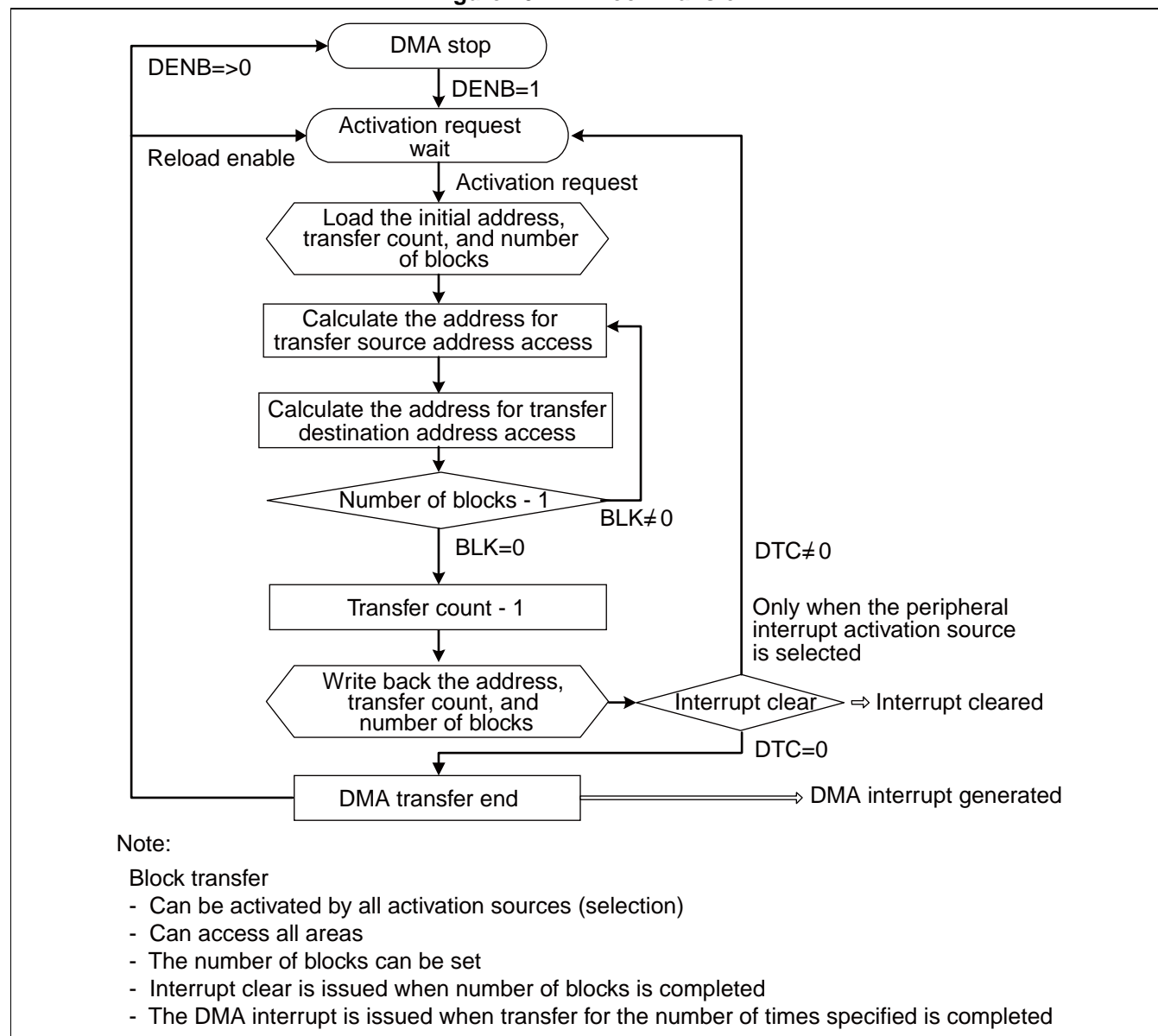
Mode	Priority	Remarks
Fixed	ch.0 > ch.1	—
Rotation	ch.0 > ch.1 ↑ ↓ ch.0 < ch.1	The initial state is the top row. If transfer occurs for the top row, the priority is reversed.

20.4 Operation Flowcharts of the DMAC

Figure 20.4-1 and Figure 20.4-2 show operation flowcharts for DMA transfer.

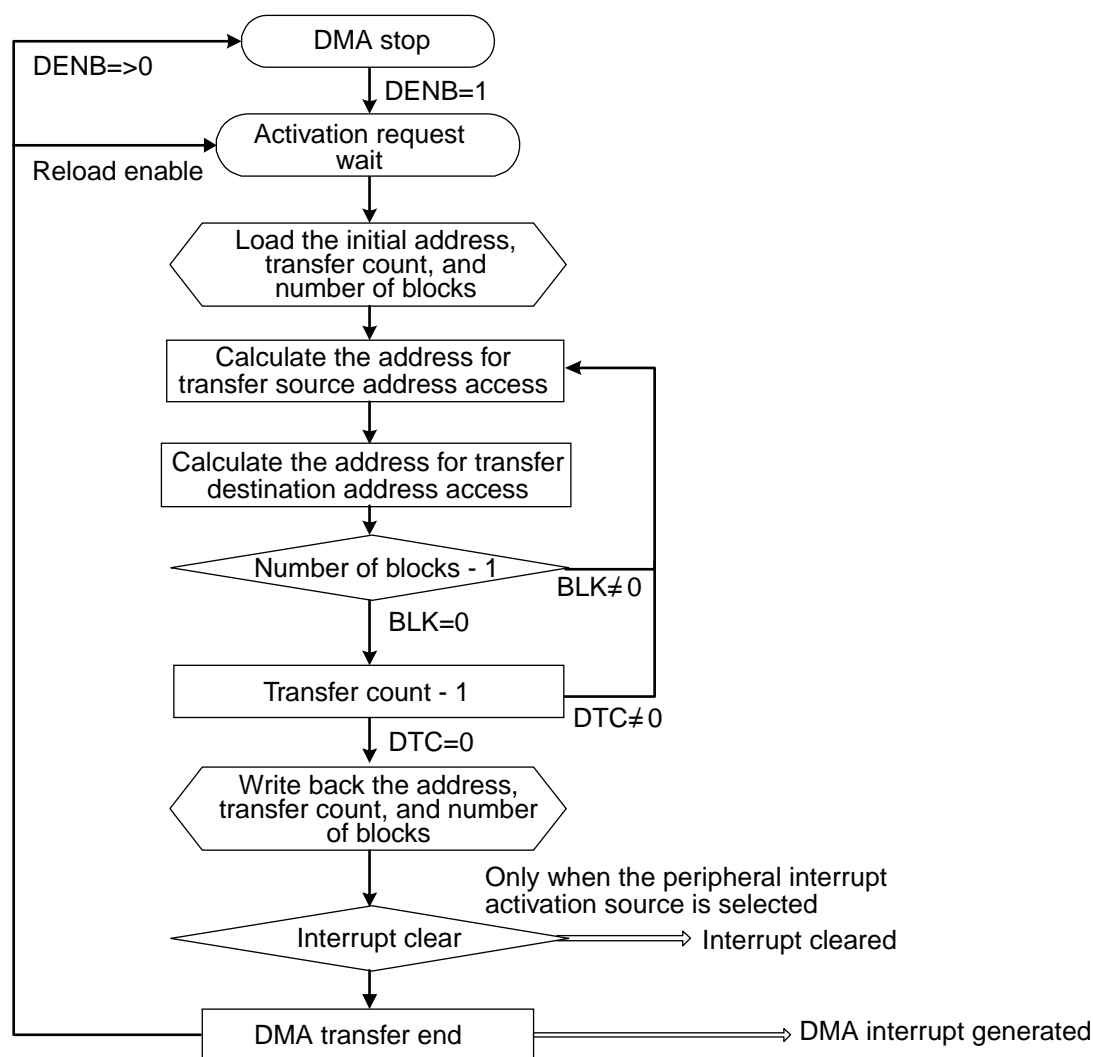
■ Operation Flowchart for Block Transfer

Figure 20.4-1 Block Transfer



■ Operation Flowchart for Burst Transfer

Figure 20.4-2 Burst Transfer



Note:

Burst transfer

- Can be activated by all activation sources (selection)
- Can access all areas
- The number of blocks can be set
- Interrupt clear and the DMA interrupt are issued when transfer for the number of times specified is completed

20.5 Data Bus of the DMAC

This section shows the flow of data during different types of transfer operation.

■ Flow of Data During 2-Cycle Transfer

Figure 20.5-1 to Figure 20.5-6 show the flow of data during 2-cycle transfer.

Figure 20.5-1 External Area → External Area Transfer

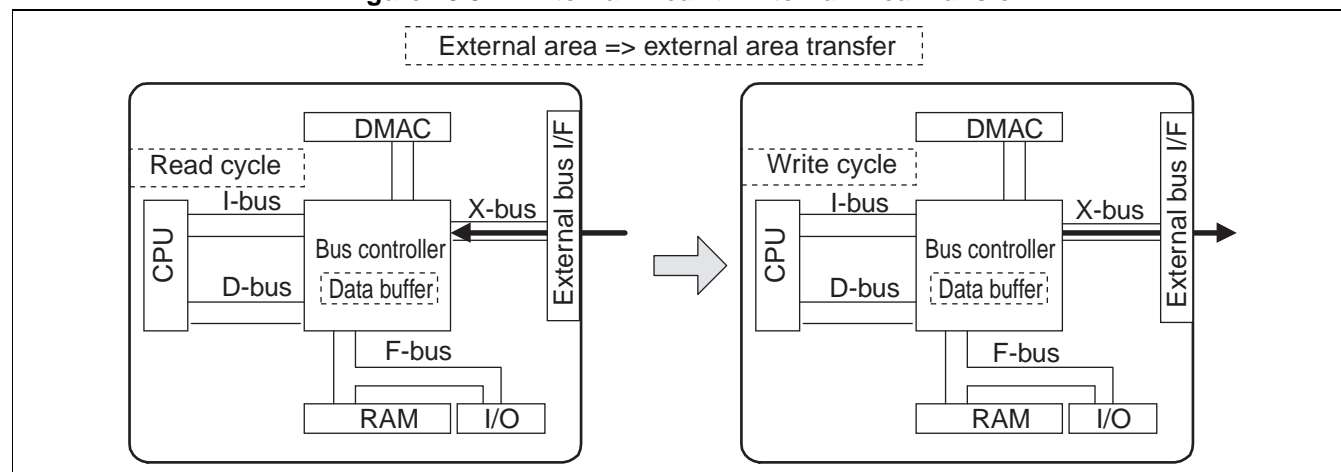


Figure 20.5-2 External Area → Internal RAM Area Transfer

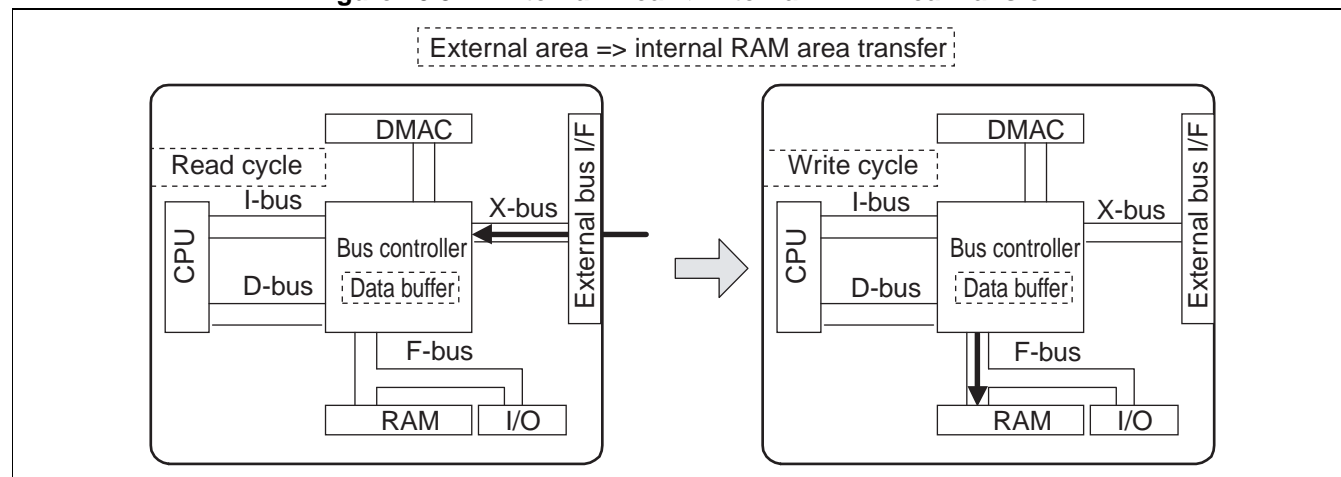


Figure 20.5-3 External Area → Built-in I/O Area Transfer

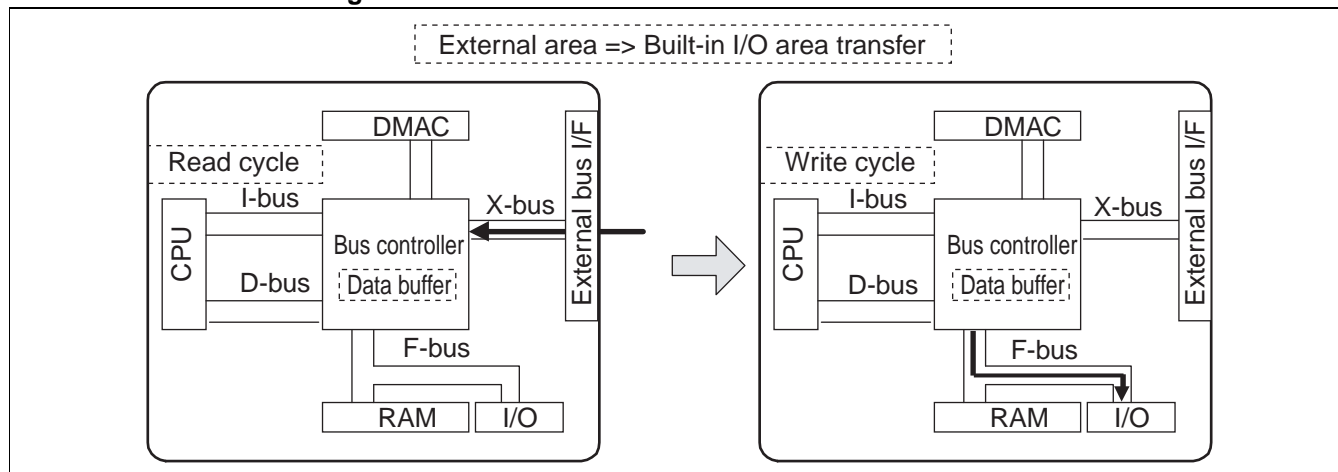


Figure 20.5-4 Built-in I/O Area → Built-in RAM Area Transfer

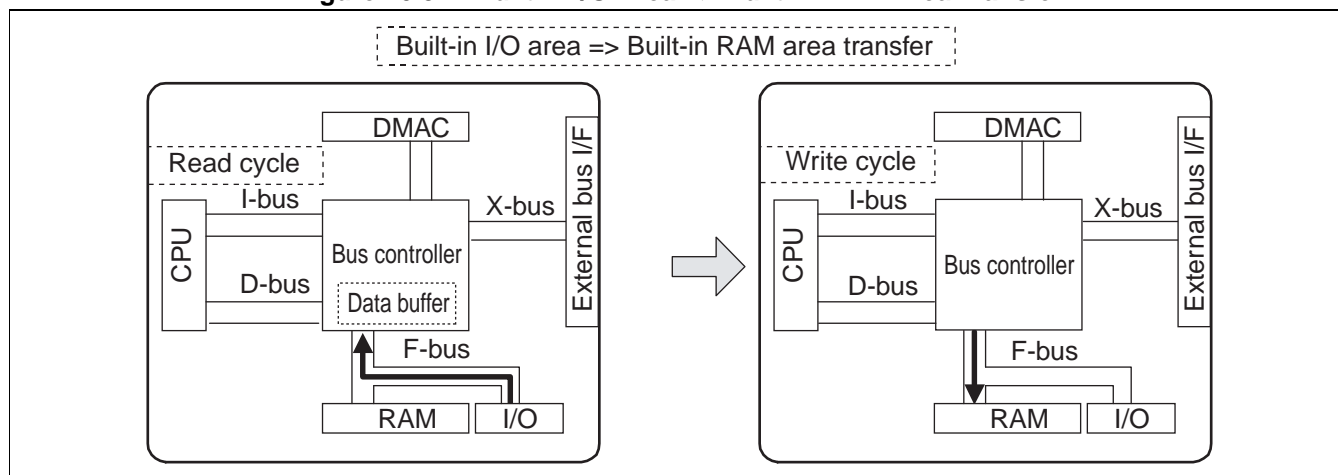


Figure 20.5-5 Internal RAM Area → External Area Transfer

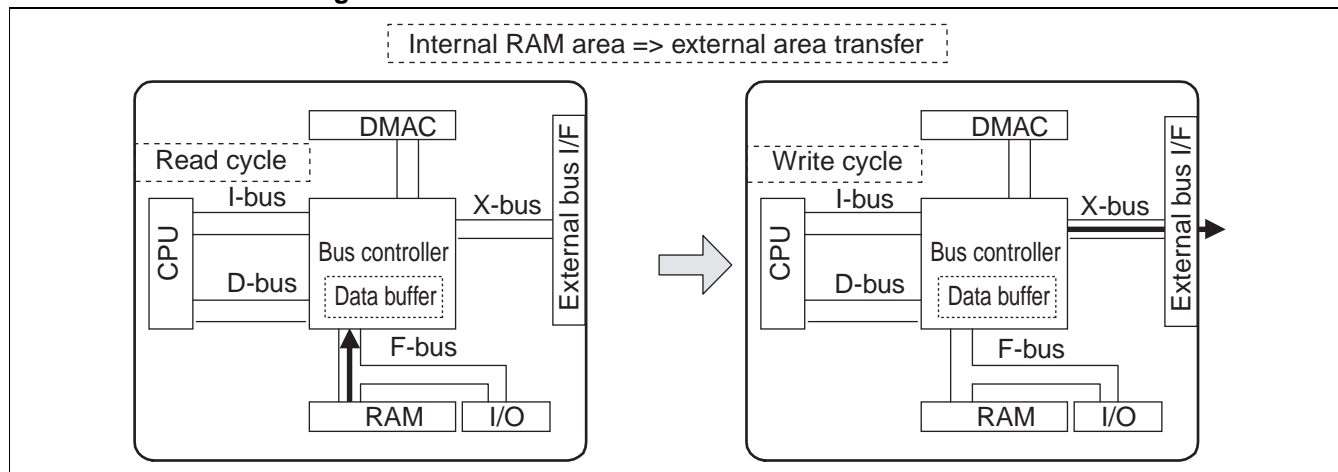
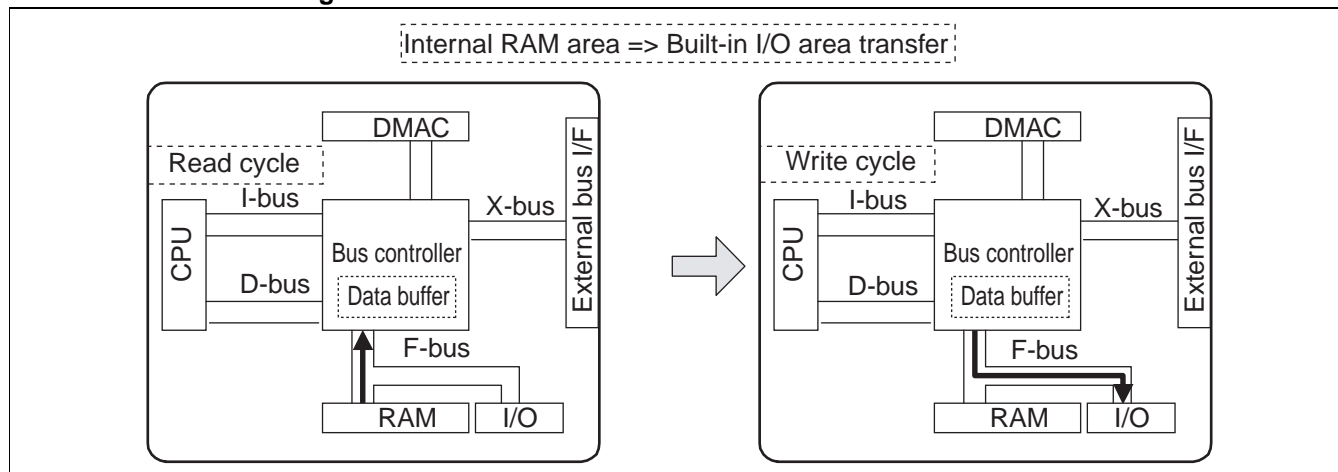


Figure 20.5-6 Internal RAM Area → Built-in I/O Area Transfer



CHAPTER 21

FLASH MEMORY

This chapter explains the overview of the flash memory, the configuration and functions of registers, and the flash memory operation.

- 21.1 Overview of Flash Memory
- 21.2 Flash Memory Registers
- 21.3 Explanation of Flash Memory Operation
- 21.4 Flash Memory Automatic Algorithms
- 21.5 Details of Programming and Erasing Flash Memory
- 21.6 Restrictions on Data Polling Flag (DQ7) and How to Avoid Problems
- 21.7 Notes on Flash Memory Programming

21.1 Overview of Flash Memory

The MB91470/480 series contains 512-Kbyte flash memory which can be erased either for all sectors collectively or sector by sector with a + 5.0 V single power supply and can be programmed by the FR-CPU in half-words (in 16 bits).

■ Overview of Flash Memory Outline

The flash memory is the same as the discrete flash memory which can be programmed from outside the device by using a Flash programmer. In addition, the flash memory has discrete flash memory-equivalent functions and it allows instructions and data to be read from in words (32 bits), contributing to high-speed operation of the device when used as the built-in ROM for FR-CPU.

The combination of the flash memory macro and the FR-CPU interface circuit provides the following functions:

- Serving as CPU's memory for storing programs and data (hereafter, refer to as CPU mode)
 - Accessible at 32-bit bus width when used as ROM
 - Capable of being read/programmed/erased by the CPU instruction (using the automatic algorithm *)
- Functions equivalent to those of a discrete flash memory product (hereafter, refer to as Flash mode)
 - Capable of being read/programmed/erased by a Flash programmer (using the automatic algorithm *)

This section describes the use of the flash memory from the FR-CPU.

For details on using this flash memory with a Flash programmer, refer to the instruction manual for the Flash programmer.

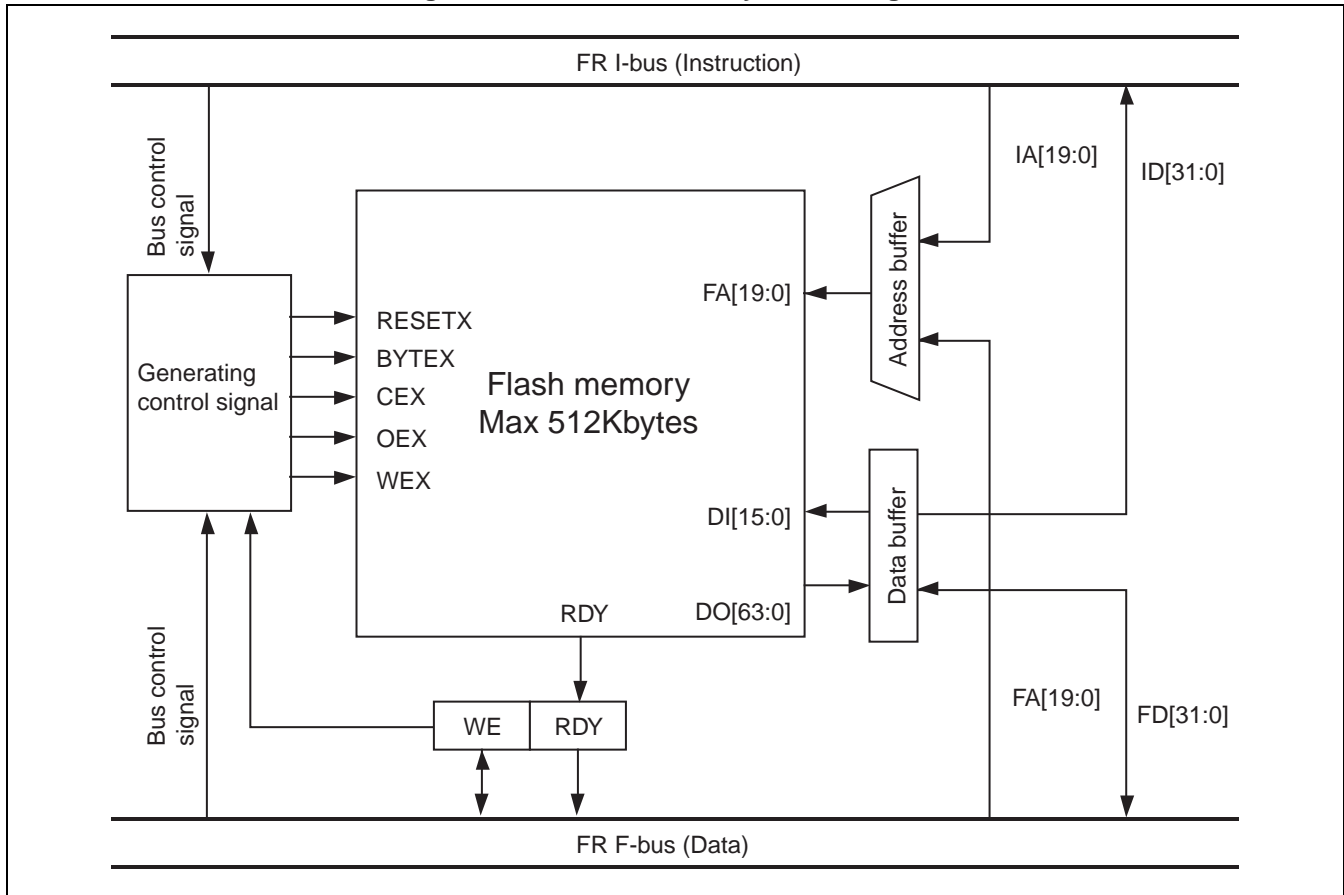
* : Automatic algorithm equivalent to Embedded Algorithm

MB91470/480 Series

Flash Memory Block Diagram

Figure 21.1-1 shows the flash memory block diagram.

Figure 21.1-1 Flash Memory Block Diagram



■ **Memory Map for Flash Memory**

Figure 21.1-2 shows the memory map for flash memory.

Figure 21.1-2 Memory Map for Flash Memory (CPU mode)

	512 Kbytes 64bits		
0010_0000 _H			
000F_C000 _H	8KB (SA7)	8KB (SA6)	
000F_8000 _H	8KB (SA5)	8KB (SA4)	
000F_4000 _H	8KB (SA3)	8KB (SA2)	
000F_0000 _H	8KB (SA1)	8KB (SA0)	
000E_0000 _H	32KB (SA15)	32KB (SA14)	
000C_0000 _H	64KB (SA13)	64KB (SA12)	← 256 Kbytes start address
	64KB (SA11)	64KB (SA10)	
000A_0000 _H	64KB (SA9)	64KB (SA8)	← 384 Kbytes start address
0008_0000 _H			
Bit row	31 to 24, 23 to 16, 15 to 8, 7 to 0	31 to 24, 23 to 16, 15 to 8, 7 to 0	
Address row	+0/+1/+2/+3	+4/+5/+6/+7	

MB91470/480 Series

21.2 Flash Memory Registers

This section explains a configuration and a function of registers used in the flash memory.

■ Overview of Flash Memory Registers

The flash memory has two registers:

- FLCR: Flash control/status register (CPU mode)
- FLWC: Flash wait register

Figure 21.2-1 Flash Memory Registers

FLCR

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
007000 _H	-	-	-	-	RDY	-	WE	-	----X-0 _B
	R/W	R/W	R/W	R	R	R/W	R/W	R/W	

FLWC

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
007004 _H	-	-	-	-	-	WTC2	WTC1	WTC0	-----011 _B
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable

R: Read only

21.2.1 Flash Control/Status Register (FLCR)

This register shows the operation of the Flash memory. The register controls writing in the Flash memory.
Do not access this register in the read modify write (RMW) instruction.

■ **Bit Configuration of Flash Control/Status Register (FLCR)**

Figure 21.2-2 shows the bit configuration of the flash control/status register (FLCR).

Figure 21.2-2 Bit Configuration of Flash Memory Control/Status Register (FLCR)

FLCR									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
007000 _H	-	-	-	-	RDY	-	WE	-	----X-0-B
	R/W	R/W	R/W	R	R	R/W	R/W	R/W	

R/W: Readable/writable
R: Read only

[bit7 to bit5] Reserved: Reserved bits

These bits are reserved bits. Always write "011_B" to these bits.

[bit4] Reserved: Reserved bit

This bit is a reserved bit. Resetting initializes to "0".

[bit3] RDY: Ready

This bit shows the operation of the automatic algorithm (data write/erase).

When this bit is "0", data write or erase by the automatic algorithm is done. Therefore, a new data write/erase command is not accepted to this bit. And, read from the Flash memory area also cannot be done.

The read data shows the status of the Flash memory.

RDY	Function
0	The read/write/erase command of data cannot be accepted while the write/erase is operating.
1	The read/write/erase command of data can be accepted.

- This bit is not initialized at reset (shows the state of the Flash memory at that time).
- Only read is possible. Writing doesn't effect to this bit value.

[bit2] Reserved: Reserved bit

This bit is a reserved bit. Always write "0" to the bit.

[bit1] WE: Write enabled.

This bit controls the writing (programming) data and command into Flash memory.

While this bit is "0", any attempt to program data or commands into Flash memory is ignored.

While this bit is "1", the programming of data and commands into Flash memory is valid, where the automatic algorithm can be started.

Before updating this bit, make sure that the automatic algorithm (data write/erase) has been stopped by the RDY bit. It is impossible to rewrite this bit value while the RDY bit is "0".

WE	Function
0	Disable write access to Flash memory. [Initial value]
1	Enable write access to Flash memory.

- Resetting initializes to "0".
- Readable and writable.

Note:

When the RDY bit of the FLCR register is "0", this bit cannot be rewritten. Rewrite it after confirming the RDY bit is "1". Moreover, please execute the rewriting program of this bit in F-bus RAM or an external area. The program example is described as follows.

● Sample program (At the change to WE=0→1)

```

-----
LDI #_FLCR,R0      // FLCR register (0x7000)
LDI #0b01101010, R1 // Writing data of FLCR register
STB R1,@R0         // Write to FLCR (WE=0→1)
NOP                //NOP for timing adjustment × 2
NOP
-----

```

● Sample program (At the change to WE=1→0)

```

-----
LDI #_FLCR,R0      // FLCR register (0x7000)
LDI #0b01101000, R1 // Writing data of FLCR register
STB R1,@R0         //Write to FLCR (WE=1→0)
NOP                // NOP for timing adjustment × 2
NOP
-----

```

[bit0] Reserved: Reserved bit

This bit is a reserved bit. Always write "0" to the bit.

21.2.2 Flash Wait Register (FLWC)

The flash wait register (FLWC) controls wait states of flash memory access.

■ Bit Configuration of Flash Wait Register (FLWC)

Figure 21.2-3 shows the bit configuration of the flash wait register (FLWC).

Figure 21.2-3 Bit Configuration of Flash Wait Register (FLWC)

FLWC									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
007004 _H	-	-	-	-	-	WTC2	WTC1	WTC0	-----011 _B
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable
R: Read only

[bit7, bit6] Reserved: Reserved bits

These bits are reserved bits. Always write "00_B" to these bits.

[bit5, bit4] Reserved: Reserved bits

These bits are reserved bits. Always write "11_B" to these bits.

[bit3] Reserved: Reserved bit

This bit is a reserved bit. Always write "0" to the bit.

[bit2 to bit0] WTC2 to WTC0: Wait cycle bits

WTC2	WTC1	WTC0	Wait Cycle	Read from flash memory	Write to flash memory
0	0	0	-	Setting disabled	Setting disabled
0	0	1	1	Setting disabled	Setting disabled
0	1	0	2	Setting disabled	Setting disabled
0	1	1	3 [Initial value]		Setting disabled
1	0	0	4	Setting disabled	Setting disabled
1	0	1	5	Setting disabled	Setting disabled
1	1	0	6	Setting disabled	Setting disabled
1	1	1	7	Setting disabled	

- Resetting initializes to "011_B".
- Setting WTC2 to WTC0="011_B" reads from the flash memory (3 wait cycles).
Setting WTC2 to WTC0="111_B" writes to the flash memory (7 wait cycles).

21.3 Explanation of Flash Memory Operation

This section explains the operation of flash memory.

■ Access Mode of Flash Memory

The following two types of access modes are available to the FR-CPU to access flash memory:

- ROM mode

The CPU can read word (32 bits) data collectively from, but cannot write data into flash memory.

- Programming mode

The CPU cannot access flash memory in word (32 bits) but can write data into flash memory in half-words (16 bits).

■ FR-CPU ROM Mode (Read Only in 32 Bits)

In this mode, the flash memory serves as internal ROM for the FR-CPU. The CPU can read a word (32 bits) data collectively but can neither program data into flash memory nor start the automatic algorithm.

● Specification method of mode

- This mode is established when the WE bit of the FLCR register is set to "0".
- The flash memory remains in this mode whenever a reset is cancelled during CPU operation.

● Operation content

When reading a flash memory area, the CPU can read word (32 bits) data collectively.

● Restrictions

- The address mapping and the endian method in this mode are different from those in Flash programmer writing mode.
- Note also that, in this mode, you cannot write command/data into flash memory.

■ FR-CPU Programming Mode (Read/Write in 16 Bits)

In this mode, the CPU can erase or write data. The CPU cannot access word (32 bits) data collectively so that program in flash memory cannot be executed during this mode.

● Specification method of mode

- This mode is established when the WE bit of the FLCR register is set to "1".
- The WE bit is set to "0" after a reset is cancelled during CPU operation. To specify this mode, write "1" to the bit. It returns to ROM mode when the WE bit becomes "0" by rewriting "0" to this bit or generating reset.
- When the RDY bit of the FLCR register is "0", the WE bit cannot be rewritten. Rewrite the WE bit after confirming the RDY bit became "1".

● Operation content

- When reading a flash memory area, the CPU can read half-word (16 bits) data collectively.
- You can start the automatic algorithm by writing the command into flash memory. You can erase data from or write data into flash memory by starting the automatic algorithm. For details on the automatic algorithm, see Sections "21.4 Flash Memory Automatic Algorithms".

● Restrictions

- The address mapping and the endian method are different from those in ROM programmer programming mode.
- The data read in word (32 bits) length is prohibited in this mode.

■ Automatic Algorithm Execution States

When FR-CPU programming mode starts the automatic algorithm, the operation of the automatic algorithm can be read by the RDY bit in FLCR register.

When the RDY bit is "0", data write/erase by the automatic algorithm is being performed. Therefore, a new data write/erase command cannot be accepted. And, the data read from the Flash memory area cannot be done.

The data read when the RDY bit is "0" is a hardware sequence flag that shows the status of the Flash memory.

21.4 Flash Memory Automatic Algorithms

This section details the command sequence for flash memory automatic algorithms, the method of checking their execution status, and programming/erasing flash memory.

■ Overview of Flash Memory Automatic Algorithms

There are four types of commands to invoke their respective flash memory automatic algorithms: Reset, Data Write, Chip Erase, and Sector Erase. For the Sector Erase command, it is possible to control the suspending and resuming of its execution.

21.4.1 Command Sequence

This section explains command sequence to start the automatic algorithm.

■ Command Sequence of Automatic Algorithm

To start the automatic algorithm, program a half-word (16 bits) data into flash memory continuously for once to six times. That is called the commands. The flash memory is reset to the read/reset state if invalid addresses and data are programmed or addresses and data are programmed in a wrong order.

Table 21.4-1 lists the commands used for flash memory write/erase.

Write data by half word (16 bits) when written with FR-CPU. (The address are for CPU mode.)

Table 21.4-1 Command Sequence Table

Command sequence	Bus write access	1st cycle		2nd cycle		3rd cycle		4th cycle		5th cycle		6th cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	FXXXX _H	F0F0 _H	--	--	--	--	--	--	--	--	--	--
Reset	3	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	F0F0 _H	--	--	--	--	--	--
Data write	4	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	A0A0 _H	PA	PD	--	--	--	--
Chip erase	6	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	8080 _H	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	1010 _H
Sector erase	6	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	8080 _H	F5557 _H	AAAA _H	FAAAB _H	5555 _H	SA	3030 _H
Sector erase temporary stop	Sector erase operation is temporary stopped by specifying the address = FXXXX _H , data = B0B0 _H .												
Sector erase restart	Sector erase operation is resumed after being temporary stopped by specifying the address = FXXXX _H , data = 3030 _H .												
Continuous mode	3	F5557 _H	AAAA _H	FAAAB _H	5555 _H	F5557 _H	2020 _H	--	--	--	--	--	--
Continuous write	2	FXXXX _H	A0A0 _H	PA	PD	--	--	--	--	--	--	--	--
Continuous mode reset	2	FXXXX _H	9090 _H	FXXXX _H	F0F0 _H or 0000 _H	--	--	--	--	--	--	--	--

PA: Writing address

SA: Sector address (Specify address 4n+2 or 4n+3)

PD: Write data

■ Reset Command

The flash memory is set to the read/reset state. The flash memory remains in the read state until another command is input. The flash memory is set to the read/reset state automatically when the power is turned on. In this case, no command is required to read data.

To return to the read/reset state when the timing limit is exceeded, issue a reset command sequence. Data is read from flash memory in the read cycle.

■ Data Write Command

In CPU programming mode, programming is performed in half-word. Data writing is performed by programming the command sequence with four cycles. Programming into memory starts after the last programming cycle of the command sequence.

Once data writing command sequence is executed, the flash memory does not require further external control.

After the flash memory activates the automatic algorithm and set the data polling flag (DQ7) to the reverse value of bit7, the flash memory generates an internally produced appropriate programming pulse to verify the margins of programmed cells. When the automatic algorithm finishes, the value of data polling flag (DQ7) matches the data written into bit7. Afterward, the operation state returns to the read/reset state. In this way, data polling flag (DQ7) indicates that data is being written in the flash memory.

During running the automatic algorithm for data writing, any command programmed into flash memory is ignored. If a hardware reset is activated during programming at an address, the data at that address is not guaranteed. Programming is allowed in any order of addresses and beyond the boundaries of sectors. Data "0" which is already programmed into flash memory cannot be returned to data "1" by writing. If data "1" is programmed over data "0", either the data polling algorithm determines that the element is defective or data "1" apparently looks as if it were programmed.

When the data is read in read/reset state, however, it remains as "0". Data "0" can be updated to data "1" only by erasing.

■ Chip Erase Command

The command sequence for chip erasure (erasing all of the sectors collectively) is performed by programming flash memory six times. Chip erasing is started by inputting the chip erase command.

Before chip erasing, the user need not perform programming to flash memory. When the automatic algorithm for chip erasing is executed, the flash memory automatically verifies its cells by programming patterns of 0s (preprogramming) before erasing all the cells. During preprogramming, the flash memory requires no external control.

Automatic algorithm for chip erasing is started by programming of the command sequence and terminates when the data polling flag (DQ7) is set to "1", and the flash memory returns to the read/reset state. The chip erase time is "sector erase time \times the number of all sectors + chip program (preprogram) time".

■ Sector Erase Command

Sector erasure is performed by writing the command sequence with six cycles. The sector erase command is written in the sixth cycle to start erasing a sector. During a sector erasure time-out period of 50 μ s or more after the last sector erase command is programmed, the next sector erase command can be accepted.

The erasing of multiple sectors can be accepted at the same time by programming the sixth cycle of the command sequence. This sequence is executed by programming the sector erase command (3030_H) continuously at the addresses of the sectors to be erased at the same time. When the 50 μ s or more sector erasure time-out after the last sector erase command is programmed expires, the flash memory starts erasing the sectors. To erase multiple sectors at the same time, therefore, the sector erase command for each of the sectors must be input within a time-out period of 50 μ s, or the command may not be accepted if it expires. You can check whether the successive sector erase command is valid by monitoring the sector erase timer flag (DQ3) (See ■ Hardware Sequence Flag in section "21.4.2 Confirming Automatic Algorithm Execution States").

If any command other than the sector erase command and erasure pause command is input in a sector erasure time-out period, the flash memory is reset to the read mode and invalidates the preceding command sequence. In this case, the relevant sector is erased completely by erasing it again. Sector addresses can be input to the sector erase buffer for any number of sectors in any combination.

The sector erasure is started after a sector erasure time-out period of 50 μ s or more after the last sector erase command is programmed and terminates when the data polling flag (DQ7) is set to "1". The flash memory returns to the read/reset state. Data polling (DQ7) works for any address in the sectors erased. The multiple-sector erase time is "(sector erase time + sector program (preprogram) time) \times the number of sectors erased".

■ Sector Erase Pause Command

The sector erase pause command allows the user to suspend the flash memory automatic algorithm during erasure of sectors in order to read data from or program data into other sectors. This command is valid only during sector erasure; it is ignored during chip erasure or programming. The sector erasure pause command (B0B0_H) is valid only during the period of sector erasure, including the sector erasure time-out period after the sector erase command (3030_H). When this command is input during the time-out period, the flash memory terminates the time-out and suspends erasure. The flash memory restarts erasure when the erasure restart command is programmed. The erasure pause and restart commands can be programmed with any address.

When the erasure pause command is input during sector erasure, it takes a maximum of 20 μs for the flash memory to pause erasure. When the flash memory enters the erasure pause mode, RDY bit in FLCR register and the data polling flag (DQ7) output "1", and the toggle bit flag (DQ6) stops toggling. You can check whether erasure is suspended by inputting the sector address being erased to monitor the values read from the toggle bit flag (DQ6) and the data polling flag (DQ7). An attempt to program another erasure pause command is ignored. When erasure is paused, the flash memory enters the erasure pause read mode. Data reading in this mode is the same as typical data reading, except that it is effective for sectors not being erasure-paused. In erasure pause read mode, the toggle bit 2 flag (DQ2) toggles for continuous reading from the sector being erasure-paused.

In erasure pause read mode, the user can program into flash memory by programming the command sequence. This program mode is the erasure pause program mode. Programming in this mode is the same as normal data writing, except that it is effective for sectors containing data not being erasure-paused. In erasure pause program mode, the toggle bit 2 flag (DQ2) toggles for continuous reading from the sector being erasure-paused. The erasure pause state can be detected by checking the erasure pause bit (bit6).

Note that the data polling flag (DQ7) must be read for the address being programmed while the toggle bit flag (DQ6) can be read for any address. To restart sector erasure, input the sector erasure restart command (3030_H). Another restart command is ignored if input at the point of restarting sector erasure. In contrast, the erasure pause command can be input after the flash memory restarts erasure.

21.4.2 Confirming Automatic Algorithm Execution States

As this flash memory uses the automatic algorithm for program/erase flow. This automatic algorithm enables confirmation of the operating state of the built-in flash memory using hardware sequences.

■ RDY Bit

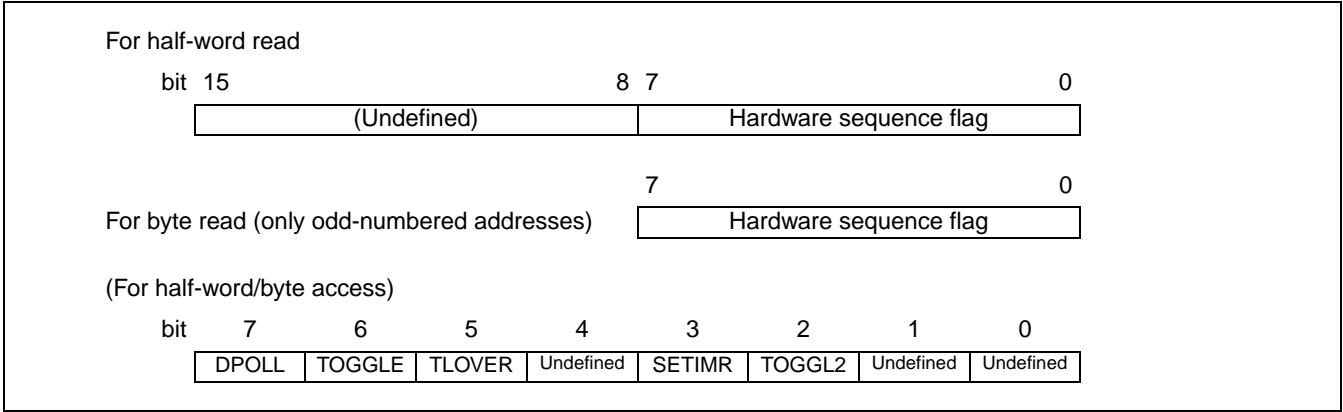
Besides the hardware sequence flag, the Flash memory has RDY bit of the Flash memory status register (FLCR) as a method of informing whether an internal automatic algorithm is now executing or ended.

When the read value of the RDY bit is "0", the Flash memory is now writing data or erasing. At this time, neither the data writing command nor the erase command are accepted. When the read value of the RDY bit is "1", the flash memory is in the state of the read/data write/erase operation waiting.

■ Hardware Sequence Flag

Figure 21.4-1 shows the structure of the hardware sequence flag.

Figure 21.4-1 Structure of Hardware Sequence Flag



Note:

The word reading is prohibited. Use only the FR-CPU programming mode.

A hardware sequence flag is obtained as data by reading an address (an odd-numbered address during byte access) of flash memory during execution of the automatic algorithm. The obtained data contains five effective bits, each of which indicates a state of the automatic algorithm.

It doesn't have the meaning about these flags at the FR-CPU ROM mode. In the FR-CPU programming mode, execute the half word read or the byte read.

Table 21.4-2 lists hardware sequence flag states.

Table 21.4-2 Hardware Sequence Flag Status List

State		DPOLL	TOGGLE	TLOVER	SETIMR	TOGGL2
During execution	Programming	Inverted data	Toggle	0	0	1
	Sector erasing	Time-out period	1	Toggle	0	1
		Erasing period	0	Toggle	0	1
	Erasing pause mode	Reading (Sector being erased)	1	1	0	0
		Reading (Sector unerased)	Data	Data	Data	Data
		Programming (Sector unerased)	Inverted data	Toggle* ²	0	0
Time limit is exceeded	Data erasing Operation	Inverted data	Toggle	1	0	1
	Temporary erase stop mode	0	Toggle	1	1	*4
	Chip/sector erasing	0	Toggle	1	1	*4

*1: TOGGL2 operates as a toggle at the time of the continuous read from the temporary erase stop sector.

*2: TOGGLE operates as a toggle even at continuous reading time from whatever address

*3: When the written address is read at writing the temporary erase stop, TOGGL2 becomes "1". However, TOGGL2 operates as a toggle at the time of the continuous reading from the temporary erase stop sector.

*4: When TLOVER is "1" (over the time limit), TOGGL2 operate as a toggle to a continuous reading to the sector in the write/erase, and does not operate as a toggle to read to other sectors.

Each bit in the table is explained below.

[bit7] :DPOLL : Data polling flag (DQ7)

[bit6] :TOGGLE : Toggle bit flag (DQ6)

[bit5] :TLOVER : Timing limit exceeded flag (DQ5)

[bit3] :SETIMR : Sector erase timer flag (DQ3)

[bit2] :TOGGL2 : Toggle bit flag 2

The following describes the function of each bit:

[bit7] DPOLL: Data polling flag (DQ7)

The data polling flag uses a data polling function to indicate that the execution of the automatic algorithm is currently in progress or completed.

- During programming

When the flash memory is read during execution of the automatic algorithm, it outputs the inverted version of data finally written to bit7 without accessing the address located by the address signal. When the flash memory is read-accessed upon completion of the automatic algorithm, it outputs bit7 of the data read from the address located by the address signal.

- During chip erasure

When the flash memory is read during execution of erasure or erasure sector algorithm, it outputs "0" regardless the current address. In the same way, 1 is output when it ends.

- During sector erasure

When the flash memory is read-accessed from the sector being erased during execution of the sector erase automatic algorithm, it outputs "0". Due to restrictions on the function in this series, the flash memory outputs "1" for 50 μ s to 160 μ s after the sector erase command is issued, and then outputs "0". After the sector erase is terminated, the flash memory outputs "1".

For restrictions on the data polling flag (DQ7) and how to avoid erroneous judgment of sector erase completion, see section "21.6 Restrictions on Data Polling Flag (DQ7) and How to Avoid Problems".

- During sector erasure suspended

When the flash memory is read during suspended sector erasure, it outputs "1" if the address located by the address signal belongs to the sector being erased. If the address does not belong to the sector being erased, the flash memory outputs bit7 of data read at the address located by the address signal. By referring this bit along with bit6 described below as the toggle bit, you can check whether sector erasure is currently being suspended and which sector is being erased.

Note:

Any read access to the specified address is ignored during the startup of the automatic algorithm. In relation to reading data, the data polling flag must be completed before data can be output from any other bit.

Therefore, upon completion of the automatic algorithm, data should be read after the read access which confirms the completion of the data polling.

[bit6] TOGGLE: Toggle bit flag (DQ6)

Like the data polling flag, the toggle bit flag uses a toggle bit function mainly to indicate that the execution of the automatic algorithm is currently in progress or completed.

- During programming, chip/sector erasure

When the flash memory is continuously read during execution of the automatic algorithm for programming, chip/sector erasure, it outputs the result of toggling between "1" and "0" for each read operation without bit6 of reading the address located by the address signal. When the automatic algorithm terminates, the flash memory stops toggling bit6 for read access and outputs bit6 (DATA:6) of data read at the address located by the address signal.

- During sector erasure suspended

When the flash memory is read during suspended sector erasure, it outputs "1" if the address located by the address signal belongs to the sector being erased. If the address does not belong to the sector being erased, the flash memory outputs bit6 of data read at the address located by the address signal.

Reference:

When the sector that tries to be written is a sector that the rewriting protection is done at writing, the toggle operation is ended without rewriting data after the toggle of about 2ms operates. When the sector that tries to be written is protected from rewriting in the erasing, it returns to the state of read/reset without rewriting data after the toggle of about 100ms operates.

[bit5] TLOVER: Timing limit exceeded flag (DQ5)

This flag is used to report that a time (number of internal pulses) specified internally with flash memory is exceeded while the automatic algorithm is being executed.

- During programming, chip/sector erasure

Read-accessing after the startup of the automatic algorithm for programming, chip erasure, or sector erasure operation outputs "0" if the specified time (time required for write/erase operation) has not been exceeded, or "1" if the time has been exceeded.

As this is not affected by whether the automatic algorithm is currently being executed or completed, you can determine if the data write/erase operation has been successful. In other words, you can determine that the data write operation has been unsuccessful, if the automatic algorithm is still being executed by the data polling function or the toggle bit function when this flag has output "1".

For example, a failure will occur if an attempt is made to write "1" to the flash memory address which contains "0". In this case, the flash memory will be locked; therefore, the automatic algorithm will not be completed.

On rare occasions, it can be completed properly as if "1" had been written successfully. As a consequence, valid data cannot be output from the data polling flag. Also, the toggle bit flag does not suspend the toggle operation, resulting in an exceeded time limit. Then, the timing limit exceeded flag outputs "1".

Note that this indicates that the flash memory was not used correctly rather than any defect with the flash memory. If this event occurs, the reset command should be executed.

[bit3] SETIMR: Sector erase timer flag (DQ3)

The sector erasure timer flag indicates whether or not the sector erasure time-out period has passed after the execution of the sector erasure command.

- During programming, chip/sector erasure

When a read access is performed after the execution of the sector erasure command, the flash memory outputs "0" within the sector erasure time-out period, or "1" after that period, rather than accessing the address specified by the address signal of the sector which has issued the command.

If this flag is "1" when the data polling function or the toggle bit function is indicating that the erase algorithm is currently being executed, that means that an internally controlled erasure operation has started. After that, any command other than the ones for programming the sector erasure code or suspending the erasure is ignored until the erasure is completed.

If this flag is "0", the flash memory accepts the additional sector deletion code to be written. To confirm this, it is advisable to check the state in this flag by software before programming succeeding sector erase code. If 1 is shown at the 2nd status check, the additional sector deletion code may not have been accepted.

- During sector erasure

When reading is carried out while sector deletion is suspended, the flash memory outputs 1 if the address indicated by the address signal belongs to the sector during deletion. If the address does not belong to the sector being erased, the flash memory outputs bit3 of data read at the address located by the address signal.

[bit2] TOGGL2: Toggle bit flag 2 (DQ2)

This toggle bit flag 2 (DQ2) is used, along with another toggle bit flag (DQ6) of bit6, to detect whether the flash memory is executing automatic erasure and whether erasure is currently being suspended.

- During programming, chip/sector erasure

Same toggle operation as the toggle bit flag (DQ6) is performed.

- During sector erasure suspended

If the flash memory is under reading mode during erasure suspension, bit2 operates the toggle by continuously reading addresses from the sector in which deletion is suspended. If the flash memory is under the execution of the automatic algorithm for programming during erasure suspension, 1 is read by bit2 by continuously reading addresses from the sector in which deletion is not suspended. Unlike bit2, toggle bit flag (DQ6) toggles only during normal programming, erasure, or erasure pause programming.

Reference:

Bit2 and bit6 are used at the same time to detect the reading the temporary erase stop mode. (As for bit2, the toggle operates. But as for bit6, the toggle doesn't operate.)
In addition, bit2 is used to detect the erasing sector. As for bit2, the toggle operates to read from the erasing sector, when the Flash memory is on the erase operating.

21.5 Details of Programming and Erasing Flash Memory

This section describes the procedures to issue the Reset, Program, Chip Erase, Sector Erase, Sector Erase Suspend, and Sector Erase Resume commands to flash memory for invoking their respective automatic algorithms to execute their operations.

■ Overview of Programming and Erasing Flash Memory

The following automatic algorithms can be performed for flash memory by carrying out the writing of the command sequences:

- Reset
- Program
- Chip Erase
- Sector Erase
- Sector Erase Suspend
- Sector Erase Resume
- Continuous mode

Each series of path write cycles must be executed continuously. The completion of each automatic algorithm can be checked, for example, by the data polling function. When the automatic algorithm terminates normally, the flash memory returns to the read/reset state.

21.5.1 Read/Reset State

This section describes the procedure for issuing the Reset command to place the flash memory in the read/reset state.

■ Placing the Flash Memory in the Reset State

To place the flash memory in the read/reset state, issue the Reset command in the command sequence table continuously to the target sector in flash memory.

There are two different command sequences available to the Reset command: one for a single bus operation and the other for three writings. The two command sequences are basically the same.

The read/reset state is the default state of the flash memory. The flash memory always enters the read/reset state when the power is turned on and upon normal termination of a command. In the read/reset state, the flash memory is waiting for input of another command.

In the read/reset state, data can be read by normal read access. Like masked ROM, flash memory is program-accessible from the CPU. The Read/Reset command is therefore not required for reading data normally. Use the command to initialize an automatic algorithm, for example, when the command has failed to terminate normally for some reason.

21.5.2 Programming Data

This section describes the procedure for issuing the Program command to program data into flash memory.

■ Programming Data into Flash Memory

To invoke the automatic algorithm for programming data into flash memory, issue the Program command in the command sequence table continuously to the target sector in flash memory. Upon completion of a data write to the target address in the fourth cycle, the automatic algorithm is activated to start programming.

■ Addressing

The write address specified in the programming data cycle must be an even-numbered address. Programming fails if an odd-numbered address is specified. That is, programming must be performed in halfwords to even-numbered addresses.

Although programming can be performed in any order of addresses and beyond a sector boundary, each Program command can write only one halfword (16 bits) of data.

■ Notes on Programming Data

Programming cannot restore data from "0" to "1".

If you attempt to write data "1" to data "0", the data polling algorithm or toggle operation does not terminate, the flash memory device is regarded as defective, and the specified programming time is exceeded, resulting in an error detected by the timing limit excess flag. Otherwise, the data "1" appears to have been written normally. If the data is then read in the read/reset state, however, the value will still be "0". Only erasing data "0" can set it to "1".

During execution of the automatic algorithm for programming, all commands are ignored. Note that, if a hardware reset occurs during programming, the data at the address currently being programmed is not guaranteed.

■ Flash Memory Programming Procedure

Figure 21.5-1 shows an example of the flash memory programming procedure.

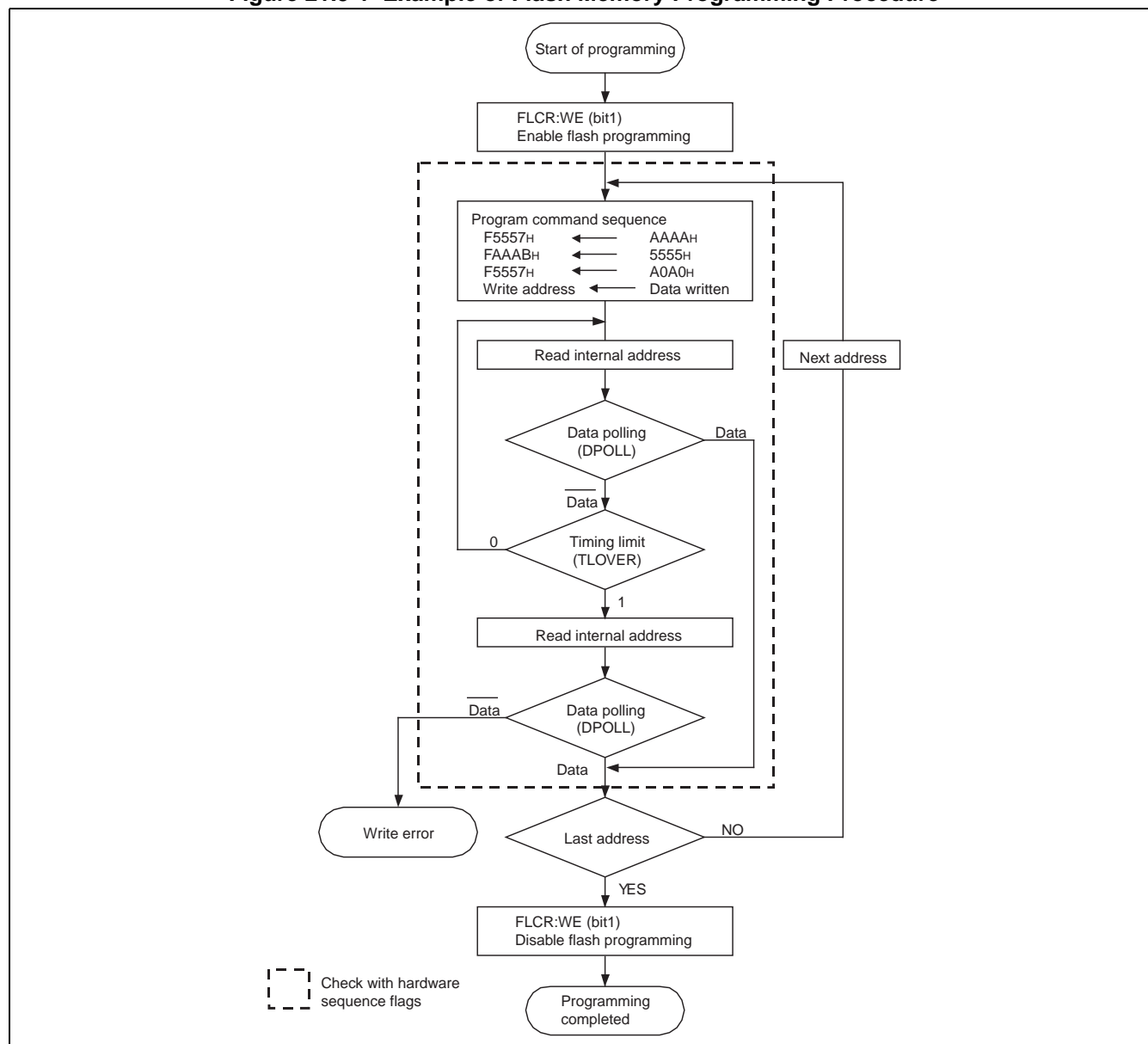
The states of the automatic algorithm in flash memory can be checked by referencing the hardware sequence flags. In the example, the data polling flag (DQ7) is used to determine whether programming has been completed.

The data to be used for checking the flag is read from the last write address.

Since the data polling flag (DQ7) changes the setting with the timing limit excess flag (DQ5) with the skew almost at the same time, the data polling flag (DQ7) must be checked again even when the timing limit excess flag (DQ5) contains "1".

The toggle bit flag (DQ6) stops the toggle operation with the skew almost at the same time when the timing limit excess flag (DQ5) is set to "1". The toggle bit flag (DQ6) must therefore be checked again.

Figure 21.5-1 Example of Flash Memory Programming Procedure



21.5.3 Erasing Data (Chip Erase)

This section describes the procedure for issuing the Chip Erase command to erase all data from flash memory.

■ Erasing Data from Flash Memory (Chip Erase)

To erase all data from flash memory, issue the Chip Erase command in the command sequence table continuously to the target sectors in flash memory.

The Chip Erase command is executed in six writings.

The chip erase operation starts upon completion of the write in the sixth cycle. The user does not have to program into flash memory before performing chip erasure. During execution of the automatic algorithm for chip erasure, the flash memory performs verification by automatically writing "0"s to all cells before erasing them.

21.5.4 Erasing Data (Sector Erase)

This section describes the procedure for issuing the Sector Erase command to erase one or more arbitrary sectors in flash memory. The Sector Erase command can erase data from flash memory in sector units. It allows two or more sectors to be specified at the same time.

To erase an arbitrary sector in flash memory, issue the Sector Erase command in the command sequence table continuously to the target sector in flash memory.

■ Specifying One or More Sectors

The Sector Erase command is executed in six writings. The sector erase time-out period of 50 μ s or more is started by writing the sector erase code (3030_H) to any address accessible in the target sector in the sixth cycle.

To erase more than one sector, continue the above sequence by writing further sector erase codes (3030_H) to the addresses in the sectors to be erased.

■ Notes on Specifying Multiple Sector

Erasing sectors starts at the end of the sector erase time-out period of 50 μ s or more after writing the last sector erase code. That is, to erase more than one sector at a time, the address in each sector to be erased and the sector erase code (in the sixth cycle of the command sequence) must be input within 50 μ s after writing the sector erase code for the previous sector. Sectors specified after this time may not be accepted.

Whether subsequent sector erase code writes are effective or not can be monitored by using the sector erase timer flag (DQ3). Note that the address to read the Sector Erase command must point to the sector to be erased.

■ Sector Erasing Procedure

The state of the automatic algorithm in flash memory can be checked by referencing hardware sequence flags. Figure 21.5-2 shows an example of the flash memory sector erasing procedure.

In the example, the toggle bit flag (DQ6) is used to determine whether the automatic algorithm for sector erasure has been completed.

Note that the data to be used for checking the flag is read from the sector to be erased.

Since the toggle bit flag (DQ6) stops toggle operation with the skew almost at the same time when the timing limit excess flag (DQ5) is set to "1", the toggle bit flag (DQ6) must be checked again even when the timing limit excess flag (DQ5) contains "1".

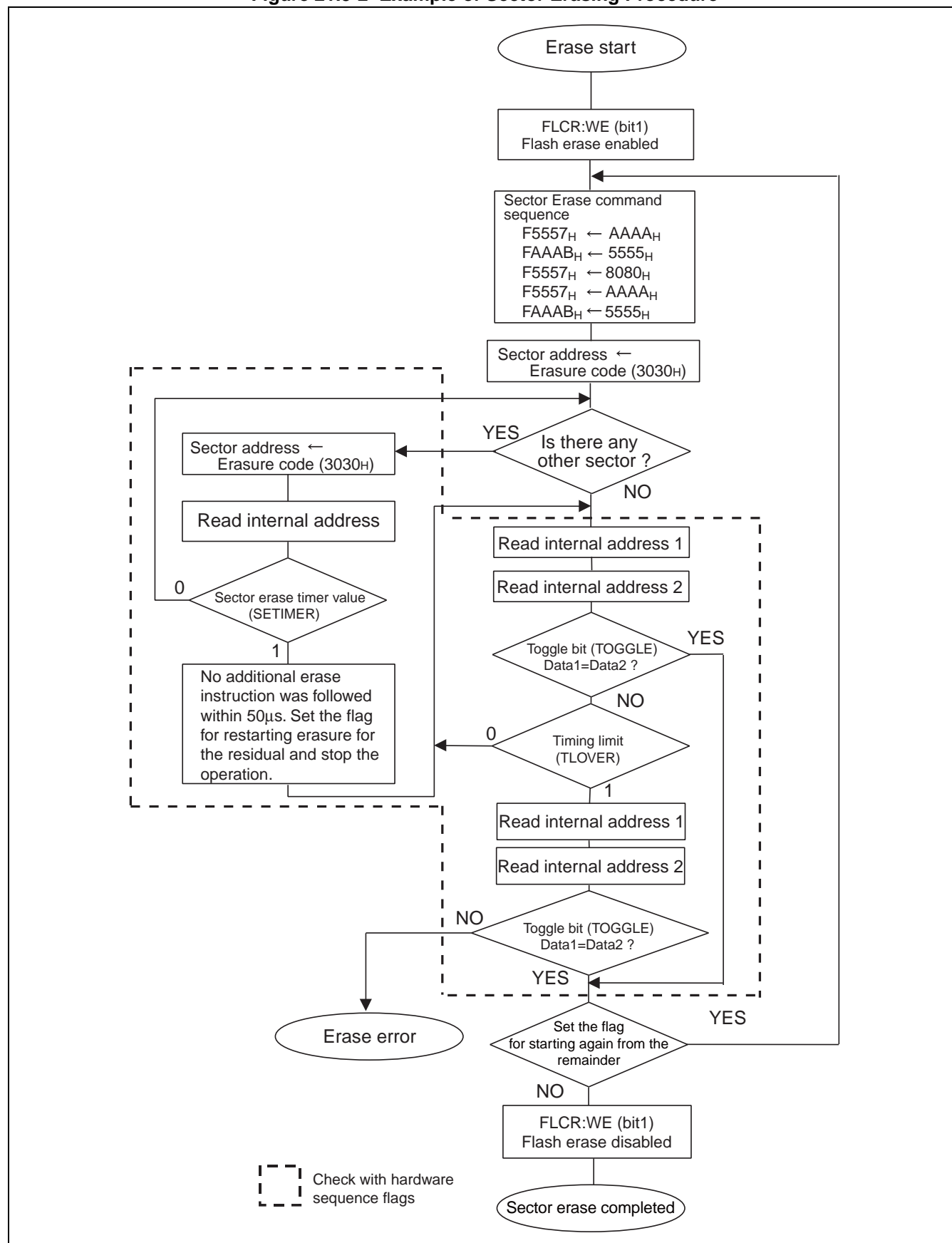
Similarly, as the data polling flag (DQ7) changes the setting with the timing limit excess flag (DQ5) with the skew almost at the same time, the data polling flag (DQ7) must also be checked again.

■ Restrictions on Data Polling Flag (DQ7)

Due to restrictions on the function in this series, the data polling flag(DQ7) outputs "1" for 50 μ s to 160 μ s after the sector erase command is issued, and then outputs "0". After the sector erase is terminated, the flash memory outputs "1".

For restrictions on the data polling flag (DQ7) and how to avoid erroneous judgment of sector erase completion, see section "1.7 Restrictions on Data Polling Flag (DQ7) and How to Avoid Erroneous Judgment".

Figure 21.5-2 Example of Sector Erasing Procedure



21.5.5 Suspending Sector Erasure

This section describes the procedure for issuing the Sector Erase Suspend command to suspend erasing one or more sectors. The command allows data to be read from sectors currently not being erased.

■ Suspending Sector Erasure in Flash Memory

To suspend erasing sectors in flash memory, issue the Sector Erase Suspend command in the table in Table 21.4-1 to flash memory.

The Sector Erase Suspend command suspends sector erasure in process, allowing data to be read from sectors currently not being erased. When sector erasure is being suspended, such sectors can only be read from; they cannot be written to. The command is valid only during the period of sector erasure including the sector erase time-out period; it is ignored during chip erasure or programming.

If the Sector Erase Suspend command is input during a sector erase time-out period, the flash memory terminates the time-out period immediately to halt erasure and enters the erase suspended state. If the command is input during sector erase operation after a sector erase time-out period, the flash memory enters the erase suspended state after a maximum of 20 μ s. Issue the Sector Erase Suspend command at least 20 μ s after issuing the Sector Erase command or Sector Erase Resume command.

21.5.6 Resuming Sector Erasure

This section describes the procedure for issuing the Sector Erase Resume command to resume suspended erasure of one or more sectors.

■ Resuming Sector Erasure in Flash Memory

To resume suspend sector erasure, issue the Sector Erase Resume command in Table 21.4-1 to flash memory.

The Sector Erase Resume command resumes sector erasure suspended by the Sector Erase Suspend command. The Sector Erase Resume command is executed by writing the erase resume code (3030_H) to any address in the flash memory area.

Note that the Sector Erase Resume command is ignored if issued during sector erasure in process.

21.5.7 Continuous Mode Operation

This section explains the procedure to write data continuously to the flash memory by issuing continuous mode commands.

■ Continuous mode

If the continuous mode commands are continuously written to the flash memory area, the state transits to the continuous mode state. It is possible to write data by the command sequence of two times instead of the command sequence of four times in the continuous mode. Moreover, data is read using the normal read access.

To end this mode, write the continuous mode reset command. Therefore, it is not possible to terminate the continuous mode even when the reset command (F0F0H) is written in this mode.

When the continuous mode reset command is written, the state returns to the read/reset state.

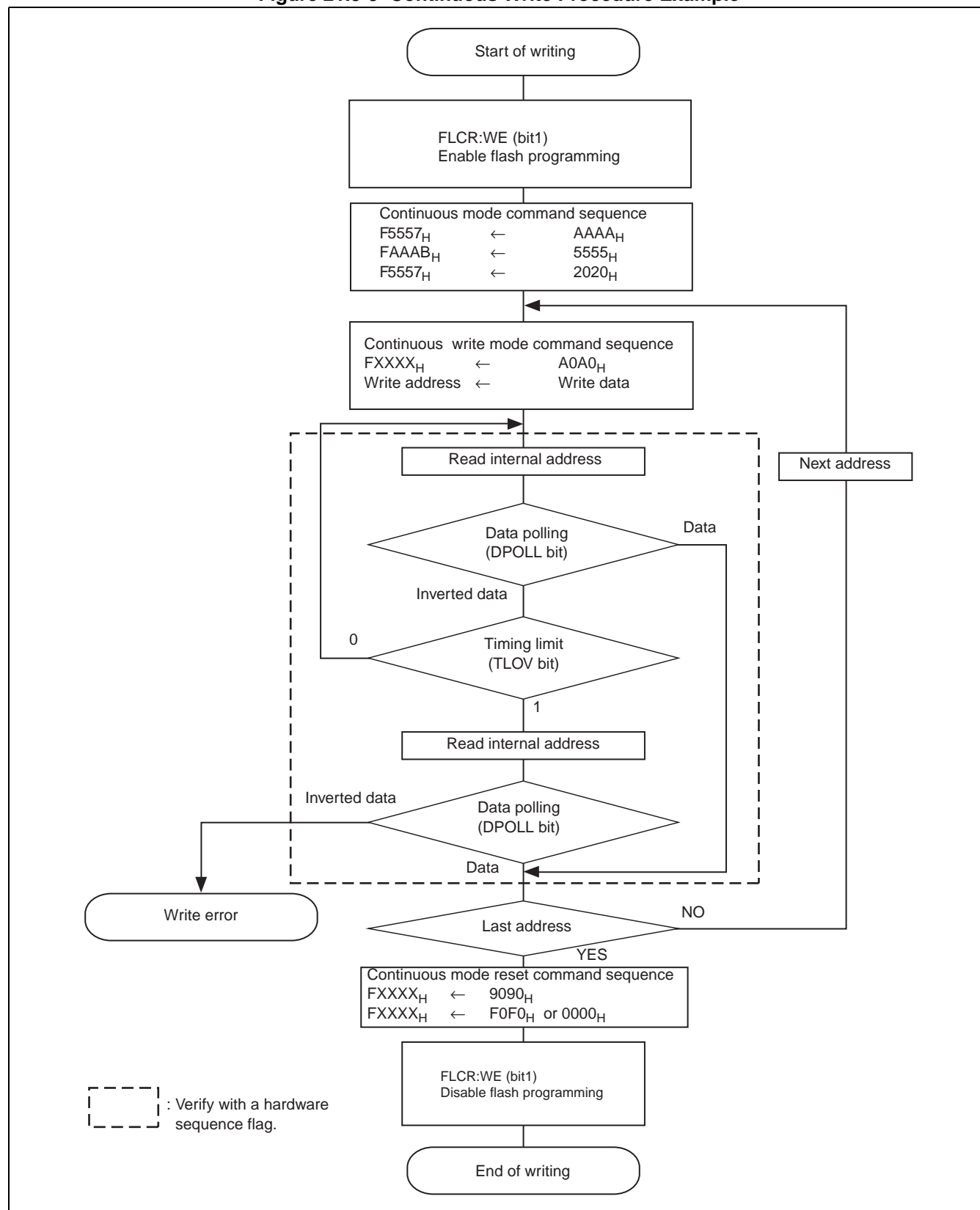
Note:

Do not write the commands other than the continuous write command/the continuous mode reset command during the continuous mode.

■ Continuous write mode

It is possible to write data by the command sequence of two times in the continuous mode. The automatic algorithm is started by writing the continuous write command in the continuous mode. This command has the same function as usual write operation except data writing by the command sequence of two times.

Figure 21.5-3 Continuous Write Procedure Example



21.6 Restrictions on Data Polling Flag (DQ7) and How to Avoid Problems

This series has some restrictions on how to use the data polling flag (DQ7) during execution of the automatic sector erase algorithm. This section describes such restrictions and how to avoid related problems.

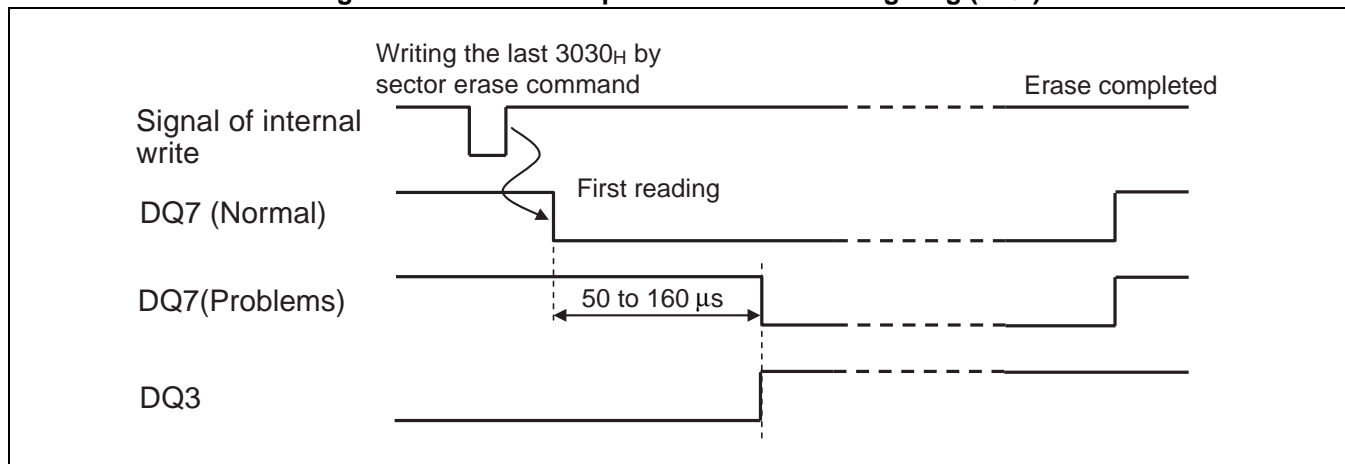
■ Description of Problems due to Restrictions

The data polling flag (DQ7) is used to indicate that the execution of the automatic algorithm is currently in progress or completed, by using the data polling function. In its original operation, as shown in Figure 21.6-1, DQ7 outputs "0" after the sector erase command is issued when the automatic algorithm is being started, and returns to "1" upon the completion of the erase operation. Therefore, the DQ7 polling algorithm indicates the completion of the erase operation by outputting "1".

In this series, DQ7 continues to output "1" for 50 to 160 μ s, after the Sector Erase command is issued, and then it outputs "0". When the erase operation is completed, it then returns to "1". For this reason, if the sector erase polling is started while "1" is still being output immediately after the sector erase command is issued, the erroneous judgment that the erase operation has been completed may occur, although the erase operation has not actually started.

The timing for DQ7 to change from "1" to "0" after the sector erase command is accepted is the same as the timing for the sector erase timer flag (DQ3), which indicates the sector erase timeout period, to change from "0" to "1".

Figure 21.6-1 Actual Operation of Data Polling Flag (DQ7)



The following or other problems may occur, as a result of the erroneous judgment that the erase operation has been completed,

- (1) Runaway or abnormal operation may occur, because the value of the sequence flag is read from the flash memory even when the CPU attempts to fetch instruction/data; therefore, the value of the program cannot be read properly.
- (2) If the next command is issued after the erroneous judgment that the sector erase operation has been completed occurs, the first command may be cancelled, resulting in a return to the read state, or the next command may not be accepted.

■ How to Avoid Problems

Use one of the following methods to avoid the problems.

- Polling using the toggle bit flag (DQ6)

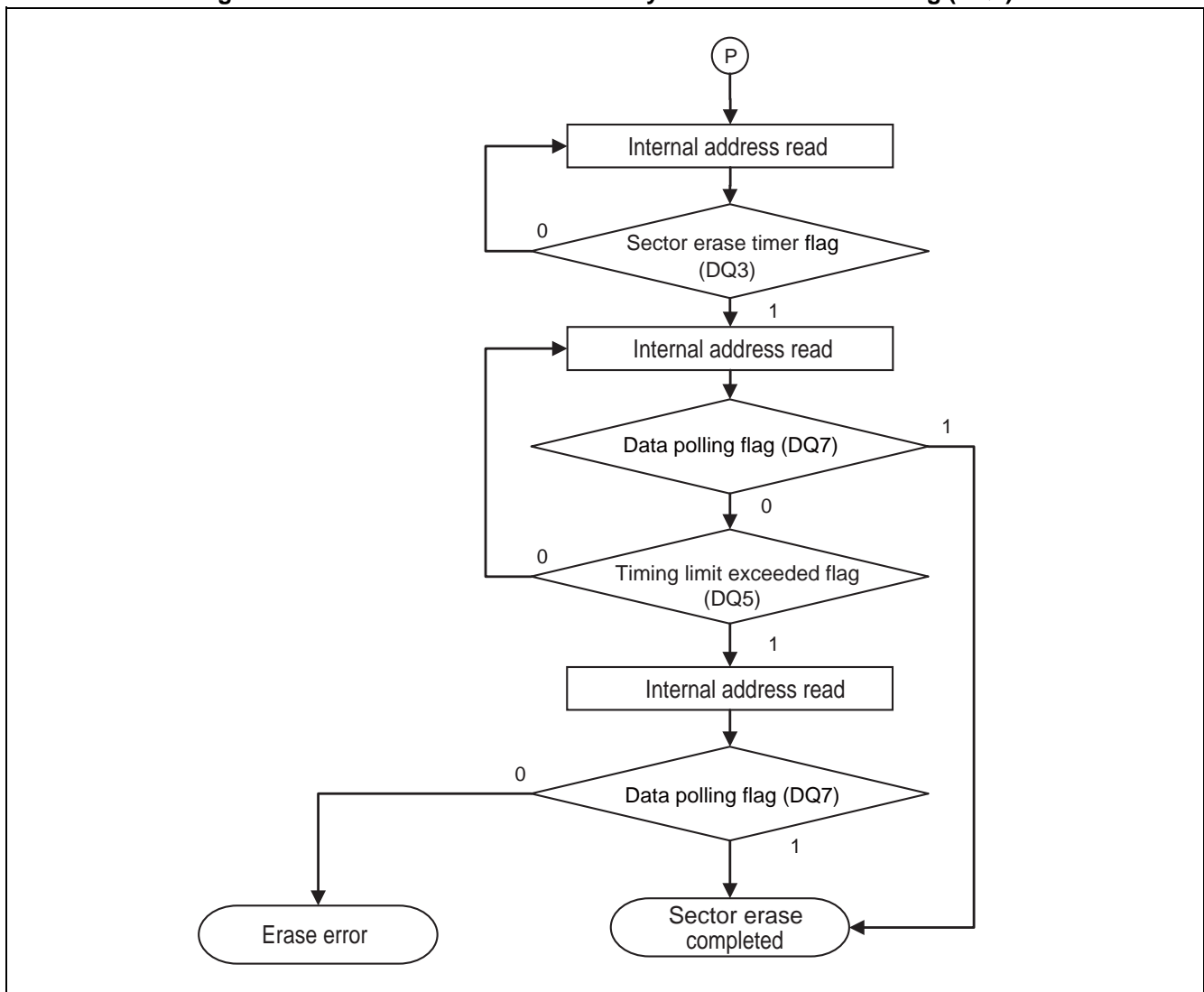
Determine the state of the automatic algorithm using DQ6, as shown in Figure 21.5-2 in "21.5.4 Erasing Data (Sector Erase)".

In the same manner as the data polling flag (DQ7), the toggle bit flag (DQ6) indicates that the automatic algorithm is being executed or has terminated by the toggle bit function.

- Starting polling of DQ7 after the sector erase timeout period elapses

Before starting the polling of DQ7, wait for 160μs or more by software after the sector erase command is issued, or wait until DQ3 is set to "1" (end of the sector erase timeout period). Figure 21.6-2 shows the judgment method using DQ3 after the sector erase command is issued.

Figure 21.6-2 How to Avoid Problems by Sector Erase Timer Flag (DQ3)

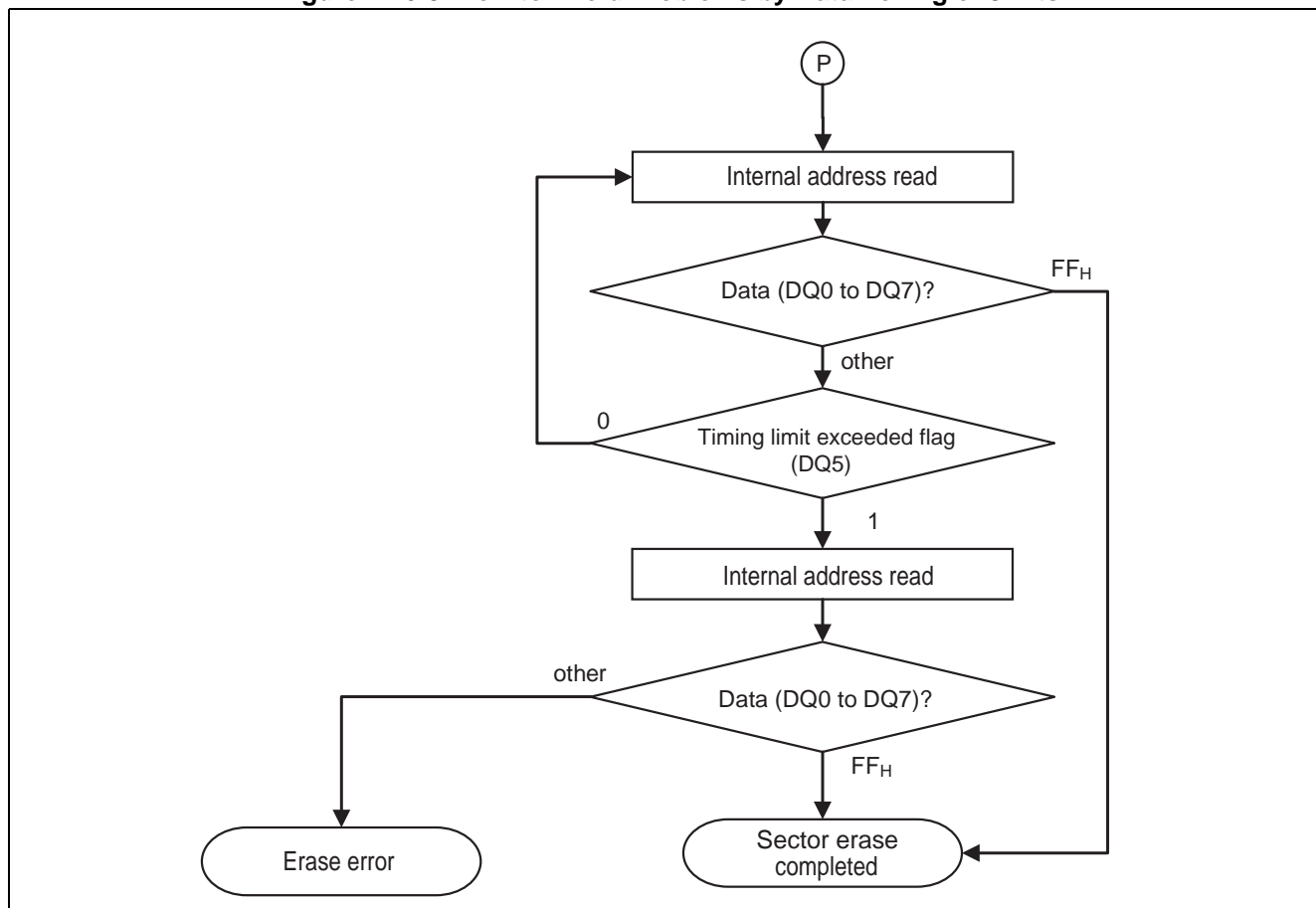


● Data polling using the 8 bits of hardware sequence flags

Make a judgment by data polling using the 8 bits of hardware sequence flags, rather than using only the polling of DQ7.

Figure 21.6-3 shows the judgment method using the data polling of the 8 bits after the sector erase command is issued.

Figure 21.6-3 How to Avoid Problems by Data Polling of 8 Bits



21.7 Notes on Flash Memory Programming

This section provides notes on programming into flash memory.

■ Notes on Flash Memory Programming

Take the following precautions when reprogramming flash memory using a program:

- If a reset occurs during programming of flash memory, the data being written is not guaranteed.
- In FR-CPU programming mode (WE = 1 in the FLCR register), do not run any program in flash memory. If an interrupt vector table resides in flash memory in that mode, do not generate an interrupt. Doing either causes the program to run out of control as it fails to fetch normal values from flash memory.
- To check whether programming into flash memory has been completed, reference the toggle bit flag (TOGGLE, DQ6) as well as the RDY flag.
If flash memory is defective, the RDY flag that indicates the completion of the automatic algorithm for programming is not set. If referencing only the RDY flag, therefore, the program will enter an infinite loop.
- In FR-CPU programming mode (WE = 1 in the FLCR register), do not enter any low-power consumption mode.
- Do not write-access the flash memory with WE = 0 in the FLCR register.
- Do not write-access the flash memory continuously with WE = 1 in the FLCR register. In that case, be sure to insert at least two "NOP" instructions.

[Example] Write commands (a command sequence) to flash memory. => Read flash memory.

```
ldi    #0xAAAA,    r0
ldi    #0x5555,     r1
ldi    #0xF5557,    r6
ldi    #0xFAAAB,    r7
ldi    #0xA0A0,     r8
ldi    # PA,        r2
ldi    # PD,        r3

sth    r0,@r6
NOP                                     // Be sure to insert at least two NOP instructions.
NOP                                     // Be sure to insert at least two NOP instructions.
sth    r1,@r7
NOP                                     // Be sure to insert at least two NOP instructions.
NOP                                     // Be sure to insert at least two NOP instructions.
sth    r8,@r6
NOP                                     // Be sure to insert at least two NOP instructions.
NOP                                     // Be sure to insert at least two NOP instructions.
sth    r3,@r2
```

NOP // Be sure to insert at least two NOP instructions.

NOP // Be sure to insert at least two NOP instructions.

- In CPU mode, write access to flash memory is allowed only in halfwords.
Do not write-access the flash memory in bytes.
- The value read immediately after writing to flash memory cannot be guaranteed. Before reading data after writing, be sure to insert a dummy read as follows:

sth r0,@r1 // Write to flash memory.

lduh @r2,r4 // Dummy read

lduh @r3,r4 // Read polling data.

CHAPTER 22

SERIAL PROGRAMMING CONNECTION

This chapter explains the basic configuration for serial programming, pins used for serial on-board programming, the connection example for serial programming, and system configuration for flash microcontroller programmer.

22.1 Overview of the Serial Programming Connection

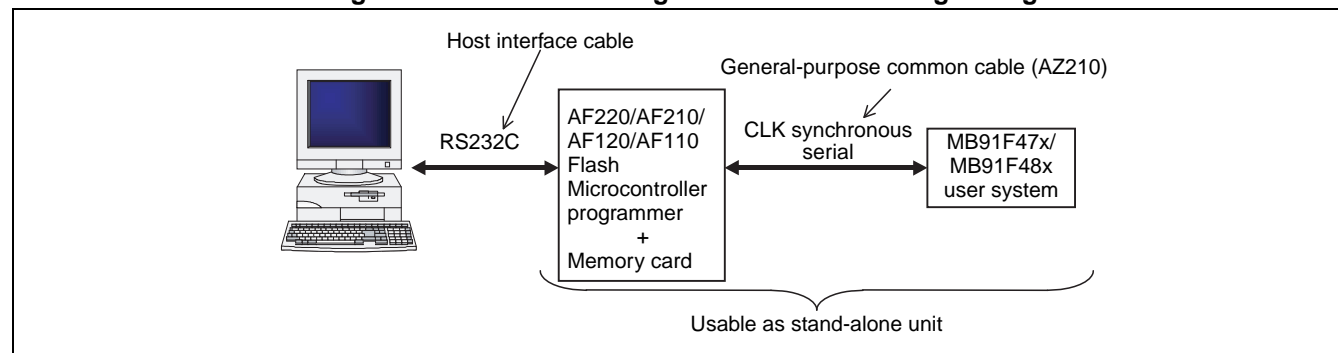
22.1 Overview of the Serial Programming Connection

The MB91F47x/MB91F48x supports serial on-board programming (Fujitsu standard) into flash ROM. This section summarizes its specifications.

■ Basic Configuration for Serial Programming

Fujitsu-standard serial on-board programming uses the AF220/AF210/AD120/AF110 flash microcontroller programmer manufactured by Yokogawa Digital Computer Corporation. It is possible to write it in the program that operates by the single-chip mode. Either of the operating program can be written by selecting it in the single-chip mode or internal ROM external bus mode. Figure 22.1-2 illustrates the basic configuration for MB91F47x/MB91F48x serial programming connection.

Figure 22.1-1 Basic Configuration for Serial Programming



Note: For the functions and usage of the AF210 flash microcontroller programmer or its general-purpose common connection cable (AZ210) and connector, contact Yokogawa Digital Computer Corporation.

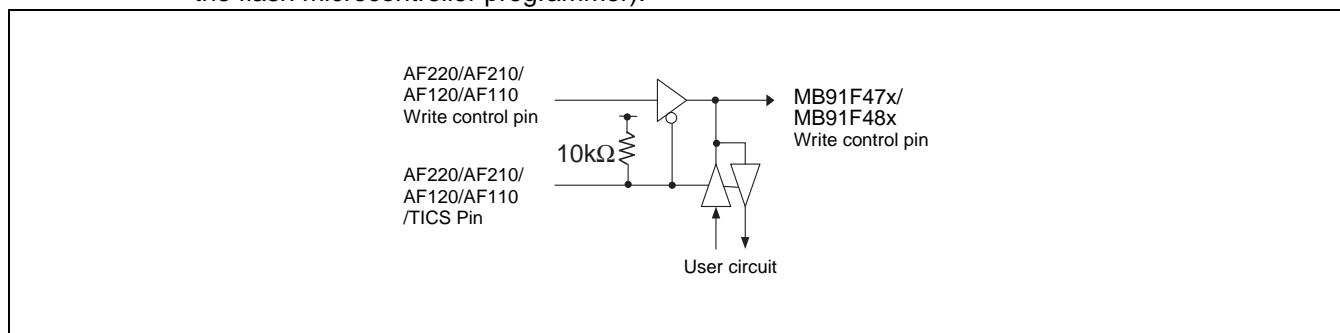
■ Pins Used for Fujitsu-standard Serial On-board Programming

Table 22.1-1 Pins Used for Fujitsu-standard Serial On-board Programming

Pin	Function	Supplementary Information
MD2,MD1,MD0	Mode Pins	Controlled for program mode. Flash serial program mode: MD2,MD1,MD0=1,0,0 Reference: Single-chip mode: MD2,MD1,MD0=0,0,0
P80, P81	Write program startup pins	Input "L" level to P80, and input "H" level to P81. Reference: Asynchronous UART mode in case of P80="L" and P81="L"
INITX	Reset pin	-
SIN0	Serial data input pin	Uses the UART ch.0 resource as clock synchronous mode.
SOT0	Serial data output pin	
SCK0	Serial clock input pin	
VCC	Power voltage supply pin	Supply a program voltage from the user system. Do not connect to the power supply of the user side when connecting.
VSS	GND pin	GND pin is common to GND of the flash microcontroller programmer.

Notes:

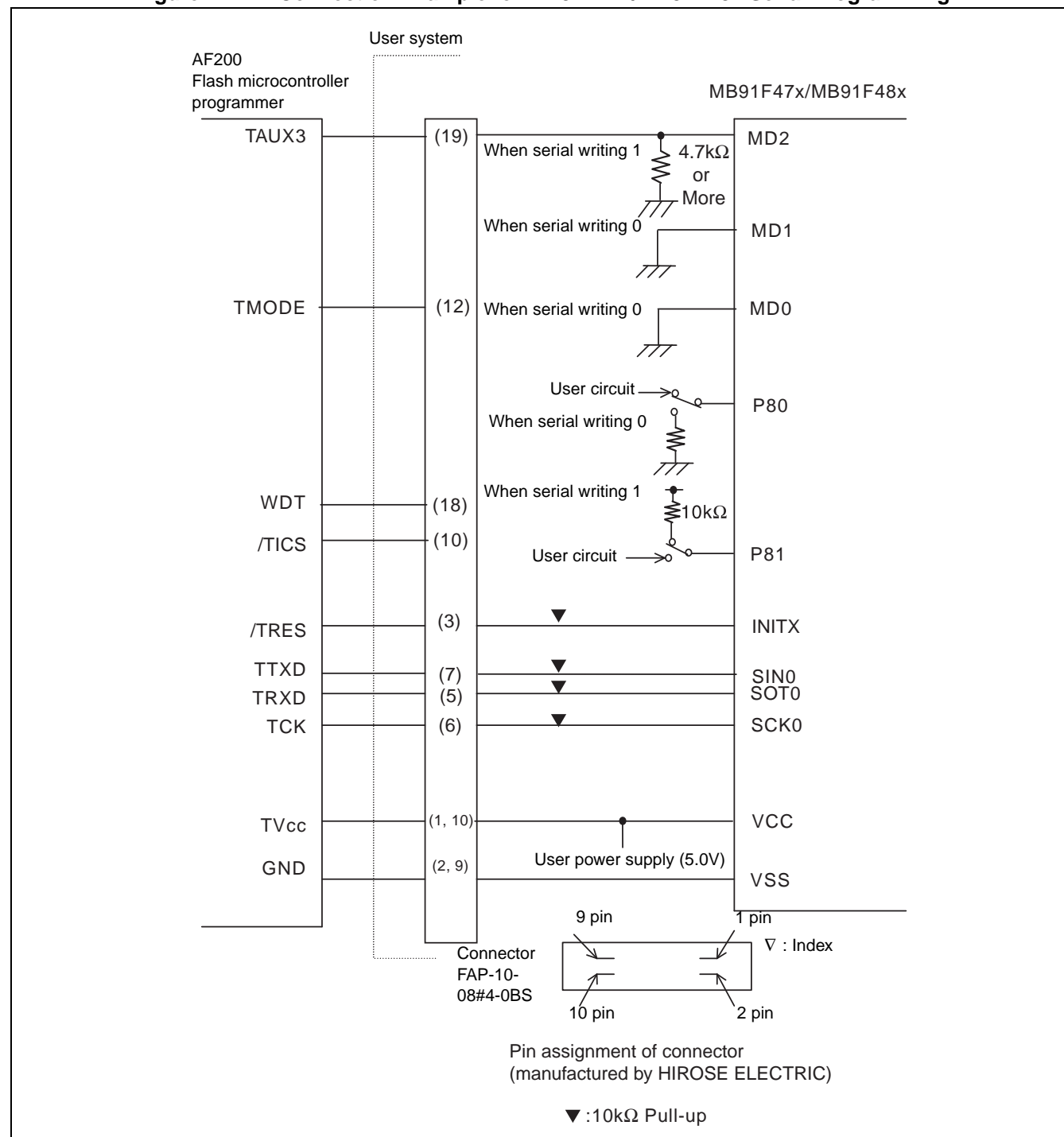
- The control circuit shown below is required for the user system to use the P80, P81, SIN0, SOT0, and SCK0 pins (During serial programming, the user circuit can be isolated by the /TICS signal of the flash microcontroller programmer).



- The AF200 must be connected with the user power supply off.

■ **Connection Example for MB91F47x/MB91F48x Serial Programming**

Figure 22.1-2 Connection Example for MB91F47x/MB91F48x Serial Programming



Note:

The values of the pull-up/pull-down resistors shown in the connection examples are informational. Appropriate values for your actual system may vary.

MB91470/480 Series

■ System Configuration for AF200 Flash Microcontroller Programmer (Manufactured by Yokogawa Digital Computer Corporation)

Model		Function	
Unit	AF220 /AC4P	Ethernet interface model	/100V to 220V power supply adapter
	AF210 /AC4P	Standard model	/100V to 220V power supply adapter
	AF120 /AC4P	Single key Ethernet interface model	/100V to 220V power supply adapter
	AF110 /AC4P	Single key model	/100V to 220V power supply adapter
AZ221		PC/AT RS232C cable for writer	
AZ210		Standard target probe (a) length: 1 m	
FF201		Control module for Fujitsu FR flash microcontroller	
AZ290		Remote controller	
		/P4	4MB PC Card (Option) Flash memory capacity is up to 512 KB.

Contact: Yokogawa Digital Computer Corporation:

Phone: + 81-42-333-6224

■ Oscillation Clock Frequency

The oscillation clock frequency available for programming into Flash memory is 10 MHz or 20 MHz.

■ Other Notes

The port state during programming into Flash memory using a serial programmer is the same as the reset state, except for the pin being used for programming.

CHAPTER 23

WILD REGISTER CONTROL BLOCK

This chapter describes the register configuration, functions and timer operations of the wild register control block.

- 23.1 Overview of Wild Register Control Block
- 23.2 Registers of Wild Register Control Block
- 23.3 Operations of Wild Register Control Block
- 23.4 Restrictions and Notes

23.1 Overview of Wild Register Control Block

The wild register function replaces the data of the patch target address set in an address register with the data set in a data register.

■ Overview of Wild Register Control Block

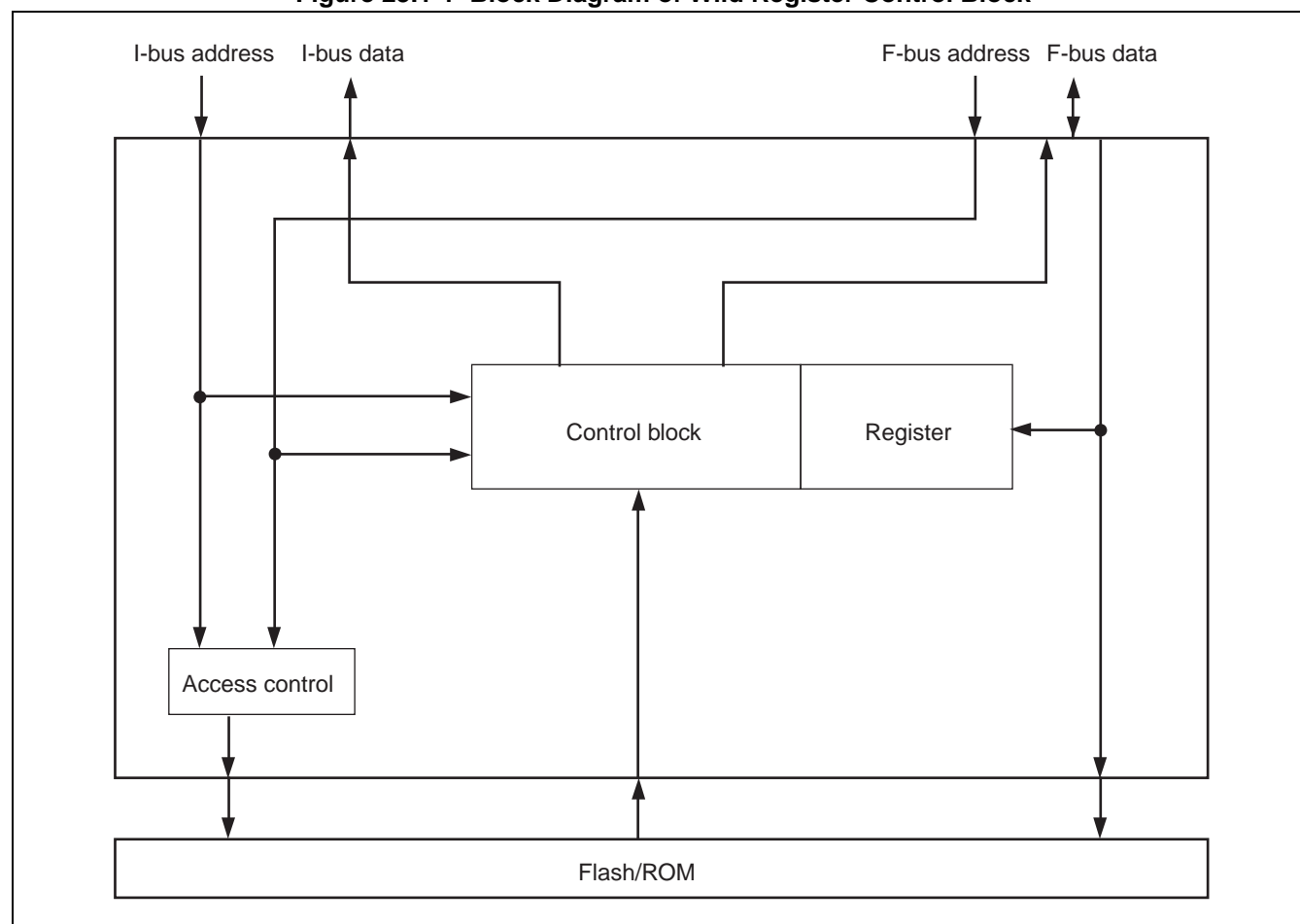
The wild register control block consists of a control register, 16 address setting registers and 16 data setting registers, a total of 33 registers.

The range of replaceable target address is limited within the internal Flash/ROM area only.

■ Block Diagram of Wild Register Control Block

Figure 23.1-1 shows a block diagram of the wild register control block.

Figure 23.1-1 Block Diagram of Wild Register Control Block



23.2 Registers of Wild Register Control Block

This section describes the configuration and functions of the registers used in the wild register control block.

■ List of Registers in Wild Register Control Block

Figure 23.2-1 List of Registers in Wild Register Control Block

Address	bit31	bit24 bit23	bit16 bit15	bit8 bit7	bit0
0000 7020 _H	WREN				
0000 7030 _H			WA00		
0000 7034 _H			WD00		
0000 7038 _H			WA01		
0000 703C _H			WD01		
0000 7040 _H			WA02		
0000 7044 _H			WD02		
0000 7048 _H			WA03		
0000 704C _H			WD03		
0000 7050 _H			WA04		
0000 7054 _H			WD04		
0000 7058 _H			WA05		
0000 705C _H			WD05		
0000 7060 _H			WA06		
0000 7064 _H			WD06		
0000 7068 _H			WA07		
0000 706C _H			WD07		
0000 7070 _H			WA08		
0000 7074 _H			WD08		
0000 7078 _H			WA09		
0000 707C _H			WD09		
0000 7080 _H			WA10		
0000 7084 _H			WD10		
0000 7088 _H			WA11		
0000 708C _H			WD11		
0000 7090 _H			WA12		
0000 7094 _H			WD12		
0000 7098 _H			WA13		
0000 709C _H			WD13		
0000 70A0 _H			WA14		
0000 70A4 _H			WD14		
0000 70A8 _H			WA15		
0000 70AC _H			WD15		

23.2.1 Wild Register Enable Register (WREN)

The wild register enable register (WREN) is a register that enables the replacement function corresponding to each channel (ch.0 to ch.15).

■ Bit Configuration of Wild Register Enable Register (WREN)

WREN (upper)									
Address	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	Initial value
007020 _H	WREN15	WREN14	WREN13	WREN12	WREN11	WREN10	WREN09	WREN08	00000000 _B
007021 _H									
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable									

WREN (lower)									
Address	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	Initial value
007020 _H	WREN07	WREN06	WREN05	WREN04	WREN03	WREN02	WREN01	WREN00	00000000 _B
007021 _H									
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable									

[bit31 to bit16] WREN15 to WREN00: Replacement function enable bits

These bits enable the replacement function corresponding to each channel.

0: Wild register function is disabled. [Initial value]

1: Wild register function is enabled.

Note:

Be sure to use halfword access when accessing to this register.

23.2.2 Wild Register Address Register (WA)

The wild register address register (WA) is a register that sets the replacement target address.

■ Bit Configuration of Wild Register Address Register (WA)

WA

Address

ch.0 007030_H

ch.1 007038_H

ch.2 007040_H

ch.3 007048_H

ch.4 007050_H

ch.5 007058_H

ch.6 007060_H

ch.7 007068_H

ch.8 007070_H

ch.9 007078_H

ch.10 007080_H

ch.11 007088_H

ch.12 007090_H

ch.13 007098_H

ch.14 0070A0_H

ch.15 0070A8_H

bit31 to bit21

bit20 to bit2

bit1 to bit0

-

A20 to A2

-

-

R/W

-

Initial value: ----- ----XXXX XXXXXXXX XXXXXXX--_B

R/W: Readable/writable

[bit31 to bit21] Reserved: Reserved bits

These bits are reserved. Be sure to set them to "0".

[bit20 to bit2] A20 to A2: Replacement target address setting bits

Set a replacement target address.

[bit1, bit0] Reserved: Reserved bits

These bits are reserved. Be sure to set them to "0".

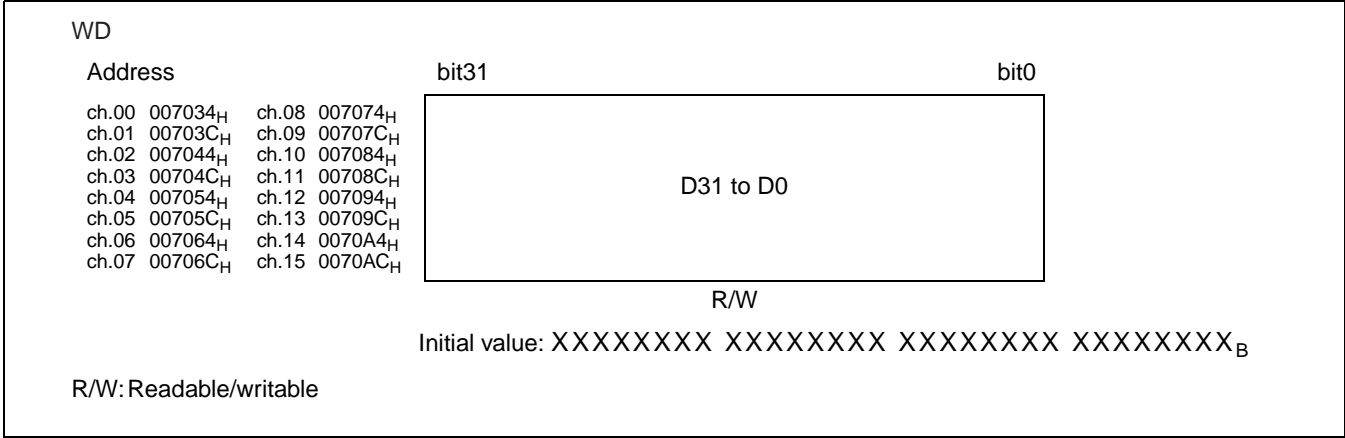
Notes:

- Be sure to use word access when accessing to this register.
- Replacement target address is limited within the internal Flash/ROM area only. Therefore, set an address located within the internal Flash/ROM area for A20 to A2.

23.2.3 Wild Register Data Register (WD)

The wild register data register (WD) is a register that sets the replacement data.

■ Bit Configuration of Wild Register Data Register (WD)



[bit31 to bit0] D31 to D0: Replacement data setting bits

Set replacement data.

Note:

Be sure to use word access when accessing to this register.

23.3 Operations of Wild Register Control Block

This section describes the operations of the wild register control block.

■ Operations of Wild Register Control Block

The wild register function replaces data with any data set in WD_x. The replacement data/register corresponding to each channel is shown in the table Table 23.3-1.

Table 23.3-1 Replacement Data/Register

Channel	Replacement data/register			
	Address +0	Address +1	Address +2	Address +3
ch.x (x=00 to 15)	WD _x (D31-D24)	WD _x (D23-D16)	WD _x (D15-D8)	WD _x (D7-D0)

23.4 Restrictions and Notes

This section summarizes the restrictions and notes on the wild register control block.

■ Wild Register Enable Register (WREN)

Be sure to use halfword access when accessing to this register.

■ Wild Register Address Register (WA)

- Be sure to use word access when accessing to this register.
- Replacement target address is limited within the internal Flash/ROM area only. Therefore, set an address located within the internal Flash/ROM area for A20 to A2.

■ Wild Register Data Register (WD)

Be sure to use word access when accessing to this register.

■ Overall Restrictions and Notes

- The wild register function prioritizes the specification of register with the lower number (Example: ch.0 > ch.1).
- Place the program that sets the wild register in an area other than the internal Flash/ROM area.
- Do not enable the wild register function while executing Flash memory automatic algorithm (or when RDY (bit3 = 0) in flash control/status register: FLCR).
- If replacement data is set for an address at which an instruction longer than 16-bit length (a 32/48-bit long instruction) is located, CPU may fail to interpret the instruction correctly, causing malfunctions. Therefore, do not set replacement data in the middle of the instruction when setting it for an address at which a 32/48-bit long instruction is located.

APPENDIX

The appendix describes pin states in each CPU state, notes on using the little-endian areas, a list of FR family instructions, and notes on using MB91470/480 series.

APPENDIX A I/O Map

APPENDIX B Interrupt Vector

APPENDIX C Pin States in Each CPU State

APPENDIX D Notes when Little Endian Region is used

APPENDIX E Instruction List

APPENDIX F Precautions when Using

APPENDIX A I/O Map

This section shows the correspondence between the various peripheral resource registers and the memory space area.

I/O Map

Figure A-1 View in Table

Address	Register				Block
	+0	+1	+2	+3	
000000 _H	PDR0 [R/W]B XXXXXXXX	PDR1 [R/W]B XXXXXXXX	PDR2 [R/W]B XXXXXXXX	PDR3 [R/W]B XXXXXXXX	T-unit Port Data Register

Read/write attribute, unit of access
(B:Byte,H:Half-word,W:Word)

Initial register value after a reset

Register name (registers in column 1 are located at 4n addresses,
registers in column 2 are located at 4n + 2 addresses, and so on)

The leftmost register address (When word access is used, data from
the register in the first column becomes the MSB.)

Note :

The bit value of the register shows the initial value as follows.

" 1 " : Initial value " 1 "

" 0 " : Initial value " 0 "

" X " : Initial value " X "

- : There is physically no register in the position.

The access by the data access attribute not described is disabled.

Table A-1 I/O Map (1 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000000 _H	PDR0 [R/W] B,H,W XXXXXXXXXX	PDR1 [R/W] B,H,W XXXXXXXXXX	PDR2 [R/W] B,H,W XXXXXXXXXX	PDR3 [R/W] B,H,W XXXXXXXXXX	Port data register
000004 _H	PDR5 [R/W] B,H,W -XXXXXXXXX	PDR6 [R/W] B,H,W -----XX	PDR8 [R/W] B,H,W XXXXXXXXXX	PDR9 [R/W] B,H,W XXXXXXXXXX	
000008 _H	PDRA [R/W] B,H,W ---XXXXX	PDRB [R/W] B,H,W XXXXXXXXXX	PDRC [R/W] B,H,W XXXXXXXXXX	PDRD [R/W] B,H,W ----XXXX	
00000C _H	PDRE [R/W] B,H,W XXXXXXXXXX	PDRF [R/W] B,H,W XXXXXXXXXX	PDRG [R/W] B,H,W --XXXXXX	PDRH [R/W] B,H,W --XXXXXX	
000010 _H	PDRJ [R/W] B,H,W XXXXXXXXXX	—	PDRL [R/W] B,H,W ----XXX	PDRM [R/W] B,H,W ----XXXX	
000014 _H	PDRP [R/W] B,H,W --XXXXXX	PDRQ [R/W] B,H,W --XXXXXX	PDRR [R/W] B,H,W --XXXXXX	PDRS [R/W] B,H,W --XXXXXX	
000018 _H to 00003C _H	—				(Reserved)
000040 _H	EIRR0 [R/W] B,H,W 00000000	ENIR0 [R/W] B,H,W 00000000	ELVR0 [R/W] B,H,W 00000000 00000000		External Interrupt (INT0 to INT7)
000044 _H	DICR [R/W] B,H,W -----0	HRCL [R/W,R] B,H,W 0--11111	—		Delay Interrupt/ Hold request
000048 _H	TMRLR0 [W] H,W XXXXXXXXXX XXXXXXXXX		TMR0 [R] H,W XXXXXXXXXX XXXXXXXXX		Reload Timer 0
00004C _H	—		TMCSR0 [R/W,R] B,H,W ----00-- ---00000		
000050 _H	TMRLR1 [W] H,W XXXXXXXXXX XXXXXXXXX		TMR1 [R] H,W XXXXXXXXXX XXXXXXXXX		Reload Timer 1
000054 _H	—		TMCSR1 [R/W,R] B,H,W ----00-- ---00000		
000058 _H to 00005C _H	—				(Reserved)

Table A-1 I/O Map (2 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000060 _H	SSR0 [R/W, R] B,H,W 00000011	ESCR0 [R/W]/ IBSR0 [R/W, R] B,H,W 00000000	SCR0 [R/W]/ IBCR0 [R/W, R]B,H,W 00000000	SMR0 [R/W] B,H,W 000-0000	Multi-function serial interface0
000064 _H	BGR01 [R/W] B,H,W 00000000	BGR00 [R/W] B,H,W 00000000	RDR0 [R]/ TDR0 [W]H,W -----0 00000000		
000068 _H	—		ISMK0 [R/W] B,H,W 01111111	ISBA0 [R/W] B,H,W 00000000	
00006C _H	FBYTE02 [R/W]B,H,W 00000000	FBYTE01 [R/W]B,H,W 00000000	FCR01 [R/W] B,H,W ---00100	FCR00 [R/W, R] B,H,W -0000000	
000070 _H	SSR1 [R/W, R] B,H,W 00000011	ESCR1 [R/W]/ IBSR1 [R/W, R]B,H,W 00000000	SCR1 [R/W]/ IBCR1 [R/W, R]B,H,W 00000000	SMR1 [R/W] B,H,W 000-0000	Multi-function serial interface1
000074 _H	BGR11 [R/W] B,H,W 00000000	BGR10 [R/W] B,H,W 00000000	RDR1 [R]/ TDR1 [W]H,W -----0 00000000		
000078 _H	—		ISMK1 [R/W] B,H,W 01111111	ISBA1 [R/W] B,H,W 00000000	
00007C _H	FBYTE12 [R/W] B,H,W 00000000	FBYTE11 [R/W] B,H,W 00000000	FCR11 [R/W] B,H,W ---00100	FCR10 [R/W, R] B,H,W -0000000	
000080 _H	SSR2 [R/W, R] B,H,W 00000011	ESCR2 [R/W]/ IBSR2 [R/W, R] B,H,W 00000000	SCR2 [R/W]/ IBCR2 [R/W, R]B,H,W 00000000	SMR2 [R/W, R] B,H,W 000-0000	Multi-function serial interface2
000084 _H	BGR21 [R/W] B,H,W 00000000	BGR20 [R/W] B,H,W 00000000	RDR2 [R]/ TDR2 [W]H,W -----0 00000000		
000088 _H	—		ISMK2 [R/W] B,H,W 01111111	ISBA2 [R/W] B,H,W 00000000	
00008C _H	FBYTE22 [R/W] B,H,W 00000000	FBYTE21 [R/W] B,H,W 00000000	FCR21 [R/W] B,H,W ---00100	FCR20 [R/W, R] B,H,W -0000000	
000090 _H	SSR3 [R/W, R] B,H,W 00000011	ESCR3 [R/W]/ IBSR3 [R/W, R] B,H,W 00000000	SCR3 [R/W]/ IBCR3 [R/W, R]B,H,W 00000000	SMR3 [R/W] B,H,W 000-0000	Multi-function serial interface3
000094 _H	BGR31 [R/W] B,H,W 00000000	BGR30 [R/W] B,H,W 00000000	RDR3 [R]/ TDR3 [W]H,W -----0 00000000		
000098 _H	—		ISMK3 [R/W] B,H,W 01111111	ISBA3 [R/W] B,H,W 00000000	
00009C _H	FBYTE32 [R/W] B,H,W 00000000	FBYTE31 [R/W] B,H,W 00000000	FCR31 [R/W] B,H,W ---00100	FCR30 [R/W, R] B,H,W -0000000	

Table A-1 I/O Map (3 / 19)

Address	Register				Block
	+0	+1	+2	+3	
0000A0 _H	OCCPBH0, OCCPBL0 [W]/ OCCPH0, OCCPL0 [R]H,W 00000000 00000000		OCCPBH1, OCCPBL1 [W]/ OCCPH1, OCCPL1 [R]H,W 00000000 00000000		OCU0
0000A4 _H	OCCPBH2, OCCPBL2 [W]/ OCCPH2, OCCPL2 [R]H,W 00000000 00000000		OCCPBH3, OCCPBL3 [W]/ OCCPH3, OCCPL3 [R]H,W 00000000 00000000		
0000A8 _H	OCCPBH4, OCCPBL4 [W]/ OCCPH4, OCCPL4 [R]H,W 00000000 00000000		OCCPBH5, OCCPBL5 [W]/ OCCPH5, OCCPL5 [R]H,W 00000000 00000000		
0000AC _H	OCSH1 [R/W] B,H,W -110--00	OCSL0 [R/W] B,H,W 00001100	OCSH3 [R/W] B,H,W -110--00	OCSL2 [R/W] B,H,W 00001100	
0000B0 _H	OCSH5 [R/W] B,H,W -110--00	OCSL4 [R/W] B,H,W 00001100	OCMOD0 [R/W] B,H,W --000000	—	
0000B4 _H	CPCLRBH0, CPCLRBL0 [W] / CPCLRH0, CPCLRL0 [R] H,W 11111111 11111111		TCDTH0,TCDTL0 [R/W] H,W 00000000 00000000		Free-run Timer 0
0000B8 _H	TCCSH0 [R/W] B,H,W 00000000	TCCSL0 [R/W] B,H,W 01000000	TCCSM0 [R/W] B,H,W ----0000	ADTRGC0 [R/W] B,H,W -000-000	
0000BC _H	CPCLRBH1,CPCLRBL1 [W] / CPCLRH1, CPCLRL1 [R] H,W 11111111 11111111		TCDTH1,TCDTL1 [R/W] H,W 00000000 00000000		Free-run Timer 1
0000C0 _H	TCCSH1 [R/W]B,H,W 00000000	TCCSL1 [R/W] B,H,W 01000000	TCCSM1 [R/W] B,H,W ----0000	ADTRGC1 [R/W] B,H,W -000-000	
0000C4 _H	CPCLRBH2,CPCLRBL2 [W] / CPCLRH2, CPCLRL2 [R] H,W 11111111 11111111		TCDTH2,TCDTL2 [R/W] H,W 00000000 00000000		Free-run Timer 2
0000C8 _H	TCCSH2 [R/W]B,H,W 00000000	TCCSL2 [R/W] B,H,W 01000000	TCCSM2 [R/W] B,H,W ----0000	ADTRGC2 [R/W] B,H,W -000-000	
0000CC _H	—	FRS2 [R/W] B,H,W -000-000	FRS1 [R/W] B,H,W -000-000	FRS0 [R/W] B,H,W -000-000	Free-run Timer Selector 0
0000D0 _H	—	—	FRS4 [R/W] B,H,W -000-000	FRS3 [R/W] B,H,W -000-000	
0000D4 _H	IPCPH0, IPCPL0 [R] H,W XXXXXXXXXX XXXXXXXXXX		IPCPH1, IPCPL1 [R] H,W XXXXXXXXXX XXXXXXXXXX		ICU0
0000D8 _H	IPCPH2, IPCPL2 [R] H,W XXXXXXXXXX XXXXXXXXXX		IPCPH3, IPCPL3 [R] H,W XXXXXXXXXX XXXXXXXXXX		
0000DC _H	PICSH01 [W,R] B,H,W 00000000	PICSL01 [R/W] B,H,W 00000000	ICSH23 [R] B,H,W -----00	ICSL23[R/W] B,H,W 00000000	

Table A-1 I/O Map (4 / 19)

Address	Register				Block
	+0	+1	+2	+3	
0000E0 _H	TMRRH0, TMRRL0 [R/W] H,W XXXXXXXXXX XXXXXXXXXX		TMRRH1, TMRRL1 [R/W] H,W XXXXXXXXXX XXXXXXXXXX		Waveform Generator 0
0000E4 _H	TMRRH2, TMRRL2 [R/W] H,W XXXXXXXXXX XXXXXXXXXX		—		
0000E8 _H	DTCR0 [R/W] B,H,W 00000000	DTCR1 [R/W] B,H,W 00000000	DTCR2 [R/W] B,H,W 00000000	—	
0000EC _H	—	SIGCR10 [R/W] B,H,W 00000000	—	SIGCR20 [R/W] B,H,W 000000-1	
0000F0 _H	ADCOMP0 [W]/ ADCOMPB0 [R] H,W 00000000 00000000		ADCOMP0 [W]/ ADCOMPDB0 [R] H,W 00000000 00000000		A/D activation compare 0
0000F4 _H	ADCOMP1 [W] / ADCOMPB1 [R] H,W 00000000 00000000		ADCOMP1 [W] / ADCOMPDB1 [R] H,W 00000000 00000000		
0000F8 _H	ADCOMP2 [W] / ADCOMPB2 [R] H,W 00000000 00000000		ADCOMP2 [W]/ ADCOMPDB2 [R] H,W 00000000 00000000		
0000FC _H	—	ADTGBUF0 [R/W] B,H,W -000-111	ADTGSEL0 [R/W] B,H,W --000000	ADTGCE0 [R/W] B,H,W --000000	
000100 _H	PRLH0 [R/W] B,H,W XXXXXXXXXX	PRLL0 [R/W] B,H,W XXXXXXXXXX	PRLH1 [R/W] B,H,W XXXXXXXXXX	PRLL1 [R/W] B,H,W XXXXXXXXXX	PPG
000104 _H	PRLH2 [R/W] B,H,W XXXXXXXXXX	PRLL2 [R/W] B,H,W XXXXXXXXXX	PRLH3 [R/W] B,H,W XXXXXXXXXX	PRLL3 [R/W] B,H,W XXXXXXXXXX	
000108 _H	PPGC0 [R/W] B,H,W 00000000	PPGC1 [R/W] B,H,W 00000000	PPGC2 [R/W] B,H,W 00000000	PPGC3 [R/W] B,H,W 00000000	
00010C _H	PRLH4 [R/W] B,H,W XXXXXXXXXX	PRLL4 [R/W] B,H,W XXXXXXXXXX	PRLH5 [R/W] B,H,W XXXXXXXXXX	PRLL5 [R/W] B,H,W XXXXXXXXXX	
000110 _H	PRLH6 [R/W] B,H,W XXXXXXXXXX	PRLL6 [R/W] B,H,W XXXXXXXXXX	PRLH7 [R/W] B,H,W XXXXXXXXXX	PRLL7 [R/W] B,H,W XXXXXXXXXX	
000114 _H	PPGC4 [R/W] B,H,W 00000000	PPGC5 [R/W] B,H,W 00000000	PPGC6 [R/W] B,H,W 00000000	PPGC7 [R/W] B,H,W 00000000	
000118 _H	PRLH8 [R/W] B,H,W XXXXXXXXXX	PRLL8 [R/W] B,H,W XXXXXXXXXX	PRLH9 [R/W] B,H,W XXXXXXXXXX	PRLL9 [R/W] B,H,W XXXXXXXXXX	
00011C _H	PRLH10 [R/W] B,H,W XXXXXXXXXX	PRLL10 [R/W] B,H,W XXXXXXXXXX	PRLH11 [R/W] B,H,W XXXXXXXXXX	PRLL11 [R/W] B,H,W XXXXXXXXXX	
000120 _H	PPGC8 [R/W] B,H,W 00000000	PPGC9 [R/W] B,H,W 00000000	PPGC10 [R/W] B,H,W 00000000	PPGC11 [R/W] B,H,W 00000000	
000124 _H	PRLH12 [R/W] B,H,W XXXXXXXXXX	PRLL12 [R/W] B,H,W XXXXXXXXXX	PRLH13 [R/W] B,H,W XXXXXXXXXX	PRLL13 [R/W] B,H,W XXXXXXXXXX	

Table A-1 I/O Map (5 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000128 _H	PRLH14 [R/W] B,H,W XXXXXXXXXX	PRLL14 [R/W] B,H,W XXXXXXXXXX	PRLH15 [R/W] B,H,W XXXXXXXXXX	PRLL15 [R/W] B,H,W XXXXXXXXXX	PPG
00012C _H	PPGC12 [R/W] B,H,W 00000000	PPGC13 [R/W] B,H,W 00000000	PPGC14 [R/W] B,H,W 00000000	PPGC15 [R/W] B,H,W 00000000	
000130 _H	TRG [R/W] B,H 00000000 00000000		—	GATEC0 [R/W] B --00--00	
000134 _H	REVC [R/W] B,H 00000000 00000000		—	GATEC4 [R/W] B -----00	
000138 _H	—		—	GATEC8 [R/W] B --00--00	
00013C _H	—		—	GATEC12 [R/W] B -----00	
000140 _H	—				(Reserved)
000144 _H	TTCR0 [R/W,W,R] B,H,W 11110000	—	—	—	Timing Generator 0
000148 _H	COMP0 [R/W] B,H,W 00000000	COMP2 [R/W] B,H,W 00000000	COMP4 [R/W] B,H,W 00000000	COMP6 [R/W] B,H,W 00000000	
00014C _H	TTCR1 [R/W,W,R] B,H,W 11110000	—	—	—	Timing Generator 1
000150 _H	COMP1 [R/W] B,H,W 00000000	COMP3 [R/W] B,H,W 00000000	COMP5 [R/W] B,H,W 00000000	COMP7 [R/W] B,H,W 00000000	
000154 _H	EIRR1 [R/W] B,H,W 00000000	ENIR1 [R/W] B,H,W 00000000	ELVR1 [R/W] B,H,W 00000000 00000000		External Interrupt (INT8 to INT15)
000158 _H	—				(Reserved)
00015C _H	—	—	—	CMCLKR [R/W] B ----0000	Clock Monitor
000160 _H	BT0TMR [R] B,H,W 00000000 00000000		BT0TMCR [R/W] B,H,W -00000000 00000000		Base Timer 0
000164 _H	—	BT0STC [R/W] B 00000000	—		
000168 _H	BT0PCSR/BT0PRL [R/W] H,W XXXXXXXXXX XXXXXXXXXX		BT0PDUT/BT0PRLH/BT0DTBF [R/W] B,H,W XXXXXXXXXX XXXXXXXXXX		
00016C _H	—				(Reserved)

Table A-1 I/O Map (6 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000170 _H	AICR2 [R/W] B,H,W ----1111 11111111		—		8/10-bit A/D converter 2 (with 12 channels)
000174 _H	ADCS2 [R/WW] B,H,W 0000000-	—	ADCH2 [R/W] B,H,W 0000000	ADMD2 [R/W] B,H,W 00001111	
000178 _H	ADCD002 [R] B,H,W 10---XX XXXXXXXXX		ADCD012 [R] B,H,W 10---XX XXXXXXXXX		
00017C _H	ADCD022 [R] B,H,W 10---XX XXXXXXXXX		ADCD032 [R] B,H,W 10---XX XXXXXXXXX		
000180 _H	ADCD042 [R] B,H,W 10---XX XXXXXXXXX		ADCD052 [R] B,H,W 10---XX XXXXXXXXX		
000184 _H	ADCD062 [R] B,H,W 10---XX XXXXXXXXX		ADCD072 [R] B,H,W 10---XX XXXXXXXXX		
000188 _H	ADCD082 [R] B,H,W 10---XX XXXXXXXXX		ADCD092 [R] B,H,W 10---XX XXXXXXXXX		
00018C _H	ADCD102 [R] B,H,W 10---XX XXXXXXXXX		ADCD112 [R] B,H,W 10---XX XXXXXXXXX		
000190 _H to 00019C _H	—				(Reserved)
0001A0 _H	OCCPBH6, OCCPBL6 [W]/ OCCPH6, OCCPL6 [R] H,W 00000000 00000000		OCCPBH7, OCCPBL7 [W]/ OCCPH7, OCCPL7 [R]H,W 00000000 00000000		OCU1
0001A4 _H	OCCPBH8, OCCPBL8 [W]/ OCCPH8, OCCPL8 [R] H,W 00000000 00000000		OCCPBH9, OCCPBL9 [W]/ OCCPH9, OCCPL9 [R] H,W 00000000 00000000		
0001A8 _H	OCCPBH10, OCCPBL10 [W]/ OCCPH10, OCCPL10 [R] H,W 00000000 00000000		OCCPBH11, OCCPBL11 [W]/ OCCPH11, OCCPL11 [R] H,W 00000000 00000000		
0001AC _H	OCSH7 [R/W] B,H,W -110--00	OCSL6 [R/W] B,H,W 00001100	OCSH9 [R/W] B,H,W -110--00	OCSL8 [R/W] B,H,W 00001100	
0001B0 _H	OCSH11 [R/W] B,H,W -110--00	OCSL10 [R/W] B,H,W 00001100	OCMOD1 [R/W] B,H,W --000000	—	
0001B4 _H	CPCLRBL3, CPCLRBL3 [W]/ CPCLRH3, CPCLRL3 [R] H,W 11111111 11111111		TCDTH3,TCDTL3 [R/W] H,W 00000000 00000000		Free-run Timer 3
0001B8 _H	TCCSH3 [R/W] B,H,W 00000000	TCCSL3 [R/W] B,H,W 01000000	TCCSM3 [R/W] B,H,W ----0000	ADTRGC3 [R/W] B,H,W -000-000	

Table A-1 I/O Map (7 / 19)

Address	Register				Block
	+0	+1	+2	+3	
0001BC _H	CPCLRBH4,CPCLRBL4 [W] / CPCLRH4, CPCLRL4 [R] H,W 11111111 11111111		TCDTH4,TCDTL4 [R/W] H,W 00000000 00000000		Free-run Timer 4
0001C0 _H	TCCSH4 [R/W] B,H,W 00000000	TCCSL4 [R/W] B,H,W 01000000	TCCSM4 [R/W] B,H,W ----0000	ADTRGC4 [R/W]B,H,W -000-000	
0001C4 _H	CPCLRBH5,CPCLRBL5 [W] / CPCLRH5, CPCLRL 5 [R] H,W 11111111 11111111		TCDTH5,TCDTL5 [R/W] H,W 00000000 00000000		Free-run Timer 5
0001C8 _H	TCCSH5 [R/W] B,H,W 00000000	TCCSL5 [R/W] B,H,W 01000000	TCCSM5 [R/W] B,H,W ----0000	ADTRGC5 [R/W] B,H,W -000-000	
0001CC _H	—	FRS7 [R/W] B,H,W -011-011	FRS6 [R/W] B,H,W -011-011	FRS5 [R/W] B,H,W -011-011	Free-run Timer Selector 1
0001D0 _H	—	—	FRS9 [R/W] B,H,W -011-011	FRS8 [R/W] B,H,W -011-011	
0001D4 _H	IPCPH4, IPCPL4 [R] H,W XXXXXXXXXX XXXXXXXXX		IPCPH5, IPCPL5 [R] H,W XXXXXXXXXX XXXXXXXXX		ICU1
0001D8 _H	IPCPH6, IPCPL6 [R] H,W XXXXXXXXXX XXXXXXXXX		IPCPH7, IPCPL7 [R] H,W XXXXXXXXXX XXXXXXXXX		
0001DC _H	PICSH45 [W,R] B,H,W 00000000	PICSL45 [R/W] B,H,W 00000000	ICSH67 [R] B,H,W -----00	ICSL67[R/W] B,H,W 00000000	
0001E0 _H	TMRRH3, TMRRL3 [R/W] H,W XXXXXXXXXX XXXXXXXXX		TMRRH4, TMRRL4 [R/W] H,W XXXXXXXXXX XXXXXXXXX		Wave form Generator 1
0001E4 _H	TMRRH5, TMRRL5 [R/W] H,W XXXXXXXXXX XXXXXXXXX		—		
0001E8 _H	DTCR3 [R/W] B,H,W 00000000	DTCR4 [R/W] B,H,W 00000000	DTCR5 [R/W] B,H,W 00000000	—	
0001EC _H	—	SIGCR11 [R/W] B,H,W 00000000	—	SIGCR21 [R/W] B,H,W 000000-1	
0001F0 _H	ADCOMP3 [W]/ADCOMPB3 [R] H,W 00000000 00000000		ADCOMPD3 [W] /ADCOMPDB3 [R] H,W 00000000 00000000		A/D activation compare 1
0001F4 _H	ADCOMP4 [W] ADCOMP B4 [R] H,W 00000000 00000000		ADCOMPD4 [W]/ADCOMPDB4 [R] H,W 00000000 00000000		
0001F8 _H	ADCOMP5 [W] /ADCOMPB5 [R] H,W 00000000 00000000		ADCOMPD5 [W] / ADCOMPDB5 [R] H,W 00000000 00000000		
0001FC _H	—	ADTGBUF1[R/W] B,H,W -000-111	ADTGSEL1[R/W] B,H,W --000000	ADTGCE1[R/W] B,H,W --000000	

Table A-1 I/O Map (8 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000200 _H	DMACA0 [R/W] B,H,W ^{*1} 00000000 ----XXXX XXXXXXXXXX XXXXXXXXXX				DMAC
000204 _H	DMACB0 [R/W] B,H,W 00000000 00000000 XXXXXXXXXX XXXXXXXXXX				
000208 _H	DMACA1 [R/W] B,H,W ^{*1} 00000000 ----XXXX XXXXXXXXXX XXXXXXXXXX				
00020C _H	DMACB1 [R/W] B,H,W 00000000 00000000 XXXXXXXXXX XXXXXXXXXX				
000210 _H	DMACA2 [R/W] B,H,W ^{*1} 00000000 ----XXXX XXXXXXXXXX XXXXXXXXXX				
000214 _H	DMACB2 [R/W] B,H,W 00000000 00000000 XXXXXXXXXX XXXXXXXXXX				
000218 _H	DMACA3 [R/W] B,H,W ^{*1} 00000000 ----XXXX XXXXXXXXXX XXXXXXXXXX				
00021C _H	DMACB3 [R/W] B,H,W 00000000 00000000 XXXXXXXXXX XXXXXXXXXX				
000220 _H	DMACA4 [R/W] B,H,W ^{*1} 00000000 ----XXXX XXXXXXXXXX XXXXXXXXXX				
000224 _H	DMACB4 [R/W] B,H,W 00000000 00000000 XXXXXXXXXX XXXXXXXXXX				
000228 _H to 00023C _H	—				(Reserved)
000240 _H	DMACR [R/W] B,H,W 0--00000 -----				DMAC
000244 _H to 00039C _H	—				(Reserved)

Table A-1 I/O Map (9 / 19)

Address	Register				Block
	+0	+1	+2	+3	
0003A0 _H	DSP-PC [R/W] B,H,W 000000-0	DSP-CSR[R/W,R,W] B,H,W 00000000	—	—	Multiplication and addition calculator circuit
0003A4 _H	DSP-LY [R/W], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003A8 _H	DSP-OT0 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003AC _H	DSP-OT1 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003B0 _H	DSP-OT2 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003B4 _H	DSP-OT3 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003B8 _H	DSP-OT4 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003BC _H	DSP-OT5 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003C0 _H	DSP-OT6 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003C4 _H	DSP-OT7 [R], W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003C8 _H	DSP-AC0 [R], W ----- 00000000				
0003CC _H	DSP-AC1 [R], W 00000000 00000000 00000000 00000000				
0003D0 _H	DSP-AC2 [R], W 00000000 00000000 00000000 00000000				
0003D4 _H to 0003EC _H	—				(Reserved)
0003F0 _H	BSD0 [W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Bit search module
0003F4 _H	BSD1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003F8 _H	BSDC [W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003FC _H	BSRR [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table A-1 I/O Map (10 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000400 _H	DDR0 [R/W] B,H,W 00000000	DDR1 [R/W] B,H,W 00000000	DDR2 [R/W] B,H,W 00000000	DDR3 [R/W] B,H,W 00000000	Port direction register
000404 _H	DDR5 [R/W] B,H,W -0000000	DDR6 [R/W] B,H,W -----00	DDR8 [R/W] B,H,W 00000000	DDR9 [R/W] B,H,W 00000000	
000408 _H	DDRA [R/W] B,H,W ---00000	DDRB [R/W] B,H,W 00000000	DDRC [R/W] B,H,W 00000000	DDRD [R/W] B,H,W ----0000	
00040C _H	DDRE [R/W] B,H,W 00000000	DDRF [R/W] B,H,W 00000000	DDRG [R/W] B,H,W --000000	DDRH [R/W] B,H,W --000000	
000410 _H	DDRJ [R/W] B,H,W 00000000	—	DDRL [R/W] B,H,W -----000	DDRM [R/W] B,H,W ----0000	
000414 _H	DDRP [R/W] B,H,W --000000	DDRQ [R/W] B,H,W --000000	DDRR [R/W] B,H,W --000000	DDRS [R/W] B,H,W --000000	
000418 _H to 00041C _H	—				(Reserved)
000420 _H	PFR0 [R/W] B,H,W 11111111	PFR1 [R/W] B,H,W 11111111	PFR2 [R/W] B,H,W 11111111	PFR3 [R/W] B,H,W 11111111	Port function register
000424 _H	PFR5 [R/W] B,H,W -1111111	PFR6 [R/W] B,H,W -----11	PFR8 [R/W] B,H,W 0000---	PFR9 [R/W] B,H,W 00000000	
000428 _H	—	—	PFRC [R/W] B,H,W --0-00-0	—	
00042C _H	—	PFRF [R/W] B,H,W -----0	PFRG [R/W] B,H,W --0-00-0	PFRH [R/W] B,H,W --0-00-0	
000430 _H	PFRJ [R/W] B,H,W 0-0-0-0-	—	—	PFRM [R/W] B,H,W ----0000	
000434 _H	—	PFRQ [R/W] B,H,W --000000	—	PFRS [R/W] B,H,W --000000	
000438 _H to 00043C _H	—				(Reserved)

Table A-1 I/O Map (11 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000440 _H	ICR00 [R/W, R] B,H,W ---11111	ICR01 [R/W, R] B,H,W ---11111	ICR02 [R/W, R] B,H,W ---11111	ICR03 [R/W, R] B,H,W ---11111	Interrupt Controller
000444 _H	ICR04 [R/W, R] B,H,W ---11111	ICR05 [R/W, R] B,H,W ---11111	ICR06 [R/W, R] B,H,W ---11111	ICR07 [R/W, R] B,H,W ---11111	
000448 _H	ICR08 [R/W, R] B,H,W ---11111	ICR09 [R/W, R] B,H,W ---11111	ICR10 [R/W, R] B,H,W ---11111	ICR11 [R/W, R] B,H,W ---11111	
00044C _H	ICR12 [R/W, R] B,H,W ---11111	ICR13 [R/W, R] B,H,W ---11111	ICR14 [R/W, R] B,H,W ---11111	ICR15 [R/W, R] B,H,W ---11111	
000450 _H	ICR16 [R/W, R] B,H,W ---11111	ICR17 [R/W, R] B,H,W ---11111	ICR18 [R/W, R] B,H,W ---11111	ICR19 [R/W, R] B,H,W ---11111	
000454 _H	ICR20 [R/W, R] B,H,W ---11111	ICR21 [R/W, R] B,H,W ---11111	ICR22 [R/W, R] B,H,W ---11111	ICR23 [R/W, R] B,H,W ---11111	
000458 _H	ICR24 [R/W, R] B,H,W ---11111	ICR25 [R/W, R] B,H,W ---11111	ICR26 [R/W, R] B,H,W ---11111	ICR27 [R/W, R] B,H,W ---11111	
00045C _H	ICR28 [R/W, R] B,H,W ---11111	ICR29 [R/W, R] B,H,W ---11111	ICR30 [R/W, R] B,H,W ---11111	ICR31 [R/W, R] B,H,W ---11111	
000460 _H	ICR32 [R/W, R] B,H,W ---11111	ICR33 [R/W, R] B,H,W ---11111	ICR34 [R/W, R] B,H,W ---11111	ICR35 [R/W, R] B,H,W ---11111	
000464 _H	ICR36 [R/W, R] B,H,W ---11111	ICR37 [R/W, R] B,H,W ---11111	ICR38 [R/W, R] B,H,W ---11111	ICR39 [R/W, R] B,H,W ---11111	
000468 _H	ICR40 [R/W, R] B,H,W ---11111	ICR41 [R/W, R] B,H,W ---11111	ICR42 [R/W, R] B,H,W ---11111	ICR43 [R/W, R] B,H,W ---11111	
00046C _H	ICR44 [R/W, R] B,H,W ---11111	ICR45 [R/W, R] B,H,W ---11111	ICR46 [R/W, R] B,H,W ---11111	ICR47 [R/W, R] B,H,W ---11111	
000470 _H to 00047C _H	—				(Reserved)
000480 _H	RSRR [R/W] B,H,W 1-0-0-00	STCR [R/W] B,H,W 001100-1	TBCR [R/W] B,H,W 00XXX-00	CTBR [W] B,H,W XXXXXXXXXX	Clock Control unit
000484 _H	CLKR [R/W] B,H,W -000-000	—	DIVR0 [R/W] B,H,W 00000011	DIVR1 [R/W] B,H,W 00000000	
000488 _H to 0004FC _H	—				(Reserved)

Table A-1 I/O Map (12 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000500 _H	—	AICR0 [R/W] B,H,W ----1111	—		8/10-bit A/D converter 0 (with 4 channels)
000504 _H	ADCS0 [R/W,W] B,H,W 0000000-	—	ADCH0 [R/W] B,H,W --00--00	ADMD0 [R/W] B,H,W 00001111	
000508 _H	ADCD000 [R] B,H,W 10---XX XXXXXXXXX		ADCD010 [R] B,H,W 10---XX XXXXXXXXX		
00050C _H	ADCD020 [R] B,H,W 10---XX XXXXXXXXX		ADCD030 [R] B,H,W 10---XX XXXXXXXXX		
000510 _H	—	AICR1 [R/W] B,H,W ----1111	—		8/10-bit A/D converter 1 (with 4 channels)
000514 _H	ADCS1 [R/W,W] B,H,W 0000000-	—	ADCH1 [R/W] B,H,W --00--00	ADMD1 [R/W] B,H,W 00001111	
000518 _H	ADCD001 [R] B,H,W 10---XX XXXXXXXXX		ADCD011 [R] B,H,W 10---XX XXXXXXXXX		
00051C _H	ADCD021 [R] B,H,W 10---XX XXXXXXXXX		ADCD031 [R] B,H,W 10---XX XXXXXXXXX		
000520 _H	—	AICR3 [R/W] B,H,W ----1111	—		12-bit A/D converter 3 (with 4 channels)
000524 _H	ADCS3 [R/W,W] B,H,W 0000000-	—	ADCH3 [R/W] B,H,W --00--00	ADMD3 [R/W] B,H,W 00001111	
000528 _H	ADCD003 [R] B,H,W 10--XXXX XXXXXXXXX		ADCD013 [R] B,H,W 10--XXXX XXXXXXXXX		
00052C _H	ADCD023 [R] B,H,W 10--XXXX XXXXXXXXX		ADCD033 [R] B,H,W 10--XXXX XXXXXXXXX		
000530 _H	—	AICR4 [R/W] B,H,W ----1111	—		12-bit A/D converter 4 (with 4 channels)
000534 _H	ADCS4 [R/W,W] B,H,W 0000000-	—	ADCH4 [R/W] B,H,W --00--00	ADMD4 [R/W] B,H,W 00001111	
000538 _H	ADCD004 [R] B,H,W 10--XXXX XXXXXXXXX		ADCD014 [R] B,H,W 10--XXXX XXXXXXXXX		
00053C _H	ADCD024 [R] B,H,W 10--XXXX XXXXXXXXX		ADCD034 [R] B,H,W 10--XXXX XXXXXXXXX		
000540 _H	RCR10 [W] B,H,W XXXXXXXX	RCR00 [W] B,H,W XXXXXXXX	UDCR10 [R] B,H,W 00000000	UDCR00 [R] B,H,W 00000000	Up/down counter 0
000544 _H	CCR0 [R/W] B,H,W 00000000	CCRL0 [R/W, R] B,H,W -0001000	—	CSR0 [R/W, R] B,H,W 00000000	
000548 _H to 00055C _H	—				(Reserved)

Table A-1 I/O Map (13 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000560 _H	SSR4 [R/W, R] B,H,W 00000011	ESCR4 [R/W]/ IBSR4 [R/W, R] B,H,W 00000000	SCR4 [R/W] / IBCR4 [R/W, R] B,H,W 00000000	SMR4 [R/W] B,H,W 000-0000	Multi-function serial interface4
000564 _H	BGR41 [R/W] B,H,W 00000000	BGR40 [R/W] B,H,W 00000000	RDR4 [R] / TDR [W] H,W -----0 00000000		
000568 _H	—		ISMK4 [R/W] B,H,W 01111111	ISBA4 [R/W] B,H,W 00000000	
00056C _H	FBYTE42 [R/W] B,H,W 00000000	FBYTE41 [R/W] B,H,W 00000000	FCR41 [R/W] B,H,W ---00100	FCR40 [R/W,R] B,H,W -0000000	
000570 _H	SSR5 [R/W, R] B,H,W 00000011	ESCR5 [R/W]/ IBSR5 [R/W, R] B,H,W 00000000	SCR5 [R/W] / IBCR5 [R/W, R] B,H,W 00000000	SMR5 [R/W] B,H,W 000-0000	Multi-function serial interface5
000574 _H	BGR51 [R/W] B,H,W 00000000	BGR50 [R/W] B,H,W 00000000	RDR5 [R] / TDR5 [W] H,W -----0 00000000		
000578 _H	—		ISMK5 [R/W] B,H,W 01111111	ISBA5 [R/W] B,H,W 00000000	
00057C _H	FBYTE52 [R/W] B,H,W 00000000	FBYTE51 [R/W] B,H,W 00000000	FCR51 [R/W] B,H,W ---00100	FCR50 [R/W, R] B,H,W -0000000	
000580 _H	BT1TMR [R] B,H,W 00000000 00000000		BT1TMCR [R/W] B,H,W -0000000 00000000		Base Timer 1
000584 _H	—	BT1STC [R/W] B 00000000	—		
000588 _H	BT1PCSR/BT1PRL [R/W] H,W XXXXXXXX XXXXXXXX		BT1PDUT/BT1PRLH/BT1DTBF [R/W] H,W XXXXXXXX XXXXXXXX		
00058C _H	—				(Reserved)
000590 _H	BT2TMR [R] B,H,W 00000000 00000000		BT2TMCR [R/W] B,H,W -0000000 00000000		Base Timer 2
000594 _H	—	BT2STC [R/W] B 00000000	—		
000598 _H	BT2PCSR/BT2PRL [R/W] H,W XXXXXXXX XXXXXXXX		BT2PDUT/BT2PRLH/BT2DTBF [R/W] H,W XXXXXXXX XXXXXXXX		
00059C _H	—				(Reserved)
0005A0 _H	BT3TMR [R] B,H,W 00000000 00000000		BT3TMCR [R/W] B,H,W -0000000 00000000		Base Timer 3
0005A4 _H	—	BT3STC [R/W] B 00000000	—		
0005A8 _H	BT3PCSR/BT3PRL [R/W] H,W XXXXXXXX XXXXXXXX		BT3PDUT/BT3PRLH/BT3DTBF [R/W] H,W XXXXXXXX XXXXXXXX		

Table A-1 I/O Map (14 / 19)

Address	Register				Block
	+0	+1	+2	+3	
0005AC _H to 0005FC _H	—				(Reserved)
000600 _H	PCR0 [R/W] B,H,W 00000000	PCR1 [R/W] B,H,W 00000000	PCR2 [R/W] B,H,W 00000000	PCR3 [R/W] B,H,W 00000000	Pull-up resistor control register
000604 _H	PCR5 [R/W] B,H,W -0000000	PCR6 [R/W] B,H,W -----00	PCR8 [R/W] B,H,W 00000000	PCR9 [R/W] B,H,W 00000000	
000608 _H	PCRA [R/W] B,H,W ---00000	PCRB [R/W] B,H,W 00000000	PCRC [R/W] B,H,W 00000000	PCRD [R/W] B,H,W ---0000	
00060C _H	PCRE [R/W] B,H,W 00000000	PCRF [R/W] B,H,W 00000000	PCRG [R/W] B,H,W --000000	PCRH [R/W] B,H,W --000000	
000610 _H	PCRJ [R/W] B,H,W 00000000	—	PCRL [R/W] B,H,W ----000	PCRM [R/W] B,H,W ----0000	
000614 _H	PCRP [R/W] B,H,W --000000	PCRQ [R/W] B,H,W --000000	PCRR [R/W] B,H,W --000000	PCRS [R/W] B,H,W --000000	
000618 _H to 00063C _H	—				(Reserved)

Table A-1 I/O Map (15 / 19)

Address	Register				Block
	+0	+1	+2	+3	
000640 _H	ASR0 [R/W] H,W 00000000 00000000 *2		ACR0 [R/W] H,W 1111XX-- --000000 *2		External bus interface
000644 _H	ASR1 [R/W] H,W XXXXXXXXXX XXXXXXXXXX *2		ACR1 [R/W] H,W XXXXXXX-- --XXXXXXX *2		
000648 _H	ASR2 [R/W] H,W XXXXXXXXXX XXXXXXXXXX *2		ACR2 [R/W] H,W XXXXXXX-- --XXXXXXX *2		
00064C _H	—				
000650 _H					
000654 _H					
000658 _H					
00065C _H					
000660 _H	AWR0 [R/W] H,W 0111---- 1111-111 *2		AWR1 [R/W] H,W XXXX---- XXXX-XXX *2		
000664 _H	AWR2 [R/W] H,W XXXX---- XXXX-XXX *2		—		
000668 _H	—				
00066C _H					
000670 _H					
000674 _H					
000678 _H					
00067C _H					
000680 _H	CSER [R/W] B,H ----001	—	—	—	
000684 _H to 0007F8 _H	—				(Reserved)
0007FC _H	—	MODR [W] XXXXXXXX	—	—	Mode register
000800 _H to 000FFC _H	—				(Reserved)

Table A-1 I/O Map (16 / 19)

Address	Register				Block
	+0	+1	+2	+3	
001000 _H	DMASA0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMAC
001004 _H	DMADA0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001008 _H	DMASA1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00100C _H	DMADA1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001010 _H	DMASA2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001014 _H	DMADA2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001018 _H	DMASA3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00101C _H	DMADA3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001020 _H	DMASA4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001024 _H	DMADA4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001028 _H to 006FFC _H	—				(Reserved)
007000 _H	FLCR [R/W, R] B ----X-0-	—	—	—	Flash memory
007004 _H	FLWC [R/W] B -----011	—	—	—	
007008 _H	—				
00700C _H					
007010 _H					
007014 _H to 00701C _H	—				(Reserved)
007020 _H	WREN [R/W] H 00000000 00000000		—		Wild register control block
007024 _H	—				
007028 _H					
00702C _H					

Table A-1 I/O Map (17 / 19)

Address	Register				Block
	+0	+1	+2	+3	
007030 _H	WA00 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				Wild register control block
007034 _H	WD00 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007038 _H	WA01 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00703C _H	WD01 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007040 _H	WA02 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
007044 _H	WD02 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007048 _H	WA03 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00704C _H	WD03 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007050 _H	WA04 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
007054 _H	WD04 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007058 _H	WA05 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00705C _H	WD05 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007060 _H	WA06 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
007064 _H	WD06 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007068 _H	WA07 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00706C _H	WD07 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				

Table A-1 I/O Map (18 / 19)

Address	Register				Block
	+0	+1	+2	+3	
007070 _H	WA08 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				Wild register control block
007074 _H	WD08 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007078 _H	WA09 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00707C _H	WD09 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007080 _H	WA10 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
007084 _H	WD10 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007088 _H	WA11 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00708C _H	WD11 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007090 _H	WA12 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
007094 _H	WD12 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
007098 _H	WA13 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
00709C _H	WD13 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
0070A0 _H	WA14 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
0070A4 _H	WD14 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
0070A8 _H	WA15 [R/W] W ----- ----XXXX XXXXXXXXXX XXXXXXX--				
0070AC _H	WD15 [R/W] W XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
0070B0 _H to 00BFFC _H	—				(Reserved)

Table A-1 I/O Map (19 / 19)

Address	Register				Block
	+0	+1	+2	+3	
00C000 _H to 00C0FC _H	X-RAM (coefficient RAM) [R/W] 64 x32-bit				Multiplication and addition calculator
00C100 _H to 00C1FC _H	Y-RAM (variable RAM) [R/W] 64 x32-bit				
00C200 _H to 00C3FC _H	I-RAM (instruction RAM) [R/W] 128 x32-bit				
00C400 _H to 00FFFC _H	—				(Reserved)
010000 _H to 0FFFFC _H	—				(Reserved)

*1: The lower 16 bits (DTC[15:0]) of DMACA0 to DMACA4 cannot be accessed in bytes.

*2: The initial value depends on the reset level. Therefore, an initial value has been described.

Notes:

- Data is undefined in reserved or (-) area.
- Do not execute read-modify-write (RMW) instruction on registers having a write-only bit.
- Different product series may have different setting of initial values. Details please refer to the related chapters.

APPENDIX B Interrupt Vector

This section shows the vector table of the MB91470/480 series.

■ Interrupt Vector Table

Table B-1 Interrupt Vector (1 / 4)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address
	Decimal	Hexadecimal			
Reset	0	00	-	3FC _H	000FFFFC _H
Mode vector	1	01	-	3F8 _H	000FFFF8 _H
System reserved	2	02	-	3F4 _H	000FFFF4 _H
System reserved	3	03	-	3F0 _H	000FFFF0 _H
System reserved	4	04	-	3EC _H	000FFFE4 _H
System reserved	5	05	-	3E8 _H	000FFFE8 _H
System reserved	6	06	-	3E4 _H	000FFFE4 _H
Coprocessor absent trap	7	07	-	3E0 _H	000FFFE0 _H
Coprocessor error trap	8	08	-	3DC _H	000FFFD4 _H
INTE instruction	9	09	-	3D8 _H	000FFFD8 _H
System reserved	10	0A	-	3D4 _H	000FFFD4 _H
System reserved	11	0B	-	3D0 _H	000FFFD0 _H
Step trace trap	12	0C	-	3CC _H	000FFFCC _H
NMI demand (tool)	13	0D	-	3C8 _H	000FFFC8 _H
Undefined instruction exception	14	0E	-	3C4 _H	000FFFC4 _H
NMI demand	15	0F	-	3C0 _H	000FFFC0 _H
External interrupt 0	16	10	ICR00	3BC _H	000FFBFC _H
External interrupt 1	17	11	ICR01	3B8 _H	000FFB8 _H
External interrupt 2	18	12	ICR02	3B4 _H	000FFB4 _H
External interrupt 3	19	13	ICR03	3B0 _H	000FFB0 _H
External interrupt 4	20	14	ICR04	3AC _H	000FFAC _H
External interrupt 5	21	15	ICR05	3A8 _H	000FFA8 _H

Table B-1 Interrupt Vector (2 / 4)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address
	Decimal	Hexadecimal			
External interrupt 6	22	16	ICR06	3A4 _H	000FFFA4 _H
External interrupt 7	23	17	ICR07	3A0 _H	000FFFA0 _H
Reload timer 0	24	18	ICR08	39C _H	000FFF9C _H
Reload timer 1	25	19	ICR09	398 _H	000FFF98 _H
Base timer 0 (Factor 0/Factor 1)	26	1A	ICR10	394 _H	000FFF94 _H
Multi-function serial interface0 (UART send completed/ receive completed/I ² C status)	27	1B	ICR11	390 _H	000FFF90 _H
Multi-function serial interface1 (UART send completed/ receive completed/I ² C status)	28	1C	ICR12	38C _H	000FFF8C _H
Base timer 1 (Factor 0/Factor 1)	29	1D	ICR13	388 _H	000FFF88 _H
Base Timer 2/3 (Factor 0/Factor 1) Up-Down Counter 0	30	1E	ICR14	384 _H	000FFF84 _H
DTTI0/DTTI1	31	1F	ICR15	380 _H	000FFF80 _H
DMAC0 (end, error)	32	20	ICR16	37C _H	000FFF7C _H
DMAC1 (end, error)	33	21	ICR17	378 _H	000FFF78 _H
DMAC2/3/4 (end, error)	34	22	ICR18	374 _H	000FFF74 _H
Multi-function serial interface2 (UART send completed/ receive completed/I ² C status)	35	23	ICR19	370 _H	000FFF70 _H
Multi-function serial interface3 (UART send completed/ receive completed/I ² C status)	36	24	ICR20	36C _H	000FFF6C _H
Multi-function serial interface4 (UART send completed/ receive completed/I ² C status)	37	25	ICR21	368 _H	000FFF68 _H
Multi-function serial interface5 (UART send completed/ receive completed/I ² C status)	38	26	ICR22	364 _H	000FFF64 _H
Multiplication and addition calculator	39	27	ICR23	360 _H	000FFF60 _H
PPG0/PPG1	40	28	ICR24	35C _H	000FFF5C _H

Table B-1 Interrupt Vector (3 / 4)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address
	Decimal	Hexadecimal			
PPG2/PPG3/PPG8/PPG9	41	29	ICR25	358 _H	000FFF58 _H
PPG4/PPG5/PPG10/PPG11	42	2A	ICR26	354 _H	000FFF54 _H
PPG6/PPG7/PPG12/PPG13/PPG14/ PPG15	43	2B	ICR27	350 _H	000FFF50 _H
Waveform generator 0/3 (underflow)	44	2C	ICR28	34C _H	000FFF4C _H
Waveform generator 1/4 (underflow)	45	2D	ICR29	348 _H	000FFF48 _H
Waveform generator 2/5 (underflow)	46	2E	ICR30	344 _H	000FFF44 _H
time-base timer overflow	47	2F	ICR31	340 _H	000FFF40 _H
External interrupt 8/9/10/11/12/13/ 14/15	48	30	ICR32	33C _H	000FFF3C _H
Free-run timer 0/3 (Compare clear)	49	31	ICR33	338 _H	000FFF38 _H
Free-run timer 0/3 (zero detection)	50	32	ICR34	334 _H	000FFF34 _H
Free-run timer 1/4 (Compare clear)	51	33	ICR35	330 _H	000FFF30 _H
Free-run timer 1/4 (zero detection)	52	34	ICR36	32C _H	000FFF2C _H
Free-run timer 2/5 (Compare clear)	53	35	ICR37	328 _H	000FFF28 _H
Free-run timer 2/5 (zero detection)	54	36	ICR38	324 _H	000FFF24 _H
8/10-bit A/D Converter 2	55	37	ICR39	320 _H	000FFF20 _H
8/10-bit A/D Converter 0 / 12-bit A/D Converter 3	56	38	ICR40	31C _H	000FFF1C _H
8/10-bit A/D Converter 1 / 12-bit A/D Converter 4	57	39	ICR41	318 _H	000FFF18 _H
ICU0/ICU1/ICU4/ICU5 (capture)	58	3A	ICR42	314 _H	000FFF14 _H
ICU2/ICU3/ICU6/ICU7 (capture)	59	3B	ICR43	310 _H	000FFF10 _H
OCU0/OCU1/OCU6/OCU7 (match)	60	3C	ICR44	30C _H	000FFF0C _H
OCU2/OCU3/OCU8/OCU9 (match)	61	3D	ICR45	308 _H	000FFF08 _H
OCU4/OCU5/OCU10/OCU11 (match)	62	3E	ICR46	304 _H	000FFF04 _H

Table B-1 Interrupt Vector (4 / 4)

Interrupt factor	Interrupt number		Interrupt level	Offset	TBR default address
	Decimal	Hexadecimal			
Delay interrupt trigger bit	63	3F	ICR47	300 _H	000FFF00 _H
System reserved (used for REALOS)	64	40	-	2FC _H	000FFEFC _H
System reserved (used for REALOS)	65	41	-	2F8 _H	000FFEF8 _H
System reserved	66	42	-	2F4 _H	000FFEF4 _H
System reserved	67	43	-	2F0 _H	000FFEF0 _H
System reserved	68	44	-	2EC _H	000FFEEC _H
System reserved	69	45	-	2E8 _H	000FFEE8 _H
System reserved	70	46	-	2E4 _H	000FFEE4 _H
System reserved	71	47	-	2E0 _H	000FFEE0 _H
System reserved	72	48	-	2DC _H	000FFEDC _H
System reserved	73	49	-	2D8 _H	000FFED8 _H
System reserved	74	4A	-	2D4 _H	000FFED4 _H
System reserved	75	4B	-	2D0 _H	000FFED0 _H
System reserved	76	4C	-	2CC _H	000FFECC _H
System reserved	77	4D	-	2C8 _H	000FFEC8 _H
System reserved	78	4E	-	2C4 _H	000FFEC4 _H
System reserved	79	4F	-	2C0 _H	000FFEC0 _H
Used in INT instruction	80 to 255	50 to FF	-	2BC _H to 000 _H	000FFEBC _H to 000FFC00 _H

APPENDIX C Pin States in Each CPU State

This section defines the following terms related to pin states:

■ Pin States in Each CPU State

Input enabled

Means that an input function can be used.

Input disabled

Means that an input function cannot be used.

Input fixed to "0"

Input level is internally fixed to 0 to prevent leakage due to the input open.

Output Hi-Z

Means to place a pin in a high impedance state by disabling the pin driving transistor from driving.

Retention of the immediately prior state

Means to output the state existing immediately prior to entering this mode.

That is, to output according to an internal resource with an output when it is operating or to preserve an output when the output is provided, for example, as a part.

- Input enabled when enabling external interrupt function selection

Set the pin function to an external interrupt request input pin and can only input if an external interrupt request is enabled.

Table C-1 Single-chip Mode (1 / 3)

Pin Name	Function	At initialization		At sleep	In stop mode	
		INITX = "L"*1	INITX = "H"*2		HIZ = 0	HIZ = 1
P00 to P07	D16 to D23	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
P10 to P17	D24 to D31					
P20 to P27	A00 to A07					
P30 to P37	A08 to A15					
P50 to P52	CS0X to CS2X					
P53	ASX					
P54	RDX					
P55 to P56	WR0X, WR1X					
P60	SYSCLK					
P61	RDY					
NMIX	NMIX	Input disabled	Input enabled	Input enabled	Input enabled	Input enabled

MB91470/480 Series

APPENDIX C Pin States in Each CPU State

Table C-1 Single-chip Mode (2 / 3)

Pin Name	Function	At initialization		At sleep	In stop mode	
		INITX = "L"*1	INITX = "H"*2		HIZ = 0	HIZ = 1
P80 to P83	INT0 to INT3	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Input enabled	Input enabled	Output Hi-Z/ Input "0" fixed Input enabled when enabling interrupt function selection
P84	INT4/PPG4					
P85	INT5/PPG5					
P86	INT6/PPG6					
P87	INT7/PPG7					
P90	INT8/PPG8					
P91	INT9/PPG9					
P92	INT10/PPG10					
P93	INT11/PPG11					
P94	INT12/PPG12					
P95	INT13/PPG13					
P96	INT14/PPG14					
P97	INT15/PPG15					
PA0 to PA4	ADTG0 to ADTG4	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PB0 to PB3	AN0-0 to AN0-3	Output Hi-Z/ Input disabled	Output Hi-Z/ Input "0" fixed	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PB4 to PB7	AN1-0 to AN1-3					
PC0	AN2-0/SCK4					
PC1	AN2-1/SIN4					
PC2	AN2-2/SOT4					
PC3	AN2-3/SCK5					
PC4	AN2-4/SIN5					
PC5	AN2-5/SOT5					
PC6, PC7	AN2-6, AN2-7					
PD0 to PD3	AN2-8 to AN2-11					
PE0 to PE3	AN3-0 to AN3-3					
PE4 to PE7	AN4-0 to AN4-3					
PF0	CLKPOUT	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PF1 to PF6	GPIO					

Table C-1 Single-chip Mode (3 / 3)

Pin Name	Function	At initialization		At sleep	In stop mode	
		INITX = "L"*1	INITX = "H"*2		HIZ = 0	HIZ = 1
PG0,PG3	SCK0, SCK1	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PG1,PG4	SIN0, SIN1					
PG2,PG5	SOT0, SOT1					
PH0,PH3	SCK2, SCK3					
PH1,PH4	SIN2, SIN3					
PH2,PH5	SOT2, SOT3					
PJ0,PJ2,PJ4,PJ6	TIN0 to TIN3	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PJ1,PJ3,PJ5,PJ7	TOUT0 to TOUT3					
PL0	AIN0	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PL1	BIN0					
PL2	ZIN0					
PM0 to PM3	PPG0 to PPG3	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PP0 to PP3	IC0 to IC3	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PP4	CKI0					
PP5	DTTI0					
PQ0 to PQ5	RTO0 to RTO5					
PR0 to PR3	IC4 to IC7	Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z/ Input "0" fixed
PR4	CKI1					
PR5	DTTI1					
PS0 to PS5	RTO6 to RTO11					

Table C-2 External Bus Mode

Pin Name	Function	At initialization		At sleep	In stop mode	
		INITX = L* ¹	INITX = H* ²		HIZ = 0	HIZ = 1
P00 to P07	D16 to D23	Output Hi-Z	Output Hi-Z	Retention of the immediately prior state	Retention of the immediately prior state	Output Hi-Z
P10 to P17	D24 to D31					
P20 to P27	A00 to A07					
P30 to P37	A08 to A15					
P50 to P52	CS0X to CS2X					
P53	ASX					
P54	RDX					
P55, P56	WR0X, WR1X					
P60	SYSCLK	Input disabled	Input disabled			Input "0" fixed
P61	RDY					

*1: INITX = "L": Indicates the pin status with INITX remaining at the "L" level.

*2: INITX = "H": Indicates the pin status existing immediately after INITX transition from "L" to "H" level.

APPENDIX D Notes when Little Endian Region is used

Here, notes of each the following items when the little endian region is used are explained.

- **C compiler**
 - **Assembler**
 - **Linker**
 - **Debugger**
-

■ C Compiler (fcc911)

Care must be taken as operation cannot be guaranteed if the following are performed on the little endian area when programming in C language.

- Arrangement of variable with initial value
- Structure substitution
- Operations other than character type array which uses character-string handling function
- Specifying the -K lib option when using a string handling function
- Use of double type and long double type
- Arrangement in little endian region of stack

● Arrangement of variable with initial value

The variable with the initial value cannot be arranged in the little endian region.

The compiler does not generate the initial value of the little endian. The initial value cannot be set though the variable can be arranged in the little endian region.

Please do processing by which the initial value is set at the head of the program.

Example: When you set the initial value in variable `little_data` of the Little endian region

```
extern int little_data;

void little_init(void) {
    little_data = Initial value;
}

void main(void) {
    little_init();
    ...
}
```

● Structure substitution

When structures are substituted, the compiler selects the optimal transfer method, and transfers are executed per byte, half-word, or word. Thus, errors will result if structure substitution is performed between a structure variable allocated to the ordinary area and another allocated to the little endian area.

Please substitute the member of structure respectively.

Example: When you substitute structure for structure variable `little_st` of the Little endian region

```
struct tag { char c; int i; } normal_st;
extern struct tag little_st;

#define STRMOVE(DEST,SRC) DEST.c=SRC.c;DEST.i=SRC.i;

void main(void) {
    STRMOVE(little_st,normal_st);
}
```

As the layout of structure members differs per compiler, it should be assumed that the member layout is different from structures compiled by other compilers. A correct result is not obtained in the above-mentioned method this time.

When the layout of structure members is unmatched, do not allocate structure variables to the little endian area.

● Operations other than character type array which uses character-string handling function

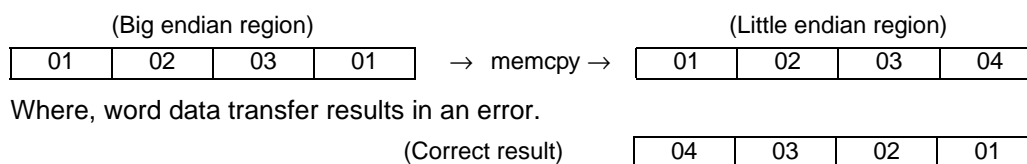
The string handling functions provided in the standard library handle strings in bytes. Thus, errors will result if processing using character string operation functions is performed on areas with other than the "char", "unsigned char", and "signed char" types that are allocated in the little endian area.

Please do not do such processing.

[Example of trouble] Forwarding of word data by wrong example memcpy

```
int big = 0x01020304;    /* Big endian region    */
extern int little;        /* Little endian region    */
memcpy(&little,&big,4);  /* Forwarding by memcpy    */
```

Execution result



● Specifying the -K lib option when using a string handling function

When a -K lib option is specified, the compiler performs inline development on some character string operation functions. At this time, the compiler may change the processing of the function into half-words or words to select the optimum process. Therefore, processing to the little endian region is not correctly executed.

While processing using character string operation functions is performed on the little endian area, do not specify the -K lib option.

Similarly, do not specify the -O 4 or -K speed options that include the -K lib option either.

● Use of double type and long double type

Accessing double or long double forms can be performed by accessing the upper or lower 1 word respectively. Thus, errors will result if accessing the double or long double form variables allocated to the little endian area.

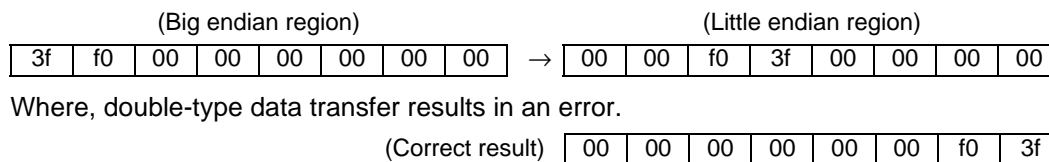
Substitution between the same type of little endian-allocated variables is possible, but as a result of optimization, such substitutions may be replaced by substitution of constant numbers.

Do not place a double or long-double variable in the little endian area.

[Example of trouble] Forwarding of wrong example double type data

```
double big = 1.0;      /* Big endian region      */
extern int little;     /* Little endian region    */
little = big;          /* Forwarding of double type data*/
```

Execution result



● Arrangement in Little endian region of stack

Placing part or whole of the stack in the little endian area yields unpredictable results of operation.

■ Assembler (fasm911)

Points to note regarding the little endian area when programming in FR assembler language are described as follows.

● About the section

The little endian region has aimed to do little endian system CPU and the data exchange chiefly. The little endian area should therefore be defined as a data section with no initial value. If a code, stack, or data section with an initial value is specified for the little endian area, access operations under cannot be guaranteed.

[Example]

```
/* Section definition of correct Little endian region */
```

```
.SECTION Little_Area, DATA, ALIGN=4
```

```
Little_Word:
```

```
.RES.W 1
```

```
Little_Half:
```

```
.RES.H 1
```

```
Little_Byte:
```

```
.RES.B 1
```

● About the access of data

When data access is performed on the little endian area, the data value can be coded without consideration for the endian status. However, to access little endian area data, access must be performed on the same size as the data size.

[Example]

```
LDI    #0x01020304, r0
```

```
LDI    #Little_Word, r1
```

```
LDI    #0x0102, r2
```

```
LDI    #Little_Half, r3
```

```
LDI    #0x01, r4
```

```
LDI    #Little_Byte, r5
```

```
/* Access 32-bit data using the ST instruction (or LD instruction, etc.).*/
```

```
ST     r0, @r1
```

```
/* Access 16-bit data using the STH instruction (or LDH instruction, etc.).*/
```

```
STH    r2, @r3
```

```
/* Access 8-bit data using the STB instruction (or LDB instruction, etc.).*/
```

```
STB    r4, @r5
```

If the little endian area is accessed in a size different from the data size on MB91470/480 series, the resulting value is not guaranteed. For example, if two sequential blocks of 16-bit data are simultaneously accessed using a 32-bit access command, the data value cannot be guaranteed.

■ Linker (flnk911)

Points to note regarding the section layout while linking when programs using the little endian area are created are described as follows.

● Limitation of section type

In the little endian area, only data sections with no initial value can be placed.

When a data section with initial value, stack section, or code section is allocated in the little endian area, program operation cannot be guaranteed since the arithmetic processing, such as address solutions, is performed on the big endian area within the linker.

● Undetection of error

As the linker does not recognize the little endian area, no error message will be sent even if an allocation that violates the above restrictions is performed. Sufficiently check the contents of the sections placed in the little endian area before use.

■ Debugger (sim911, eml911, mon911)

● Simulator debugger

There is no memory space specification command by which the little endian region is shown. Memory manipulation commands and instructions which involve memory manipulation are therefore handled as those using the big endian method.

● Emulator debugger and monitor debugger

Care must be taken that data will not be treated as having a normal value when the little endian area is accessed using the following command.

- set memory / show memory / enter / examine / set watch Command

When floating point (single/double) data is handled, specific values cannot be set or displayed.

- search memory Command

When searching data of half-word or word, the search cannot be executed with specific values.

- The line/the disassembly (The disassembly display of the source window is included).

A normal instruction code cannot be set and displayed (Do not place instruction code in the little endian area).

- call / show call Command

Arrangement of the stack area in the little endian area does not lead to normal operation (Do not place the stack area in the little endian area).

APPENDIX E Instruction List

It is an instruction table of the FR family.

■ Instruction List of FR Family

Figure E-1 How to Read the Instruction List

Mnemonic	Type	OP	CYC	NZVC	Operation	Remarks
ADD Rj, Rj	A	AG	1	CCCC	Ri + Rj →Rj	
* ADD #s5, Rj	C	A4	1	CCCC	Ri + s5 →Ri	
,	,	,	,	,	,	
,	,	,	,	,	,	

(1)
(2)
(3)
(4)
(5)
(6)
(7)

(1) The instruction name is shown.

Instructions marked with * are extended instructions implemented either by extending existing instructions or by coding from scratch using the assembler, which are not native to the CPU.

(2) A specifiable Addressing mode is shown in the operand by the sign.

Please refer to the sign of the Addressing mode (next item) for the meaning of the sign.

(3) The instruction format is shown.

(4) The hexadecimal number is displayed to the instruction code.

(5) The number of machine cycles is shown.

a: It is a memory access cycle, and there is a possibility to postpone by the Ready function.

b: It is a memory access cycle, and there is a possibility to postpone by the Ready function. When the immediately succeeding instruction references the register to be subject to LD operation, however, an interlock is applied and the number of execution cycles is incremented by 1.

c: If the succeeding instruction reads or writes to R15, SSP, or USP or if the instruction is in instruction format A, an interlock is applied and the number of execution cycles is incremented by 1 to become 2.

d: If the succeeding instruction references MDH/MDL, an interlock is applied and the number of execution cycles is incremented to become 2. a, b, c, d, and the minimum are one cycle.

(6) The flag change is shown.

Flag change	Meaning of flag
C : Change	N : Negative flag
- : No change	Z : Zero flag
0 : Clear	V : Overflow flag
1 : Set	C : Carry flag

(7) The instruction operation is written.

■ **Addressing Mode Symbols**

Ri : register direct (R0 to R15, AC, FP, SP)
Rj : register direct (R0 to R15, AC, FP, SP)
R13 : register direct (R13, AC)
Ps : Register direct (program status register)
Rs : register direct (TBR, RP, SSP, USP, MDH, MDL)
Cri : register direct (CR0 to CR15)
CRj : register direct (CR0 to CR15)
#i8 : Unsigned 8-bit value immediately(-128 to 255)
Attention: - 128 to -1 is treated as 128 to 255.
#i20 : Unsigned 20-bit value immediately(-0X80000 to 0XFFFFFF)
Attention: -0X7FFFF to -1 is treated as 0X7FFFF to 0XFFFFFF.
#i32 : Unsigned 32-bit value immediately(-0X80000000 to 0xFFFFFFFF)
Attention: 0X80000000 to -1 is treated as 0X80000000 to 0xFFFFFFFF.
#s5 : Signed 5-bit immediate value (-16 to 15)
#s10 : Signed 10-bit immediate value (only multiples of 4, - 512 to 508)
#u4 : Unsigned 4-bit value immediately (0 to 15)
#u5 : Unsigned 5-bit value immediately (0 to 31)
#u8 : Unsigned 8-bit value immediately (0 to 255)
#u10 : Unsigned ten bit value immediately (Only the multiple of 4, 0 to 1020)
@dir8 : Unsigned 8-bit direct address (0 to 0XFF)
@dir9 : Unsigned 9-bit direct address (Only the multiple of 2, 0 to 0X1FE)
@dir10: Unsigned ten-bit direct address (Only the multiple of 4, 0 to 0X3FC)
label9: Signed 9-bit branch address (only multiples of 2, -0X100 to 0XFC)
label12 : Signed 12-bit branch address (only multiples of 2, -0X800 to 0X7FC)
label20: Divergence address of signed 20 bits (-0X80000 to 0X7FFFF)
label32: Divergence address of signed 32 bits (-0X80000000 to 0X7FFFFFFF)
@Ri : Register indirect (R0 to R15, AC, FP, SP)
@Rj : Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj) : Relativity is register indirect (Rj: R0 to R15, AC, FP, SP)
@(R14,disp10): Relative indirectly register (Only the multiple of disp10: -0X200 to 0X1FC 4)
@(R14,disp9): Relative indirectly register (Only the multiple of disp9: -0X100 to 0XFE 2)
@(R14,disp8): Relativity is register indirect (disp8: -0X80 to 0X7F)
@(R15,udisp6): Relative indirectly register (Only the multiple of 4, udisp6: 0 to 60)
@Ri+: Register indirect with post increment (R0 to R15, AC, FP, SP)
@R13+: Register indirect with post increment (R13, AC)
@SP+: Stack pop
@-SP: Stack push
(reglist) : Register list

■ Instruction Format

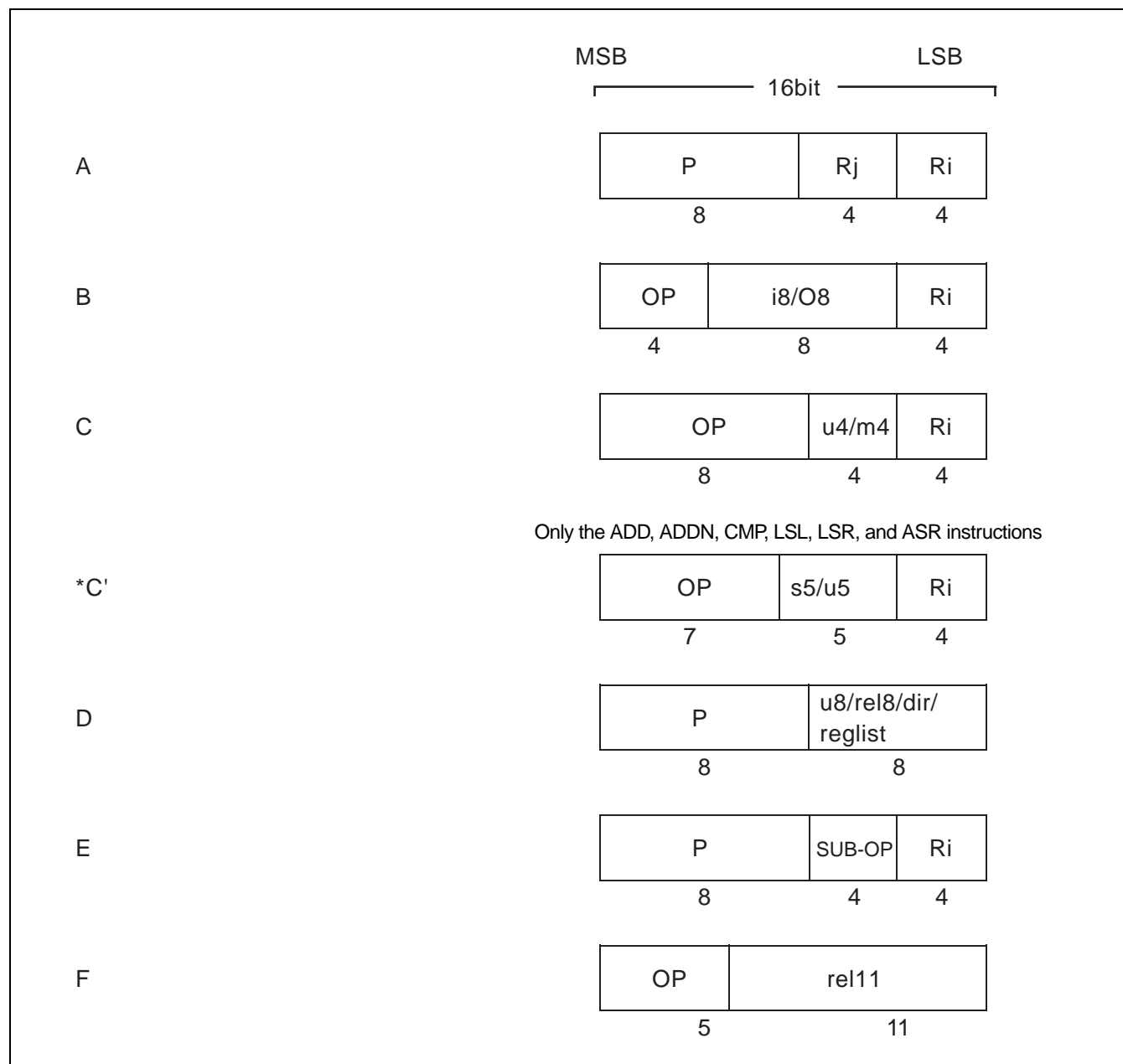


Table E-1 Addition and Subtraction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	Ri+Rj→ Ri	I think high rank 1bit to be a sign in the assembler. 0 expansions Minus expansion
ADD #s5, Ri	C'	A4	1	CCCC	Ri+s5→ Ri	
ADD #u4, Ri	C	A4	1	CCCC	Ri+extu(i4)→ Ri	
ADD2 #u4, Ri	C	A5	1	CCCC	Ri+extu(i4)→ Ri	
ADDN Rj, Ri	A	A7	1	CCCC	Ri+Rj+c→ Ri	Addition with carry
ADDN Rj, Ri	A	A2	1	----	Ri+Rj→ Ri	I think high rank 1bit to be a sign in the assembler. 0 expansions Minus expansion
*ADDN #s5, Ri	C'	A0	1	----	Ri+s5→ Ri	
ADDN #u4, Ri	C	A0	1	----	Ri+extu(i4)→ Ri	
ADDN2 #u4, Ri	C	A1	1	----	Ri+extu(i4)→ Ri	
SUB Rj, Ri	A	AC	1	CCCC	Ri-Rj→ Ri	
SUBC Rj, Ri	A	AD	1	CCCC	Ri-Rj-c→ Ri	Reduction with carry
SUBN Rj, Ri	A	AE	1	----	Ri-Rj→ Ri	

Table E-2 Comparison Operation

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	Ri-Rj	I think high rank 1bit to be a sign in the assembler. 0 expansions Minus expansion
*CMP #s5, Ri	C'	A8	1	CCCC	Ri-s5	
CMP #u4, Ri	C	A8	1	CCCC	Ri-extu(i4)	
CMP2 #u4, Ri	C	A9	1	CCCC	Ri-extu(i4)	

Table E-3 Logical Operation

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
AND Rj, Ri	A	82	1	CC--	Ri &= Rj	-	word
AND Rj, @Ri	A	84	1+2a	CC--	(Ri) &= Rj	○	word
ANDH Rj, @Ri	A	85	1+2a	CC--	(Ri) &= Rj	○	half-word
ANDB Rj, @Ri	A	86	1+2a	CC--	(Ri) &= Rj	○	Bytes
OR Rj, Ri	A	92	1	CC--	Ri = Rj	-	word
OR Rj, @Ri	A	94	1+2a	CC--	(Ri) = Rj	○	word
ORH Rj, @Ri	A	95	1+2a	CC--	(Ri) = Rj	○	half-word
ORB Rj, @Ri	A	96	1+2a	CC--	(Ri) = Rj	○	Bytes
EOR Rj, Ri	A	9A	1	CC--	Ri ^= Rj	-	word
EOR Rj, @Ri	A	9C	1+2a	CC--	(Ri) ^= Rj	○	word
EORH Rj, @Ri	A	9D	1+2a	CC--	(Ri) ^= Rj	○	half-word
EORB Rj, @Ri	A	9E	1+2a	CC--	(Ri) ^= Rj	○	Bytes

Table E-4 Bit Manipulation Instructions

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
BANDL #u4, @Ri	C	80	1+2a	----	(Ri)&=(0xF0+u4)	○	The subordinate position four bits are operated.
BANDH #u4, @Ri	C	81	1+2a	----	(Ri)&=((u4<<4)+0x0F)	○	The high rank four bits are operated.
*BAND #u8, @Ri ^{*1}				----	(Ri)&=u8	-	
BORL #u4, @Ri	C	90	1+2a	----	(Ri) = u4	○	The subordinate position four bits are operated.
BORH #u4, @Ri	C	91	1+2a	----	(Ri) = (u4<<4)	○	The high rank four bits are operated.
*BOR #u8, @Ri ^{*2}				----	(Ri) = u8	-	
BEORL #u4, @Ri	C	98	1+2a	----	(Ri) ^= u4	○	The subordinate position four bits are operated.
BEORH #u4, @Ri	C	99	1+2a	----	(Ri) ^= (u4<<4)	○	The high rank four bits are operated.
*BEOR #u8, @Ri ^{*3}				----	(Ri) ^= u8	-	
BTSTL #u4, @Ri	C	88	2+a	0C--	(Ri) & u4	-	The subordinate position four bits are tested.
BTSTH #u4, @Ri	C	89	2+a	CC--	(Ri) & (u4<<4)	-	The high rank four bits are tested.

*1: The assembler generates BANDL or BANDH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BANDL and BANDH are occasionally generated.

*2: The assembler generates BORL or BORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BORL and BORH are occasionally generated.

*3: The assembler generates BEORL or BEORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BEORL and BEORH are occasionally generated.

Table E-5 Multiplication and Division

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
MUL Rj,Ri	A	AF	5	CCC-	Ri * Rj → MDH,MDL	32bit × 32bit=64bit
MULU Rj,Ri	A	AB	5	CCC-	Ri * Rj → MDH,MDL	Unsigned
MULH Rj,Ri	A	BF	3	CC--	Ri * Rj → MDL	16bit × 16bit=32bit
MULUH Rj,Ri	A	BB	3	CC--	Ri * Rj → MDL	Unsigned
DIV0S Ri	E	97-4	1	----		Step operation
DIV0U Ri	E	97-5	1	----		32bit/32bit=32bit
DIV1 Ri	E	97-6	d	-C-C		
DIV2 Ri	E	97-7	1	-C-C		
DIV3	E	9F-6	1	----		
DIV4S	E	9F-7	1	----		
*DIV Ri *1			36	-C-C	MDL / Ri → MDL , MDL % Ri → MDH	
*DIVU Ri *2				-C-C	MDL / Ri → MDL , MDL % Ri → MDH	

Table E-6 Shift

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	Ri << Rj → Ri	Logical shift
*LSL #u5, Ri(u5:0 to 31)	C'	B4	1	CC-C	Ri << u5 → Ri	
LSL #u4, Ri	C	B4	1	CC-C	Ri << u4 → Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	Ri <<(u4+16) → Ri	
LSR Rj, Ri	A	B2	1	CC-C	Ri >> Rj → Ri	Logical shift
*LSR #u5, Ri(u5:0 to 31)	C'	B0	1	CC-C	Ri >> u5 → Ri	
LSR #u4, Ri	C	B0	1	CC-C	Ri >> u4 → Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	Ri >>(u4+16) → Ri	
ASR Rj, Ri	A	BA	1	CC-C	Ri >> Rj → Ri	Arithmetic shift
*ASR #u5, Ri (u5:0 to 31)	C'	B8	1	CC-C	Ri >> u5 → Ri	
ASR #u4, Ri	C	B8	1	CC-C	Ri >> u4 → Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	Ri >>(u4+16) → Ri	

Table E-7 Value Move Operation of Value Sets/16 bits/32 Bits Immediately

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LDI:32 #i32, Ri	E	9F-8	3	----	i32 → Ri	0 expansions of high rank 12bit.
LDI:20 #i20, Ri	C	9B	2	----	i20 → Ri	0 expansions of high rank 24bit.
LDI:8 #i8, Ri	B	C0	1	----	i8 → Ri	
*LDI # {i8 i20 i32} ,Ri *3					{i8 i20 i32} → Ri	

*1: DIV0S, DIV1x32, DIV2, DIV3, and DIV4S are generated. The instruction code length becomes 72 bytes.

*2: DIV0S, DIV1x32 are generated. The instruction code length becomes 66 bytes.

*3: When the immediate value is an absolute value, i8, i20, or i32 is selected automatically by the assembler. When the immediate value is a relative value or includes an externally referenced symbol, i32 is selected.

Table E-8 Memory Loading

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LD @Rj, Ri	A	04	b	----	(Rj)→ Ri	
LD @(R13,Rj), Ri	A	00	b	----	(R13+Rj)→ Ri	
LD @(R14,disp10), Ri	B	2	b	----	(R14+disp10)→ Ri	
LD @(R15,udisp6), Ri	C	03	b	----	(R15+udisp6)→ Ri	
LD @R15+, Ri	E	07-0	b	----	(R15)→ Ri, R15+=4	
LD @R15+, Rs	E	07-8	b	----	(R15)→ Rs, R15+=4	Rs: Special register
LD @R15+, PS	E	07-9	1+a+b	CCCC	(R15)→ PS, R15+=4	*
LDUH @Rj, Ri	A	05	b	----	(Rj)→ Ri	0 expansions
LDUH @(R13,Rj), Ri	A	01	b	----	(R13+Rj)→ Ri	0 expansions
LDUH @(R14,disp9), Ri	B	4	b	----	(R14+disp9)→ Ri	0 expansions
LDUB @Rj, Ri	A	06	b	----	(Rj)→ Ri	0 expansions
LDUB @(R13,Rj), Ri	A	02	b	----	(R13+Rj)→ Ri	0 expansions
LDUB @(R14,disp8), Ri	B	6	b	----	(R14+disp8)→ Ri	0 expansions

* : In the hard-spec o8 and o4 fields, the assembler sets values through the following calculation:

disp10/4→ o8, disp9/2→ o8, disp8→ o8, disp10, disp9, disp8: signed. udisp6/4→ o4udisp6: unsigned.

Table E-9 Memory Store

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
ST Ri, @Rj	A	14	a	----	Ri→ (Rj)	word
ST Ri, @(R13,Rj)	A	10	a	----	Ri→ (R13+Rj)	word
ST Ri, @(R14,disp10)	B	3	a	----	Ri→ (R14+disp10)	word
ST Ri, @(R15,udisp6)	C	13	a	----	Ri→ (R15+udisp6)	
ST Ri, @-R15	E	17-0	a	----	R15-=4, Ri→ (R15)	
ST Rs, @-R15	E	17-8	a	----	R15-=4, Rs→ (R15)	Rs: Special register
ST PS, @-R15	E	17-9	a	----	R15-=4, PS→ (R15)	*
STH Ri, @Rj	A	15	a	----	Ri→ (Rj)	half-word
STH Ri, @(R13,Rj)	A	11	a	----	Ri→ (R13+Rj)	half-word
STH Ri, @(R14,disp9)	B	5	a	----	Ri→ (R14+disp9)	half-word
STB Ri, @Rj	A	16	a	----	Ri→ (Rj)	Bytes
STB Ri, @(R13,Rj)	A	12	a	----	Ri→ (R13+Rj)	Bytes
STB Ri, @(R14,disp8)	B	7	a	----	Ri→ (R14+disp8)	Bytes

*: In the hard-spec o8 and o4 fields, the assembler sets values through the following calculation:

disp10/4→ o8, disp9/2→ o8, disp8→ o8, disp10, disp9, disp8: signed. udisp6/4→ o4udisp6: unsigned.

Table E-10 Transfer between Registers

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	Rj → Ri	Transfer between general-purpose registers
MOV Rs, Ri	A	B7	1	----	Rs → Ri	Rs: Special register
MOV Ri, Rs	E	B3	1	----	Ri → Rs	Rs: Special register
MOV PS, Ri	E	17-1	1	----	PS → Ri	*
MOV Ri, PS	E	07-1	c	CCCC	Ri → PS	

* : Special register Rs: TBR,RP,USP,SSP,MDH,MDL

Table E-11 Normal Branch (without delay)

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
JMP @Ri	E	97-0	2	----	Ri → PC	
CALL label12	E	D0	2	----	PC+2→ RP,PC+2+(label12-PC-2)→ PC	
CALL @Ri	F	97-1	2	----	PC+2→ RP,Ri→ PC	
RET	E	97-2	2	----	RP → PC	Return
INT #u8	D	1F	3+3a	----	SSP-=4,PS→ (SSP),SSP-=4,PC+2→ (SSP), 0→ I flag, 0→ S flag, (TBR+0x3FC-u8× 4)→ PC	
INTE	E	9F-3	3+3a	----	SSP-=4,PS→ (SSP),SSP-=4,PC+2→ (SSP), 0→ S Flag,(TBR+10x3D8)→ PC	
					-	for emulator
RETI	E	97-3	2+2A	CCCC	(R15)→ PC,R15-=4,(R15)→ PS,R15-=4	
BRA label9	D	E0	2	----	PC+2+(label9-PC-2)→ PC	
BNO label9	D	E1	1	----	Point-to-point	
BEQ label9	D	E2	2/1	----	if(Z==1) then PC+2+(label9-PC-2)→ PC	
BNE label9	D	E3	2/1	----	↑ s/Z==0	
BC label9	D	E4	2/1	----	↑ s/C==1	
BNC label9	D	E5	2/1	----	↑ s/C==0	
BN label9	D	E6	2/1	----	↑ s/N==1	
BP label9	D	E7	2/1	----	↑ s/N==0	
BV label9	D	E8	2/1	----	↑ s/V==1	
BNV label9	D	E9	2/1	----	↑ s/V==0	
BLT label9	D	EA	2/1	----	↑ s/V xor N==1	
BGE label9	D	EB	2/1	----	↑ s/V xor N==0	
BLE label9	D	EC	2/1	----	↑ s/(V xor N) or Z==1	
BGT label9	D	ED	2/1	----	↑ s/(V xor N) or Z==0	
BLS label9	D	EE	2/1	----	↑ s/C or Z==1	
BHI label9	D	EF	2/1	----	↑ s/C or Z==0	

Notes : • "2/1" in the CYCLE column means "2" when a branch occurs or "1" when no branch occurs.

- In the hard-spec rel11 and rel8 fields, the assembler sets values through the following calculation:
(label12-PC-2)/2→ rel11, (label9-PC-2)/2→ rel8, label12,label9 are signed.
- The execution of the RETI instruction assumes that the S flag is "0".

Table E-12 Delayed Branch

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
JMP:D @Ri	E	9F-0	1	----	Ri → PC	
CALL:D label12	F	D8	1	----	PC+4→ RP, PC+2+(label12-PC-2)→ PC	
CALL:D @Ri	E	9F-1	1	----	PC+4→ RP, Ri→ PC	
RET:D	E	9F-2	1	----	RP → PC	Return
BRA:D label9	D	F0	1	----	PC+2+(label9-PC-2)→ PC	
BNO:D label9	D	F1	1	----	Point-to-point	
BEQ:D label9	D	F2	1	----	if(Z==1) then PC+2+(label9-PC-2)→ PC	
BNE:D label9	D	F3	1	----	↑ s/Z==0	
BC:D label9	D	F4	1	----	↑ s/C==1	
BNC:D label9	D	F5	1	----	↑ s/C==0	
BN:D label9	D	F6	1	----	↑ s/N==1	
BP:D label9	D	F7	1	----	↑ s/N==0	
BV:D label9	D	F8	1	----	↑ s/V==1	
BNV:D label9	D	F9	1	----	↑ s/V==0	
BLT:D label9	D	FA	1	----	↑ s/V xor N==1	
BGE:D label9	D	FB	1	----	↑ s/V xor N==0	
BLE:D label9	D	FC	1	----	↑ s/(V xor N) or Z==1	
BGT:D label9	D	FD	1	----	↑ s/(V xor N) or Z==0	
BLS:D label9	D	FE	1	----	↑ s/C or Z==1	
BHI:D label9	D	FF	1	----	↑ s/C or Z==0	

- Notes :
- In the hard-spec rel11 and rel8 fields, the assembler sets values through the following calculation:
(label12-PC-2)/2→ rel11, (label9-PC-2)/2→ rel8, label12, label9 are signed.
 - A delayed branch always takes place after the instruction that follows (delay slot) is executed.
 - All of 1-cycle, a-, b-, c-, and d-cycle instructions can be placed in the delay slot.
Two or more-cycle instruction cannot be put.

Table E-13 Others

Mnemonic	Type	OP	CYCLE	NZVC	Operation	RMW	Remarks
NOP	E	9F-A	1	----	Anything does not change either.	-	
ANDCCR #u8	D	83	c	cccc	CCR and u8 → CCR	-	
ORCCR #u8	D	93	c	cccc	CCR or u8 → CCR	-	
STILM #u8	D	87	1	----	i8 → ILM	-	Value set of ILM immediately
ADDSP #s10 ^{*1}	D	A3	1	----	R15 += s10	-	ADD SP instruction
EXTSB Ri	E	97-8	1	----	Sign extension 8→ 32bit	-	
EXTUB Ri	E	97-9	1	----	0 expansions 8→ 32bit	-	
EXTSH Ri	E	97-A	1	----	Sign extension 16→ 32bit	-	
EXTUH Ri	E	97-B	1	----	0 expansions 16→ 32bit	-	
LDM0 (reglist)	D	8C	^{*5}	----	(R15)→ reglist, R15 increment	-	loading multi R0-R7
LDM1 (reglist)	D	8D	^{*5}	----	(R15)→ reglist, R15 increment	-	loading multi R8-R15
*LDM (reglist) ^{*2}				----	(R15)→ reglist, R15 increment	-	loading multi R0-R15
STM0 (reglist)	D	8E	^{*6}	----	R15 decrement, reglist→ (R15)	-	store multi R0-R7
STM1 (reglist)	D	8F	^{*6}	----	R15 decrement, reglist→ (R15)	-	store multi R8-R15
*STM (reglist) ^{*3}				----	R15 decrement, reglist→ (R15)	-	store multi R0-R15
ENTER #u10 ^{*4}	D	0F	1+a	----	R14 → (R15 - 4), R15 - 4 → R14, R15 - u10 → R15	-	Entrance processing of function
LEAVE	E	9F-9	b	----	R14 + 4 → R15, (R15 - 4) → R14	-	Exit processing of function
XCHB @Rj, Ri	A	8A	2a	----	Ri → TEMP (Rj) → Ri TEMP → (Rj)	○	For semaphore control Byte data

*1 : S10 calculates s10/4 by the assembler, makes to s8, and sets the value. s10 is signed.

*2 : If reglist specifies any of R0 to R7, LDM0 is generated. If it specifies any of R8 to R15, LDM1 is generated. Both LDM0 and LDM1 are occasionally generated.

*3 : If reglist specifies any of R0 to R7, STM0 is generated. If it specifies any of R8 to R15, STM1 is generated. Both STM1 and STM0 are occasionally generated.

*4 : u10 calculates u10/4 by the assembler and the value is set as u8. u10 is unsigned.

*5 : The number of execution cycles for LDM0 (reglist) and LDM1 (reglist) is $a \times (n-1) + b + 1$, where the specified number of registers is n.

*6 : The number of execution cycles for STM0(reglist) and STM1(reglist) is $a \times n + 1$, where the specified number of registers is n.

Table E-14 Usual Branch Macro Instruction of 20bits

Mnemonic	Operation	Remarks
*CALL20 label20,Ri	Address of the following instruction→ RP, label20→ PC	Ri: Temporary register (See Reference 1)
*BRA20 label20,Ri	label20→ PC	Ri: Temporary register (See Reference 2)
*BEQ20 label20,Ri	if(Z==1) then label20→ PC	Ri: Temporary register (See Reference 3)
*BNE20 label20,Ri	↑ s/Z==0	↑
*BC20 label20,Ri	↑ s/C==1	↑
*BNC20 label20,Ri	↑ s/C==0	↑
*BN20 label20,Ri	↑ s/N==1	↑
*BP20 label20,Ri	↑ s/N==0	↑
*BV20 label20,Ri	↑ s/V==1	↑
*BNV20 label20,Ri	↑ s/V==0	↑
*BLT20 label20,Ri	↑ s/V xor N==1	↑
*BGE20 label20,Ri	↑ s/V xor N==0	↑
*BLE20 label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20 label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20 label20,Ri	↑ s/C or Z==1	↑
*BHI20 label20,Ri	↑ s/C or Z==0	↑

Reference 1: CALL20

(1) When label20-PC-2 is -0x800 to +0x7fe, the instruction is generated as follows.

```
CALL    label12
```

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
LDI:20  #label20,Ri
CALL    @Ri
```

Reference 2: BRA20

(1) When label20-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

```
BRA     label9
```

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
LDI:20  #label20,Ri
JMP      @Ri
```

Reference 3: Bcc20

(1) When label20-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

```
Bcc      label9
```

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
Bxcc      false      xcc is a contradiction condition of cc.
LDI:20    #label20,Ri
JMP        @Ri
false:
```

Table E-15 20-bit Delay Divergence Macro Instruction

Mnemonic	Operation	Remarks
*CALL20:D label20,Ri	Address of the following instruction+2→ RP, label20→ PC	Ri: Temporary register (See Reference 1)
*BRA20:D label20,Ri	label20→ PC	Ri: Temporary register (See Reference 2)
*BEQ20:D label20,Ri	if(Z==1) then label20→ PC	Ri: Temporary register (See Reference 3)
*BNE20:D label20,Ri	↑ s/Z==0	↑
*BC20:D label20,Ri	↑ s/C==1	↑
*BNC20:D label20,Ri	↑ s/C==0	↑
*BN20:D label20,Ri	↑ s/N==1	↑
*BP20:D label20,Ri	↑ s/N==0	↑
*BV20:D label20,Ri	↑ s/V==1	↑
*BNV20:D label20,Ri	↑ s/V==0	↑
*BLT20:D label20,Ri	↑ s/V xor N==1	↑
*BGE20:D label20,Ri	↑ s/V xor N==0	↑
*BLE20:D label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20:D label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20:D label20,Ri	↑ s/C or Z==1	↑
*BHI20:D label20,Ri	↑ s/C or Z==0	↑

Reference 1: CALL20:D

(1) When label20-PC-2 is -0x800 to +0x7fe, the instruction is generated as follows.

CALL:D label12

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

LDI:20 #label20,Ri

CALL:D @Ri

Reference 2: BRA20:D

(1) When label20-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

BRA:D label9

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

LDI:20 #label20,Ri

JMP:D @Ri

Reference 3: Bcc20:D

(1) When label20-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

Bcc:D label9

(2) When label20-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

Bxcc false xcc is a contradiction condition of cc.

LDI:20 #label20,Ri

JMP:D @Ri

false:

Table E-16 Usual Branch Macro Instruction of 32bits

Mnemonic	Operation	Remarks
*CALL32label32,Ri	Address of the following instruction→ RP, label32→ PC	Ri: Temporary register (See Reference 1)
*BRA32 label32,Ri	label32→ PC	Ri: Temporary register (See Reference 2)
*BEQ32 label32,Ri	if(Z==1) then label32→ PC	Ri: Temporary register (See Reference 3)
*BNE32 label32,Ri	↑ s/Z==0	↑
*BC32 label32,Ri	↑ s/C==1	↑
*BNC32 label32,Ri	↑ s/C==0	↑
*BN32 label32,Ri	↑ s/N==1	↑
*BP32 label32,Ri	↑ s/N==0	↑
*BV32 label32,Ri	↑ s/V==1	↑
*BNV32 label32,Ri	↑ s/V==0	↑
*BLT32 label32,Ri	↑ s/V xor N==1	↑
*BGE32 label32,Ri	↑ s/V xor N==0	↑
*BLE32 label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32 label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32 label32,Ri	↑ s/C or Z==1	↑
*BHI32 label32,Ri	↑ s/C or Z==0	↑

Reference 1: CALL32

(1) When label32-PC-2 is -0x800 to +0x7fe, the instruction is generated as follows.

```
CALL    label12
```

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
LDI:32  #label32,Ri
CALL    @Ri
```

Reference 2: BRA32

(1) When label32-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

```
BRA     label9
```

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
LDI:32  #label32,Ri
JMP     @Ri
```

Reference 3: Bcc32

(1) When label32-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

```
Bcc     label9
```

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

```
Bxcc    false      xcc is a contradiction condition of cc.
LDI:32  #label32,Ri
JMP     @Ri
false:
```


Table E-17 32-bit Delay Delayed Branch Macro Instruction

Mnemonic	Operation	Remarks
*CALL32D label32,Ri	Address of the following instruction+2→ RP, label32→ PC	Ri: Temporary register (See Reference 1)
*BRA32:D label32,Ri	label32→ PC	Ri: Temporary register (See Reference 2)
*BEQ32:D label32,Ri	if(Z==1) then label32→ PC	Ri: Temporary register (See Reference 3)
*BNE32:D label32,Ri	↑ s/Z==0	↑
*BC32:D label32,Ri	↑ s/C==1	↑
*BNC32:D label32,Ri	↑ s/C==0	↑
*BN32:D label32,Ri	↑ s/N==1	↑
*BP32:D label32,Ri	↑ s/N==0	↑
*BV32:D label32,Ri	↑ s/V==1	↑
*BNV32:D label32,Ri	↑ s/V==0	↑
*BLT32:D label32,Ri	↑ s/V xor N==1	↑
*BGE32:D label32,Ri	↑ s/V xor N==0	↑
*BLE32:D label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32:D label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32:D label32,Ri	↑ s/C or Z==1	↑
*BHI32:D label32,Ri	↑ s/C or Z==0	↑

Reference 1: CALL32:D

(1) When label32-PC-2 is -0x800 to +0x7fe, the instruction is generated as follows.

CALL:D label12

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

LDI:32 #label32,Ri

CALL:D @Ri

Reference 2: BRA32:D

(1) When label32-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

BRA:D label9

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

LDI:32 #label32,Ri

JMP:D @Ri

Reference 3: Bcc32:D

(1) When label32-PC-2 is -0x100 to +0xfe, the instruction is generated as follows.

Bcc:D label9

(2) When label32-PC-2 falls outside the range in (1) or contains an externally referenced symbol, the following instructions are generated:

Bxcc false xcc is a contradiction condition of cc.

LDI:32 #label32,Ri

JMP:D @Ri

false:

Table E-18 Direct Addressing

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	(dir10)→ R13	word
DMOV R13, @dir10	D	18	a	----	R13 → (dir10)	word
DMOV @dir10, @R13+	D	0C	2a	----	(dir10)→ (R13),R13+=4	word
DMOV @R13+, @dir10	D	1C	2a	----	(R13)→ (dir10),R13+=4	word
DMOV @dir10, @-R15	D	0B	2a	----	R15-=4,(R15)→ (dir10)	word
DMOV @R15+, @dir10	D	1B	2a	----	(R15)→ (dir10),R15+=4	word
DMOVH @dir9, R13	D	09	b	----	(dir9)→ R13	half-word
DMOVH R13, @dir9	D	19	a	----	R13 → (dir9)	half-word
DMOVH @dir9, @R13+	D	0D	2a	----	(dir9)→ (R13),R13+=2	half-word
DMOVH @R13+, @dir9	D	1D	2a	----	(R13)→ (dir9),R13+=2	half-word
DMOV B @dir8, R13	D	0A	b	----	(dir8)→ R13	Bytes
DMOV B R13, @dir8	D	1A	a	----	R13 → (dir8)	Bytes
DMOV B @dir8, @R13+	D	0E	2a	----	(dir8)→ (R13),R13++	Bytes
DMOV B @R13+, @dir8	D	1E	2a	----	(R13)→ (dir8),R13++	Bytes

Note : The assembler calculates in dir8, dir9, and the dir10 field as follows and the value is set.
 Dir8 →dir, dir9/2 →dir, dir10/4 →dir
 dir8, dir9, and dir10 are unsigned.

Table E-19 Resource Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
LDRES @Ri+, #u4	C	BC	a	----	(Ri)→ Resource of u4 Ri+=4	u4: Channel number
STRES #u4, @Ri+	C	BD	a	----	Resource of u4 → (Ri) Ri+=4	u4: Channel number

Note : Not available to this model as it has no channel-numbered resource.

Table E-20 Coprocessor Control Instruction

{CRi|CRj} := CR0|CR1|CR2|CR3|CR4|CR5|CR6|CR7|CR8|CR9|CR10|CR11|CR12|CR13|CR14|CR15|

u4: channel specification

u8: command specification

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remarks
COPOP #u4, #u8, CRj, Cri	E	9F-C	2+a	----	Operation instruction	No error trap
COPLD #u4, #u8, Rj, Cri	E	9F-D	1+2a	----	Rj → CRi	
COPST #u4, #u8, CRj, Ri	E	9F-E	1+2a	----	CRj → Ri	
COPSV #u4, #u8, CRj, Ri	E	9F-F	1+2a	----	CRj → Ri	

Note : Not available to this model as it contains no coprocessor.

APPENDIX F Precautions when Using

This appendix describes precautions when using the MB91470/480 series.

■ Common Matter

● Clock control block

For "L" input to INITX, reserve the oscillation stabilization wait time.

● Switching shared ports between functions

Switching between a port and a shared pin relies on the PFR (port function register). Note, however, that bus pins are switched depending on the external bus setting.

● D-bus Memory

Do not set a code area in memory on the D-bus.

No instruction fetch applies to the D-bus. Applying an instruction fetch to the D-bus area causes wrong data to be interpreted as code, involving the risk of running out of control.

● Low-power Consumption Mode

- To place the device in standby mode, use the synchronous standby mode (set with bit8 (SYNCS bit) of the time-base counter control register, TBCR) and be sure to use the following sequence:

```
-----  
// -- STCR write  
LDI      #_STCR,R0          // STCR register (0x0481)  
LDI      #value_of_standby, R1 // value_of_standby is the data to write to STCR.  
STB      R1,@R0             // Write to STCR  
  
// -- CTBR write  
LDI      #_CTBR,R2          // CTBR register (0x0483)  
LDI      #0xA5,R1           // Clear command (1)  
STB      R1,@R2             // Write A5H to CTBR  
LDI      #0x5A,R1           // Clear command (2)  
STB      R1,@R2             // Write 5AH to CTBR(Clear time-base counter)  
  
LDUB     @R0,R1              // Read STCR (Synchronous standby transition starting)  
LDUB     @R0,R1              // Read dummy STCR  
NOP      // NOP for timing adjusting × 5  
NOP  
NOP  
NOP  
NOP  
-----
```

- Please do not do the following when the monitor debugger is used;
 - Please do not set the break point to the above-mentioned instruction row.
 - Please do not execute the step for the above-mentioned instruction row.

● Notes of PS register

As the PS register is processed speculatively for some instructions, the interrupt service routine may break or the PS flag display may be updated when the debugger is used if exceptions are handled as follows: The device is designed such that processing is performed again correctly after recovery from EIT in either case. Appropriate actions are therefore taken before and after EIT as designed.

- The following operations may occur when (a) user interrupt/NMI is received, (b) step execution is performed, (c) break occurs in a data event or emulator menu in an immediately preceding DIV0U/DIV0S instruction.
 - (1) D0 and D1 flags precede and are renewed.
 - (2) EIT processing routine (user interruption, NMI or emulator) is executed.
 - (3) After returning from EIT, DIV0U/DIV0S instructions are executed and the D0 and D1 flags are updated to the same value as (1).
- When each ORCCR/STILM/MOV Ri and PS instruction is executed to permit interrupting with the user interruption and the NMI factor generated, the following operations are done.
 - (1) The PS register precedes and is updated.
 - (2) Execute an EIT processing routine (user interrupt or NMI).
 - (3) After returning from EIT, the above instructions are executed and the PS register is updated to the same value as (1).

● About watchdog timer function

The watchdog timer integrated in MB91470/480 series monitors the program to check that it delays a reset within a certain period of time and, if the program runs out of control and fails to delay the reset, resets the CPU in place. Once the watchdog timer is enabled, it keeps running until reset.

As an exception, the watchdog timer delays the reset automatically when a condition which stops program execution by the CPU develops. For those conditions which correspond to this exception, see the description of [Temporarily halting the watchdog timer (automatically postpone generation)] in "3.11.8 Peripheral Circuit Functions in the Clock Controller ■ Time-base Counter ● Watchdog timer"

■ Notes on Debugging

● Single stepping of RETI instructions

In an environment where interrupts frequently occur during single-step execution, only the relevant interrupt processing routines are executed repeatedly during single-step execution of the RETI instruction. As a result, a low program of the main routine and interrupt levels is not executed.

To avoid it, do not single-step RETI instructions. When the relevant interrupt routine no longer requires being debugged, disable the relevant interrupt and perform debugging.

● Operand break

Do not apply a data even break to access of the area containing the address of a system stack pointer.

● Execution of an unused area of Flash memory

If an unused area (data at 0xFFFF_B) of Flash memory is executed accidentally, no break can be accepted. To prevent this, it is recommended to use the code event address mask feature of the debugger to break at instruction access to the unused area.

● Interrupt handler for NMI request (tool)

Add the following program to the interrupt handler to prevent the device from malfunctioning when the source flag is set accidentally with no ICE connected, for example, due to noise to the DSU pin, which is to be set only at the break request of the ICE.

Note that the ICE can be used normally with this program added.

Added place

Next interrupt handler

Interrupt source	: NMI request (tool)
Interrupt number	: 13 (decimal), 0D (Hexadecimal)
Offset	: 3C8 _H
TBR default address	: 000FFFC8 _H

Added program

```

STM    (R0, R1)
LDI    #B00H,R0    ; B00H is the address of the DSUs break source register.
LDI    #0, R1
STB    R1,@R0      ; Clear the break source register.
LDM    (R0, R1)
RETI

```

INDEX

**The index follows on the next page.
This is listed in alphabetic order.**

Index

Numerics

12-bit A/D Converter	
12-bit A/D Converter	
(successive approximation type)	4
Block Diagram of 12-bit A/D Converter	734
Function of 12-bit A/D Converter	732
Interrupt of 12-bit A/D Converter	750
Notes on Using 12-bit A/D Converter.....	758
Pins of 12-bit A/D Converter	737
Register List of 12-bit A/D Converter.....	738
16/32-bit PWC	
16/32-bit PWC Timer	423
Block Diagram of 16/32-bit PWC Timer	
(ch.1, ch.0)	426
16/32-bit Reload Timer	
16/32-bit Reload Timer	423
Block Diagram of 16/32-bit Reload Timer	
(ch.1, ch.0)	425
16-bit Dead Timer Control Register	
16-bit Dead Timer Control Register, Lower Byte	
(DTCR1/DTCR4).....	348
16-bit Dead Timer Control Register, Upper Byte	
(DTCR0/DTCR3).....	346
16-bit Dead Timer Control Register, Upper Byte	
(DTCR2/DTCR5).....	350
16-bit Dead Timer Register	
16-bit Dead Timer Register (TMRRH0 to TMRRH5,	
TMRRL0 to TMRRL5).....	345
16-bit Free-run Timer	
16-bit Free-run Timer Interrupt.....	364
16-bit Free-run Timer Register	303
Block Diagram of 16-bit Free-run Timer	288
Example Program for the 16-bit Free-run	
Timer	417
Notes on Using the 16-bit Free-run Timers	415
16-bit Input Capture	
16-bit Input Capture Interrupt	365
16-bit Input Capture Operation	392
Block Diagram of the 16-bit Input Capture	293
Cautions for Use of 16-bit Input Capture	415
Input Timing for 16-bit Input Capture	393
16-bit Input Capture Register	
16-bit Input Capture Register.....	307
16-bit Output Compare	
16-bit Output Compare Interrupt.....	365
16-bit Output Compare Operation	
(Inversion Mode, MOD1x=0).....	379
16-bit Output Compare Operation	
(Set/Reset Mode, MOD1x=1).....	383
16-bit Output Compare Timing.....	384
Block Diagram of the 16-bit Output	
Compare	290
Example Program for the 16-bit Output	
Compare	418
Notes on Using the 16-bit Output Compare	415
Operation of 16-bit Output Compare and Free-run	
Timer.....	385
16-bit Output Compare Register	
16-bit Output Compare Register	306
16-bit PPG	
16-bit PPG Timer.....	423
Block Diagram of 16-bit PPG Timer.....	424
Reload Registers: 16-bit PPG Mode.....	266
16-bit PWM Timer	
16-bit PWM Timer.....	422
Block Diagram of 16-bit PWM Timer.....	424
16-bit PWM/PPG/Reload Timer	
Notes on Using the 16-bit PWM/PPG/Reload	
Timer.....	435
16-bit Reload Register	
Bit Configuration of 16-bit Reload Register	
(TMRLR).....	241
16-bit Reload Timer	
16-bit Reload Timer	3
Block Diagram of the 16-bit Reload Timer	236
Overview of 16-bit Reload Timer	236
16-bit Reload Timer Register	
List of 16-bit Reload Timer Registers	237
16-bit Timer Register	
Bit Configuration of 16-bit Timer Register	
(TMR)	240
1st Byte	
Receiving Acknowledge after Transmitting	
1st Byte.....	661
2-Cycle Transfer	
Burst 2-Cycle Transfer	808
Flow of Data During 2-Cycle Transfer	828
Step/Block Transfer 2-Cycle Transfer	809
32-bit↔16-bit Bus Converter	
32-bit↔16-bit Bus Converter	45
32-bit Mode	
Functions of 32-bit Mode.....	433
Operations in 32-bit Mode	434
Setting the 32-bit Mode	433
7-bit Slave Address Mask Register	
7-bit Slave Address Mask Register (ISMK)	644
7-bit Slave Address Register	
7-bit Slave Address Register (ISBA).....	645

MB91470/480 Series

8/10-bit A/D Converter	
8/10-bit A/D Converter	
(Successive Comparison Type).....	4
Block Diagram of 8/10-bit A/D Converter	701
Function of 8/10-bit A/D Converter	698
Interrupt of 8/10-bit A/D Converter	718
Notes on Using 8/10-bit A/D	
Converter	725
Pins of 8/10-bit A/D Converter.....	704
Register List of 8/10-bit A/D Converter	705
8/16-bit PPG	
8/16-bit PPG Timer	3
8/16-bit Up-Down Counter	
8/16-bit Up-Down Counter	3
8-bit Counter	
8-bit Counter	255
8-bit PPG	
Block Diagram of the 8-bit PPG ch.0, ch.2, ch.4, ch.6,	
ch.8, ch.10, ch.12 and ch.14	260
Block Diagram of the 8-bit PPG ch.1, ch.5, ch.9 and	
ch.13.....	261
Block Diagram of the 8-bit PPG ch.3, ch.7, ch.11 and	
ch.15.....	262
8-bits and 16-bit	
Operations for 8-bits and 16-bit	515

A

A/D

A/D Activation	406
A/D Activation by Free-run Timer	375
A/D Activation by Zero Detection of Free-run Timer	
or Compare Clear	408
A/D Activation Compare Register	309
A/D Channel Control Register	
(ADCH: ADCH0 to ADCH2)	707
A/D Channel Control Register	
(ADCH: ADCH3, ADCH4)	739
A/D Compare Activation Enabled	406
A/D Compare Activation Mode	407
A/D Control Status Register	
(ADCS: ADCS1, ADCS2)	712
A/D Control Status Register	
(ADCS: ADCS3, ADCS4)	744
A/D Conversion Data Protection	
Function	723, 754
A/D Data Register	
(ADCD: ADCD000 to ADCD030,	
ADCD001 to ADCD031, ADCD002 to	
ADCD112).....	715
A/D Data Register (ADCD: ADCD003 to ADCD033,	
ADCD004 to ADCD034).....	747
A/D Mode Setting Register	
(ADMD: ADMD0 to ADMD2)	709
A/D Mode Setting Register	
(ADMD: ADMD3, ADMD4).....	741

A/D Trigger Control Register	
(ADTRGC0 to ADTRGC5)	320
Block Diagram of 12-bit A/D Converter	734
Block Diagram of 8/10-bit A/D Converter.....	701
Block Diagram of the A/D Activation	
Compare	297
Function of 12-bit A/D Converter.....	732
Function of 8/10-bit A/D Converter.....	698
Interrupt of 12-bit A/D Converter	750
Interrupt of 8/10-bit A/D Converter	718
Notes on Using 12-bit A/D Converter	758
Notes on Using 8/10-bit A/D	
Converter	725
Notes on Using the A/D Activation	
Compare	416
Pins of 12-bit A/D Converter	737
Pins of 8/10-bit A/D Converter	704
Register List of 12-bit A/D Converter	738
Register List of 8/10-bit A/D Converter	705
A/D Activation Compare	
Block Diagram of the A/D Activation	
Compare	297
Notes on Using the A/D Activation	
Compare	416
A/D Activation Compare Register	
A/D Activation Compare Register	309
A/D Channel Control Register	
A/D Channel Control Register	
(ADCH: ADCH0 to ADCH2)	707
A/D Channel Control Register	
(ADCH: ADCH3, ADCH4)	739
A/D Compare	
A/D Compare Activation Enabled	406
A/D Compare Activation Mode	407
A/D Control Status Register	
A/D Control Status Register	
(ADCS: ADCS1, ADCS2).....	712
A/D Control Status Register	
(ADCS: ADCS3, ADCS4).....	744
A/D Conversion	
A/D Conversion Data Protection	
Function	723, 754
A/D Converter	
12-bit A/D Converter	
(successive approximation type).....	4
8/10-bit A/D Converter	
(Successive Comparison Type)	4
Block Diagram of 12-bit A/D Converter	734
Block Diagram of 8/10-bit A/D Converter.....	701
Function of 12-bit A/D Converter.....	732
Function of 8/10-bit A/D Converter.....	698
Interrupt of 12-bit A/D Converter	750
Interrupt of 8/10-bit A/D Converter	718
Notes on Using 12-bit A/D Converter	758
Notes on Using 8/10-bit A/D	
Converter	725

Pins of 12-bit A/D Converter	737	A/D Channel Control Register	
Pins of 8/10-bit A/D Converter	704	(ADCH: ADCH3, ADCH4)	739
Register List of 12-bit A/D Converter.....	738	ADCOMP	
Register List of 8/10-bit A/D Converter.....	705	Compare Register 0/3,1/4,2/5 (ADCOMP0/ ADCOMP3, ADCOMP1/ADCOMP4, ADCOMP2/ADCOMP5)	357
A/D Data Register		ADCOMPB	
A/D Data Register		Compare Buffer Register 0/3,1/4,2/5 (ADCOMPB0/ ADCOMPB3, ADCOMPB1/ADCOMPB4, ADCOMPB2/ADCOMPB5)	356
(ADCD: ADCD000 to ADCD030, ADCD001 to ADCD031, ADCD002 to ADCD112)	715	ADCS	
A/D Data Register (ADCD: ADCD003 to ADCD033, ADCD004 to ADCD034)	747	A/D Control Status Register	
A/D Mode Setting Register		(ADCS: ADCS1, ADCS2)	712
A/D Mode Setting Register		A/D Control Status Register	
(ADMD: ADMD0 to ADMD2)	709	(ADCS: ADCS3, ADCS4)	744
A/D Mode Setting Register		Addition	
(ADMD: ADMD3, ADMD4)	741	Multiplication and Addition Calculator	4
A/D Trigger Control Register		Register List of Multiplication and Addition Calculator	766
A/D Trigger Control Register		Address Register	
(ADTRGC0 to ADTRGC5).....	320	Address Register Specifications	812
Acceptable Baud Rate Range		Addressing	
Acceptable Baud Rate Range for Reception	563	Addressing	853
Access Address		Direct Addressing Area	40, 57
Access Address	813	ADMD	
Access Mode		A/D Mode Setting Register	
Access Mode of Flash Memory	839	(ADMD: ADMD0 to ADMD2)	709
Accumulator		A/D Mode Setting Register	
Accumulator Output	777	(ADMD: ADMD3, ADMD4).....	741
Accumulator Output Registers		ADMD Register Setting	724
(DSP-AC0 to DSP-AC2).....	773	Register ADMD Setting	757
Accumulator Output Register		ADMD Register	
Accumulator Output Registers		ADMD Register Setting	724
(DSP-AC0 to DSP-AC2).....	773	ADTGBUF	
Acknowledge		Buffer Control Register	
Receiving Acknowledge after Transmitting		(ADTGBUF0/ADTGBUF1).....	362
1st Byte	661	ADTGCE	
ACR		Compare Enable Register	
Configuration of Area Configuration Registers		(ADTGCE0/ADTGCE1)	358
(ACR0 to ACR2)	133	ADTGSEL	
Activation Compare		Count Direction Selection (for Comparison) Register	
Block Diagram of the A/D Activation		(ADTGSEL0/ADTGSEL1).....	360
Compare.....	297	ADTRGC	
Activation Compare Register		A/D Trigger Control Register	
A/D Activation Compare Register.....	309	(ADTRGC0 to ADTRGC5)	320
ADCD		AF200	
A/D Data Register		System Configuration for AF200 Flash	
(ADCD: ADCD000 to ADCD030, ADCD001 to ADCD031, ADCD002 to ADCD112)	715	Microcontroller Programmer (Manufactured by Yokogawa Digital Computer Corporation)	871
A/D Data Register (ADCD: ADCD003 to ADCD033, ADCD004 to ADCD034)	747	AICR	
ADCH		Analog Input Control Register	
A/D Channel Control Register		(AICR: AICR0 to AICR2)	717
(ADCH: ADCH0 to ADCH2).....	707		

MB91470/480 Series

Analog Input Control Register (AICR: AICR3, AICR4)	749
Analog Input Control Register	
Analog Input Control Register (AICR: AICR0 to AICR2)	717
Analog Input Control Register (AICR: AICR3, AICR4)	749
Analog Input Pin	
Analog Input Pin in Differential Input Mode	755
Arbitration Lost	
Arbitration Lost Condition	675
Architecture	
Features of Internal Architecture	43
Internal Architecture	44
Area Configuration Registers	
Configuration of Area Configuration Registers (ACR0 to ACR2)	133
Area Select Registers	
Configuration of Area Select Registers (ASR0 to ASR2)	132
Area Wait Registers	
Configuration of Area Wait Registers (AWR0 to AWR2)	139
Arithmetic Function	
Arithmetic Functions	775
Arithmetic Result	
Transfer of Arithmetic Result	776
ASR	
Configuration of Area Select Registers (ASR0 to ASR2)	132
ASRs	
Example of Setting ASRs and ASZ[3:0]	144
Assembler	
Assembler (fasm911)	913
Asynchronous	
Functions of UART (Asynchronous Serial Interface)	521
List of Registers of UART (Asynchronous Serial Interface)	522
ASZ	
Example of Setting ASRs and ASZ[3:0]	144
Automatic Algorithm	
Automatic Algorithm Execution States	840
Command Sequence of Automatic Algorithm	842
Overview of Flash Memory Automatic Algorithms	841
Auto-Wait	
Auto-Wait Timing (TYP[3:0]=0000 _B , AWR=2008 _H)	164
AWR	
Auto-Wait Timing (TYP[3:0]=0000 _B , AWR=2008 _H)	164

Basic Timing (For Successive Accesses) (TYP[3:0]=0000 _B , AWR=0008 _H)	161
Configuration of Area Wait Registers (AWR0 to AWR2)	139
CSX ->RDX/WR0X, WR1X Setup (TYP[3:0]=0101 _B , AWR=100B _H)	170
CSX Delay Setting (TYP[3:0]=0000 _B , AWR=000C _H)	166
External Wait Timing (TYP[3:0]=0001 _B , AWR=2008 _H)	165
Read ->Write Timing (TYP[3:0]=0000 _B , AWR=0048 _H)	162
Setting of CSX ->RDX/WR0X, WR1X Setup and of RDX/WR0X, WR1X ->CSX Hold (TYP[3:0]=0000 _B , AWR=000B _H)	167
With External Wait (TYP[3:0]=0101 _B , AWR=1008 _H)	169
Without External Wait (TYP[3:0]=0100 _B , AWR=0008 _H)	168
Write ->Write Timing (TYP[3:0]=0000 _B , AWR=0018 _H)	163

B

Backup	
Backup/Restore Processing	234
Base Timer	
Base Timer	3
Base Timer Function	438
Basic Programming Model	
Basic Programming Model	48
Basic Timing	
Basic Timing (For Successive Accesses) (TYP[3:0]=0000 _B , AWR=0008 _H)	161
Baud Rate	
Acceptable Baud Rate Range for Reception	563
Baud Rate Selection	681
Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)	617
Calculating the Baud Rate	561, 618, 681
Reload Values and Baud Rates for Different Peripheral Clock Frequencies	562, 619, 682
UART Baud Rate Selection	560
Baud Rate Generator Registers	
Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	538, 590, 646
BGA-144	
Package Dimension (BGA-144P-M06)	14, 15
BGR	
Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	538, 590, 646
Big Endian	
Big Endian Data Format	147
Data Bus Width of Big Endian	149

Bit Ordering		Block Diagram of the Multi-function Timer	286
Bit Ordering	55	Block Diagram of the Waveform Generator	295
Bit Search Module		Block Diagram of Up/Down Counter	500
Bit Search Module (for REALOS)	3	Block Diagram of Wild Register Control	
Bit Search Module Registers	230	Block	874
Block Diagram of Bit Search Module	230	Block Diagrams of Ports	175
Bit Search Module Registers		Flash Memory Block Diagram	833
Bit Search Module Registers	230	Block Size	
Block Diagram		Block Size	810
Block Diagram	763	Block Transfer	
Block Diagram of 12-bit A/D Converter	734	Block Transfer	809
Block Diagram of 16/32-bit PWC Timer		Operation Flowchart for Block Transfer	826
(ch.1, ch.0)	426	Branch Instruction	
Block Diagram of 16/32-bit Reload Timer		JMP Instruction (Branch Instruction)	782
(ch.1, ch.0)	425	BTnDTBF	
Block Diagram of 16-bit Free-run Timer	288	Bit Configuration of the Data Buffer Register	
Block Diagram of 16-bit PPG Timer	424	(BTnDTBF)	487
Block Diagram of 16-bit PWM Timer	424	BTnPCSR	
Block Diagram of 8/10-bit A/D Converter	701	Bit Configuration of the Period Setting Register	
Block Diagram of Analog Input I/O Port	180	(BTnPCSR)	475
Block Diagram of Bit Search Module	230	Bit Configuration of the PWM Period Setting Register	
Block Diagram of Clock Generation Control		(BTnPCSR)	446
Unit	96	BTnPDUT	
Block Diagram of Clock Monitor	729	Bit Configuration of the PWM Duty Setting Register	
Block Diagram of Delayed Interrupt		(BTnPDUT)	447
Module	228	BTnPRLH	
Block Diagram of DMAC	788	Bit Configuration of the "H"-width Setting Reload	
Block Diagram of External Bus Interface		Register (BTnPRLH)	461
I/O Port	182	BTnPRLL	
Block Diagram of External Interrupt Input		Bit Configuration of the "L"-width Setting Reload	
I/O Port	178	Register (BTnPRLL)	460
Block Diagram of External Interrupt/MMI		BTnSTC	
Controller	219	Status Control Register	
Block Diagram of Gate Function	264	(BTnSTC)	444, 458, 473, 485
Block Diagram of MB91470 Series (144 Pins)	7	BTnTMCR	
Block Diagram of MB91480 Series (100 Pins)	8	Timer Control Register (BTnTMCR	
Block Diagram of Multi-Function Timer		Lower Byte)	442, 456, 471, 483
I/O Port	184	Timer Control Register (BTnTMCR	
Block Diagram of Normal I/O Port	176	Upper Byte)	440, 454, 469, 481
Block Diagram of the 16-bit Input Capture	293	BTnTMR	
Block Diagram of the 16-bit Output		Bit Configuration of the Timer Register	
Compare	290	(BTnTMR)	448, 462, 476
Block Diagram of the 16-bit Reload Timer	236	Buffer	
Block Diagram of the 8-bit PPG ch.0, ch.2, ch.4, ch.6,		Compare Clear Buffer	369
ch.8, ch.10, ch.12 and ch.14	260	Buffer Control Register	
Block Diagram of the 8-bit PPG ch.1, ch.5, ch.9 and		Buffer Control Register	
ch.13	261	(ADTGBUF0/ADTGBUF1)	362
Block Diagram of the 8-bit PPG ch.3, ch.7, ch.11 and		Buffer Register	
ch.15	262	Notes at Accessing the Buffer Registers	414
Block Diagram of the A/D Activation		Built-in Peripheral Request	
Compare	297	Built-in Peripheral Request	807
Block Diagram of the External Bus		Burst	
Interface	129	Burst 2-Cycle Transfer	808
Block Diagram of the Free-run Timer			
Selector	300		
Block Diagram of the Interrupt Controller	209		

MB91470/480 Series

Burst Transfer		PPG Channels Corresponding to Each Mode	259
Operation Flowchart for Burst Transfer	827	Priority among Channels	824
Bus Converter		Chip Erase	
32-bit↔16-bit Bus Converter	45	Erasing Data from Flash Memory (Chip Erase) ...	855
Harvard↔Princeton Bus Converter	45	Chip Erasure	
Bus Error		Chip Erasure Command	843
Bus Error Operation	680	Chip Select Enable Register	
Conditions for the Occurrence of Bus Errors	680	Configuration of the Chip Select Enable Register	
Bus Interface		(CSER)	143
Block Diagram of the External Bus		Clear	
Interface	129	Clearing the Counters and Their Initial	
Control Signals on the Ordinary Bus		Values	491
Interface	146	Count Clear/Gate Function	514
Features of the External Bus Interface	128	Timer Clear	368
List of External Bus Interface Registers	130	Timing for Clearing Interrupts During DMA	818
Procedure for Setting the External Bus		CLKB	
Interface	171	CPU Clock (CLKB)	94
Register Overview of External Bus Interface	131	CLKP	
Bus Mode		Peripheral Clock (CLKP)	94
Bus Mode 0 (Single-Chip Mode)	73	CLKR	
Bus Mode 1		Clock Source Control Register (CLKR)	105
(Internal ROM/External Bus Mode)	73	CLKT	
Bus Mode 2		External Bus Clock (CLKT)	94
(External ROM/External Bus Mode)	73	Clock	
Busy		Baud Rate Selection for CSIO	
Ready/Busy Signal(RDY/BUSYX)	846	(Clock Synchronous Serial	
BUSYX		Interface)	617
Ready/Busy Signal(RDY/BUSYX)	846	Block Diagram of Clock Monitor	729
Byte Access		Clock Divider	95
Byte Access	159	Counting with the Internal Clock Selected	477
Byte Ordering		CPU Clock (CLKB)	94
Byte Ordering	55	External Bus Clock (CLKT)	94
		External Clock	564
C		Functions of CSIO (Clock Synchronous Serial	
C		Interface)	573
C Compiler (fcc911)	910	Internal Clock Operation	242
Calculating the Baud Rate		List of Registers of CSIO (Clock Synchronous Serial	
Calculating the Baud Rate	561, 618, 681	Interface)	574
Calculator		Operation of CSIO (Clock Synchronous Serial	
Multiplication and Addition Calculator	4	Interface)	605
Cancel		Oscillation Clock Frequency	871
Example of Using the Function to Generate a Request		Output Frequency from Clock Monitor	728
to Cancel a Hold Request (HRCR)	215	Peripheral Clock (CLKP)	94
Caution		Reload Values and Baud Rates for Different	
Cautions for Use of 16-bit Input Capture	415	Peripheral Clock Frequencies	
Cautions for Use of Free-run Timer Selector	415	562, 619, 682
CCR		Selected External Count Clock	374
Bit Configuration of Counter Control Register		Selecting the Count Clock	489
(CCR)	506	Selection of the Source Clock	88
Condition Code Register (CCR)	50	Clock Frequency	
Channel		Oscillation Clock Frequency	871
Channel Group	825	Reload Values and Baud Rates for Different	
Disabling All Channels	820	Peripheral Clock Frequencies	
Enabling Operation for All Channels	816	562, 619, 682

Clock Generation	
Block Diagram of Clock Generation Control	
Unit	96
Overview of Clock Generation Control	87
Clock Monitor	
Clock Monitor Function	4
Clock Output Enable Register	
Bit Configuration of Clock Output Enable	
Register	730
Clock Source Control Register	
Clock Source Control Register (CLKR)	105
Command	
Chip Erasure Command	843
Data Writing Command	843
Reset Command	842
Sector Erasure Command	844
Command Sequence	
Command Sequence of Automatic Algorithm	842
Common	
Common Matter	930
Common Notes on Using Each Type	
of Timer	435
COMP	
Compare Register	
(COMP0/COMP2/COMP4/COMP6,	
COMP1/COMP3/COMP5/COMP7)	253
Compare	
Reload/Compare Function	511
Synchronous Start of Reload/Compare	
Function	512
Compare Buffer Register	
Compare Buffer Register 0/3,1/4,2/5 (ADCOMPB0/	
ADCOMPB3, ADCOMP1/ADCOMP4,	
ADCOMP2/ADCOMP5)	356
Compare Clear	
A/D Activation by Zero Detection of Free-run Timer	
or Compare Clear	408
Compare Clear Buffer	369
Compare Clear Buffer Register	
Compare Clear Buffer Register (CPCLRBH0 to	
CPCLRBH5, CPCLRBL0 to	
CPCLRBL5)	310
Compare Clear Register	
Compare Clear Register (CPCLRH0 to CPCLRH5,	
CPCLRL0 to CPCLRL5)	311
Compare Control Register	
Compare Control Register, Lower Byte	
(OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/	
OCSL10)	332
Compare Control Register, Upper Byte	
(OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/	
OCSH11)	329
Compare Detection	
Compare Detection Flag	515
Compare Enable Register	
Compare Enable Register	
(ADTGCE0/ADTGCE1)	358
Compare Mode Control Register	
Compare Mode Control Register	
(OCMOD0/OCMOD1)	334
Compare Register	
Compare Register	
(COMP0/COMP2/COMP4/COMP6,	
COMP1/COMP3/COMP5/COMP7)	253
Compare Register 0/3,1/4,2/5 (ADCOMP0/	
ADCOMP3, ADCOMP1/ADCOMP4,	
ADCOMP2/ADCOMP5)	357
Compare Register Buffering	408
Comparison	
Count Direction Selection (for Comparison) Register	
(ADTGSEL0/ADTGSEL1)	360
Component	
Components of Each Model	5
Condition Code Register	
Condition Code Register (CCR)	50
Configuration	
Configuration	174
Connection	
Connection among CPUs	567
Connection between CPUs	565, 621
Connection between MB91470/480 Series and the	
Endian Areas	156
Connection Diagram between PPG and Multi-function	
Timer	263
Connection Example for MB91F47x/MB91F48x	
Serial Programming	870
Example of Connection with External	
Devices	152
Continuous	
Continuous Operation	449, 464
Operation of Continuous Conversion	
Mode	721, 752
Control Register	
Timer Control Register (BTnTMCR	
Upper Byte)	469
Control Signal	
Control Signals on the Ordinary Bus	
Interface	146
Control Signals on the Time Division I/O	
Interface	146
Control Status Register	
Bit Configuration of Control Status Register	
(TMCSR)	238
Controlling	
Controlling the Enabling of Input	179, 181
Conversion	
A/D Conversion Data Protection	
Function	723, 754

MB91470/480 Series

Operation of Continuous Conversion	
Mode	721, 752
Operation of Pause-Conversion Mode	722, 753
Operation of Single Conversion Mode	719, 751
Corresponding	
PPG Channels Corresponding to Each Mode	259
Count Clear	
Count Clear/Gate Function.....	514
Count Clock	
Selected External Count Clock	374
Selecting the Count Clock.....	489
Count Direction	
Count Direction Change Flag	515
Count Direction Flag	514
Count Direction Selection (for Comparison) Register (ADTGSEL0/ADTGSEL1)	360
Setting of Free-run Timer Count Direction Selection	407
Count Direction Selection (for Comparison) Register	
Count Direction Selection (for Comparison) Register (ADTGSEL0/ADTGSEL1)	360
Counter	
8-bit Counter	255
Block Diagram of Up/Down Counter	500
Clearing the Counters and Their Initial Values.....	491
Counter Operation States	244
Features of Up/Down Counter	498
List of Registers of Up/Down Counter	501
Counter Control Register	
Bit Configuration of Counter Control Register (CCR)	506
Counter Operation	
Counter Operation States	244
Counter Status Register	
Bit Configuration of Counter Status Register (CSR)	504
Counting	
Counting with the Internal Clock Selected	477
Measurement Mode and Counting	493
CPCLRBH	
Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5).....	310
CPCLRBL	
Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5).....	310
CPCLRH	
Compare Clear Register (CPCLRH0 to CPCLRH5, CPCLRL0 to CPCLRL5)	311
CPCLRL	
Compare Clear Register (CPCLRH0 to CPCLRH5, CPCLRL0 to CPCLRL5)	311

CPU	
Connection among CPUs.....	567
Connection between CPUs	565, 621
CPU	45
Features of FR CPU	2
FR-CPU Programming Mode (Read/Write in 16 Bits)	840
FR-CPU ROM Mode (Read Only in 32 Bits)	839
Pin States in Each CPU State	906
CPU Clock	
CPU Clock (CLKB).....	94
CSER	
Configuration of the Chip Select Enable Register (CSER)	143
CSIO	
Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)	617
Functions of CSIO (Clock Synchronous Serial Interface)	573
Interrupts of CSIO	599
List of Registers of CSIO (Clock Synchronous Serial Interface)	574
CSR	
Bit Configuration of Counter Status Register (CSR)	504
CSX	
CSX ->RDX/WR0X, WR1X Setup (TYP[3:0]=0101 _B , AWR=100B _H).....	170
CSX Delay Setting (TYP[3:0]=0000 _B , AWR=000C _H).....	166
Setting of CSX ->RDX/WR0X, WR1X Setup and of RDX/WR0X, WR1X ->CSX Hold (TYP[3:0]=0000 _B , AWR=000B _H).....	167
CTBR	
Time-base Counter Clear Register (CTBR)	105
D	
Data Buffer Register	
Bit Configuration of the Data Buffer Register (BTnDTBF)	487
Data Bus Width	
Data Bus Width of Little Endian	155
Data Bus Width of Big Endian	149
Data Direction	
Data Direction Bit.....	677
Data Direction Register	
Data Direction Registers (DDR: DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS)	189
Data Format	
Big Endian Data Format	147
Little Endian Data Format	154

Data Protection Function	
A/D Conversion Data Protection	
Function	723, 754
Data Writing	
Data Writing Command	843
DDR	
Data Direction Registers (DDR: DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS).....	189
Dead Time Timer Mode	
Note in Using Dead Time Timer Mode.....	403
Operation during Dead Time Timer Mode	401
Dead Timer Control Register	
16-bit Dead Timer Control Register, Lower Byte (DTCR1/DTCR4).....	348
16-bit Dead Timer Control Register, Upper Byte (DTCR0/DTCR3).....	346
16-bit Dead Timer Control Register, Upper Byte (DTCR2/DTCR5).....	350
Dead Timer Register	
16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5).....	345
Debugger	
Debugger (sim911, em1911, mon911).....	914
Debugging	
Notes on Debugging	932
Delay	
CSX Delay Setting (TYP[3:0]=0000 _B , AWR=000C _H)	166
DSP Delay Register (DSP-LY)	771
Delay Slot	
Operation with Delay Slot.....	58
Operation without Delay Slot	60
Delayed Interrupt	
Block Diagram of Delayed Interrupt	
Module.....	228
Delayed Interrupt Module Registers	228
Delayed Interrupt Module Registers	
Delayed Interrupt Module Registers	228
Delayed Write Function	
Delayed Write Function	776
Determining	
Determining the Priority	212
Device	
State of Device and Each Transition.....	116
Difference	
Difference between Multi-function Timers	
0 and 1	285
Difference Point of Multi-function Timer	
Construction in Each Series.....	285
Differences between Timing Generators	
0 and 1	246
Differential Input Mode	
Analog Input Pin in Differential Input	
Mode	755
Example of Differential Input Mode	
Operation	756
Register Setting in Differential Input Mode	755
Direct Addressing	
Direct Addressing Area	40, 57
Disabling	
Disabling All Channels.....	820
Division	
Control Signals on the Time Division I/O	
Interface.....	146
DIVR	
Internal Clock Dividing Frequency Set Register 0 (DIVR0)	108
Internal Clock Dividing Frequency Set Register 1 (DIVR1)	110
DMA	
DMA Transfer and Interrupts	815
DMAC (DMA Controller)	4
Notes on DMA Transfer in Sleep Mode	823
Suppressing DMA.....	815
Timing for Clearing Interrupts During DMA	818
DMA Transfer	
DMA Transfer and Interrupts	815
Notes on DMA Transfer in Sleep Mode	823
DMAC	
DMAC (DMA Controller)	4
Hardware Configuration of the DMAC	786
Interrupts That Enable DMAC Interrupt Control	
Outputs	822
Main DMAC Functions	786
Main DMAC Operations	805
Overview of DMAC.....	804
Register List of the DMAC Registers.....	787
DMACA	
Functions of the [DMACA0 to DMACA4]	
Bits.....	790
DMACB	
Functions of the DMACB0 to DMACB4	
Bits.....	794
DMACR	
Functions of the [DMACR] Bits	802
DMADA	
Functions of the DMASA0 to DMASA4 and DMADA0 to DMADA4	
Bits.....	800
DMASA	
Functions of the DMASA0 to DMASA4 and DMADA0 to DMADA4	
Bits.....	800
DSP	
DSP Control/Status Register (DSP-CSR)	767
DSP Delay Register (DSP-LY).....	771

MB91470/480 Series

DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7).....	772
DSP Control/Status Register DSP Control/Status Register (DSP-CSR)	767
DSP Delay Register DSP Delay Register (DSP-LY).....	771
DSP Variable Monitor Registers DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7).....	772
DSP-AC Accumulator Output Registers (DSP-AC0 to DSP-AC2)	773
DSP-CSR DSP Control/Status Register (DSP-CSR)	767
DSP-LY DSP Delay Register (DSP-LY).....	771
DSP-OT DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7).....	772
DSP-PC Program Counter (DSP-PC)	770
DTCR 16-bit Dead Timer Control Register, Lower Byte (DTCR1/DTCR4)	348
16-bit Dead Timer Control Register, Upper Byte (DTCR0/DTCR3)	346
16-bit Dead Timer Control Register, Upper Byte (DTCR2/DTCR5)	350
DTTI DTTI Interrupt	405
DTTI Operation of Waveform Control Register 2 (SIGCR2).....	405
DTTI Pin DTTI Pin Input Operation	404
DTTI Pin Noise Cancellation Feature.....	405
E	
Each Timer Function Bit Configurations in Each Timer Function Mode	428
EIRR External Interrupt Source Register (EIRR (EIRR0, EIRR1): External Interrupt Request Register)	220
EIT EIT Factor	61
EIT Operation.....	68
EIT Vector Table	65
Feature of EIT	61
Multiple EIT Processing	66
Priority of EIT Factor	66
Return from EIT	61
ELVR External Interrupt Request Level Setting Register (ELVR (ELVR0, ELVR1): External LeVel Register)	221
eml911 Debugger (sim911, eml911, mon911).....	914
Enable Interrupt Request Register Interrupt Enable Register (ENIR (ENIR0, ENIR1): ENable Interrupt Request Register)	220
Enabling Enabling Operation for All Channels	816
Endian Overview of Endian	145
Endian Areas Connection between MB91470/480 Series and the Endian Areas.....	156
ENIR Interrupt Enable Register (ENIR (ENIR0, ENIR1): ENable Interrupt Request Register)	220
Erasing Erasing Data from Flash Memory (Chip Erase) ...	855
Overview of Programming and Erasing Flash Memory	851
Sector Erasing Procedure.....	856
Error Bus Error Operation	680
Conditions for the Occurrence of Bus Errors	680
Occurrence of an Address Error	821
ESCR Bit Structure of the Extended Communication Control Register (ESCR)	532, 585
Example Examples	757
Explanation Operation Explanation	229, 232
Register Details Explanation.....	228, 231
Extended Communication Control Register Bit Structure of the Extended Communication Control Register (ESCR)	532, 585
External Bus Bus Mode 1 (Internal ROM/External Bus Mode).....	73
Bus Mode 2 (External ROM/External Bus Mode).....	73
External Bus Access	150
External Bus Clock (CLKT)	94
External Bus Interface	2
External Bus Interface Block Diagram of the External Bus Interface	129
Features of the External Bus Interface.....	128
Procedure for Setting the External Bus Interface	171
Register Overview of External Bus Interface.....	131

External Bus Interface Registers	
List of External Bus Interface Registers	130
External Clock	
External Clock	564
External Count Clock	
Selected External Count Clock	374
External Devices	
Example of Connection with External Devices	152
External Interrupt	
External Interrupt Request Level.....	223
External Interrupt Request Level Setting Register (ELVR (ELVR0, ELVR1): External LeVel Register).....	221
External Interrupt Source Register (EIRR (EIRR0, EIRR1): External Interrupt Request Register)	220
Notes If Restoring from STOP Status Performed Using an External Interrupt	225
Operating Procedure for an External Interrupt	222
Operation of an External Interrupt.....	222
External Interrupt Request Level Setting Register	
External Interrupt Request Level Setting Register (ELVR (ELVR0, ELVR1): External LeVel Register).....	221
External Interrupt Source Register	
External Interrupt Source Register (EIRR (EIRR0, EIRR1): External Interrupt Request Register)	220
External ROM	
Bus Mode 2 (External ROM/External Bus Mode)	73
External Wait	
External Wait Timing (TYP[3:0]=0001 _B , AWR=2008 _H)	165
With External Wait (TYP[3:0]=0101 _B , AWR=1008 _H)	169
Without External Wait (TYP[3:0]=0100 _B , AWR=0008 _H)	168
F	
fasm911	
Assembler (fasm911).....	913
FBGA-144	
FBGA-144 (MB91470 Series)	10
FBYTE	
Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)	546, 597, 653
fcc911	
C Compiler (fcc911).....	910
FCR	
Bit Structure of FIFO Control Register 0 (FCR0).....	543, 594, 649

Bit Structure of FIFO Control Register 1 (FCR1)	540, 592, 647
--	---------------

Feature	
DTTI Pin Noise Cancellation Feature	405
Features of FR CPU	2
Features of Internal Architecture	43
Features of the External Bus Interface.....	128
Features of Up/Down Counter.....	498
Other Features	4
Pulse Width Measurement Feature.....	488
FIFO	
Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2).....	546, 597, 653
Bit Structure of FIFO Control Register 0 (FCR0)	543, 594, 649
Bit Structure of FIFO Control Register 1 (FCR1)	540, 592, 647
Functions of FIFO.....	625
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing.....	551, 601
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing.....	554, 604
Transmission/Reception FIFO.....	519
FIFO Byte Register	
Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2).....	546, 597, 653
FIFO Control Register	
Bit Structure of FIFO Control Register 0 (FCR0)	543, 594, 649
Bit Structure of FIFO Control Register 1 (FCR1)	540, 592, 647
Flag	
Count Direction Change Flag	515
Count Direction Flag.....	514
Hardware Sequence Flag	846
Occurrence of Reception Interrupts and Flag Set Timing.....	550, 600
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing.....	551, 601
Occurrence of Transmission Interrupts and Flag Set Timing.....	553, 603
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing.....	554, 604
Flash Control/Status Register	
Bit Configuration of Flash Control/Status Register (FLCR)	836
Flash Memory	
Access Mode of Flash Memory	839
Erasing Data from Flash Memory (Chip Erase)...	855
Flash Memory Block Diagram	833
Flash Memory Programming Procedure	853
Memory Map for Flash Memory	834

MB91470/480 Series

Notes on Flash Memory Programming	865	Setting of Free-run Timer Count Direction Selection	407
Overview of Flash Memory Automatic Algorithms	841	Setting of Free-run Timer Input Selection	406
Overview of Flash Memory Outline	832	Free-run Timer Register 16-bit Free-run Timer Register	303
Overview of Flash Memory Registers	835	Free-run Timer Selection Register Free-run Timer Selection Register (Lower) for Input Capture(FRS3, FRS8)	326
Overview of Programming and Erasing Flash Memory	851	Free-run Timer Selection Register (Lower) for Output Compare (FRS0, FRS2, FRS5, FRS7)	324
Placing the Flash Memory in the Reset State	852	Free-run Timer Selection Register (Upper) for Input Capture (FRS4, FRS9)	325
Programming Data into Flash Memory	853	Free-run Timer Selection Register (Upper) for Output Compare (FRS1, FRS6)	322
Resuming Sector Erasure in Flash Memory	859	Free-run Timer Selection Registers	305
Suspending Sector Erasure in Flash Memory	858	FRS Free-run Timer Selection Register (Lower) for Input Capture(FRS3, FRS8)	326
Flash Microcontroller Programmer System Configuration for AF200 Flash Microcontroller Programmer (Manufactured by Yokogawa Digital Computer Corporation)	871	Free-run Timer Selection Register (Lower) for Output Compare (FRS0, FRS2, FRS5, FRS7)	324
Flash Wait Register Bit Structure of Flash Wait Register (FLWC)	838	Free-run Timer Selection Register (Upper) for Input Capture (FRS4, FRS9)	325
FLCR Bit Configuration of Flash Control/Status Register (FLCR)	836	Free-run Timer Selection Register (Upper) for Output Compare (FRS1, FRS6)	322
flnk911 Linker (flnk911)	914	Function Function Selection	567
Flowchart Flowchart	565, 621	Major Functions	204
Flowcharts	569	Function Mode Function Mode Bit Settings and Timer Function Modes Assigned	422
FLWC Bit Structure of Flash Wait Register (FLWC)	838	G Gate Count Clear/Gate Function	514
FR Features of FR CPU	2	Output Status of RTO0 to RTO5 and GATE	394
FR-CPU Programming Mode (Read/Write in 16 Bits)	840	PPG Output by Gate Trigger	396
FR-CPU ROM Mode (Read Only in 32 Bits)	839	Gate Function Block Diagram of Gate Function	264
Instruction List of FR Family	915	GATE Function Control Registers (GATEC0/ GATEC4/GATEC8/GATEC12)	273
Free-run Timer 16-bit Free-run Timer Interrupt	364	GATE Function Control Registers GATE Function Control Registers (GATEC0/ GATEC4/GATEC8/GATEC12)	273
16-bit Free-run Timer Register	303	GATEC GATE Function Control Registers (GATEC0/ GATEC4/GATEC8/GATEC12)	273
A/D Activation by Free-run Timer	375	General-purpose Register General-purpose Register	49
A/D Activation by Zero Detection of Free-run Timer or Compare Clear	408	Generating Generating a Start Condition	658
Block Diagram of 16-bit Free-run Timer	288		
Block Diagram of the Free-run Timer Selector	300		
Cautions for Use of Free-run Timer Selector	415		
Example Program for the 16-bit Free-run Timer	417		
Notes on Using the 16-bit Free-run Timers	415		
Operation of 16-bit Output Compare and Free-run Timer	385		
Operation of Free-run Timer Selector in MB91470 Series	376		
Operation of Free-run Timer Selector in MB91480 Series	377		

H		
Halfword Access		
Halfword Access	158	
Handling		
Precautions on Handling Device	34	
Hardware Configuration		
Hardware Configuration of the DMAC.....	786	
Hardware Configuration of the Interrupt		
Controller	204	
Hardware Sequence Flag		
Hardware Sequence Flag.....	846	
Harvard↔Princeton Bus Converter		
Harvard↔Princeton Bus Converter	45	
Hold		
Example of Using the Function to Generate a Request		
to Cancel a Hold Request (HRCR)	215	
Hold Request Cancel Level Register		
(HRCL)	211	
Hold Request Cancel Request.....	213	
NMI/Hold Suppress Level Interrupt		
Processing	819	
Setting of CSX ->RDX/WR0X, WR1X Setup and of		
RDX/WR0X, WR1X ->CSX Hold		
(TYP[3:0]=0000 _B , AWR=000B _H)	167	
Hold Request		
Example of Using the Function to Generate a Request		
to Cancel a Hold Request (HRCR)	215	
Hold Request Cancel Level Register		
(HRCL)	211	
Hold Request Cancel Request.....	213	
Hold Request Cancel Level Register		
Hold Request Cancel Level Register		
(HRCL).....	211	
Hold Request Cancel Request		
Hold Request Cancel Request.....	213	
HRCL		
Hold Request Cancel Level Register		
(HRCL).....	211	
HRCR		
Example of Using the Function to Generate a Request		
to Cancel a Hold Request (HRCR)	215	
"H"-width Setting Reload Register		
Bit Configuration of the "H"-width Setting Reload		
Register (BTnPRLH)	461	
I		
I Flag		
I Flag	63	
I/O		
Control Signals on the Time Division I/O		
Interface	146	
I/O Map.....	882	
I/O Pins.....	130	
Modes of I/O Port.....	177, 179, 181, 183, 185	
I/O Circuit		
I/O Circuit Type	29	
I/O Map		
I/O Map	882	
I/O Port		
I/O Port.....	3	
Modes of I/O Port	177, 179, 181, 183, 185	
I ² C		
Functions of I ² C Interface.....	625	
I ² C Bus Control Register (IBCR)	628	
Interrupts of I ² C Interface	655	
List of Registers of I ² C Interface	626	
Repeated Start Condition for I ² C Bus.....	657	
Start Condition for I ² C Bus.....	657	
Stop Condition for I ² C Bus	657	
I ² C Bus Control Register		
I ² C Bus Control Register (IBCR)	628	
I ² C Bus Status Register		
I ² C Bus Status Register (IBSR)	635	
IBCR		
I ² C Bus Control Register (IBCR)	628	
IBSR		
I ² C Bus Status Register (IBSR)	635	
ICR		
Interrupt Control Register (ICR).....	63, 210	
ICSH		
Input Capture State Control Register		
(ch.2/ch.6, ch.3/ch.7), Upper Byte (ICSH23/		
ICSH67)	337	
ICSL		
Input Capture State Control Register		
(ch.2/ch.6, ch.3/ch.7), Lower Byte (ICSL23/		
ICSL67).....	339	
ILM		
ILM.....	52, 63	
INIT		
Set Initialization Reset (INIT)	78	
Set Initialization Reset (INIT) Release		
Sequence.....	81	
Initial Values		
Clearing the Counters and Their Initial		
Values	491	
Initialization		
INITX Terminal Input (System Initialization Reset		
Terminal)	79	
Operation Initialization Reset (RST).....	78	
Operation Initialization Reset (RST) Release		
Sequence.....	82	
Set Initialization Reset (INIT)	78	
Set Initialization Reset (INIT) Release		
Sequence.....	81	
System Initialization Reset		
(SINIT).....	78	

MB91470/480 Series

System Initialization Reset (SINIT)		Internal Clock Dividing Frequency Set Register	
Release Sequence	81	Internal Clock Dividing Frequency Set Register 0	
Initialize		(DIVR0)	108
Wait Time after System/Setting is Initialized	92	Internal Clock Dividing Frequency Set Register 1	
INITX		(DIVR1)	110
INITX Terminal Input (System Initialization Reset		Internal Memory	
Terminal)	79	Internal Memory	2
Input		Internal ROM	
Controlling the Enabling of Input	179, 181	Bus Mode 1	
Setting of Free-run Timer Input Selection	406	(Internal ROM/External Bus Mode)	73
Input Capture		Interrupt	
16-bit Input Capture Interrupt	365	16-bit Free-run Timer Interrupt	364
16-bit Input Capture Operation	392	16-bit Input Capture Interrupt	365
Block Diagram of the 16-bit Input Capture	293	16-bit Output Compare Interrupt	365
Cautions for Use of 16-bit Input Capture	415	Block Diagram of Delayed Interrupt	
Free-run Timer Selection Register (Lower) for Input		Module	228
Capture(FRS3, FRS8)	326	Block Diagram of the Interrupt Controller	209
Free-run Timer Selection Register (Upper) for Input		Delayed Interrupt Module Registers	228
Capture (FRS4, FRS9)	325	DMA Transfer and Interrupts	815
Input Capture Data Register (IPCPH0 to IPCPH7,		DTTI Interrupt	405
IPCPL0 to IPCPL7)	336	External Interrupt Input	3
Input Timing for 16-bit Input Capture	393	External Interrupt Request Level	223
Input Capture Data Register		External Interrupt Request Level Setting Register	
Input Capture Data Register (IPCPH0 to IPCPH7,		(ELVR (ELVR0, ELVR1): External LeVel	
IPCPL0 to IPCPL7)	336	Register)	221
Input Capture Register		Hardware Configuration of the Interrupt	
16-bit Input Capture Register	307	Controller	204
Input Capture State Control Register		Interrupt Control Bits and Interrupt Factors by Timer	
Input Capture State Control Register (ch.0/ch.4,ch.1/		Function Mode	437
ch.5) Lower Byte (PICSL01/PICSL45)		Interrupt Control Register (ICR)	210
.....	343	Interrupt Controller Registers	205
Input Capture State Control Register		Interrupt Enable Register (ENIR (ENIR0, ENIR1):	
(ch.2/ch.6, ch.3/ch.7), Lower Byte (ICSL23/		ENable Interrupt Request Register)	220
ICSL67)	339	Interrupt Factors and Timing Chart	
Input Capture State Control Register		(PPG Output: Normal Polarity)	467
(ch.2/ch.6, ch.3/ch.7), Upper Byte (ICSH23/		Interrupt Factors and Timing Chart	
ICSH67)	337	(PWM Output: Normal Polarity)	451
Input Pin		Interrupt Generation Timing	516
Input Pin Operation	479	Interrupt Level	62
Input Timing		Interrupt Mask Function	372
Input Timing for 16-bit Input Capture	393	Interrupt of 12-bit A/D Converter	750
Instruction		Interrupt of 8/10-bit A/D Converter	718
Instruction Definitions	765	Interrupt Vector Table	902
Instruction List of FR Family	915	Interrupts of CSIO	599
Instruction Operations	775	Interrupts of I ² C Interface	655
MAC Instruction	778	Interrupts of UART	548
NOP Instruction	783	Interrupts That Enable DMAC Interrupt Control	
Overview of Instructions	46	Outputs	822
STR Instruction (Transfer Instruction)	780	NMI (Non Maskable Interrupt)	212
Interface		NMI/Hold Suppress Level Interrupt	
Interface Mode	518	Processing	819
Switching the Interface Mode	518	Notes If Restoring from STOP Status Performed	
Internal Clock		Using an External Interrupt	225
Counting with the Internal Clock Selected	477	Occurrence of Reception Interrupts and Flag Set	
Internal Clock Operation	242	Timing	550, 600

Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set	
Timing	551, 601
Occurrence of Transmission Interrupts and Flag Set	
Timing	553, 603
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set	
Timing	554, 604
Operating Procedure for an External Interrupt	222
Operation of an External Interrupt	222
Timer Interrupt	371
Timing for Clearing Interrupts During DMA	818
Waveform Generator Interrupts	366
Interrupt Control	
Interrupt Control Bits and Interrupt Factors by Timer Function Mode	437
Interrupt Control Register (ICR)	210
Interrupts That Enable DMAC Interrupt Control	
Outputs	822
Interrupt Control Register	
Interrupt Control Register (ICR)	63, 210
Interrupt Controller	
Block Diagram of the Interrupt Controller	209
Hardware Configuration of the Interrupt Controller	204
Interrupt Controller Registers	205
Interrupt Controller Registers	
Interrupt Controller Registers	205
Interrupt Enable Register	
Interrupt Enable Register (ENIR (ENIR0, ENIR1): ENable Interrupt Request Register)	220
Interrupt Stack	
Interrupt Stack	65
Interrupt Vector	
Interrupt Vector Table	902
Interruption	
Level Mask to Interruption and NMI	63
IPCPH	
Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)	336
IPCPL	
Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)	336
ISBA	
7-bit Slave Address Register (ISBA)	645
ISMK	
7-bit Slave Address Mask Register (ISMK)	644
J	
JMP	
JMP Instruction (Branch Instruction)	782

L	
Level Mask	
Level Mask to Interruption and NMI	63
Linker	
Linker (flnk911)	914
Little Endian	
Data Bus Width of Little Endian	155
Little Endian Data Format	154
Overview of Little Endian Method	153
Low-power Consumption	
Low-power Consumption Mode	120
LQFP-100	
LQFP-100 (MB91480 Series)	11
Package Dimension (LQFP-100P-M20)	16
LQFP-144	
LQFP-144 (MB91470 Series)	9
Package Dimension (LQFP-144P-M12)	13
"L"-width Setting Reload Register	
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	460
M	
MAC	
MAC Instruction	778
Main DMAC	
Main DMAC Functions	786
Main DMAC Operations	805
Major Functions	
Major Functions	204
Master	
Master Data Reception	672
Master Data Transmission	665
Wait in Master Mode	675
MB91470	
Block Diagram of MB91470 Series (144 Pins)	7
FBGA-144 (MB91470 Series)	10
LQFP-144 (MB91470 Series)	9
Memory Map (MB91470 Series)	41
Operation of Free-run Timer Selector in MB91470 Series	376
MB91480	
Block Diagram of MB91480 Series (100 Pins)	8
LQFP-100 (MB91480 Series)	11
Memory Map (MB91480 Series)	42
Operation of Free-run Timer Selector in MB91480 Series	377
QFP-100 (MB91480 Series)	12
MB91F47x	
Connection Example for MB91F47x/MB91F48x Serial Programming	870
MB91F48x	
Connection Example for MB91F47x/MB91F48x Serial Programming	870

MB91470/480 Series

MDH		Multiplication	
Multiplication and Division Register (Multiply & Divide Register) (MDH/MDL).....	54	Multiplication and Addition Calculator.....	4
MDL		PLL Multiplication Rate.....	89
Multiplication and Division Register (Multiply & Divide Register) (MDH/MDL).....	54	Multiplication and Addition Calculator	
Measurement		Register List of Multiplication and Addition Calculator	766
Measurement Mode and Counting	493	Multiplication and Division Register	
Memory		Multiplication and Division Register (Multiply & Divide Register) (MDH/MDL)	54
Internal Memory	2	N	
Memory Map		NMI	
Memory Map.....	57	Level Mask to Interruption and NMI	63
Memory Map (MB91470 Series)	41	NMI	224
Memory Map (MB91480 Series)	42	NMI (Non Maskable Interrupt)	212
Memory Map for Flash Memory.....	834	NMI/Hold Suppress Level Interrupt Processing.....	819
MOD		Noise Cancellation	
16-bit Output Compare Operation (Inversion Mode, MOD1x=0).....	379	DTTI Pin Noise Cancellation Feature	405
16-bit Output Compare Operation (Set/Reset Mode, MOD1x=1).....	383	NOP	
Mode Data		NOP Instruction	783
Mode Data	74	Normal Access	
Mode Pin		Normal Access and Address/Data Multiplex Access	140
Mode Pin	74	Normal Polarity	
Module		Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	467
Block Diagram of Delayed Interrupt Module	228	Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	451
Delayed Interrupt Module Registers.....	228	Normal Transfer	
mon911		Normal Transfer (I).....	605
Debugger (sim911, eml911, mon911)	914	Normal Transfer (II)	608
Monitor		Normal Transfer (II) Timing Chart	609
Block Diagram of Clock Monitor	729	Note	
DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7).....	772	Note	71, 76, 516
Output Frequency from Clock Monitor	728	Other Notes.....	871
Variable Monitor Output.....	777	O	
Multi-function		OCCPBH	
Block Diagram of the Multi-function Timer	286	Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11)	327
Connection Diagram between PPG and Multi-function Timer.....	263	OCCPBL	
Difference Point of Multi-function Timer Construction in Each Series	285	Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11)	327
Operation of the Multi-function Timer	367	OCCPH	
Pins of the Multi-function Timer	301	Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11).....	328
Structure of Multi-function Timer.....	282	OCCPL	
Multi-function Timer		Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11).....	328
Difference between Multi-function Timers 0 and 1	285		
Multi-function Timer.....	3		
Multiplex Access			
Normal Access and Address/Data Multiplex Access	140		

Occurrence	
Conditions for the Occurrence of Bus Errors	680
Occurrence of an Address Error	821
Occurrence of Reception Interrupts and Flag Set	
Timing	550, 600
Occurrence of Reception Interrupts when Reception	
FIFO is Used and Flag Set	
Timing	551, 601
Occurrence of Transmission Interrupts and Flag Set	
Timing	553, 603
Occurrence of Transmission Interrupts when	
Transmission FIFO is Used and Flag Set	
Timing	554, 604
OCMOD	
Compare Mode Control Register	
(OCMOD0/OCMOD1)	334
OCSH	
Compare Control Register, Upper Byte	
(OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/	
OCSH11)	329
OCSL	
Compare Control Register, Lower Byte	
(OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/	
OCSL10)	332
One-shot Operation	
One-shot Operation	450, 465
Operating Mode	
Operating Mode	72
Operating Procedure	
Operating Procedure for an External	
Interrupt	222
Operation	
Operation Explanation	229, 232, 274
Principles of Operation	463
Operation Flow	
Pulse Width Measurement Operation Flow	495
Operation Mode	
Operation Mode	523
Operation Modes	774
Selecting the Operation Mode	490
Ordinary Bus Interface	
Control Signals on the Ordinary Bus	
Interface	146
Oscillation Clock Frequency	
Oscillation Clock Frequency	871
Oscillation Stabilization Wait	
Triggers for the Oscillation Stabilization	
Wait	83
Oscillation Stabilization Wait Time	
Select Oscillation Stabilization Wait Time	84
Output Compare	
16-bit Output Compare Interrupt	365
16-bit Output Compare Operation	
(Inversion Mode, MOD1x=0)	379
16-bit Output Compare Operation	
(Set/Reset Mode, MOD1x=1)	383
16-bit Output Compare Timing	384
Block Diagram of the 16-bit Output	
Compare	290
Example Program for the 16-bit Output	
Compare	418
Free-run Timer Selection Register (Lower) for Output	
Compare	
(FRS0, FRS2, FRS5, FRS7)	324
Free-run Timer Selection Register (Upper) for Output	
Compare (FRS1, FRS6)	322
Notes on Using the 16-bit Output Compare	415
Operation of 16-bit Output Compare and Free-run	
Timer	385
Output Compare Buffer Register	
(OCCPBH0 to OCCPBH11, OCCPBL0 to	
OCCPBL11)	327
Output Compare Buffer Register	
Output Compare Buffer Register	
(OCCPBH0 to OCCPBH11, OCCPBL0 to	
OCCPBL11)	327
Output Compare Register	
16-bit Output Compare Register	306
Output Compare Register (OCCPH0 to OCCPH11,	
OCCPL0 to OCCPL11)	328
Output Inversion Register	
Output Inversion Register (REVC)	272
Output Pin	
Output Pin Operation	479
Overall Restrictions	
Overall Restrictions and Notes	880
Overview	
Overview	77, 762
P	
Package	
Package Lineup	5
Package Dimension	
Package Dimension (BGA-144P-M06)	14, 15
Package Dimension (LQFP-100P-M20)	16
Package Dimension (LQFP-144P-M12)	13
Package Dimension (QFP-100P-M06)	17
Pause	
Sector Erasure Pause Command	845
Pause-Conversion Mode	
Operation of Pause-Conversion Mode	722, 753
PC	
Program Counter (PC)	53
PCR	
Pull-up Control Registers (PCR: PCR0 to PCR3,	
PCR5, PCR6, PCR8 to PCRH, PCRJ, PCRL,	
PCRM, PCRP to PCRS)	192

MB91470/480 Series

PDR		PPG	
Port Data Registers (PDR: PDR0 to PDR3, PDR5, PDR6, PDR8 to PDRH, PDRJ, PDRL, PDRM, PDRP to PDRS).....	186	16-bit PPG Timer	423
Period Setting Register		Block Diagram of 16-bit PPG Timer	424
Bit Configuration of the Period Setting Register (BTnPCSR)	475	Block Diagram of the 8-bit PPG ch.0, ch.2, ch.4, ch.6, ch.8, ch.10, ch.12 and ch.14	260
Peripheral		Block Diagram of the 8-bit PPG ch.1, ch.5, ch.9 and ch.13	261
Transfer Stop Requests from Peripheral Circuits	821	Block Diagram of the 8-bit PPG ch.3, ch.7, ch.11 and ch.15	262
Peripheral Clock		Connection Diagram between PPG and Multi-function Timer	263
Reload Values and Baud Rates for Different Peripheral Clock Frequencies	562, 619, 682	Functions of PPG.....	258
PFR		Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	467
Port Function Control Registers (PFR: PFR0 to PFR3, PFR5, PFR6, PFR8, PFR9, PFRJ, PFRF to PFRH, PFRJ, PFRM, PFRQ, PFRS)	195	Notes on Using the 16-bit PWM/PPG/Reload Timer	435
PICSH		PPG Channels Corresponding to Each Mode	259
PPG Output Control Register Upper Byte (PICSH01/ PICSH45).....	341	PPG Mode	258
PPG Output Control Upper Byte (PICSH01/PICSH45).....	341	PPG Operation Mode Control Registers (PPGC0 to PPGC15).....	267
PICSL		PPG Output by Gate Trigger	396
Input Capture State Control Register (ch.0/ch.4,ch.1/ ch.5) Lower Byte (PICSL01/PICSL45)	343	PPG Output Control	395
Pin Functions		PPG Registers	265
List of Pin Functions	18	PPG Trigger Register (TRG)	271
Pin States		Reload Registers: 16-bit PPG Mode	266
Pin States in Each CPU State	906	PPG Operation Mode Control Registers	
Placing		PPG Operation Mode Control Registers (PPGC0 to PPGC15).....	267
Placing the Flash Memory in the Reset State	852	PPG Output Control Register	
PLL		PPG Output Control Register Upper Byte (PICSH01/ PICSH45)	341
PLL Enabling Operation	89	PPG Timer	
PLL Multiplication Rate	89	16-bit PPG Timer	423
Wait Time after Changing the PLL Multiplier Ratio	92	8/16-bit PPG Timer.....	3
Wait Time after Enabling PLL Operation	92	Block Diagram of 16-bit PPG Timer	424
Port		PPG Trigger Register	
Block Diagrams of Ports	175	PPG Trigger Register (TRG)	271
Overview of Ports	174	PPGC	
Port Data Registers		PPG Operation Mode Control Registers (PPGC0 to PPGC15).....	267
Port Data Registers (PDR: PDR0 to PDR3, PDR5, PDR6, PDR8 to PDRH, PDRJ, PDRL, PDRM, PDRP to PDRS).....	186	Precautions	
Port Function Control Registers		Precautions	244
Port Function Control Registers (PFR: PFR0 to PFR3, PFR5, PFR6, PFR8, PFR9, PFRJ, PFRF to PFRH, PFRJ, PFRM, PFRQ, PFRS)	195	Prescaler	
Power Supply		Operation of Prescaler	255
Wait Time after Power Supply is Turned on	92	Principles	
		Principles of Operation.....	463
		Priority	
		Determining the Priority	212
		Priority among Channels	824
		Priority of EIT Factor	66
		PRLH	
		Reload Registers (PRLH0 to PRLH15, PRLH0 to PRLH15).....	270

PRL		Pulse Width Measurement Operation Flow	495
Reload Registers (PRLH0 to PRLH15, PRL0 to PRL15)	270	Relationship between Reload Value and Pulse Width	466
Procedure		Starting and Stopping Pulse Width Measurement	491
Flash Memory Programming Procedure	853		
Procedure for Setting the External Bus Interface	171		
Sector Erasing Procedure	856		
Processing			
Backup/Restore Processing	234		
NMI/Hold Suppress Level Interrupt Processing	819		
Program Counter		PWC	
Program Counter (DSP-PC)	770	16/32-bit PWC Timer	423
Program Counter (PC)	53	Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)	426
Program Status		Notes on Using the PWC Timer	436
Program Status (PS)	49	PWM	
Programmer		16-bit PWM Timer	422
System Configuration for AF200 Flash Microcontroller Programmer (Manufactured by Yokogawa Digital Computer Corporation)	871	Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	447
Programming		Bit Configuration of the PWM Period Setting Register (BTnPCSR)	446
Basic Configuration for Serial Programming	868	Block Diagram of 16-bit PWM Timer	424
Basic Programming Model	48	Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	451
Connection Example for MB91F47x/MB91F48x Serial Programming	870	Notes on Using the 16-bit PWM/PPG/Reload Timer	435
Flash Memory Programming Procedure	853	PWM Output at All "L" or All "H" Level	452
FR-CPU Programming Mode (Read/Write in 16 Bits)	840	PWM Duty Setting Register	
Notes on Flash Memory Programming	865	Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	447
Notes on Programming Data	853	PWM Output	
Overview of Programming and Erasing Flash Memory	851	Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	451
Pins Used for Fujitsu-standard Serial On-board Programming	869	PWM Output at All "L" or All "H" Level	452
Programming Data into Flash Memory	853	PWM Period Setting Register	
Protection Function		Bit Configuration of the PWM Period Setting Register (BTnPCSR)	446
A/D Conversion Data Protection Function	723, 754	PWM Timer	
PS		16-bit PWM Timer	422
Program Status (PS)	49	Block Diagram of 16-bit PWM Timer	424
Pull-up Control Registers		Q	
Pull-up Control Registers (PCR: PCR0 to PCR3, PCR5, PCR6, PCR8 to PCRH, PCRJ, PCRL, PCRM, PCRP to PCRS)	192	QFP-100	
Pull-up Resistor Control Register		Package Dimension (QFP-100P-M06)	17
Setting Value in Pull-up Resistor Control Register	177, 179, 181, 183, 185	QFP-100 (MB91480 Series)	12
Pulse Width		R	
Details of Pulse Width Measurement Operation	492	Range	
Pulse Width Measurement Feature	488	Acceptable Baud Rate Range for Reception	563
		RCR	
		Reload Compare Register (RCR)	503
		RDR	
		Reception Data Register (RDR)	642
		Reception Data Register (RDR0/RDR1)	534, 587
		RDX	
		CSX -> RDX/WR0X, WR1X Setup (TYP[3:0]=0101 _B , AWR=100B _H)	170

MB91470/480 Series

Setting of CSX ->RDX/WR0X, WR1X Setup and of RDX/WR0X, WR1X ->CSX Hold (TYP[3:0]=0000 _B , AWR=000B _H)	167	A/D Channel Control Register (ADCH: ADCH0 to ADCH2)	707
RDY		A/D Channel Control Register (ADCH: ADCH3, ADCH4)	739
Ready/Busy Signal(RDY/BUSYX).....	846	A/D Control Status Register (ADCS: ADCS1, ADCS2).....	712
Read		A/D Control Status Register (ADCS: ADCS3, ADCS4).....	744
FR-CPU Programming Mode (Read/Write in 16 Bits).....	840	A/D Data Register (ADCD: ADCD000 to ADCD030, ADCD001 to ADCD031, ADCD002 to ADCD112)	715
FR-CPU ROM Mode (Read Only in 32 Bits).....	839	A/D Data Register (ADCD: ADCD003 to ADCD033, ADCD004 to ADCD034)	747
Placing the Flash Memory in the Reset State	852	A/D Mode Setting Register (ADMD: ADMD0 to ADMD2).....	709
Read ->Write Timing (TYP[3:0]=0000 _B , AWR=0048 _H).....	162	A/D Mode Setting Register (ADMD: ADMD3, ADMD4).....	741
Ready		A/D Trigger Control Register (ADTRGC0 to ADTRGC5)	320
Ready/Busy Signal(RDY/BUSYX).....	846	Accumulator Output Registers (DSP-AC0 to DSP-AC2)	773
REALOS		Address Register Specifications	812
Bit Search Module (for REALOS)	3	ADMD Register Setting	724
Receiving		Analog Input Control Register (AICR: AICR0 to AICR2)	717
Receiving Acknowledge after Transmitting 1st Byte.....	661	Analog Input Control Register (AICR: AICR3, AICR4).....	749
Reception		Bit Configuration of 16-bit Reload Register (TMRLR)	241
Acceptable Baud Rate Range for Reception.....	563	Bit Configuration of Clock Output Enable Register	730
Master Data Reception	672	Bit Configuration of Control Status Register (TMCSR).....	238
Occurrence of Reception Interrupts and Flag Set Timing	550, 600	Bit Configuration of Counter Control Register (CCR).....	506
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing	551, 601	Bit Configuration of Counter Status Register (CSR)	504
Reception Data Register (RDR).....	642	Bit Configuration of Flash Control/Status Register (FLCR).....	836
Reception Data Register (RDR0/RDR1).....	534, 587	Bit Configuration of the Data Buffer Register (BTnDTBF)	487
Slave Reception	677	Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	461
Transmission/Reception FIFO	519	Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL).....	460
Reception Data Register		Bit Configuration of the Period Setting Register (BTnPCSR).....	475
Reception Data Register (RDR).....	642	Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	447
Reception Data Register (RDR0/RDR1).....	534, 587	Bit Configuration of the PWM Period Setting Register (BTnPCSR).....	446
Recovery		Bit Configuration of the Timer Register (BTnTMR).....	448, 462, 476
Recovery from a Standby Mode (Stop or Sleep).....	214	Bit Configuration of Wild Register Address Register (WA).....	877
Recovery Operations from STOP Status.....	226		
Register			
16-bit Dead Timer Control Register, Lower Byte (DTCR1/DTCR4)	348		
16-bit Dead Timer Control Register, Upper Byte (DTCR0/DTCR3)	346		
16-bit Dead Timer Control Register, Upper Byte (DTCR2/DTCR5)	350		
16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)	345		
16-bit Free-run Timer Register	303		
16-bit Input Capture Register	307		
16-bit Output Compare Register	306		
7-bit Slave Address Mask Register (ISMK)	644		
7-bit Slave Address Register (ISBA).....	645		

Bit Configuration of Wild Register Data Register (WD)	878
Bit Configuration of Wild Register Enable Register (WREN)	876
Bit Search Module Registers	230
Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)	546, 597
Bit Structure of FIFO Control Register 0 (FCR0)	543, 594, 649
Bit Structure of FIFO Control Register 1 (FCR1)	540, 592, 647
Bit Structure of Flash Wait Register (FLWC)	838
Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	538, 590, 646
Bit Structure of the Extended Communication Control Register (ESCR)	532, 585
Block Diagram of Wild Register Control Block	874
Buffer Control Register (ADTGBUF0/ADTGBUF1)	362
Clock Source Control Register (CLKR)	105
Compare Buffer Register 0/3,1/4,2/5 (ADCOMPB0/ADCOMPB3, ADCOMPBI/ADCOMPB4, ADCOMPBI/ADCOMPB5)	356
Compare Clear Buffer Register (CPCLRBH0 to CPCLRBH5, CPCLRBL0 to CPCLRBL5)	310
Compare Clear Register (CPCLRHO to CPCLRHS, CPCLRL0 to CPCLRL5)	311
Compare Control Register, Lower Byte (OCSL0/OCSL6, OCSL2/OCSL8, OCSL4/OCSL10)	332
Compare Control Register, Upper Byte (OCSH1/OCSH7, OCSH3/OCSH9, OCSH5/OCSH11)	329
Compare Enable Register (ADTGCE0/ADTGCE1)	358
Compare Mode Control Register (OCMOD0/OCMOD1)	334
Compare Register (COMP0/COMP2/COMP4/COMP6, COMP1/COMP3/COMP5/COMP7)	253
Compare Register 0/3,1/4,2/5 (ADCOMP0/ADCOMP3, ADCOMP1/ADCOMP4, ADCOMP2/ADCOMP5)	357
Condition Code Register (CCR)	50
Configuration of Area Configuration Registers (ACR0 to ACR2)	133
Configuration of Area Select Registers (ASR0 to ASR2)	132
Configuration of Area Wait Registers (AWR0 to AWR2)	139
Configuration of the Chip Select Enable Register (CSER)	143
Count Direction Selection (for Comparison) Register (ADTGSEL0/ADTGSEL1)	360

Data Direction Registers (DDR: DDR0 to DDR3, DDR5, DDR6, DDR8 to DDRH, DDRJ, DDRL, DDRM, DDRP to DDRS)	189
Delayed Interrupt Module Registers	228
DSP Control/Status Register (DSP-CSR)	767
DSP Delay Register (DSP-LY)	771
DSP Variable Monitor Registers (DSP-OT0 to DSP-OT7)	772
DTTI Operation of Waveform Control Register 2 (SIGCR2)	405
External Interrupt Request Level Setting Register (ELVR (ELVR0, ELVR1): External LeVel Register)	221
External Interrupt Source Register (EIRR (EIRR0, EIRR1): External Interrupt Request Register)	220
Free-run Timer Selection Register (Lower) for Input Capture (FRS3, FRS8)	326
Free-run Timer Selection Register (Upper) for Input Capture (FRS4, FRS9)	325
Free-run Timer Selection Register (Upper) for Output Compare (FRS1, FRS6)	322
Free-run Timer Selection Registers	305
GATE Function Control Registers (GATEC0/GATEC4/GATEC8/GATEC12)	273
General-purpose Register	49
Hold Request Cancel Level Register (HRCL)	211
I ² C Bus Control Register (IBCR)	628
I ² C Bus Status Register (IBSR)	635
Input Capture Data Register (IPCPH0 to IPCPH7, IPCPL0 to IPCPL7)	336
Input Capture State Control Register (ch.0/ch.4, ch.1/ch.5) Lower Byte (PICSL01/PICSL45)	343
Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Lower Byte (ICSL23/ICSL67)	339
Input Capture State Control Register (ch.2/ch.6, ch.3/ch.7), Upper Byte (ICSH23/ICSH67)	337
Internal Clock Dividing Frequency Set Register 0 (DIVR0)	108
Internal Clock Dividing Frequency Set Register 1 (DIVR1)	110
Interrupt Control Register (ICR)	63, 210
Interrupt Controller Registers	205
Interrupt Enable Register (ENIR (ENIR0, ENIR1): ENable Interrupt Request Register)	220
List of 16-bit Reload Timer Registers	237
List of Base Timer's Registers	427
List of External Bus Interface Registers	130
List of Registers in Wild Register Control Block	875
List of Registers of CSIO (Clock Synchronous Serial Interface)	574
List of Registers of I ² C Interface	626

List of Registers of UART (Asynchronous Serial Interface)	522	Setting of Temporary Stopping by Writing to the Control Register (Set Independently for Each Channel or All Channels Simultaneously)	819
List of Registers of Up/Down Counter	501	Setting Value in Pull-up Resistor Control Register	177, 179, 181, 183, 185
Notes on Setting Registers	789	Standby Control Register (STCR)	99
Operations of Wild Register Control Block	879	Status Control Register (BTnSTC)	444, 458, 473, 485
Output Compare Buffer Register (OCCPBH0 to OCCPBH11, OCCPBL0 to OCCPBL11)	327	System Condition Code Register (SCR)	51
Output Compare Register (OCCPH0 to OCCPH11, OCCPL0 to OCCPL11)	328	Table Base Register (TBR)	65
Output Inversion Register (REVC)	272	Time-base Counter Clear Register (CTBR)	105
Overview of Flash Memory Registers	835	Time-base Counter Control Register (TBCR)	102
Overview of Wild Register Control Block	874	Timer Control Register (BTnTMCR Lower Byte)	442, 456, 471, 483
Port Data Registers (PDR: PDR0 to PDR3, PDR5, PDR6, PDR8 to PDRH, PDRJ, PDRL, PDRM, PDRP to PDRS)	186	Timer Control Register (BTnTMCR Upper Byte)	440, 454, 469, 481
Port Function Control Registers (PFR: PFR0 to PFR3, PFR5, PFR6, PFR8, PFR9, PFR10, PFR11 to PFRH, PFRJ, PFRM, PFRQ, PFRS)	195	Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)	312
PPG Operation Mode Control Registers (PPGC0 to PPGC15)	267	Timer State Control Register Lower (TCCSL0 to TCCSL5)	316
PPG Registers	265	Timer State Control Register M (TCCSM0 to TCCSM5)	318
PPG Trigger Register (TRG)	271	Timer State Control Register Upper (TCCSH0 to TCCSH5)	313
Pull-up Control Registers (PCR: PCR0 to PCR3, PCR5, PCR6, PCR8 to PCRH, PCRJ, PCRL, PCRM, PCRP to PCRS)	192	Timing Generator Control Register (TTCR0/TTCR1)	251
Reception Data Register (RDR)	642	Transfer Count Registers and Reload Operation	814
Reception Data Register (RDR0/RDR1)	534, 587	Transmission Data Register (TDR)	643
Register ADMD Setting	757	Transmission Data Register (TDR0/TDR1)	536, 588
Register Details Explanation	228, 231	Up/Down Count Register (UDCR)	502
Register List	218	Waveform Control Register 1 (SIGCR1)	352
Register List of 12-bit A/D Converter	738	Waveform Control Register 2 (SIGCR2)	354
Register List of 8/10-bit A/D Converter	705	Waveform Generator Register	308
Register List of Multiplication and Addition Calculator	766	Wild Register Address Register (WA)	880
Register List of the DMAC Registers	787	Wild Register Data Register (WD)	880
Register Overview of External Bus Interface	131	Wild Register Enable Register (WREN)	880
Register Setting in Differential Input Mode	755	Relationship	
Registers of Timing Generator 0	249	Relationship between Reload Value and Pulse Width	466
Registers of Timing Generator 1	250	Release	
Reload Compare Register (RCR)	503	Operation Initialization Reset (RST) Release Sequence	82
Reload Register Write Timing	463	Set Initialization Reset (INIT) Release Sequence	81
Reload Registers (PRLH0 to PRLH15, PRL0 to PRL15)	270	Reload	
Reload Registers: 16-bit PPG Mode	266	Relationship between Reload Value and Pulse Width	466
Reset Source Register/Watchdog Timer Control Register (RSRR)	97	Reload Operation	810
Serial Control Register (SCR)	524, 576	Reload Values and Baud Rates for Different Peripheral Clock Frequencies	562, 619, 682
Serial Mode Register (SMR)	527, 579, 633	Reload/Compare Function	511
Serial Status Register (SSR)	529, 582, 639		

Synchronous Start of Reload/Compare Function	512	System Initialization Reset (SINIT) Release Sequence.....	81
Transfer Count Registers and Reload Operation	814	Watchdog Reset.....	79
Reload Compare Register Reload Compare Register (RCR)	503	Reset Source Register Reset Source Register/Watchdog Timer Control Register (RSRR)	97
Reload Counter Functions of Reload Counters.....	564, 619, 682	Restart Restart	564, 620
Reload Register Bit Configuration of 16-bit Reload Register (TMRLR)	241	Restarting.....	491
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	460	Restore Backup/Restore Processing.....	234
Reload Register Write Timing	463	Notes If Restoring from STOP Status Performed Using an External Interrupt.....	225
Reload Registers (PRLH0 to PRLH15, PRL0 to PRL15)	270	Restrictions Overall Restrictions and Notes	880
Reload Registers: 16-bit PPG Mode	266	Result Transfer of Arithmetic Result.....	776
Reload Timer 16/32-bit Reload Timer.....	423	Resuming Sector Erasure Resuming Sector Erasure in Flash Memory	859
16-bit Reload Timer	3	Return Return from Standby	222
Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)	425	Return Pointer Return Pointer (RP).....	53
Block Diagram of the 16-bit Reload Timer	236	REVC Output Inversion Register (REVC)	272
Notes on Using the 16-bit PWM/PPG/Reload Timer	435	ROM FR-CPU ROM Mode (Read Only in 32 Bits).....	839
Overview of 16-bit Reload Timer	236	RP Return Pointer (RP).....	53
Reload Timer (ch.1).....	408	RSRR Reset Source Register/Watchdog Timer Control Register (RSRR)	97
Reload Timer Register List of 16-bit Reload Timer Registers.....	237	RST Operation Initialization Reset (RST).....	78
Reload Value Relationship between Reload Value and Pulse Width	466	Operation Initialization Reset (RST) Release Sequence.....	82
Reload Values and Baud Rates for Different Peripheral Clock Frequencies	562, 619, 682	RTO Output Status of RTO0 to RTO5 and GATE.....	394
Repeated Repeated Start Condition for I ² C Bus	657	S	
Reset INITX Terminal Input (System Initialization Reset Terminal)	79	SCR Serial Control Register (SCR)	524, 576
Normal Reset Operation.....	85	System Condition Code Register (SCR)	51
Operation Initialization Reset (RST)	78	Sector Specifying One or More Sectors.....	856
Operation Initialization Reset (RST) Release Sequence	82	Sector Erasing Sector Erasing Procedure	856
Placing the Flash Memory in the Reset State.....	852	Sector Erasure Sector Erasure Command	844
Reset Command.....	842	Sector Erasure Pause Command	845
Reset Factor.....	79	Select Selecting Counting Mode	509
Reset Mode.....	422		
Set Initialization Reset (INIT).....	78		
Set Initialization Reset (INIT) Release Sequence	81		
Synchronous Reset Operation.....	85		
System Initialization Reset (SINIT)	78		

MB91470/480 Series

Selecting the Count Clock.....	489	Slave Address	
Selecting the Operation Mode	490	7-bit Slave Address Mask Register (ISMK).....	644
Selected External Count Clock		7-bit Slave Address Register (ISBA)	645
Selected External Count Clock	374	Outputting a Slave Address	659
Selecting Counting Mode		Slave Address Match Detection	676
Selecting Counting Mode.....	509	Slave Address Mask Register	
Selector		7-bit Slave Address Mask Register (ISMK).....	644
Cautions for Use of Free-run Timer Selector.....	415	Slave Address Register	
Sequence		7-bit Slave Address Register (ISBA)	645
System Initialization Reset (SINIT)		Sleep	
Release Sequence.....	81	Notes on DMA Transfer in Sleep Mode	823
Serial Control Register		Recovery from a Standby Mode	
Serial Control Register (SCR)	524, 576	(Stop or Sleep)	214
Serial Interface		Sleep Mode	
Multi-function Serial Interface	3	Notes on DMA Transfer in Sleep Mode	823
Serial Mode Register		SMR	
Serial Mode Register (SMR)	527, 579, 633	Serial Mode Register (SMR).....	527, 579, 633
Serial On-board Programming		Software Request	
Pins Used for Fujitsu-standard Serial On-board		Software Request.....	807
Programming.....	869	Software Reset	
Serial Programming		STCR: SRST Bit Writing (Software Reset)	80
Basic Configuration for Serial		Source Clock	
Programming.....	868	Selection of the Source Clock	88
Connection Example for MB91F47x/MB91F48x		Specifying	
Serial Programming	870	Specifying One or More Sectors.....	856
Serial Status Register		SPI	
Serial Status Register (SSR).....	529, 582, 639	SPI Transfer (I)	611
Set/Reset Mode		SPI Transfer (II)	614
16-bit Output Compare Operation		SRST	
(Set/Reset Mode, MOD1x=1).....	383	STCR: SRST Bit Writing (Software Reset)	80
SIGCR		SSP	
DTTI Operation of Waveform Control Register 2		System Stack Pointer (SSP)	53, 64
(SIGCR2).....	405	SSR	
Waveform Control Register 1 (SIGCR1)	352	Serial Status Register (SSR)	529, 582, 639
Waveform Control Register 2 (SIGCR2)	354	Standby	
sim911		Recovery from a Standby Mode	
Debugger (sim911, eml911, mon911)	914	(Stop or Sleep)	214
Single Conversion Mode		Return from Standby.....	222
Operation of Single Conversion Mode	719, 751	Standby Control Register	
Single-Chip Mode		Standby Control Register (STCR)	99
Bus Mode 0 (Single-Chip Mode)	73	Standby Mode	
SINIT		Recovery from a Standby Mode	
System Initialization Reset		(Stop or Sleep)	214
(SINIT)	78	Start	
System Initialization Reset (SINIT)		Generating a Start Condition.....	658
Release Sequence.....	81	Repeated Start Condition for I ² C Bus	657
Slave		Start Condition for I ² C Bus.....	657
7-bit Slave Address Mask Register (ISMK)	644	Starting a Count.....	564, 619, 682
7-bit Slave Address Register (ISBA)	645	Starting and Stopping Pulse Width	
Outputting a Slave Address.....	659	Measurement.....	491
Slave Address Match Detection	676	Starting from a Temporary Stop	816
Slave Reception	677	Starting Transfer.....	816
Slave Transmission	679	Synchronous Start of Reload/Compare	
		Function	512

State		Structure	
Pin States in Each CPU State.....	906	Structure of Multi-function Timer.....	282
Placing the Flash Memory in the Reset State.....	852	Successive Accesses	
Timer State Control Register Lower (TCCSL0 to TCCSL5).....	316	Basic Timing (For Successive Accesses) (TYP[3:0]=0000 _B , AWR=0008 _H).....	161
Timer State Control Register M (TCCSM0 to TCCSM5).....	318	Sum-of-Products Circuit	
Timer State Control Register Upper (TCCSH0 to TCCSH5).....	313	Notes on Using the Sum-of-Products Circuit	784
Status		Suppressing	
A/D Control Status Register (ADCS: ADCS1, ADCS2)	712	Suppressing DMA.....	815
A/D Control Status Register (ADCS: ADCS3, ADCS4)	744	Suspending Sector Erasure	
Bit Configuration of Control Status Register (TMCSR)	238	Suspending Sector Erasure in Flash Memory	858
DSP Control/Status Register (DSP-CSR).....	767	Switching	
I ² C Bus Status Register (IBSR)	635	Switching the Interface Mode.....	518
Notes If Restoring from STOP Status Performed Using an External Interrupt	225	Synchronous	
Output Status of RTO0 to RTO5 and GATE	394	Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)	617
Recovery Operations from STOP Status	226	Functions of CSIO (Clock Synchronous Serial Interface)	573
Serial Status Register (SSR)	529, 582, 639	List of Registers of CSIO (Clock Synchronous Serial Interface)	574
Status Control Register		Operation of CSIO (Clock Synchronous Serial Interface)	605
Status Control Register (BTnSTC)	444, 458, 473, 485	Synchronous Reset Operation	85
STCR		Synchronous Start of Reload/Compare Function.....	512
Standby Control Register (STCR)	99	System Condition Code Register	
STCR: SRST Bit Writing (Software Reset).....	80	System Condition Code Register (SCR)	51
Step		System Configuration	
Step Transferring.....	809	System Configuration for AF200 Flash Microcontroller Programmer (Manufactured by Yokogawa Digital Computer Corporation).....	871
Step/Block Transfer		System Stack Pointer	
Step/Block Transfer 2-Cycle Transfer	809	System Stack Pointer (SSP)	53, 64
Stop		T	
Notes If Restoring from STOP Status Performed Using an External Interrupt	225	Table Base Register	
Recovery from a Standby Mode (Stop or Sleep)	214	Table Base Register (TBR)	53, 65
Recovery Operations from STOP Status	226	TBCR	
Setting of Temporary Stopping by Writing to the Control Register (Set Independently for Each Channel or All Channels Simultaneously)	819	Time-base Counter Control Register (TBCR)	102
Starting and Stopping Pulse Width Measurement	491	TBR	
Starting from a Temporary Stop.....	816	Table Base Register (TBR)	53, 65
Stop Condition for I ² C Bus	657	TCCSH	
Stopping.....	491	Timer State Control Register Upper (TCCSH0 to TCCSH5).....	313
Transfer Stop Requests from Peripheral Circuits	821	TCCSL	
Stop Mode		Timer State Control Register Lower (TCCSL0 to TCCSL5)	316
Wait Time after Recovering from Stop Mode.....	93	TCCSM	
STR		Timer State Control Register M (TCCSM0 to TCCSM5)	318
STR Instruction (Transfer Instruction).....	780		

MB91470/480 Series

TCDTH	Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)	312	Timer State Control Register M (TCCSM0 to TCCSM5)	318
TCDTL	Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)	312	Timer State Control Register Upper (TCCSH0 to TCCSH5).....	313
TDR	Transmission Data Register (TDR)	643	Timing	
	Transmission Data Register (TDR0/TDR1)	536, 588	16-bit Output Compare Timing	384
Temporary Stop	Starting from a Temporary Stop	816	Auto-Wait Timing (TYP[3:0]=0000 _B , AWR=2008 _H)	164
Temporary Stopping	Setting of Temporary Stopping by Writing to the Control Register (Set Independently for Each Channel or All Channels Simultaneously).....	819	Basic Timing (For Successive Accesses) (TYP[3:0]=0000 _B , AWR=0008 _H)	161
Time-base Counter	Time-base Counter	112	Configuration of Timing Generator	246
Time-base Counter Clear Register	Time-base Counter Clear Register (CTBR).....	105	Differences between Timing Generators 0 and 1	246
Time-base Counter Control Register	Time-base Counter Control Register (TBCR)	102	External Wait Timing (TYP[3:0]=0001 _B , AWR=2008 _H)	165
Timer	Timer Clear	368	Input Timing for 16-bit Input Capture.....	393
	Timer Interrupt	371	Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	467
	Timer Mode	368	Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	451
Timer Clear	Timer Clear	368	Interrupt Generation Timing	516
Timer Control Register	Timer Control Register (BTnTMCR Lower Byte)	442, 456, 471, 483	Normal Transfer (II) Timing Chart	609
	Timer Control Register (BTnTMCR Upper Byte).....	440, 454, 469, 481	Occurrence of Reception Interrupts and Flag Set Timing.....	550, 600
Timer Data Register	Timer Data Register (TCDTH0 to TCDTH5, TCDTL0 to TCDTL5)	312	Occurrence of Transmission Interrupts and Flag Set Timing.....	553, 603
Timer Function Mode	Function Mode Bit Settings and Timer Function Modes Assigned	422	Read ->Write Timing (TYP[3:0]=0000 _B , AWR=0048 _H)	162
Timer Interrupt	Timer Interrupt	371	Registers of Timing Generator 0	249
Timer Mode	Note in Using Dead Time Timer Mode	403	Registers of Timing Generator 1	250
	Operation during Dead Time Timer Mode	401	Reload Register Write Timing	463
	Operation of Timer Mode	399	Timing for Clearing Interrupts During DMA	818
	Timer Mode	368	Write ->Write Timing (TYP[3:0]=0000 _B , AWR=0018 _H)	163
Timer Register	Bit Configuration of the Timer Register (BTnTMR).....	448, 462, 476	Timing Chart	
Timer State Control Register	Timer State Control Register Lower (TCCSL0 to TCCSL5)	316	Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	467
			Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	451
			Normal Transfer (II) Timing Chart	609
			Timing Generator	
			Registers of Timing Generator 0	249
			Registers of Timing Generator 1	250
			Timing Generator	3
			Timing Generator Control Register (TTCR0/TTCR1).....	251
			Timing Generator Control Register Timing Generator Control Register (TTCR0/TTCR1).....	251
			TMCSR	
			Bit Configuration of Control Status Register (TMCSR).....	238
			TMR	
			Bit Configuration of 16-bit Timer Register (TMR)	240

TMRLR		
Bit Configuration of 16-bit Reload Register (TMRLR)	241	
TMRRH		
16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)	345	
TMRRL		
16-bit Dead Timer Register (TMRRH0 to TMRRH5, TMRRL0 to TMRRL5)	345	
Transfer		
Block Transfer	809	
Burst 2-Cycle Transfer	808	
DMA Transfer and Interrupts	815	
Flow of Data During 2-Cycle Transfer	828	
Notes on DMA Transfer in Sleep Mode	823	
Operation Flowchart for Block Transfer	826	
Operation Flowchart for Burst Transfer	827	
Selection of the Transfer Sequence	808	
SPI Transfer (I)	611	
SPI Transfer (II)	614	
Starting Transfer	816	
Step/Block Transfer 2-Cycle Transfer	809	
STR Instruction (Transfer Instruction)	780	
Transfer Address	806	
Transfer Count and Transfer End	806	
Transfer Count Registers and Reload Operation	814	
Transfer End	820	
Transfer Mode	805	
Transfer of Arithmetic Result	776	
Transfer Request Acceptance and Transfer	817	
Transfer Stop Requests from Peripheral Circuits	821	
Transfer Type	806	
Transfer Count Registers		
Transfer Count Registers and Reload Operation	814	
Transfer Instruction		
STR Instruction (Transfer Instruction)	780	
Transfer Request		
Transfer Request Acceptance and Transfer	817	
Transition		
State of Device and Each Transition	116	
Transmission		
Master Data Transmission	665	
Occurrence of Transmission Interrupts and Flag Set Timing	553, 603	
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing	554, 604	
Slave Transmission	679	
Transmission Data Register (TDR)	643	
Transmission Data Register (TDR0/TDR1)	536, 588	
Transmission/Reception FIFO	519	
Transmission Data Register		
Transmission Data Register (TDR)	643	
Transmission Data Register (TDR0/TDR1)	536, 588	
TRG		
PPG Trigger Register (TRG)	271	
Trigger		
Triggers for the Oscillation Stabilization Wait	83	
TTCR		
Timing Generator Control Register (TTCR0/TTCR1)	251	
TYP		
Auto-Wait Timing (TYP[3:0]=0000 _B , AWR=2008 _H)	164	
Basic Timing (For Successive Accesses) (TYP[3:0]=0000 _B , AWR=0008 _H)	161	
CSX ->RDX/WR0X, WR1X Setup (TYP[3:0]=0101 _B , AWR=100B _H)	170	
CSX Delay Setting (TYP[3:0]=0000 _B , AWR=000C _H)	166	
External Wait Timing (TYP[3:0]=0001 _B , AWR=2008 _H)	165	
Read ->Write Timing (TYP[3:0]=0000 _B , AWR=0048 _H)	162	
Setting of CSX ->RDX/WR0X, WR1X Setup and of RDX/WR0X, WR1X ->CSX Hold (TYP[3:0]=0000 _B , AWR=000B _H)	167	
With External Wait (TYP[3:0]=0101 _B , AWR=1008 _H)	169	
Without External Wait (TYP[3:0]=0100 _B , AWR=0008 _H)	168	
Write ->Write Timing (TYP[3:0]=0000 _B , AWR=0018 _H)	163	
U		
UART		
Functions of UART (Asynchronous Serial Interface)	521	
Interrupts of UART	548	
List of Registers of UART (Asynchronous Serial Interface)	522	
Operation of UART	555	
UART Baud Rate Selection	560	
UDCR		
Up/Down Count Register (UDCR)	502	
Writing Data to UDCR	513	
Underflow Operation		
Underflow Operation	243, 478	
Up/Down Count Register		
Up/Down Count Register (UDCR)	502	
Up/Down Counter		
Block Diagram of Up/Down Counter	500	
Features of Up/Down Counter	498	
List of Registers of Up/Down Counter	501	

MB91470/480 Series

Up-Down Counter	
8/16-bit Up-Down Counter	3
User Stack Pointer	
User Stack Pointer (USP).....	54
USP	
User Stack Pointer (USP).....	54
V	
Variable	
DSP Variable Monitor Registers	
(DSP-OT0 to DSP-OT7).....	772
Variable Monitor Output	
Variable Monitor Output.....	777
Vector	
Interrupt Vector Table	902
Vector Table	
EIT Vector Table	65
Interrupt Vector Table	902
Vector Table Initial Area	57
W	
WA	
Bit Configuration of Wild Register Address Register	
(WA)	877
Wild Register Address Register (WA).....	880
Wait	
Wait in Master Mode.....	675
Wait Time	
Wait Time after Changing the PLL Multiplier	
Ratio.....	92
Wait Time after Enabling PLL Operation	92
Wait Time after Power Supply is Turned on	92
Wait Time after Recovering from Stop Mode	93
Wait Time after System/Setting is Initialized	92
Watchdog	
Watchdog Reset	79
Watchdog Timer Control Register	
Reset Source Register/Watchdog Timer Control	
Register (RSRR)	97
Waveform Control Register	
DTTI Operation of Waveform Control Register 2	
(SIGCR2).....	405
Waveform Control Register 1 (SIGCR1)	352
Waveform Control Register 2 (SIGCR2)	354
Waveform Generator	
Block Diagram of the Waveform Generator.....	295
Notes on Using the Waveform Generator	416
Waveform Generator Interrupts	366
Waveform Generator Register	308
Waveform Generator Register	
Waveform Generator Register	308

WD	
Bit Configuration of Wild Register Data Register	
(WD).....	878
Wild Register Data Register (WD)	880
Wild Register	
Block Diagram of Wild Register Control	
Block.....	874
List of Registers in Wild Register Control	
Block.....	875
Operations of Wild Register Control Block	879
Overview of Wild Register Control Block.....	874
Wild Register Address Register (WA)	880
Wild Register Data Register (WD)	880
Wild Register Enable Register (WREN)	880
Wild Register Function	4
Wild Register Address Register	
Bit Configuration of Wild Register Address Register	
(WA).....	877
Wild Register Address Register (WA)	880
Wild Register Data Register	
Bit Configuration of Wild Register Data Register	
(WD).....	878
Wild Register Data Register (WD)	880
Wild Register Enable Register	
Bit Configuration of Wild Register Enable Register	
(WREN)	876
Wild Register Enable Register (WREN)	880
With External Wait	
With External Wait	
(TYP[3:0]=0101 _B , AWR=1008 _H)	169
Without External Wait	
Without External Wait	
(TYP[3:0]=0100 _B , AWR=0008 _H)	168
Word	
Word Alignment.....	56
Word Access	
Word Access.....	157
WR	
CSX ->RDX/WR0X, WR1X Setup	
(TYP[3:0]=0101 _B , AWR=100B _H).....	170
Setting of CSX ->RDX/WR0X, WR1X Setup and of	
RDX/WR0X, WR1X ->CSX Hold	
(TYP[3:0]=0000 _B , AWR=000B _H).....	167
WREN	
Bit Configuration of Wild Register Enable Register	
(WREN)	876
Wild Register Enable Register (WREN)	880
Write	
Delayed Write Function	776
FR-CPU Programming Mode	
(Read/Write in 16 Bits)	840
Read ->Write Timing	
(TYP[3:0]=0000 _B , AWR=0048 _H)	162
Reload Register Write Timing	463

Setting of Temporary Stopping by Writing to the Control Register (Set Independently for Each Channel or All Channels Simultaneously)	819
STCR: SRST Bit Writing (Software Reset)	80
Write ->Write Timing (TYP[3:0]=0000 _B , AWR=0018 _H)	163

Writing Data Writing Data to UDCR	513
--	-----

Z

Zero Detection A/D Activation by Zero Deetction of Free-run Timer or Compare Clear	408
--	-----

CM71-10134-8E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL
32-BIT MICROCONTROLLER
FR60
MB91470/480 Series
HARDWARE MANUAL

June 2011 the eighth edition

Published **FUJITSU SEMICONDUCTOR LIMITED**
Edited Sales Promotion Dept.
