



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.

FR30
32-BIT MICROCONTROLLER
MB91F127/128
HARDWARE MANUAL

FR30
32-BIT MICROCONTROLLER
MB91F127/128
HARDWARE MANUAL

PREFACE

■ Objectives and Intended Readers

Thank you very much for using our semiconductor products.

MB91F127/128 has been developed as one of the "32-bit single-chip microcontroller FR30 series" with a new RISC architecture CPU as its core. The device is optimized for embedded applications that require high-performance CPU processing power.

This manual describes the functions and operations of MB91F127/128 for engineers engaged in the development of products using this MB91F127/128. It is important to read this manual thoroughly.

For details of the instructions, see the *"Instruction Manual."*

■ Trademarks

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

Embedded Algorithm TM is a trademark of Advanced Micro Devices, Inc.

■ Organization of this manual

This manual is organized into 17 chapters and an appendix, as listed below.

CHAPTER 1 "OVERVIEW"

Chapter 1 provides an overview of the device including the features of MB91F127/128, block diagrams, and functional outlines.

CHAPTER 2 "HANDLING THE DEVICE"

Chapter 2 explains the handling of the pins of the MB91F127/128 and precautions to take when handling the power supply.

CHAPTER 3 "CPU"

Chapter 3 provides fundamental information such as the architecture, specifications, and instructions so you can become familiar with the functions of the CPU core of the FR series.

CHAPTER 4 "BUS INTERFACE"

Chapter 4 explains the basics of the bus interface, its register configurations and functions, bus operation, and bus timing. A program example of bus operation is included.

CHAPTER 5 "I/O PORTS"

Chapter 5 gives general outlines of the I/O port, its register configuration, and conditions for using an external pin as an I/O port.

CHAPTER 6 "16-BIT RELOAD TIMER"

Chapter 6 gives a general outline of the 16-bit reload timer, its register configurations and functions, and its operation.

CHAPTER 7 "PROGRAMMABLE PULSE GENERATOR (PPG) TIMER"

Chapter 7 gives a general outline of the PPG timer, its register configurations and functions, and its operation.

CHAPTER 8 "MULTIFUNCTIONAL TIMER"

Chapter 8 gives a general outline of the multifunctional timer, its register configurations and functions, and its operation.

CHAPTER 9 "U-TIMER"

Chapter 9 gives a general outline of the U-TIMER, its register configurations and functions, and its operation.

CHAPTER 10 "EXTERNAL INTERRUPT"

Chapter 10 gives a general outline of the external interrupt controller, its register configurations and functions, and its operation.

CHAPTER 11 "DELAYED INTERRUPT MODULE"

Chapter 11 gives a general outline of the delayed interrupt module, its register configurations and functions, and its operation.

CHAPTER 12 "INTERRUPT CONTROLLER"

Chapter 12 gives a general outline of the interrupt controller, its register configurations and functions, its operation, and an example of using the hold request withdrawal request function.

CHAPTER 13 "A/D CONVERTER (SUCCESSIVE APPROXIMATION TYPE)"

Chapter 13 gives a general outline of the A/D converter, its register configurations and functions, and its operation.

CHAPTER 14 "UART"

Chapter 14 gives a general outline of the UART, its register configurations and functions, and its operation.

CHAPTER 15 "DMA CONTROLLER (DMAC)"

Chapter 15 gives a general outline of the DMA controller (DMAC), its register configurations and functions, and its operation.

CHAPTER 16 "BIT SEARCH MODULE"

Chapter 16 gives a general outline of the bit search module, its register configurations and functions, its operation, and an example of save/restore processing.

CHAPTER 17 "FLASH MEMORY"

Chapter 17 describes the functions and operation of flash memory.

This chapter includes the case in which flash memory is used from the FR-CPU. For details of when this flash memory is used with a ROM writer, see the user's manual of the ROM writer provided separately.

"APPENDIX"

The appendix provides I/O maps, interrupt vectors, and pin statuses for each CPU state.

- The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
Customers are advised to consult with FUJITSU 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 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 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 or any third party or does Fujitsu warrant non-infringement of any third-party's intellectual property right or other right by using such information. Fujitsu 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 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.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

READING THIS MANUAL

■ Notations in this manual

The following table lists the principal terms used in this manual.

Term	Meaning
I-BUS	16-bit bus for internal instructions. Since the FR series adopts the internal Harvard architecture, the bus for instructions is independent of the data bus. A bus converter is connected to I-BUS.
D-BUS	Internal 32-bit width data bus. Internal resources are connected to D-BUS
C-BUS	Internal multiplex bus. C-BUS is connected to I-BUS and D-BUS through a switch. An external interface module is connected to the C-BUS. Data and instructions are multiplexed in the external data bus.
R-BUS	Internal 16-bit width data bus. R-BUS is connected to D-BUS through an adapter. Various I/O units, the clock generator, and interrupt controller are connected to the R-BUS. Since the R-BUS has a 16-bit width and addresses and data are multiplexed, the CPU requires multiple cycles to access these resources.
E-unit	Execution unit
ϕ	System clock. This clock is output from the clock generator to each built-in resource connected to the R-BUS. At its fastest, this clock has the same frequency as the oscillation and is divided into 1, 1/2, 1/4, and 1/8 (or 1/2, 1/4, 1/8, and 1/16) by PCK1, 0 of the clock generator gear control register (GCR).
θ	System clock. This is the operating clock for resources connected to any bus other than the R-BUS and the CPU. At its fastest, this clock has the same frequency as the oscillation and is divided into 1, 1/2, 1/4, and 1/8 (or 1/2, 1/4, 1/8, and 1/16) by CCK1, 0 of the clock generator gear control register (GCR).

CONTENTS

CHAPTER 1	OVERVIEW	1
1.1	MB91F127/128 Features	2
1.2	MB91F127/128 Block Diagram	5
1.3	Package Dimensions	6
1.4	Pin Assignment	7
1.5	Pin Description	8
1.6	I/O Circuit Type	13
CHAPTER 2	HANDLING THE DEVICE	15
2.1	Precautions on Handling the Device	16
2.2	Precautions on Handling Power Supplies	18
CHAPTER 3	CPU	19
3.1	Memory Space	20
3.2	Internal Architecture	23
3.3	Dedicated Registers	26
3.3.1	Program Status Register (PS)	30
3.4	General-purpose Registers	34
3.5	Data Structure	35
3.6	Word Alignment	36
3.7	Memory Map	37
3.8	Overview of Instructions	39
3.8.1	Branch Instructions with Delay Slot	41
3.8.2	Branch Instructions without Delay Slot	44
3.9	Exception, Interrupt, and Trap (EIT)	45
3.9.1	EIT Interrupt Level	47
3.9.2	Interrupt Control Register (ICR)	49
3.9.3	System Stack Pointer (SSP)	50
3.9.4	Table Base Register (TBR)	51
3.9.5	70Concurrent EIT Processing	53
3.9.6	EIT Operations	55
3.10	Reset Sequence	59
3.11	Clock Generator and Controller	60
3.11.1	Reset Cause Register (RSRR) and Watchdog Cycle Control Register (WTCR)	62
3.11.2	Standby Control Register (STCR)	64
3.11.3	DMA Request Suppression Register (PDRR)	66
3.11.4	Time-based Timer Clear Register (CTBR)	67
3.11.5	Gear Control Register (GCR)	68
3.11.6	Watchdog Reset Suspending Register (WPR)	71
3.11.7	PLL Control Register (PCTR)	72
3.11.8	Watchdog Function	74
3.11.9	Gear Function	76
3.11.10	Reset Cause Holding	78
3.11.11	DMA Suppression	80

3.11.12 Clock doubler function	82
3.11.13 Examples of PLL Clock Settings	84
3.12 Standby Mode (Low-power Consumption Mechanism)	87
3.12.1 Stop Status	89
3.12.2 Sleep Status	92
3.12.3 Status Switch in Standby Mode	95
3.13 Operation Mode	96
CHAPTER 4 BUS INTERFACE	99
4.1 Outline of Bus Interface	100
4.2 Bus Interface Block Diagram	103
4.3 Bus Interface Registers	104
4.3.1 Area Selection Register (ASR) and Area Mask Register (AMR)	105
4.3.2 Area Mode Register 0 (AMD0)	109
4.3.3 Area Mode Register 1 (AMD1)	111
4.3.4 Area Mode Register 32 (AMD32)	113
4.3.5 Area Mode Register 4 (AMD4)	114
4.3.6 Area Mode Register 5 (AMD5)	115
4.3.7 External pin control register 0 (EPCR0)	116
4.3.8 External Pin Control Register 1 (EPCR1)	120
4.3.9 Little Endian Register (LER)	121
4.4 Bus Operations	122
4.4.1 Relationship Between Data Bus Widths and Control Signals	123
4.4.2 Big Endian Bus Access	124
4.4.3 Little Endian Bus Access	130
4.4.4 Comparison of External Access Operations of Big Endian and Little Endian Type	134
4.5 Bus Timings	140
4.5.1 Basic Read Cycle	141
4.5.2 Basic Write Cycle	143
4.5.3 Read Cycle of Each Mode	145
4.5.4 Write Cycle of Each Mode	147
4.5.5 Read/Write Mixed Cycles	149
4.5.6 Automatic Wait Cycle	150
4.5.7 External Wait Cycle	151
4.5.8 Time-division I/O Interface	152
4.5.9 External Bus Requests	154
4.6 Internal Clock Frequency Multiplier Operations (Clock Doubler)	155
4.7 Program Example of External Bus Operations	156
CHAPTER 5 I/O PORTS	159
5.1 Outline of I/O Ports	160
5.2 Port Data Registers (PDR)	161
5.3 Data Direction Registers (DDR)	162
5.4 Using External Pins as I/O Ports	163
CHAPTER 6 16-BIT RELOAD TIMER	167
6.1 Outline of the 16-Bit Reload Timer	168
6.2 16-Bit Reload Timer Registers	169

6.2.1	Control Status Register (TMCSR)	170
6.2.2	16-Bit Timer Register (TMR)	173
6.2.3	16-Bit Reload Register (TMRLR)	174
6.3	16-Bit Reload Timer Operations	175
6.4	Counter Operation States	178
CHAPTER 7 PROGRAMMABLE PULSE GENERATOR (PPG) TIMER		179
7.1	Outline of the PPG Timer	180
7.2	Block Diagram of the PPG Timer	181
7.3	Registers of the PPG Timer	183
7.3.1	Control Status Register (PCNH, PCNL)	184
7.3.2	PPG Cycle Setting Register (PCSR)	188
7.3.3	PPG Duty Setting Register (PDUT)	189
7.3.4	PPG Timer Register (PTMR)	190
7.3.5	General Control Register 1 (GCN1)	191
7.3.6	General Control Register 2 (GCN2)	194
7.4	PPG Mode	195
7.5	One-shot Mode	197
7.6	Interrupts	199
7.7	PPG Output of All-L and All-H	200
7.8	Activation of Multiple Channels of PPG Timer	201
CHAPTER 8 MULTIFUNCTIONAL TIMER		203
8.1	Overview of the Multifunctional Timer	204
8.2	Block Diagram of the Multifunctional Timer Unit	205
8.3	Registers of the Multifunctional Timer Unit	206
8.3.1	Register of the 16-bit free-running timer	207
8.3.2	Register of the Output Compare Unit	211
8.3.3	Register of the Input Capture Unit	215
8.4	Operation of the Multifunctional Timer Unit	217
8.4.1	Operation of the 16-bit Free-running Timer Unit	218
8.4.2	Operation of the 16-bit Output Compare Unit	220
8.4.3	Operation of the 16-bit Input Capture Unit	223
CHAPTER 9 U-TIMER		225
9.1	Outline of U-TIMER	226
9.2	U-TIMER Registers	227
9.3	U-TIMER Operation	230
CHAPTER 10 EXTERNAL INTERRUPT		233
10.1	Overview of the External Interrupt	234
10.2	External Interrupt Controller Registers	235
10.2.1	Interrupt Enable Register (ENIR)	236
10.2.2	External Interrupt Source Register (EIRR)	237
10.2.3	External Interrupt Request Level Setting Register (ELVR, EHVR)	238
10.3	Operation of the External Interrupt Controller	239
CHAPTER 11 DELAYED INTERRUPT MODULE		241

11.1	Outline of the Delayed Interrupt Module	242
11.2	Delayed Interrupt Module Register	243
11.3	Delayed Interrupt Module Operation	244
CHAPTER 12 INTERRUPT CONTROLLER		245
12.1	Overview of the Interrupt Controller	246
12.2	Interrupt Controller Registers	248
12.2.1	Interrupt Control Register (ICR)	250
12.2.2	Hold Request Cancellation Request Level Setting Register (HRCL)	252
12.3	Interrupt Controller Operation	253
12.4	Example of Using the Hold Request Cancellation Request Function (HRCR)	258
CHAPTER 13 A/D CONVERTER (SUCCESSIVE APPROXIMATION TYPE)		261
13.1	Outline of the A/D Converter	262
13.2	A/D Converter Registers	264
13.2.1	Control status register (ADCS)	265
13.2.2	Data register (ADCR)	271
13.2.3	Analog Input Control Register (AIC)	273
13.3	A/D Converter Operation	274
13.4	Conversion Data Protection Function	276
13.5	Notes on Using the A/D Converter	278
CHAPTER 14 UART		279
14.1	Outline of the UART	280
14.2	UART Registers	282
14.2.1	Serial Mode Register (SMR)	283
14.2.2	Serial Control Register (SCR)	285
14.2.3	Serial Input Data Register (SIDR) and Serial Output Data Register (SODR)	288
14.2.4	Serial Status Register (SSR)	289
14.3	UART Operation	292
14.3.1	Asynchronous Modes	293
14.3.2	CLK Synchronous Mode	294
14.3.3	UART Interrupts and Timings for Setting Flags	296
14.4	Example of Using UART	299
14.5	Sample Settings for Baud Rate and U-TIMER Reload Value	301
CHAPTER 15 DMA CONTROLLER (DMAC)		303
15.1	Outline of the DMA Controller	304
15.2	DMA Controller Block Diagram	305
15.3	DMA Controller Registers	306
15.3.1	DMAC Parameter Descriptor Pointer (DPDP)	307
15.3.2	DMAC Control Status Register (DACSR)	308
15.3.3	DMAC Pin Control Register (DATCR)	311
15.3.4	RAM Descriptor-Internal Register	314
15.4	DMA Controller Transfer Modes	317
15.5	DMA Controller Timing Diagrams	320
15.5.1	Timing Diagrams of the Descriptor Access Block	321
15.5.2	Timing Diagrams of the Data Transfer Block	323

15.5.3	Timing Diagrams of Transfer Stop in Continuous Transfer Mode	325
15.5.4	Timing Diagrams of Transfer End	328
15.6	Notes on the DMA Controller	330
CHAPTER 16	BIT SEARCH MODULE	333
16.1	Outline of the Bit Search Module	334
16.2	Bit Search Module Registers	335
16.3	Bit Search Module Operation and Saving/Restoring	337
CHAPTER 17	FLASH MEMORY	341
17.1	Outline of Flash Memory	342
17.2	Flash Memory Registers	348
17.2.1	Flash Memory Status Register (FSTR)	349
17.2.2	Flash Memory Wait Register (FWTC)	351
17.3	Flash Memory Access Modes	352
17.4	Starting the Automatic Algorithm	355
17.5	Execution Status of the Automatic Algorithm	359
17.6	Sector Protect Operation	364
APPENDIX	367
APPENDIX A	I/O MAP	368
APPENDIX B	INTERRUPT VECTORS	376
APPENDIX C	PIN STATES FOR EACH CPU STATE	379
INDEX	387

CHAPTER 1 OVERVIEW

This chapter describes the basic information items necessary to obtain an overall knowledge of the MB91F127/128, such as its features, its block diagram and functions.

- 1.1 "MB91F127/128 Features"
- 1.2 "MB91F127/128 Block Diagram"
- 1.3 "Package Dimensions"
- 1.4 "Pin Assignment"
- 1.5 "Pin Description"
- 1.6 "I/O Circuit Type"

1.1 MB91F127/128 Features

The MB91F127/128 is a standard single-chip microcontroller that employs a 32-bit RISC CPU (FR series) as core. The MB91F127/128 incorporates bus control mechanisms and various I/O resources for embedded control as required for allowing high-performance, high-speed CPU processing.

The MB91F127 contains 256K bytes of flash memory and 14K bytes of RAM. The MB91F128 contains 510K bytes of flash memory and 14K bytes of RAM.

The MB91F127/128 provides ideal specifications for embedded use in systems requiring high CPU performance, such as a navigation system, high-performance FAX, and printer control.

■ FR-CPU

- 32-bit RISC, load/store architecture, and 5-stage pipeline
- Operating frequency: Internal 25 MHz
- General-purpose registers: 32 bit x 16
- 16-bit fixed-length instructions (basic instructions), one instruction per cycle
- Such instructions as memory-to-memory transfer, bit processing, and barrel shift, which are ideal for embedded use
- High-level language instructions, such as instruction for function entry and exit and instructions for register
- Register interlock function, which facilitates assembler coding
- Branch instructions with delayed slot, which reduce overhead during branch processing
- Support at the built-in and instruction level of the multiplier unit
 - Signed 32-bit multiplication in 5 cycles
 - Signed 16-bit multiplication in 3 cycles
- Interrupts (PC and PS save) at 6 cycles and 16 priority levels

■ Bus interface

- Maximum operating frequency: Internal 25 MHz
- 25-bit address bus (32 Mbytes space)
- 16-bit address output and 8/16-bit data input-output
- Basic bus cycle: 2 clock cycles
- 6 chip select outputs that can be set in minimum units of 64 Kbytes
- Automatic wait cycle supports optional selection of 0 to 7 cycles for each area.
- The time-division I/O interface of addresses/data is supported (only area 1).
- Unused data and address pins can be used as I/O pins.
- Little endian mode supported (Select one of areas 1 to 5.)

■ Internal RAM: 14 KB

- D-bus RAM: 12 KB, C-bus RAM: 2 KB

■ Reload timer

- 16-bit timer: 3 channels
- Internal clock: 2-clock cycle resolution or 2/8/32 cycle and external clock can be selected.

■ Other interval timers

- 16-bit timer: 3 channels (U-TIMER)
- PPG timer: 4 channels
- 16-bit OCU: 4 channels; ICU: 4 channels; free-run timer: 1 channel
- Watchdog timer: 1 channel

■ Interrupt controller

- External interrupt input: Normal interrupt pins x 6 (INT0 to INT5)
- Internal interrupt causes: UART, DMAC, A/D, reload timer, UTIMER, and delayed interrupts, PPG, ICU, OCU
- The priority levels can be programmed (16 levels).

■ A/D converter (successive approximation conversion type)

- 8/10-bit resolution and 8 channels
- Successive approximation conversion: 5.2 μ s during operation at 25 MHz
- Internal sample and hold circuits
- Conversion mode: Single conversion, scan conversion, or repeat conversion can be selected.
- Activation: Software, external trigger, or internal timer can be selected.

■ UART

- 3 channels
- Full and half duplex double buffers
- Data lengths: 7 to 9 bits (without parity) and 6 to 8 bits (with parity)
- Asynchronous (start-stop synchronization) and CLK synchronous communication can be selected.
- Multiprocessor mode
- Using an internal 16-bit timer (U-TIMER) as baud rate generator enables creation of arbitrary baud rates.
- External clocks can be used as transfer clocks.
- Error detection for parity, frame, and overrun

■ DMA controller (DMAC)

- 8 channels

CHAPTER 1 OVERVIEW

- Transfer causes: Internal resource interrupt requests
- Transfer sequences: Step transfer, block transfer, burst transfer and continuous transfer
- Transfer data length: 8-bit, 16-bit, and 32-bit can be selected.
- Can be stopped temporarily using interrupt

■ Bit search module

- The position of the first bit change ("1" interchanged with "0") from the MSB in a word is searched for within one cycle.

■ Reset causes

- Power-on reset, watchdog timer, software reset, and external reset

■ Low-power consumption mode

- Sleep and stop mode

■ Clock control

- Internal PLL circuit, multiply-by-one, or multiply-by-two can be selected.
- Gear function: An arbitrary operating clock frequencies of the CPU and peripherals can be set independently. The gear lock can be selected from 1/1, 1/2, 1/4, and 1/8 (or 1/2, 1/4, 1/8, and 1/16). However, the operating clock frequency of the peripherals is restricted to an upper limit of 25 MHz.

■ Flash memory

- 256-KB flash ROM (MB91F127)/510-KB flash ROM (MB91F128): Read/write/erase is enabled with a single power source.

■ Other

- Package: LQFP-100
- CMOS technology: 0.35 μm
- Power supply: 3.3 V plus or minus 0.3 V

■ Series configuration

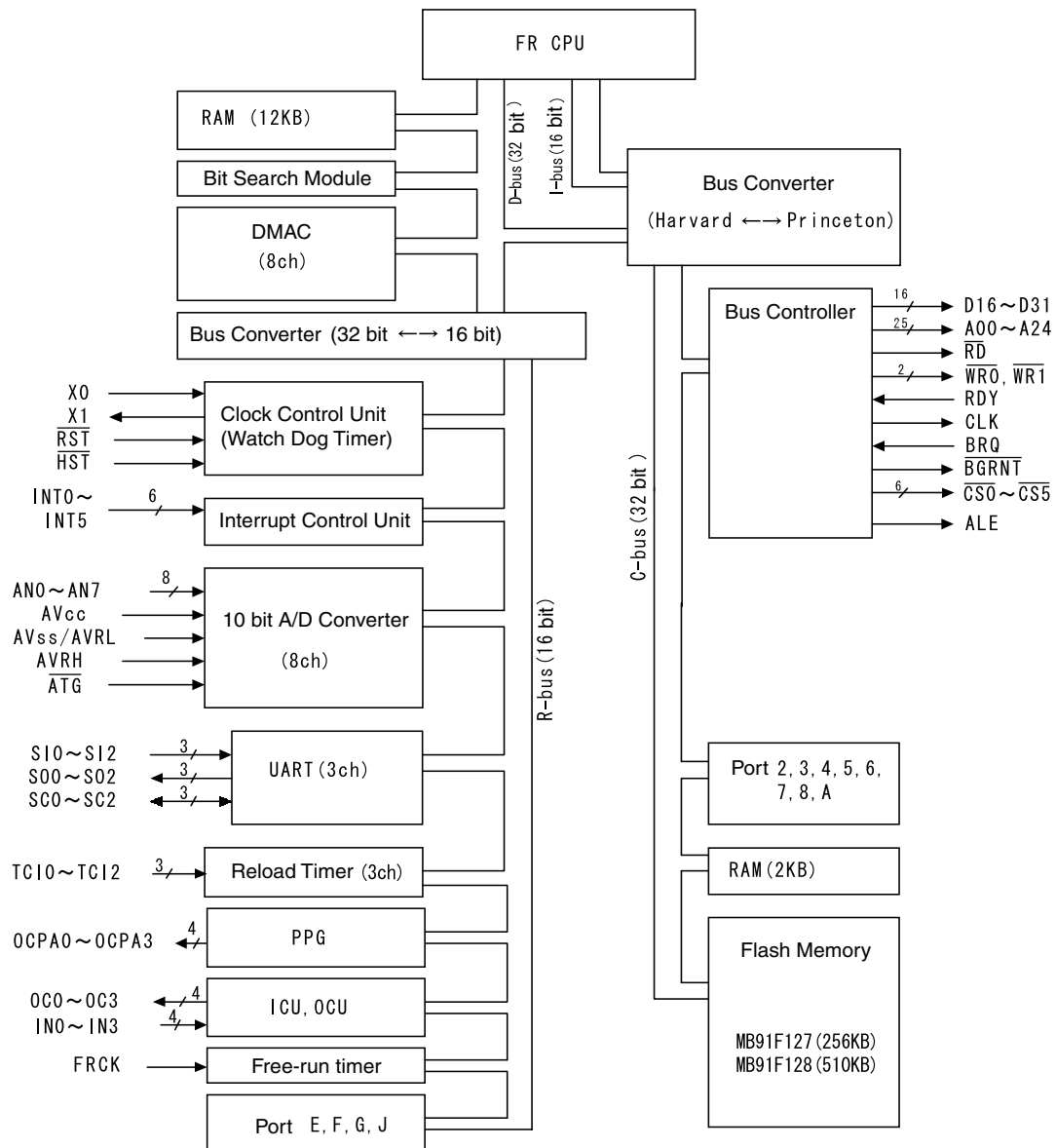
Model name	MB91FV129	MB91F127	MB91F128
Description	Evaluation chip	Mass production type	Mass production type
Flash memory	510KB	256KB	510KB
D-bus RAM	16KB	12KB	12KB
C-bus RAM	2KB	2KB	2KB

1.2 MB91F127/128 Block Diagram

Figure 1.2-1 "MB91F127/128 Block Diagram" shows a block diagram of the MB91F127/128.

■ MB91F127/128 Block Diagram

Figure 1.2-1 MB91F127/128 Block Diagram



Note:

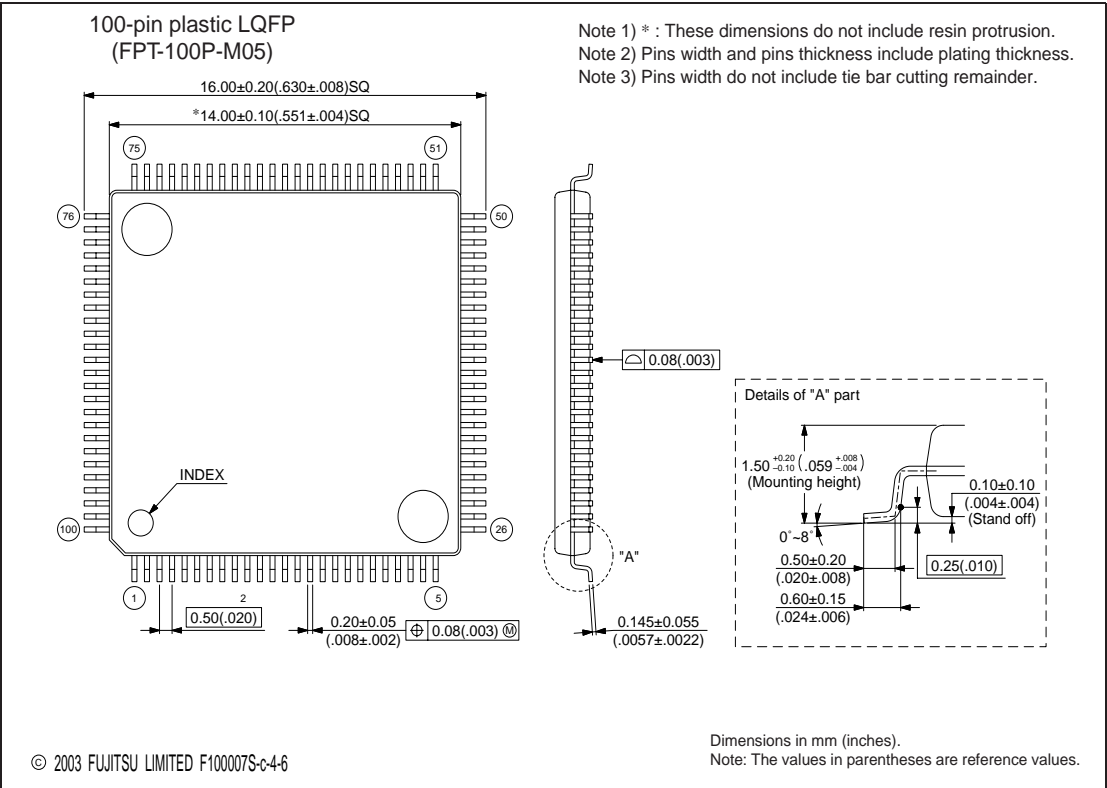
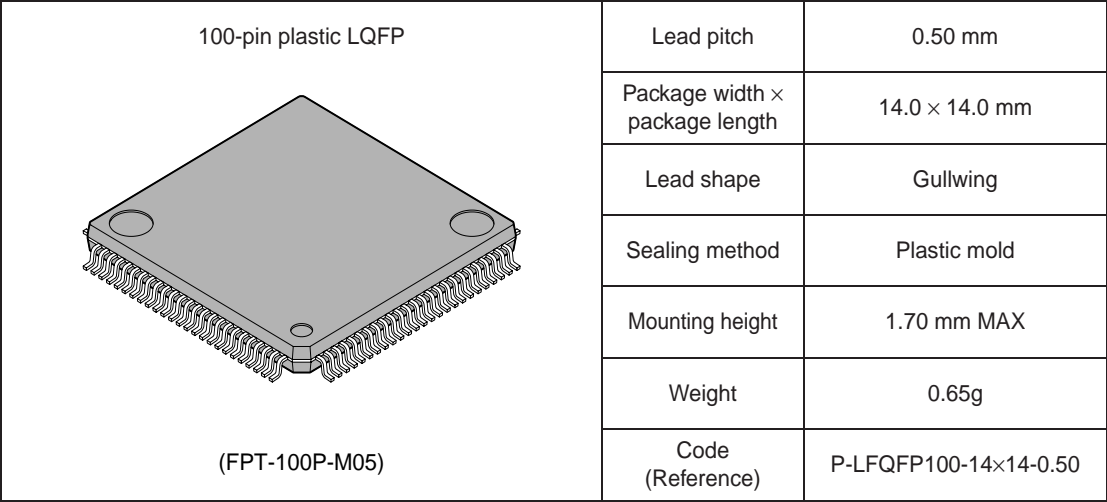
- Rather than the actual pins, their function is displayed (some of the actual pins are multiplexed).
- To use REALOS, use external interrupts or internal timers for time management.

1.3 Package Dimensions

Figure Figure 1.3-1 "Package Dimensions of FPT-100P-M05" shows a package dimensions of the MB91F127/128.

■ Package Dimensions of FPT-100P-M05

Figure 1.3-1 Package Dimensions of FPT-100P-M05

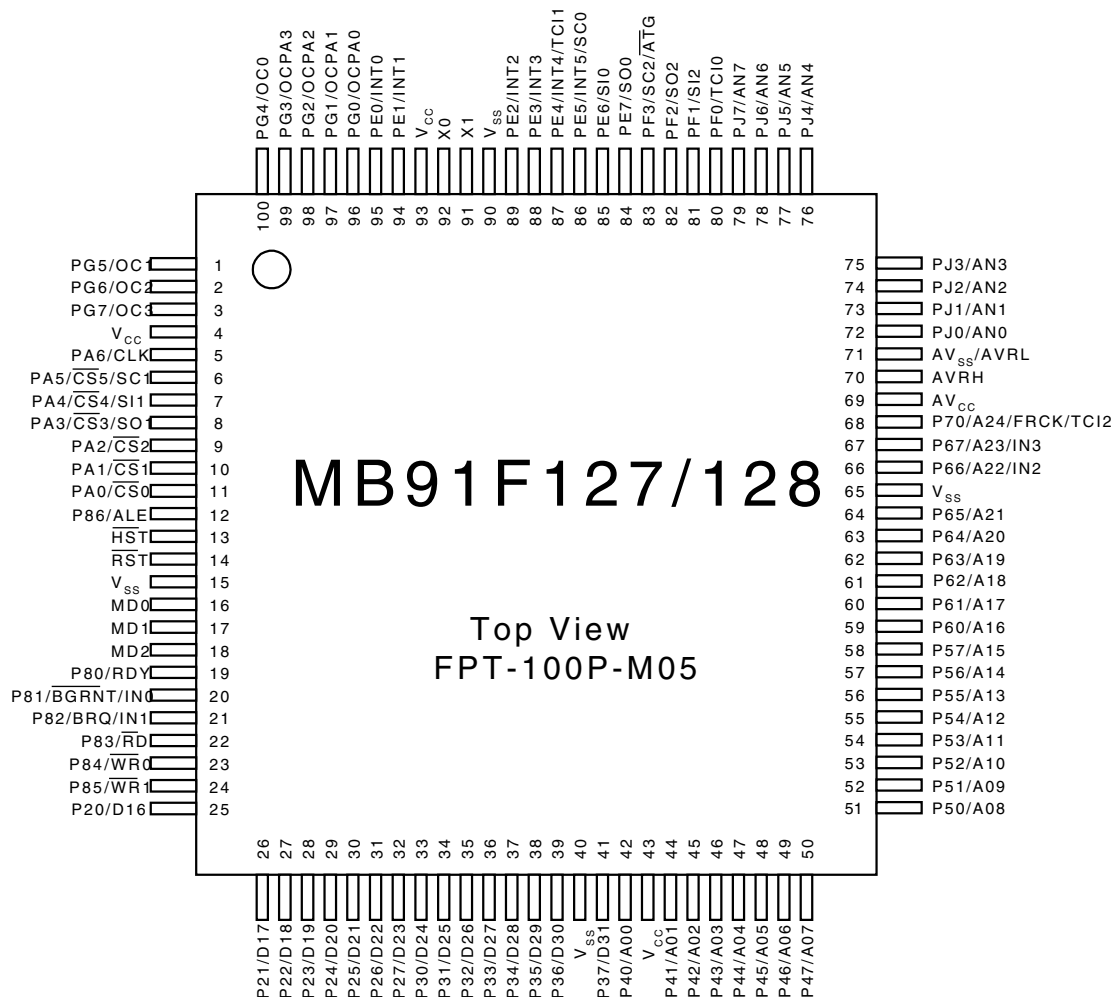


1.4 Pin Assignment

Figure 1.4-1 "MB91F127/128 Pin Assignment" shows a pin assignment of the MB91F127/128.

■ Pin Assignment of the MB91F127/128 (FPT-100P-M05)

Figure 1.4-1 MB91F127/128 Pin Assignment



1.5 Pin Description

Table 1.5-1 "Pin Description" provide pin function explanations.

■ Pin Description

Table 1.5-1 Pin Description

Pin name	I/O circuit type	Function
D16/P20 D17/P21 D18/P22 D19/P23 D20/P24 D21/P25 D22/P26 D23/P27	D	These pins transfer bits 16 to 23 of the external data bus. If the external bus width is set to 8 bits or when the microcontroller is in single chip mode, these pins can be used as a general-purpose I/O port (P20 to P27).
D24/P30 D25/P31 D26/P32 D27/P33 D28/P34 D29/P35 D30/P36 D31/P37	D	These pins transfer bits 24 to 31 of the external data bus. If these pins are not used, they can be used as a general-purpose I/O port (P30 to P37).
A00/P40 A01/P41 A02/P42 A03/P43 A04/P44 A05/P45 A06/P46 A07/P47 A08/P50 A09/P51 A10/P52 A11/P53 A12/P54 A13/P55 A14/P56 A15/P57	D	These pins transfer bits 00 to 15 of the external address bus. If these pins are not used as the address bus, they can be used as a general-purpose I/O port (P40 to P47, P50 to P57).

Table 1.5-1 Pin Description (Continued)

Pin name	I/O circuit type	Function												
A16/P60 A17/P61 A18/P62 A19/P63 A20/P64 A21/P65 A22/P66/IN2 A23/P67/IN3	D	These pins transfer bits 16 to 23 of the external address bus. If the pins are not used as an address bus, the pins can be used. These pins can be used as a general-purpose I/O port (P60 to P67). [INT2, INT3] These pins are for input of input capture. This function is enabled if input capture is in input mode.												
A24/P70/FRCK/TCI2	D	This pin transfers bit 24 of the external address bus. [P70] If A24, FRCK, and TCI2 are not used, this pin can be used as a general-purpose I/O port. [FRCK] This pin is for external clock input of the free-run timer. This function is enabled if the external clock input of the free-run timer is used. [TCI2] This pin is for external clock input of Timer 2. This function is enabled if the external clock input of Timer 2 is used.												
RDY/P80	D	This pin is for input of external ready. If the bus cycle is not completed during execution, "0" is input. If external ready input is not used, this pin can be used as a general-purpose port.												
BGRNT/P81/IN0	D	This pin is for output of external bus release acceptance. When the external bus is released, L is output. If external bus release acceptance output is not used, the pin can be used as a general-purpose port. [IN0] This pin is for input of input capture. This function is enabled if input capture is in input mode.												
BRQ/P82/IN1	D	This pin is for external bus release request input. To release the external bus, enter 1. If external bus release request input is not used, the pin can be used as a general-purpose port. [IN1] This pin is for input of input capture. This function is enabled if input capture is in input mode.												
RD/P83	D	This pin is an external bus read strobe. If this pin is not used, it can be used as a general-purpose I/O port.												
WR0/P84	D	This pin is external bus write strobe. The following table lists the relationship between the control signals and data bus byte positions. <table><tr><td></td><td>16-bit bus width</td><td>8-bit bus width</td><td>Single chip mode</td></tr><tr><td>D31 to D24</td><td>WR0</td><td>WR0</td><td>(Port enabled)</td></tr><tr><td>D23 to D16</td><td>WR1</td><td>(Port enabled)</td><td>(Port enabled)</td></tr></table> Note: WR1 is set to Hi-Z during a reset. When using a 16-bit bus, add a pull-up resistor externally. [P84 or P85] If WR0 and WR1 are not used, these pins can be used as a general-purpose I/O port.		16-bit bus width	8-bit bus width	Single chip mode	D31 to D24	WR0	WR0	(Port enabled)	D23 to D16	WR1	(Port enabled)	(Port enabled)
	16-bit bus width		8-bit bus width	Single chip mode										
D31 to D24	WR0	WR0	(Port enabled)											
D23 to D16	WR1	(Port enabled)	(Port enabled)											
WR1/P85	D													

CHAPTER 1 OVERVIEW

Table 1.5-1 Pin Description (Continued)

Pin name	I/O circuit type	Function
$\overline{CS0}/PA0$ $\overline{CS1}/PA1$ $\overline{CS2}/PA2$	D	These pins are for chip select 0, 1, and 2 output (active-low). [PA0, 1, 2] If chip select is not used, these pins can be used as a general-purpose I/O port.
$\overline{CS3}/PA3/SO1$ $\overline{CS4}/PA4/SI1$ $\overline{CS5}/PA5/SC1$	D	These pins are for chip select 3, 4, and 5 output (active-low). [PA3, 4, 5] If chip select or ch1 of the UART is not used, these pins can be used as a general-purpose I/O port. [SO1, SI1, SC1] These pins are for the data output, data input, and clock of UART1. These pins are enabled if UART1 operation is enabled.
CLK/PA6	D	This pin is for system clock output. The same clock frequency as the external bus operating frequency is output. If system clock output is not used, PA6 can be used as a general-purpose port.
OCPA0/PG0 OCPA1/PG1 OCPA2/PG2 OCPA3/PG3 OC0/PG4 OC1/PG5 OC2/PG6 OC3/PG7	D	[OCPA0 to 3] These pins are for PPG timer output. This function is enabled if specification of PPG timer output is enabled. [OC0 to 3] These pins are for output of output compare. This function is enabled if output specification of output compare is enabled. [PG0 to 7] If these pins are not used, they can be used as a general-purpose I/O port.
MD0 MD1 MD2	B	These pins are mode pins 0 to 2. These pins set the standard operating mode of the MCU. Do not connect these pins directly to the V_{CC} or V_{SS} .
X0 X1	A	This pin is for clock (oscillation) input. This pin is for clock (oscillation) output.
\overline{RST}	C	This pin is for external reset input.
\overline{HST}	C	This pin is for hardware standby input.
P86/ALE	D	[ALE] This pin is for output of an address latch signal. This function is enabled if ALE output specification of EPCR is enabled.
INT0/PE0 INT1/PE1 INT2/PE2 INT3/PE3	D	[INT0, 1, 2, 3] These pins are for input of an external interrupt request. This input is used from time to time while a corresponding external interrupt is enabled. Output by another function, unless it is intended, must be stopped. [PE0, 1, 2, 3] These pins are a general-purpose I/O port.
INT4/PE4/TCI1 INT5/PE5/SC0	D	[INT4, 5] These pins are for input of an external interrupt request. This input is used from time to time while a corresponding external interrupt is enabled. Output by another function, unless it is intended, must be stopped. [TCI1] This pin is for external clock input of Timer 1. [SC0] This pin is for clock input of UART0. [PE4, 5] These pins are a general-purpose I/O port.

Table 1.5-1 Pin Description (Continued)

Pin name	I/O circuit type	Function
SI0/PE6	D	[SI0] This pin is for data input of UART0. This function is enabled if data input specification of UART0 is enabled.
		[PE6] This pin is a general-purpose I/O port.
SO0/PE7	D	[SO0] This pin is for data output of UART0. This function is enabled if data output specification of UART0 is enabled.
		[PE7] This pin is a general-purpose I/O port.
PF0/TCI0	D	[TCI0] This pin is for external clock input of Timer 0.
		[PF0] This pin is a general-purpose I/O port.
SI2/PF1	D	[SI2] This pin is for data input of UART2. This function is enabled if data input specification of the UART2 is enabled.
		[PF1] This pin is a general-purpose I/O port.
SO2/PF2	D	[SO2] This pin is for data output of UART2. This function is enabled if data output specification of the UART2 is enabled.
		[PF2] This pin is a general-purpose I/O port. This function is enabled if data output specification of the UART2 is disabled.
SC2/PF3/ATG	D	[SC2] This pin is for clock input of UART2. [ATG] This pin is for external trigger input of the A/D converter. This input is used from time to time while a corresponding function is selected. Output by another function, unless it is intended, must be stopped.
		[PF3] This pin is a general-purpose I/O port.
AN0/PJ0 AN1/PJ1 AN2/PJ2 AN3/PJ3 AN4/PJ4 AN5/PJ5 AN6/PJ6 AN7/PJ7	E	[AN0 to 7] This is for analog input of the A/D converter. This function is enabled if analog input is specified for the AIC register.
		[PJ0 to 7] These pin are a general-purpose I/O port.
AV _{CC}	-	This pin is the V _{CC} power supply of the A/D converter.
AVRH	-	This pin provides the A/D converter reference voltage (high potential side). For turning on and off, always apply a V _{CC} potential greater than AVRH.
AV _{SS} /AVRL	-	These pins act as VSS power supply of the A/D converter and provide reference voltage (low potential side).
V _{CC}	-	These pins are used for the power supply of the digital circuits. Always use connected to the power supply.
V _{SS}	-	These pins provide ground level potential for digital circuits.

CHAPTER 1 OVERVIEW

Note:

For most of the pins listed above, input-output of the I/O ports and resources are multiplexed, which is indicated in the form xxxx/Pxx. If port output and resource output compete on these pins, the resource has priority.

1.6 I/O Circuit Type

Table Table 1.6-1 "Types of I/O Circuits" lists the I/O circuit forms.

I/O Circuit Type

Table 1.6-1 Types of I/O Circuits

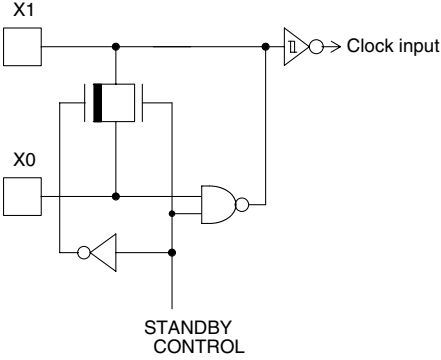
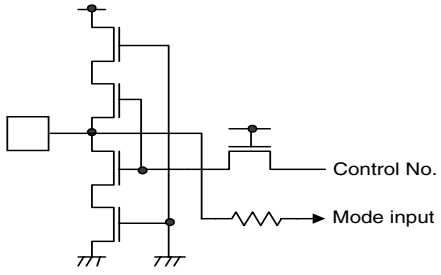
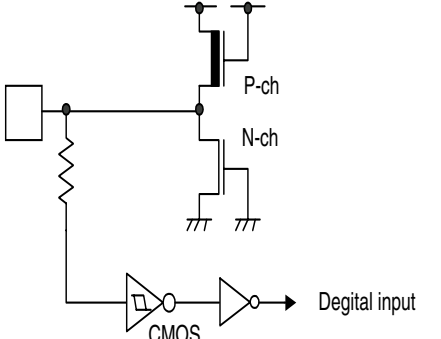
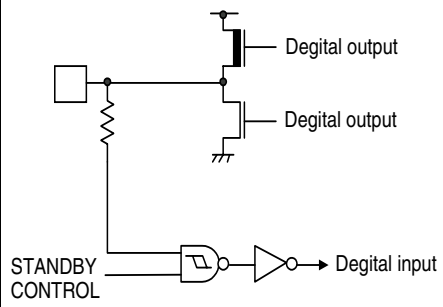
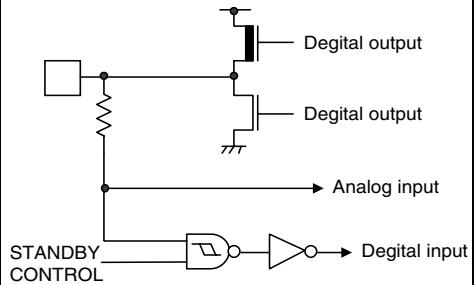
Classification	Circuit type	Remarks
A		<ul style="list-style-type: none">• With standby control• For 25 MHz oscillator• Oscillation feedback resistor: About 1 MΩ
B		<ul style="list-style-type: none">• With high-voltage control for flash test• CMOS level input
C		<ul style="list-style-type: none">• CMOS level hysteresis inputNo standby control

Table 1.6-1 Types of I/O Circuits

Classification	Circuit type	Remarks
D		<ul style="list-style-type: none">• CMOS level output• CMOS level hysteresis input with standby control
E		<ul style="list-style-type: none">• With standby control• CMOS level output• CMOS level hysteresis input• Analog input

CHAPTER 2 HANDLING THE DEVICE

This chapter provides precautions on handling the MB91F127/128.

2.1 "Precautions on Handling the Device"

2.2 "Precautions on Handling Power Supplies"

2.1 Precautions on Handling the Device

This section contains information on preventing a latchup and on the handling of pins.

■ Effective way to prevent latchup

A latchup can occur if, on a CMOS IC, a voltage higher than V_{CC} or a voltage lower than V_{SS} is applied to an input or output pin or a voltage higher than the rating is applied between V_{CC} and V_{SS} . A latchup, if it occurs, significantly increases the power supply current and may cause thermal destruction of an element. When you use a CMOS IC, be very careful not to exceed the maximum rating.

Furthermore, be very careful not to apply a voltage exceeding the digital power source rating to an analog pin.

■ Handling 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.

■ Connection of power supply pins (V_{CC} , V_{SS})

If more than one V_{CC} or V_{SS} pin exists, those that must be kept at the same potential are designed to be connected to one other inside the device to prevent malfunctions such as latchup. Be sure to connect the pins to a power supply and ground external to the device to minimize undesired electromagnetic radiation, prevent strobe signal malfunctions due to an increase in ground level, and conform to the total output current rating. Given consideration to connecting the current supply source to V_{CC} or V_{SS} of the device at the lowest impedance possible.

■ Quartz oscillation circuit

Noise near the X0 or X1 pin may cause the device to malfunction. Design printed circuit boards so that X0, X1, the quartz oscillator (or ceramic oscillator), and the bypass capacitor to ground are located as near to one another as possible. Furthermore, design printed circuit boards so that wiring does not cross other wiring if possible.

During design, use of printed circuit board artwork that surrounds the X0 and X1 pins with ground should be considered to increase the expectation of stable operation.

■ Handling of NC pins

Be sure to leave NC pins open for using the device.

■ Mode pins (MD0 to MD2)

These pins must be directly connected to V_{CC} or V_{SS} when they are used. To prevent malfunctions due to noise, keep the pattern length between a mode pin on a printed circuit board and V_{CC} or V_{SS} as short as possible so that they can be connected at a low impedance.

■ External reset input

The L level must be input to the $\overline{\text{RST}}$ terminal for at least five machine cycles so that an internal reset occurs unfailingly.

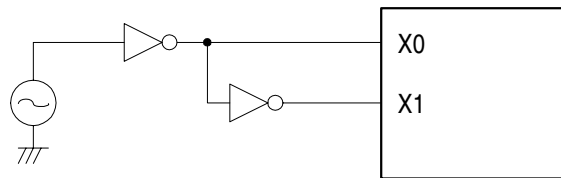
■ Precaution on using an external clock

When using an external clock, in general supply it to the X0 pin while also supplying a reverse-phase clock to the X1 pin simultaneously. In this case, do not use the STOP mode (oscillation stop mode). (This is because the X1 pin stops with output of the H level in STOP mode.)

Additionally, the X0 pin can be used only if an external clock is supplied at 12.5 MHz or less.

Figure Figure 2.1-1 "Example of External Clock Usage (Normal)" and Figure 2.1-2 "Example of External Clock Usage (Applicable at 12.5 MHz or Less)" show an example of using an external clock.

Figure 2.1-1 Example of External Clock Usage (Normal)



Note

The STOP mode (oscillation stop mode) cannot be used.

Figure 2.1-2 Example of External Clock Usage (Applicable at 12.5 MHz or Less)



■ Return from the sleep/stop state

To return a program from the sleep/stop state in C-Bus RAM, use a reset instead of an interrupt to cause the return.

■ Notes on during operation of PLL clock mode

IF the PLL clock mode is selected, the microcontroller attempt to be working with the self-oscillating circuit even when there is no external oscillator or external clock input is stopped. Performance of this operation, however, cannot be guaranteed.

■ Watchdog timer function

The watchdog timer supported by the FR family monitors the program that performs the reset delay operation for a specified time. If the program hangs and the reset delay operation is not performed, the watchdog timer resets the CPU. Therefore, once the watchdog timer is enabled, operation continues until the CPU is reset.

As an exception, a reset delay automatically occurs if the CPU stops program execution. For the conditions that apply to this exception, refer to the section that describes the watchdog function.

2.2 Precautions on Handling Power Supplies

This section provides precautions on power supplies with regard to pin handling and processing when power is turned on.

■ Power-on

Immediately after the power-on sequence, be sure to first input the L level to the $\overline{\text{RST}}$ pin. When the power supply reaches the V_{CC} level, provide a wait time of at least five cycles of the internal operation clock. Then, input the H level.

■ Pin Status after Power-on

The pin status is undefined after the power-on sequence. After the power-on sequence, oscillation starts, and the circuit is initialized.

■ Oscillation Input after Power-on

After the power-on sequence, input a clock until oscillation stabilization wait state is canceled.

■ Initialization at Power-on Reset

The device contains some internal registers that are initialized only by a power-on reset. To initialize the registers, perform a power-on reset by turning off and on the device.

CHAPTER 3 CPU

This chapter describes basic items such as the architecture, specifications, and instructions necessary for an overall understanding of the functions of the FR family.

- 3.1 "Memory Space"
- 3.2 "Internal Architecture"
- 3.3 "Dedicated Registers"
- 3.4 "General-purpose Registers"
- 3.5 "Data Structure"
- 3.6 "Word Alignment"
- 3.7 "Memory Map"
- 3.8 "Overview of Instructions"
- 3.9 "Exception, Interrupt, and Trap (EIT)"
- 3.10 "Reset Sequence"
- 3.11 "Clock Generator and Controller"
- 3.12 "Standby Mode (Low-power Consumption Mechanism)"
- 3.13 "Operation Mode"

3.1 Memory Space

The logical address space of the FR family is 4 Gbytes (2³² addresses). The CPU performs linear access.

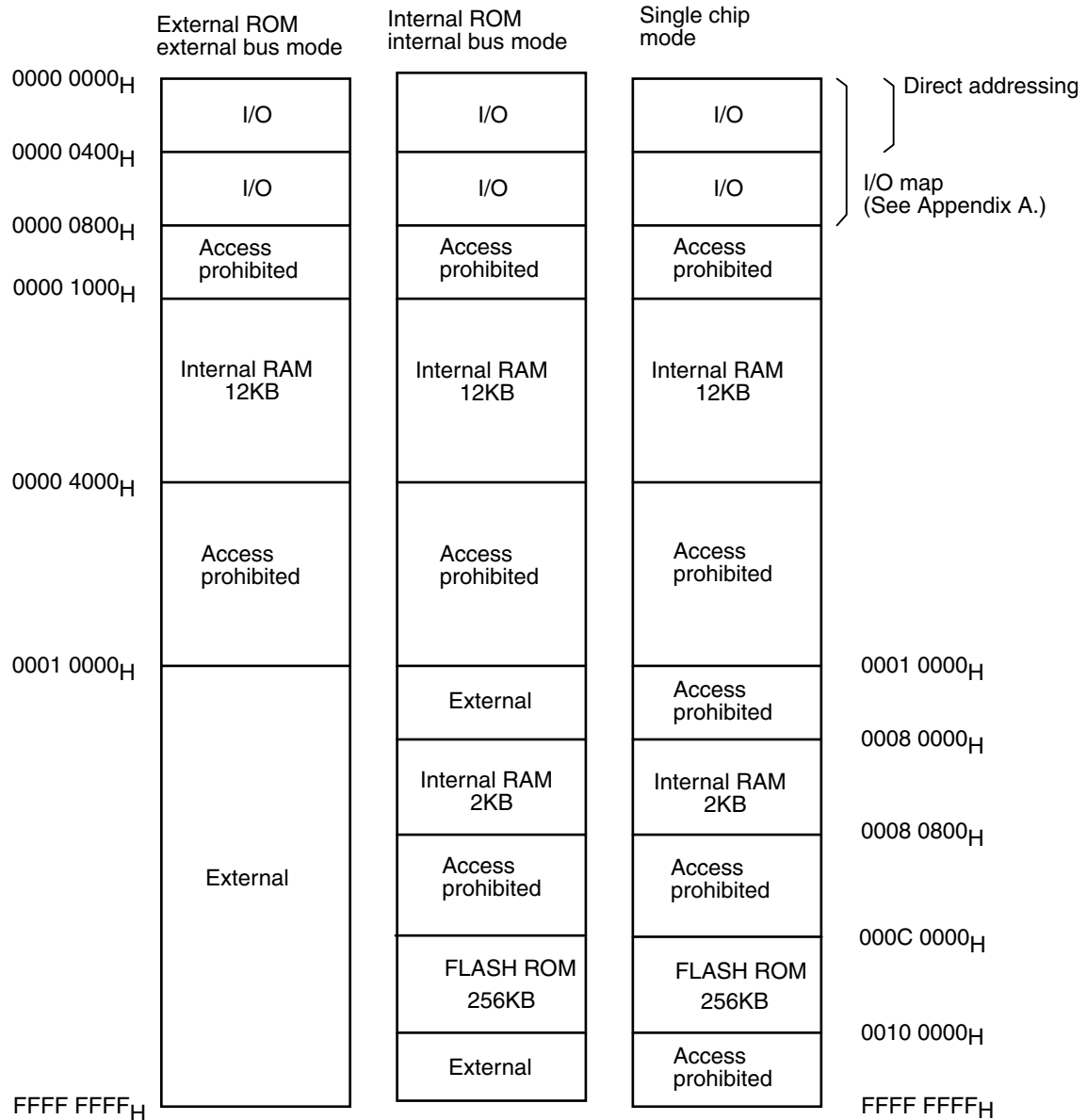
■ Direct addressing area

The following area of the address space is used for I/O and is referred to as the direct addressing area. The address of operands can be specified directly in the instructions. The direct addressing area depends in the following way on the size of the data to be accessed:

- Access in bytes: 0 to 0FF_H
- Access in halfwords: 0 to 1FF_H
- Access in words: 0 to 3FF_H

■ Memory Map

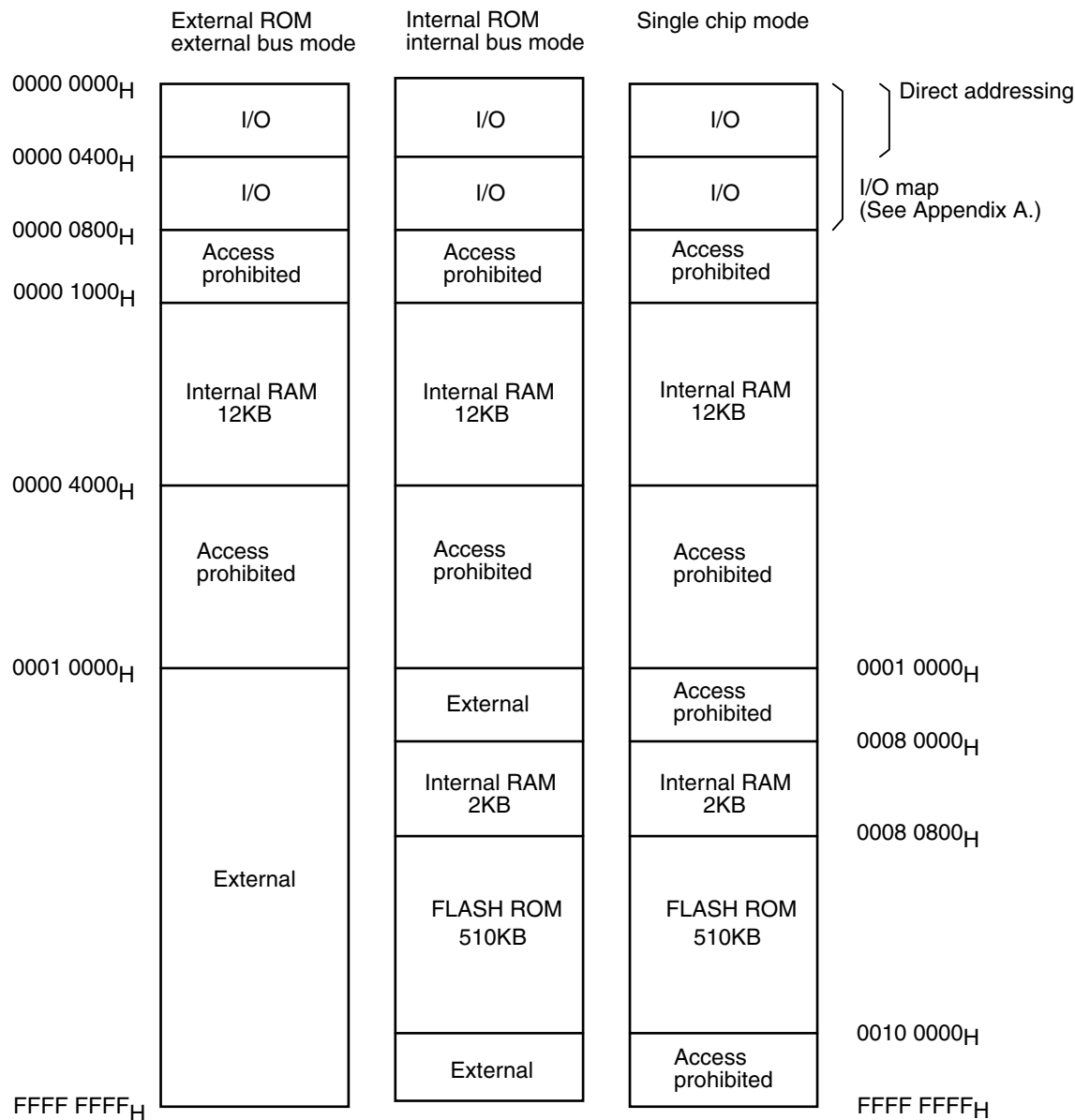
Figure 3.1-1 "MB91F127 Memory Map" shows the memory space of the MB91F127.

Figure 3.1-1 MB91F127 Memory Map**Note:**

In single chip mode, an external area cannot be accessed. Select the internal ROM external bus mode in the mode register to enable access to an external area.

Figure 3.1-2 "MB91F128 Memory Map" shows the memory spaces of the MB91F128.

Figure 3.1-2 MB91F128 Memory Map



Note

In single chip mode, an external area cannot be accessed. Select the internal ROM external bus mode in the mode register to enable access to an external area.

3.2 Internal Architecture

The CPU of the FR family employs RISC architecture. The FR CPU is a high-performance core that incorporates very powerful instructions for embedded use.

■ Features

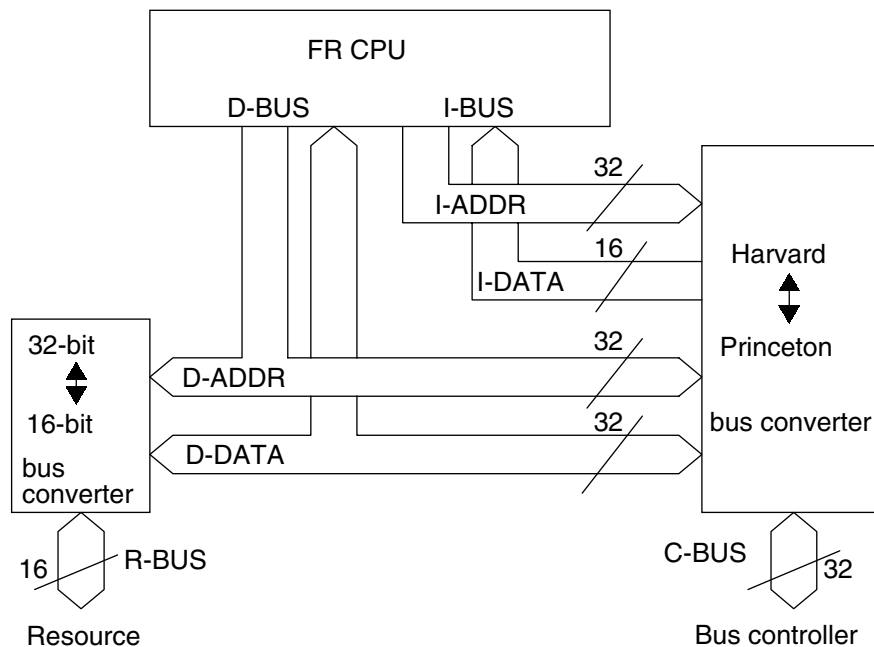
- **Use of RISC architecture**
 - Basic instructions: One instruction per cycle
- **32-bit architecture**
 - General-purpose registers: 32 bit × 16
- **4 Gbytes linear memory space**
- **Process cycles of the multiplier:**
 - 32 bit × 32 bit multiplication: 5 cycles
 - 16 bit × 16 bit multiplication: 3 cycles
- **Improved interrupt processing functions**
 - High-speed response (6 cycles)
 - Multiple interrupts supported
 - Level mask function (16 levels)
- **Improved instructions for I/O operations**
 - Memory-to-memory transfer instructions
 - Bit processing instructions
- **High code efficiency**
 - Basic instruction word length: 16 bits
- **Low power consumption**
 - Sleep and stop mode

■ Internal Architecture

The CPU of the FR family uses a Harvard architecture in which the instruction and data buses are independent.

A 32-bit <-- --> 16-bit bus converter is connected to the data bus (D-BUS) as interface between the CPU and peripheral resources. A Harvard <-- --> Princeton bus converter is connected to the I-BUS and D-BUS as interface between the CPU and the bus controller.

Figure 3.2-1 "Internal Architecture" shows the internal architecture.

Figure 3.2-1 Internal Architecture

○ CPU

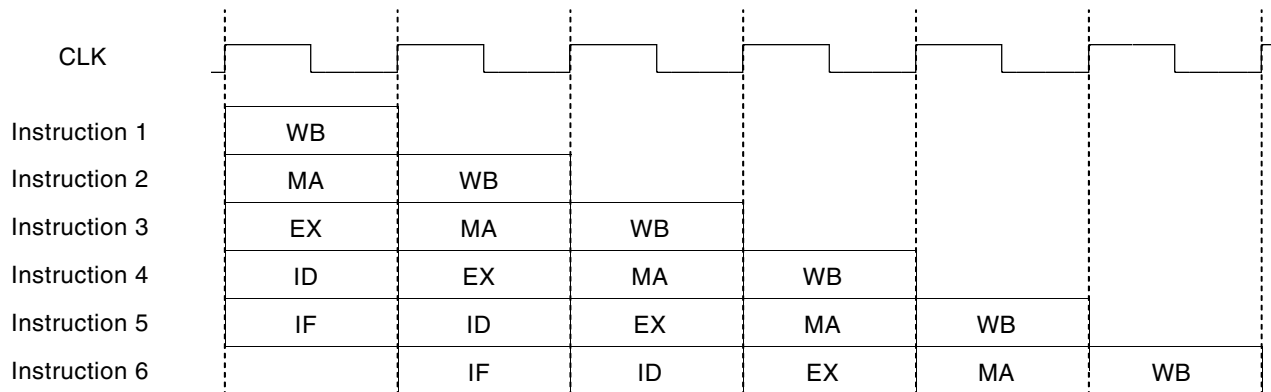
The CPU employs a compact 32-bit RISC FR architecture.

A five-stage instruction pipeline is used to execute one instruction per cycle. The pipeline is configured from the following stages:

- Instruction fetch (IF): The instruction address is output and the instruction fetched.
- Instruction decode (ID): The fetched instruction is decoded. Register read is also performed here.
- Execution (EX): An arithmetic operation is executed.
- Memory access (MA): Load or Store accesses to the memory.
- Write back (WB): Results of arithmetic operations (or loaded memory data) are written to a register.

Figure 3.2-2 "Instruction Pipeline" shows the structure of the instruction pipeline.

Figure 3.2-2 Instruction Pipeline



Instructions are executed sequentially. For example, if instruction A enters the pipeline before instruction B, instruction A reaches the write back stage before instruction B.

As a rule, instructions are executed at a speed of one instruction per cycle. However, load and store instructions that involve memory wait, branch instructions that do not have delay slots, and multiple cycle instructions will require several cycles to execute. In addition, if the supply of instructions is delayed, the execution speed of the instructions will be reduced.

○ 32-bit <-- --> 16-bit bus converter

This converter acts as interface between the D-BUS for high-speed 32-bit access and the R-BUS for 16-bit access and enables the accesses from the CPU to the built-in circuits for peripherals.

The converter converts 32-bit accesses from the CPU into two 16-bit accesses and performs the access over the R-BUS. Some of the built-in circuits for peripherals are limited with respect to their access width.

○ Harvard <-- --> Princeton bus converter

The Harvard <-- --> Princeton bus converter adjusts CPU instruction access and data access to ensure efficient interfacing with the external bus.

The CPU employs Harvard architecture, in which the instruction and data buses are independent. The bus controller that controls the external bus employs Princeton architecture, in which only a single bus is used. The Harvard <-- --> Princeton bus converter assigns an access priority for instruction access and data access to control access to the bus controller, thereby optimizing the order of external access.

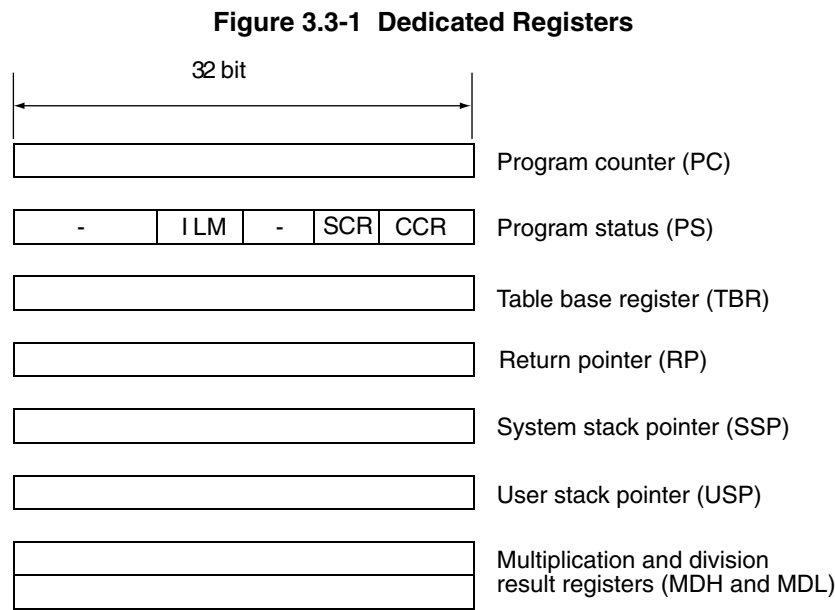
Moreover, the bus converter has a two-word write buffer to eliminate the CPU bus wait time and a one-word prefetch buffer to prefetch instructions.

3.3 Dedicated Registers

Dedicated registers are used for specific purposes. A program counter (PC), program status (PS), table base register (TBR), return pointer (RP), system stack pointer (SSP), user stack pointer (USP), and multiplication and division result registers (MDH and MDL) are provided.

■ Dedicated Registers

Figure 3.3-1 "Dedicated Registers" shows the configuration of the dedicated registers.

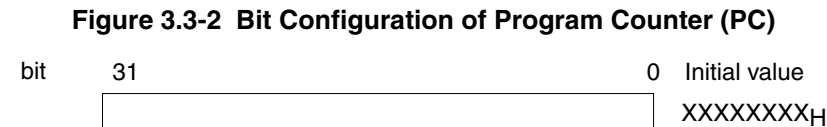


■ Program counter (PC)

This section explains the functions of the program counter (PC).

The program counter (PC) consists of 32 bits.

Figure 3.3-2 "Bit Configuration of Program Counter (PC)" shows the bit configuration of the program counter (PC).



The program counter (PC) indicates the address of the instruction being executed.

If execution of the instruction involves updating of the PC. Bit 0 could be set to "1" only when an odd-numbered address was specified as the branch address. However, bit 0 is invalid in this case, since instruction must be stored at addresses that are a multiple of two.

The initial value at a reset is undefined.

■ Program status (PS)

This register retains the program status. The register is divided into three parts: CCR, SCR, and ILM. See Section 3.3.1 "Program Status Register (PS)" for details.

All undefined bits are reserved bits. During read operations, the system always reads "0".

Writing is disabled.

■ Table base register (TBR)

This section explains the functions of the table base register (TBR).

The table base register (TBR) consists of 32 bits.

Figure 3.3-3 "Bit Configuration of Table Base Register (TBR)" shows the bit configuration of the table base register (TBR).

Figure 3.3-3 Bit Configuration of Table Base Register (TBR)



The table base register (TBR) saves the start address of the vector table for EIT processing.

The initial value at a reset is 000FFC00_H.

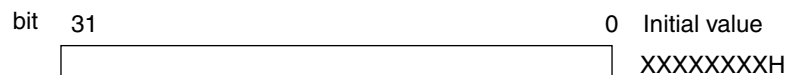
■ Return pointer (RP)

This section explains the functions of the return pointer (RP).

The return pointer (RP) consists of 32 bits.

Figure 3.3-4 "Bit Configuration of Return Pointer (RP)" shows the bit configuration of the return pointer (RP).

Figure 3.3-4 Bit Configuration of Return Pointer (RP)



The return pointer (RP) saves the address for return from the subroutine.

When the CALL instruction is executed, the PC value is transferred to this RP.

When the RET instruction is executed, the RP contents are transferred to the PC.

The initial value at a reset is undefined.

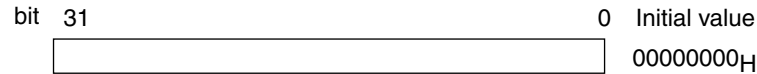
■ System stack pointer (SSP)

This section explains the functions of the system stack pointer (SSP).

The system stack pointer (SSP) consists of 32 bits.

Figure 3.3-5 "Bit Configuration of System Stack Pointer (SSP)" shows the bit configuration of the system stack pointer (SSP).

Figure 3.3-5 Bit Configuration of System Stack Pointer (SSP)



The SSP is the register for the system stack pointer.

When the S flag is "0", the SSP functions as R15.

The contents of the SSP can also be specified directly.

In addition, when EIT is issued, the SSP can be used as a stack pointer that specifies the stack for saving the contents of PS and PC.

The initial value at a reset is 00000000_H.

■ User stack pointer (USP)

This section explains the functions of the user stack pointer (USP).

The user stack pointer (USP) consists of 32 bits.

Figure 3.3-6 "Bit Configuration of User Stack Pointer (USP)" shows the bit configuration of the user stack pointer (USP).

Figure 3.3-6 Bit Configuration of User Stack Pointer (USP)



The USP is the register for the user stack pointer.

When the S flag is "1", the USP functions as R15.

The USP can also be specified.

The initial value at a reset is undefined.

The RETI instruction cannot use the USP.

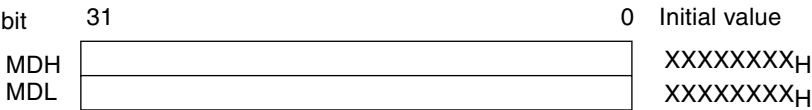
■ Multiplication and division result registers (MDH and MDL)

This section explains the functions of the multiply and divide registers (MDH and MDL).

Each of the multiply and divide registers (MDH and MDL) consist of 32 bits.

Figure 3.3-7 "Multiplication and division result registers (MDH and MDL)" shows the bit configuration of the multiply and divide registers (MDH and MDL).

Figure 3.3-7 Multiplication and division result registers (MDH and MDL)



The MDH and MDL are the registers for multiplication and division. Both registers are 32 bits long.

The initial value at a reset is undefined.

○ Functions used for multiplications

The 64-bit long results of 32 bit × 32 bit multiplication operations are stored in the multiplication and division result storage registers as follows:

- MDH: 32 high-order bits
- MDL: 32 low-order bits

For 16 bit × 16 bit multiplication operations, the results are stored as follows:

- MDH: Undefined
- MDL: Resulting 32 bits

○ Functions used for divisions

When the division operation starts, the dividend is stored in the MDL.

When the DIV0S, DIV0U, DIV1, DIV2, DIV3, and DIV4S instructions are executed for division operations, the results are stored in the MDH and MDL.

- MDH: Remainder
- MDL: Quotient

3.3.1 Program Status Register (PS)

The program status register (PS) retains the program status. The register is divided into three parts: ILM, SCR, and CCR.

All undefined bits are reserved bits. During read operations, they always return "0". Writing is disabled.

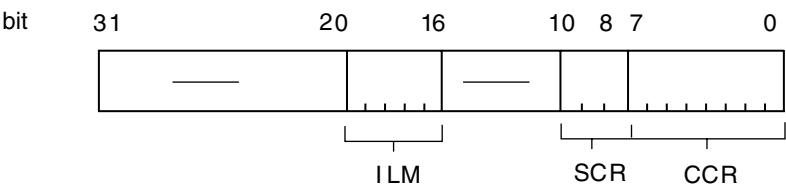
■ Program status register (PS)

○ Program status register (PS)

The program status register (PS) consists of the condition code register (CCR), system condition code register (SCR), and interrupt level mask register (ILM).

Figure 3.3-8 "Register Configuration of Program Status Register (PS)" shows the register configuration of the program status register.

Figure 3.3-8 Register Configuration of Program Status Register (PS)



○ Condition code register (CCR)

Figure 3.3-9 "Register Configuration of Condition Code Register (CCR)" shows the configuration of the condition code register (CCR).

Figure 3.3-9 Register Configuration of Condition Code Register (CCR)

bit	7	6	5	4	3	2	1	0	initial value
	-	-	S	I	N	Z	V	G	-00XXXX _B

The function of each bit is explained below.

[Bit 5] S: Stack flag

Bit 5 specifies use of the stack pointer as R15.

S	Stack flag (S) function
0	The SSP is used as R15. When EIT is issued, 0 is set automatically. (However, the value saved to the stack is the value before clearing.)
1	The USP is used as R15.

A reset sets the bit to "0".

Set this bit to "0" when the RETI instruction is executed.

[Bit 4] I: Interrupt enable flag

Bit 4 is used to allow or prohibit user interrupt requests.

I	Function of interrupt enable flag (I)
0	User interrupts are prohibited. When the INT instruction is executed, the value is set to 0. (However, the value saved to the stack is the value before clearing.)
1	User interrupts are allowed. The value saved to the ILM is used to control masking of user interrupt requests.

This bit is cleared to "0" after a reset.

[Bit 3] N: Negative flag

Bit 3 is used to indicate when the result of an arithmetic operation is a signed integer in two's complement representation.

N	Function of negative flag (N)
0	Indicates that the result of the arithmetic operation is positive.
1	Indicates that the result of the arithmetic operation is negative.

This bit has an undefined initial state after a reset.

[Bit 2] Z: Zero flag

Bit 2 is used to indicate whether the result of an arithmetic operation is zero.

Z	Function of zero flag (Z)
0	Indicates that the result of the arithmetic operation is a value other than 0.
1	Indicates that the result of the arithmetic operation is 0.

This bit has an undefined initial state after a reset.

[Bit 1] V: Overflow flag

Bit 1 is used to indicate whether an overflow occurred in the result for an arithmetic operation when the operands used for the arithmetic operation results are integers in two's complement representation.

V	Function of overflow flag (V)
0	Indicates that an overflow did not occur for the result of the arithmetic operation.
1	Indicates that an overflow occurred for the result of the arithmetic operation.

This bit has an undefined initial state after a reset.

[Bit 0] C: Carry flag

Bit 0 is used to indicate whether a carry or borrow from the highest bit occurred during an arithmetic operation.

C	Function of carry flag (C)
0	Indicates that a carry or borrow did not occur.
1	Indicates that a carry or borrow occurred.

This bit has an undefined initial state after a reset.

○ **System condition code register (SCR)**

Figure 3.3-10 "Register Configuration of System Condition Code Register (SCR)" shows the configuration of the system condition code register (SCR).

Figure 3.3-10 Register Configuration of System Condition Code Register (SCR)

bit	10	9	8	Initial value
	D1	D0	T	XX0B

The functions of bits of the system condition code register (SCR) are explained below.

[Bits 10 and 9] D1 and D0: Step division flag

Bits 10 and 9 save the intermediate data during step division.

The bits must not be changed during division.

If another operation is executed during step division, the value of the PS register is saved and later returned to restart the step division.

This bit has an undefined initial state after a reset.

Executes the DIV0S instruction, accesses and stores the values of dividend and divisor.

Executes the DIV0U instruction and clears the register.

[Bit 8] T: Step trace trap flag

Bit 8 specifies whether to enable the step trace trap.

T	Function of step trace trap (T)
0	Disables the step trace trap.
1	Enables the step trace trap. All user NMIs and user interrupts are prohibited at this time.

This bit has an undefined initial state after a reset.

The emulator uses the step trace trap function. When the emulator is used, the step trace trap function can be used in a user program.

○ **Interrupt level mask register (ILM)**

Figure 3.3-11 "Register Configuration of Interrupt Level Mask Register (ILM)" shows the configuration of the interrupt level mask register (ILM).

Figure 3.3-11 Register Configuration of Interrupt Level Mask Register (ILM)

bit	20	19	18	17	16	Initial value
	ILM4	ILM3	ILM2	ILM1	ILM0	01111 _B

This register is used to save the interrupt level mask value. The value saved in this ILM is used for the level mask.

Interrupt requests are accepted only if the interrupt level corresponding to the interrupt request input to the CPU is higher than the level indicated by this ILM.

For level values, 0 (00000_B) is the highest and 31 (11111_B) is the lowest.

The values that can be set from a program are restricted. If the original value is between 16 and 31, a value between 16 and 31 can be set as a new value. If an instruction for which a value between 0 and 15 has been set is executed, the value obtained by adding 16 to the specified value will be transferred.

If the original value is between 0 and 15, an arbitrary value between 16 and 31 can be set.

A reset initializes the register to 15 (01111_B).

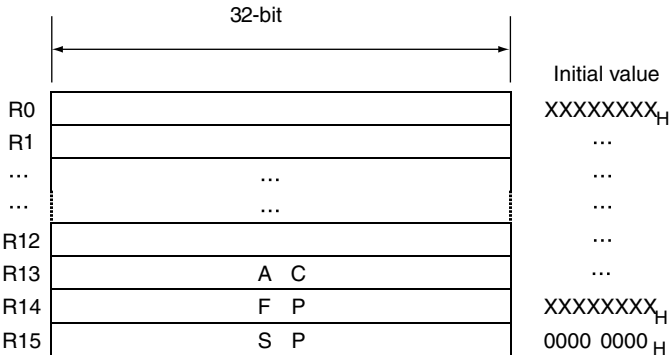
3.4 General-purpose Registers

Registers 0 to 15 are for general-purpose use and for use as accumulators in arithmetic operations and as memory access pointers.

■ General-purpose Registers

Figure 3.4-1 "Configuration of General-purpose Registers" shows the configuration of the general-purpose registers.

Figure 3.4-1 Configuration of General-purpose Registers



Among these 16 registers, specific use is assumed for the following three. Therefore, instructions for these registers are partially enhanced.

- R13: Virtual accumulator
- R14: Frame pointer
- R15: Stack pointer

The values of R0 to R14 at reset are not defined. The reset value of R15 is 00000000_H [value of the system stack pointer (SSP)].

3.5 Data Structure

TAn FR-family CPU has the following two data layouts:

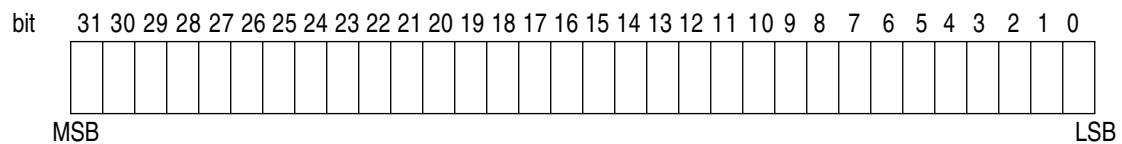
- For bit ordering: Little endian
- For byte ordering: Big endian

■ Bit Ordering

The CPU of the FR family uses little endian bit ordering.

Figure 3.5-1 "Bit Arrangement of Bit Ordering" shows the bit ordering.

Figure 3.5-1 Bit Arrangement of Bit Ordering

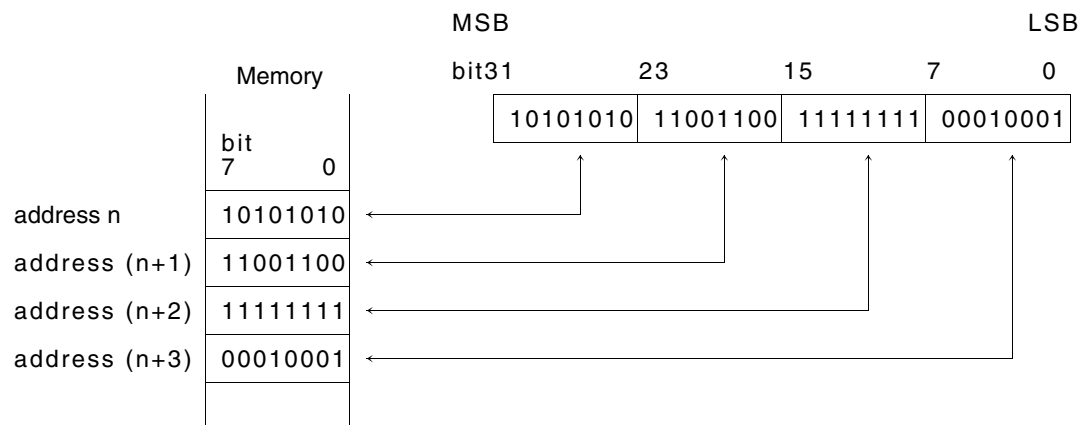


■ Byte Ordering

The CPU of the FR family uses big endian byte ordering.

Figure 3.5-2 "Arrangement of Byte Ordering" shows the arrangement of byte ordering.

Figure 3.5-2 Arrangement of Byte Ordering



3.6 Word Alignment

Instructions and data are accessed in units of bytes. The address to which instructions and data are allocated depends on instruction length and data width.

■ Program Access

Programs for the FR family must be stored at an address that is a multiple of 2.

Bit 0 of the program counter (PC) is set to "0" at PC update after instruction execution. This bit could be set to "1" only when an odd address is specified for the branch destination address. However, bit 0 is invalid in such a case and the instruction must be allocated to an address that is a multiple of 2.

There are no exceptions at which an odd address would be used.

■ Data Access

In FR family data accesses, forcible address alignment is performed depending on the access width as follows:

- Word access: The address is a multiple of 4. (The two least significant bits are forcibly set to "00".)
- Half-word access: The address is a multiple of 2. (The least significant bit is forcibly set to "0".)
- Byte access: -

Some bits are forcibly set to 0 in word and half-word data access when determining the effective address as result of a calculation. For example, in addressing mode @ (R13, Ri), the register before addition is used for calculation (even when the least significant bit is 1) and the two low-order bits of the addition result are masked. Before addition, the register is not masked.

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

	R 13	00002222 _H
	R 2	00000003 _H
+)		
<hr/>		
Addition result		00002225 _H
		↓ The two low-order bits are forcibly masked.
Address pin		00002224 _H

3.7 Memory Map

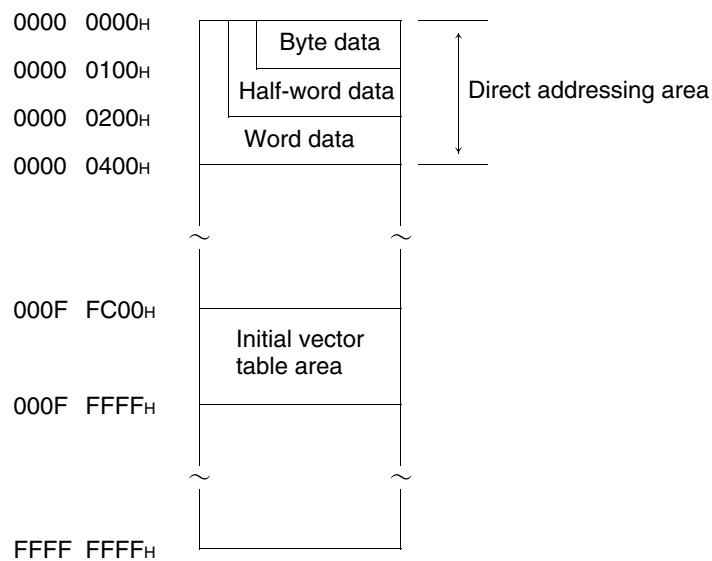
This section describes the MB91F127/128 memory map and the common memory map for the FR family.

■ MB91F127/128 Memory Map

The address space of memory is a 32-bit linear space.

Figure 3.7-1 "MB91F127/128 Memory Map" shows the MB91F127/128 memory map.

Figure 3.7-1 MB91F127/128 Memory Map



○ Direct addressing area

The address space areas below are used for input and output operations. These areas can be used to specify an operand address in an instruction using direct addressing.

The size of the address area for which direct addressing is enabled depends on the data size.

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

○ Initial vector table area

The area between addresses 000FFC00_H to 000FFFFF_H is an initial vector table area for exceptions, interrupts, and traps (EIT).

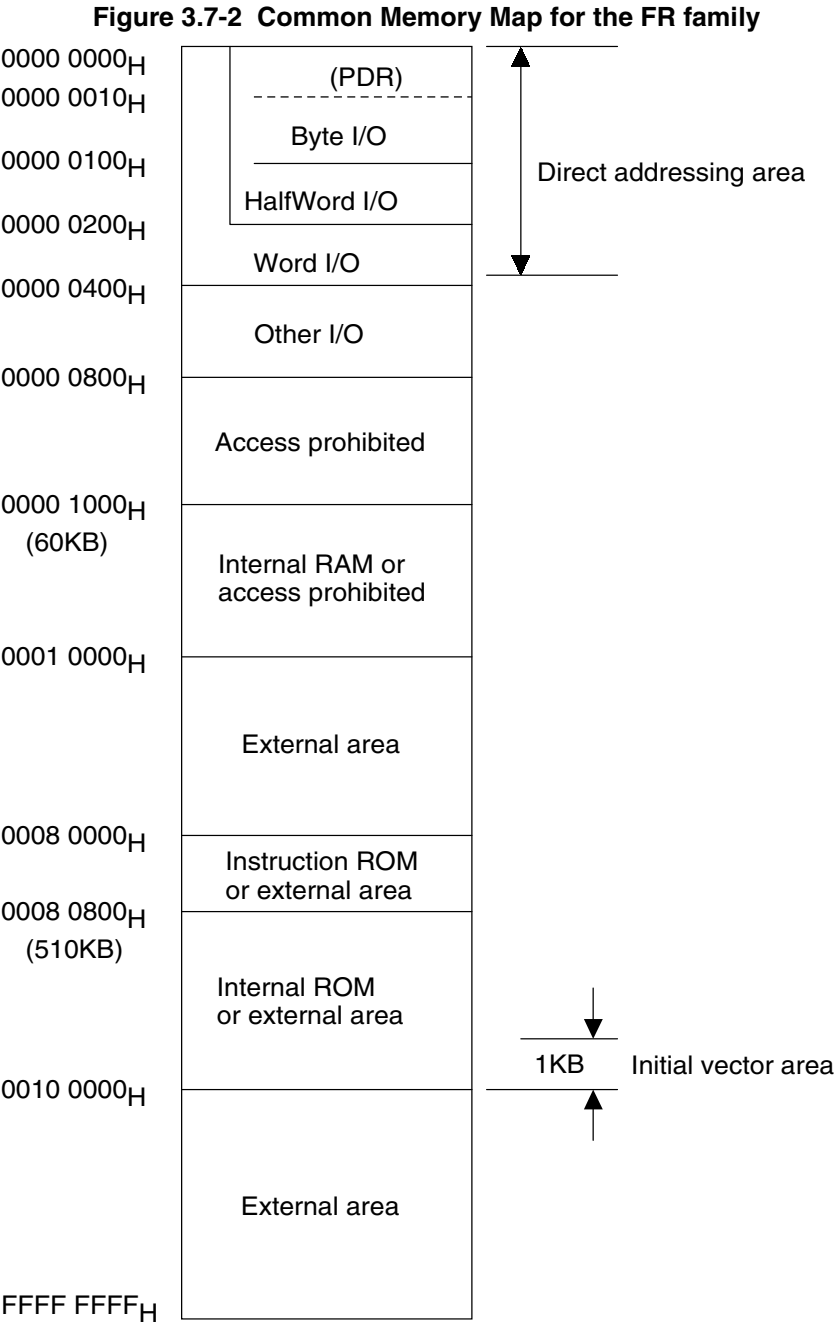
The vector table used at EIT processing can be allocated to any address by rewriting the contents of the TBR. After a reset, the table will always be stored at this address, however.

■ Common Memory Map for the FR Family

The memory map for the FR family is defined as shown in Figure 3.7-2 "Common Memory Map for the FR family".

This memory map is common to all FR family models (except single-chip models).

Figure 3.7-2 "Common Memory Map for the FR family" shows the common memory map for the FR family.



Note

In single chip mode, an external area cannot be accessed.

For the MB91F127/128, 80000 to 0807FF_H of the internal ROM area are assigned to 2 KB of internal RAM.

3.8 Overview of Instructions

The FR family supports logical operation instructions, bit operation instructions, and direct addressing instructions optimized for internal use, as well as the general RISC instruction set. Each 16-bit instruction (some of these instructions have actually a length of 32-bit or 48-bit) provides excellent memory utilization efficiency.

The instruction set can be categorized into the following functional groups:

- **Arithmetic operation instructions**
 - **Load and store instructions**
 - **Branch instructions**
 - **Logical operation and bit operation instructions**
 - **Direct addressing instructions**
 - **Other instructions**
-

■ Overview of Instructions

○ **Arithmetic operation instructions**

The instructions for arithmetic operations include the instructions for standard arithmetic operations (addition, subtraction, and compare) as well as shift instructions (logical shift and arithmetic shift). Addition and subtraction support operations with carry-over (as used in multiword operations) as well as operations without changing flag values, which is useful in address calculations.

The instructions for arithmetic operations also include 32×32 bit and 16×16 bit multiplication instructions and $32/32$ bit step division instructions.

Immediate transfer instructions (in which register values are set directly) and instructions for transfers between registers are also supported.

The instructions for arithmetic operations perform all operations using general-purpose registers, multiplication registers, and division registers in the CPU.

○ **Load and store instructions**

The load and store instructions are used for read and write access to external memory and to the built-in circuits for peripherals (I/O).

The load and store instructions support three access lengths: byte, half-word, and word. In addition to general register indirect memory addressing, some load and store instructions support register indirect memory addressing with displacement and register increment/decrement operations.

○ **Branch instructions**

The branch instructions include the instructions for branch, call, interrupt, and return. Depending on use, a branch instruction may have a delay slot to perform optimization.

See Section 3.8.1 "Branch instructions with delay slot" and Section 3.8.2 "Branch instructions without delay slot" for details.

○ **Logical operation and bit operation instructions**

The logical operation instructions can perform the logical operations AND, OR, and EOR between the contents of general-purpose registers or between the values in general-purpose registers and memory (and I-O units). The bit operation instructions allow to manipulate the contents of memory (and I-O units) directly. The instructions for logical operations and bit operations support general register-indirect memory addressing.

○ **Direct addressing instructions**

The direct addressing instructions are used for access between I-O units and general-purpose registers or between I-O units and memory. Specifying the I-O address directly in an instruction instead of using register-indirect addressing enables data access with high speed and high efficiency. Some direct addressing instructions support register indirect memory addressing and register increment/decrement operations.

○ **Other instructions**

Instructions that set flags and perform stack operations, coding, and zero expansion in the PS register are supported. Instructions for the entry/exit function of high-level languages and register multiload/store instructions are supported as well.

3.8.1 Branch Instructions with Delay Slot

For a branch instruction with a delay slot, the CPU first executes the instruction immediately after the branch instruction before branching to the instruction at the branching destination. The location immediately after the branch instruction is called the delay slot.

■ Branch Instructions with Delay Slot

The following instructions are branch instructions 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

■ CPU Operation for Branch Instructions with Delay Slot

For a branch instruction with a delay slot, the CPU first executes the instruction immediately after the branch instruction before branching to the instruction at the branching destination.

Because the CPU executes the instruction in the delay slot before executing the branch operation, execution speed appears to be delayed by one cycle. If an effective instruction cannot be set in the delay slot, the NOP instruction must be set.

[Example]

```

;      Instruction sequence
      ADD    R1,    R2      ;
      BRA:D  LABEL      ; Branch instruction
      MOV    R2,    R3      ; Delay slot: Executed before branching
      :
LABEL : ST    R3,    @R4    ; Branch destination

```

For a conditional branch instruction, the instruction in the delay slot is executed even when the branch conditions are satisfied.

The execution order can appear reversed for some branch instructions with a delay slot. However, this applies only to PC update operations. Other operations (e.g., register update or reference) are executed in the sequence they are coded.

Details are as follows:

- **Ri, which is referenced by the JMP:D @Ri or CALL:D @Ri instruction, is not affected even when the instruction in the delay slot updates Ri.**

[Example]

```
LDI:32    #Label,    R0
JMP:D     @R0         ; Branch to Label.
LDI:8     #0,        R0         ; The branch destination address is not affected
:
```

- **RP, which is referenced by the RED:D instruction, is not affected even if the instruction in the delay slot updates RP.**

[Example]

```
RET:D                                     ; Branch to the address indicated by RP set
                                           ; before this instruction.
MOV      R8,      RP         ; Return operation is not affected.
:
```

- **The flag referenced by the Bcc:D rel instruction is not affected by the instruction in the delay slot.**

[Example]

```
ADD      #1,      R0         ; Flag change
BC:D     Overflow          ; Branch based on the result of the instruction above.
ANDCCR   #0,                                     ; This flag update is not affected by the branch
                                           ; instruction above.
:
```

- **When RP is referenced by the instruction in the delay slot of the CALL:D instruction, the contents updated by the CALL:D instruction are read.**

[Example]

```
CALL:D   Label
MOV      RP,      R0         ; Branched after RP update.
                                           ; RP (execution result of the CALL:D instruction above)
                                           ; is transferred.
:
```

■ Restrictions on branch instructions with delay slot

○ Instructions that Can Be Set in the Delay Slot

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

- 1-cycle instruction
- Instruction other than a branch instruction
- Instruction for which a change of execution order does not affect operation

"1-cycle instructions" are characterized by the fact that the corresponding number in the "number of cycles"-column of the instruction list is "1", "a", "b", "c", or "d".

○ Step trace trap

No step trace trap occurs between the execution of a branch instruction with a delay slot and the delay slot itself.

○ Interrupt or non-maskable interrupt (NMI)

No interrupt or NMI is accepted between the execution of a branch instruction with a delay slot and the delay slot itself.

○ Undefined instruction exception

If the instruction in a delay slot is not defined, an undefined instruction exception is not thrown. The undefined instruction is in this case executed as if it was a NOP instruction.

3.8.2 Branch Instructions without Delay Slot

For a branch instruction without a delay slot, the CPU executes instructions according to the coded sequence.

■ Branch Instructions without Delay Slot

The following instructions are branch instructions 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

■ CPU Operation for Branch Instructions without Delay Slot

For a branch instruction without a delay slot, the CPU executes instructions according to the coded sequence. In other words, the CPU does not execute the instruction after the branch instruction before executing the branch operation itself.

[Example]

```

;      Sequence of instructions
      ADD    R1,    R2      ;
      BRA    LABEL        ; Branch instruction (without delay slot)
      MOV    R2,    R3      ; Not executed
      :
LABEL : ST    R3,    @R4    ; Branch destination

```

The number of execution cycles for a branch instruction without a delay slot is two cycles when branching is executed, and one cycle when branching is not executed.

Because an effective instruction cannot be set in the delay slot, instruction code efficiency can be improved relative to a branch instruction with a delay slot containing NOP.

When an effective instruction can be set in a delay slot, an operation with a delay slot should be selected; otherwise, an operation without the delay slot should be selected. Correct selection permits the improvement of execution speed and code efficiency.

3.9 Exception, Interrupt, and Trap (EIT)

The acronym EIT indicates that the CPU interrupts execution of the current program because of an event that occurred during program execution, and runs another program instead. EIT is the generic name for exception, interrupt, and trap.

■ EIT (exception, interrupt, trap)

An exception is an event that occurs in the context of program execution. After the exception, the CPU resumes program execution beginning again with the instruction that caused the exception.

An interrupt is an event that occurs regardless of the context of program execution and is caused on the hardware level.

A trap is an event that occurs in the context of program execution. The trap may be specified by the program in the same way as a system call. After the trap, the CPU resumes execution with the next instruction after the instruction that caused the trap.

■ Features

- Multi-interrupt. is supported.
- The level mask function for interrupts is supported (with up to 15 user levels).
- Trap instruction (INT)
- Emulator activation EIT (hardware or software)

■ EIT causes

EIT causes are as follows:

- Reset
- User interrupt (internal cause and external interrupt)
- NMI
- Delay interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap
- Coprocessor nonexistent trap
- Coprocessor error trap

■ Return from EIT

Use the RETI instruction to return from EIT.

- Use the STILM instruction to set an interrupt level mask register (ILM).

■ Notes on EIT

○ Delay slot

The delay slot for branch instructions has restrictions on EIT. See Section 3.8 "Overview of Instructions" for details.

3.9.1 EIT Interrupt Level

Interrupt levels 0 to 31 are managed with five bits.

■ Interrupt Levels

Table 3.9-1 "Interrupt Levels" shows the assignment of levels.

Table 3.9-1 Interrupt Levels

Level		Cause	Remarks
Binary	Decimal		
00000	0	(Reserved by the system)	If the original interrupt level mask (ILM) value is between 16 and 31, a value in this range cannot be set in ILM by the program.
:	:	:	
:	:	:	
00011	3	(Reserved by the system)	
		INTE instruction	
00100	4		
		Step trace trap	
00101	5	(Reserved by the system)	
:	:	:	
:	:	:	
01110	14	(Reserved by the system)	
01111	15	NMI (for user)	
10000	16	Interrupt	User interrupt is prohibited when ILM is set.
10001	17	Interrupt	
:	:	:	
:	:	:	
11110	30	Interrupt	
11111	31	-	Interrupt is prohibited when ICR is set.

Interrupt levels 16 to 31 can be used in operation.

An undefined instruction exception, coprocessor nonexistent trap, coprocessor error trap, and INT instruction are not affected by the interrupt level, and the ILM is not changed.

■ I flag

The I flag allows and prohibits interrupts. The I flag is set in CCR bit 4 of the PS register.

I	Function of the flag
0	Interrupt prohibited Set to "0" when an INT instruction is executed (The value saved to the stack is the value before being cleared.)
1	Interrupt allowed Masking an interrupt request is controlled by the ILM value.

■ Interrupt Level Mask Register (ILM)

The ILM register (one of the PS registers 20 to 16) stores an interrupt level mask value.

An interrupt request to be entered in the CPU is accepted only when the corresponding interrupt level is higher than the level indicated by this ILM.

Level value 0 (00000_B) is the highest; level value 31 (11111_B) is the lowest.

Values that can be set by the program are restricted. If the original value is between 16 and 31, a value between 16 and 31 can be set as a new value. When an instruction is executed that sets a value from 0 to 15, "specified-value + 16" is transferred.

If the original value is between 0 and 15, a value between 0 and 31 can be set.

Note:

Use the STILM instruction to set an interrupt level mask register (ILM).

■ Level Mask for Interrupt

If an interrupt request occurs, the interrupt level of the interrupt cause (as shown in Table 3.9-1 "Interrupt Levels") is compared with the level mask value stored in the ILM. When the following condition is found to be satisfied in the comparison, the interrupt level is masked and the interrupt request is not accepted:

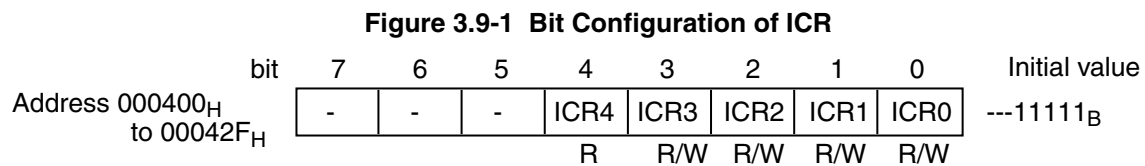
The interrupt level of the cause is equal to or greater than the level mask value.

3.9.2 Interrupt Control Register (ICR)

The ICR is provided by the interrupt controller. This register sets a level for each interrupt request. The ICR supports all types of interrupt requests. It is allocated in the I/O area and accessed by the CPU through a bus.

■ ICR Configuration

Figure 3.9-1 "Bit Configuration of ICR" shows the bit configuration of the ICR.



The following figure shows the bit configuration of the ICR.

[Bit 4] ICR4

This bit is always set to "1".

[Bits 3 to 0] ICR3 to ICR0

Four low-order bits representing the interrupt level of the corresponding interrupt cause.

These bits can be read and written.

Together with bit 4, a value between 16 and 31 can be set in the ICR.

■ ICR Mapping

Table 3.9-2 "Mapping between Interrupt Cause and Interrupt Vector" shows the mapping applied between the interrupt cause and the interrupt vector.

Table 3.9-2 Mapping between Interrupt Cause and Interrupt Vector

Interrupt cause	Interrupt control register		Corresponding interrupt vector		
	Number	Address	Number		Address
			Hexa-decimal	Decimal	
IRQ00	ICR00	00000400 _H	10 _H	16	TBR + 3BC _H
IRQ01	ICR01	00000401 _H	11 _H	17	TBR + 3B8 _H
IRQ02	ICR02	00000402 _H	12 _H	18	TBR + 3B4 _H
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
IRQ45	ICR45	0000042D _H	3D _H	61	TBR + 308 _H
IRQ46	ICR46	0000042E _H	3E _H	62	TBR + 304 _H
IRQ47	ICR47	0000042F _H	3F _H	63	TBR + 300 _H

See CHAPTER 12 "INTERRUPT CONTROLLER" for details.

3.9.3 System Stack Pointer (SSP)

The SSP is used as a pointer to the stack for saving and restoring data during EIT acceptance and return operations

■ System Stack Pointer (SSP)

The SSP consists of 32 bits.
Figure 3.9-2 "Bit Configuration of SSP" shows the bit configuration of the SSP.

Figure 3.9-2 Bit Configuration of SSP

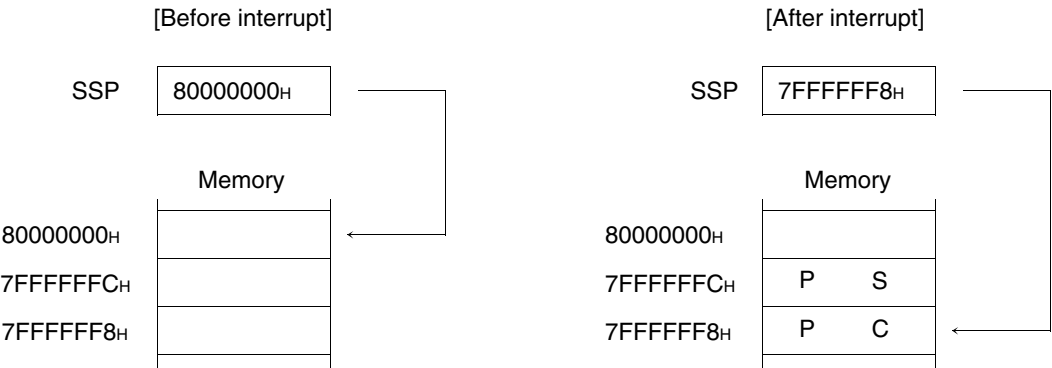


A value of 8 is subtracted from the contents of the SSP during EIT processing. A value of 8 is added during RETI processing.
A reset initializes the register to 00000000H.
The SSP serves as general-purpose register R15 if the S flag in the CCR is set to "0".

■ Interrupt Stack

The interrupt stack can be referenced from the SSP. The PC and PS values are saved or restored in this area. After an interrupt, the PC is stored at the address indicated by SSP, and PS is stored at address "SSP + 4".
Figure 3.9-3 "Example of Interrupt Stack Operation" shows an example of interrupt stack operation.

Figure 3.9-3 Example of Interrupt Stack Operation



3.9.4 Table Base Register (TBR)

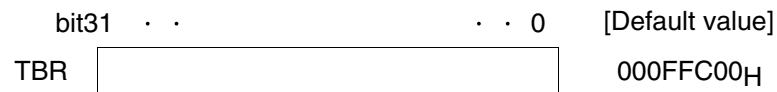
The TBR stores the starting address of the EIT vector table.

■ Table Base Register (TBR)

The TBR consists of 32 bits.

Figure 3.9-4 "Bit Configuration of TBR" shows the bit configuration of the TBR.

Figure 3.9-4 Bit Configuration of TBR



A vector address is obtained by adding the contents of the TBR and the offset value determined for the respective EIT.

A reset initializes the register to 000FFC00_H.

■ EIT Vector Table

The EIT vector table is a 1 Kbyte area beginning with the address indicated by the TBR.

One vector takes an area of 4 bytes. The relationship between the vector number and vector address is as follows:

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

vctadr : Vector address

vctofs : Vector offset

vct : Vector number

The two low-order bits of the addition result are always regarded to be "00".

The area between 000FFC00_H and 000FFFFF_H is the initial vector table area after reset.

Some special functions are allocated to certain vectors.

Table 3.9-3 "Vector Table" shows the structure of the vector table.

Table 3.9-3 Vector Table

Vector offset (hexadecimal)	Vector number		Description
	Hexa- decimal	Decimal	
3FC	00	0	Reset *1
3F8	01	1	Reserved by the system
3F4	02	2	Reserved by the system

Table 3.9-3 Vector Table (Continued)

Vector offset (hexadecimal)	Vector number		Description
	Hexa- decimal	Decimal	
3F0	03	3	Reserved by the system
:	:	:	:
:	:	:	:
:	:	:	:
3E0	07	7	Reserved by the system
3DC	08	8	Reserved by the system
3D8	09	9	INTE instruction
3D4	0A	10	Reserved by the system
3D0	0B	11	Reserved by the system
3CC	0C	12	Step trace trap
3C8	0D	13	Reserved by the system
3C4	0E	14	Undefined instruction exception
3C0	0F	15	NMI (for user)
3BC	10	16	Maskable interrupt cause #0
3B8	11	17	Maskable interrupt cause #1 *2
:	:	:	:
:	:	:	:
:	:	:	:
300	3F	63	Maskable interrupt cause/INT instruction
2FC	40	64	Reserved by the system (used by REALOS)
2F8	41	65	Reserved by the system (used by REALOS)
2F4	42	66	Maskable interrupt cause/INT instruction
:	:	:	:
:	:	:	:
:	:	:	:
000	FF	255	

*1: The fixed address 000FFFFC_H is always used for the reset vector, even when the TBR value changes.

*2: See Appendix B, "Interrupt Vectors", for details on the vector table for this model.

3.9.5 70Concurrent EIT Processing

If multiple EIT causes exist at the same time, the CPU selects and accepts first one cause and executes the respective EIT sequence. After that, the CPU determines the next EIT cause again and performs the EIT sequence for that EIT cause. When no more acceptable EIT causes can be detected, the CPU executes the handler instruction for the last accepted EIT cause.

Therefore, if multiple EIT causes exist at the same time, the execution order of handlers for each cause depends on the following two factors:

- EIT cause acceptance priority
- How other causes are masked when an EIT cause is accepted

■ EIT Cause Acceptance Priority

The EIT cause acceptance priority determines the order of selecting a cause for which to execute the EIT sequence. The cause is selected by saving the PS and PC and updating the PC (if necessary) to mask other causes.

The handler of the cause accepted first is not always the first to be executed.

Table 3.9-4 "EIT Cause Acceptance Priority and Masking Other Causes" shows the EIT cause acceptance priority.

Table 3.9-4 EIT Cause Acceptance Priority and Masking Other Causes

Acceptance priority	Cause	Masking other causes
1	Reset	Other causes are discarded.
2	Undefined instruction exception	Canceled
3	INT instruction	I flag = 0
	Coprocessor absent trap	None
	Coprocessor error trap	
4	User interrupt	ILM=level of accepted cause
5	NMI (for user)	ILM=15
6	Step trace trap	ILM=4
7	INTE instruction	ILM=4

Table 3.9-5 "EIT Handler Execution Order" lists the execution order of handlers for concurrently existing EIT causes in consideration of masking other causes after one of the EIT causes is accepted.

Table 3.9-5 EIT Handler Execution Order

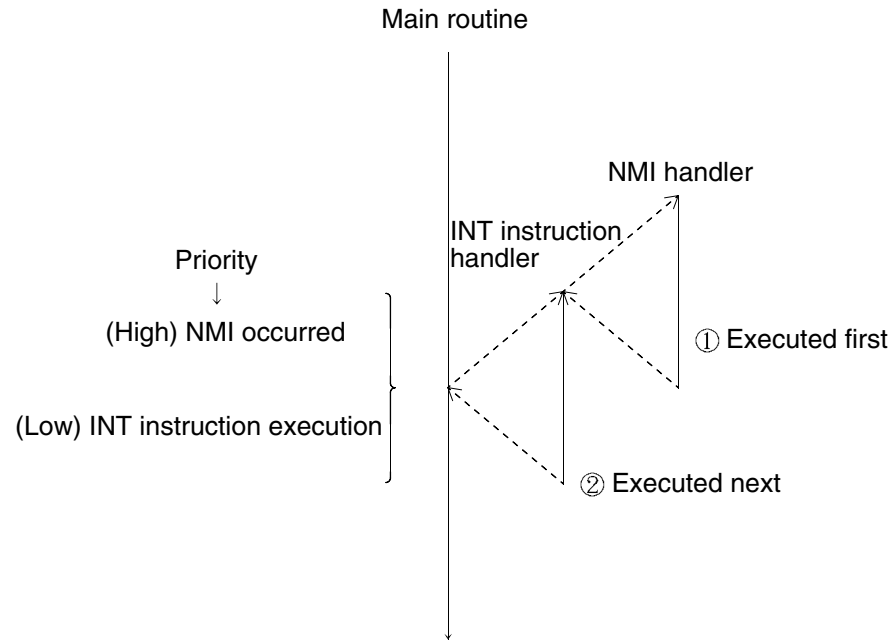
Handler execution order	Cause
1	Reset *1
2	Undefined instruction exception
3	Step trace trap *2
4	INTE instruction *2
5	NMI (for user)
6	INT instruction
7	User interrupt
8	Coprocessor absent trap, coprocessor error trap

*1: Other causes are discarded.

*2: The INTE instruction cannot be used if a step trace trap EIT is issued.

Figure 3.9-5 "Example of Concurrent EIT Processing" shows an example of concurrent EIT processing.

Figure 3.9-5 Example of Concurrent EIT Processing



3.9.6 EIT Operations

This section describes the EIT operations.

■ EIT Operations

In the description, the transfer source "PC" refers to the address of the instruction that detected the EIT cause.

The term "next instruction address" in the explanation of the operation indicates that the address of the instruction that detected EIT is one of the following:

- LDI:32: PC + 6
- LDI:20, COPOP, COPLD, COPST, or COPSV: PC + 4
- Other instructions: PC + 2"

■ User Interrupt Operation

If a user interrupt request is detected, the CPU determines whether the request can be accepted in the following steps:

○ Determining whether the interrupt request can be accepted

1. The CPU compares the interrupt levels of requests that occur concurrently and selects the request with the highest level (smallest value). The level comparison is made based on the values stored by the corresponding ICR for maskable interrupts.
2. If multiple interrupt requests with the same level are detected concurrently, the CPU selects the interrupt request with the smallest interrupt number.
3. The CPU compares the interrupt level of the selected interrupt request with the level mask value determined by ILM:
 - When the interrupt level is equal to or greater than the level mask value, the interrupt request is masked and not accepted.
 - When the interrupt level is smaller than the level mask value, proceed to step 4).
4. When the the I flag is 0 and the selected interrupt request is maskable, the interrupt request is masked and not accepted. When the I flag is 1, proceed to step 5).
- When the selected interrupt request is NMI, proceed to step 5) regardless of the I flag value.
5. When the conditions above are satisfied, the interrupt request is accepted at a break in instruction processing.

When a user interrupt request is accepted at EIT request detection, the CPU uses the interrupt number that corresponds to the accepted interrupt request and operates as follows:

In [Operation], the parenthesized item indicates the address indicated by the register:

[Operation]

1. SSP - 4 --> SSP
2. PS --> (SSP)
3. SSP - 4 --> SSP

4. Next instruction address --> (SSP)
5. Interrupt level of accepted request --> ILM
6. "0" --> S flag
7. (TBR + vector offset of accepted interrupt request) --> PC

When the interrupt sequence is terminated, the CPU determines a new EIT before executing the starting instruction of the handler. When there is an acceptable EIT, the CPU moves to the respective EIT processing sequence.

■ INT Instruction Operation

The CPU processes the INT #u8 instruction as shown below.

It branches to the interrupt handler of the vector indicated by u8.

[Operation]

1. $SSP - 4 \rightarrow SSPX$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. $PC + 2 \rightarrow (SSP)$
5. "0" --> I flag
6. "0" --> S flag: $(TBR + 3FC_H - 4 \times u8) \rightarrow PC$
7. $(TBR + 3FC_H - 4 \times u8) \rightarrow PC$

■ INTE Instruction Operation

The CPU processes the INTE instruction as shown below.

It branches to the interrupt handler of vector #9.

[Operation]

1. $SSP - 4 \rightarrow SSP$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. $PC + 2 \rightarrow (SSP)$
5. "00100" --> ILM
6. "0" --> S flag
7. $(TBR + 3D8_H) \rightarrow PC$

Do not use an INTE instruction within another INTE instruction or within a processing routine for a step trace trap.

No EIT is caused by INTE instructions during step execution.

■ Step Trace Trap Operation

When the T flag is set in the SCR of the PS and the step trace function is set to "enabled", a trap is issued for each instruction, and a break occurs.

The step trace trap is detected under the following conditions:

- T flag = 1

- Not the delay branch instruction
- An instruction other than the INTE instruction and processing routine of the step trace trap is being executed.

When the conditions above are satisfied, a "break" is performed at the break in instruction operation.

[Operation]

1. $SSP - 4 \rightarrow SSP$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. Next instruction address $\rightarrow (SSP)$
5. "00100" $\rightarrow ILM$
6. "0" $\rightarrow S$ flag
7. $(TBR + 3CC_H) \rightarrow PC$

When the T flag is set and step trace trap is set to "enabled", user interrupts are prohibited.

In this case, no EIT is caused by the INTE instruction.

■ **Operation for undefined instruction exception**

When the CPU detects that the instruction being decoded is undefined, an undefined instruction exception is thrown.

The undefined instruction exception is detected under the following conditions:

- The instruction being decoded is detected as being undefined.
- The instruction is not located in a delay slot (not immediately after a delay branch instruction).

If the conditions above are satisfied, an undefined instruction exception is thrown and a break occurs.

[Operation]

1. $SSP - 4 \rightarrow SSP$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. $PC \rightarrow (SSP)$
5. "0" $\rightarrow S$ flag
6. $(TBR + 3C4_H) \rightarrow PC$

The address of the instruction in which an undefined instruction exception is detected is saved in the PC.

■ **Coprocessor Absent Trap**

A coprocessor absent trap occurs if an executed coprocessor instruction attempts to use a nonexistent coprocessor.

[Operation]

1. $SSP - 4 \rightarrow SSP$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. Address of next instruction $\rightarrow (SSP)$
5. "0" \rightarrow S flag
6. $(TBR + 3C4_H) \rightarrow PC$

■ Coprocessor Error Trap

If an error occurs during use of a coprocessor and an executed coprocessor instruction then attempts to manipulate the coprocessor, a coprocessor error trap occurs. (Note that the MB91F128 contains no coprocessor.)

[Operation]

1. $SSP - 4 \rightarrow SSP$
2. $PS \rightarrow (SSP)$
3. $SSP - 4 \rightarrow SSP$
4. Address of next instruction $\rightarrow (SSP)$
5. "0" \rightarrow S flag
6. $(TBR + 3C4_H) \rightarrow PC$

■ RETI instruction operation

The RETI instruction is used to return from the EIT processing routine.

[Operation]

1. $(R15) \rightarrow PC$
2. $R15 + 4 \rightarrow R15$
3. $(R15) \rightarrow PS$
4. $R15 + 4 \rightarrow R15$

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

3.10 Reset Sequence

This section describes the reset operation that returns the CPU to initial state.

■ Reset Causes

The reset causes are as follows:

- Input from external reset pins
- Software reset by using the SRST bit in the standby control register (STCR)
- Count-up of watchdog timer
- Power-on reset

■ Initializing the CPU by Reset

When a reset cause is detected, the CPU is initialized.

○ Release by external reset pins/software reset

- Set the pins to the defined states.
- Reset resources in the device. The control register is initialized to the defined value.
- The slowest gear is selected for the clock.

■ Reset Sequence

When the reset cause is canceled, the CPU executes the following reset sequence:

- (000FFFFC_H) --> PC

Note:

After reset, use the mode register to set the operation mode accurately.

See the description of the mode register in Section 3.13 "Operation Mode" for details.

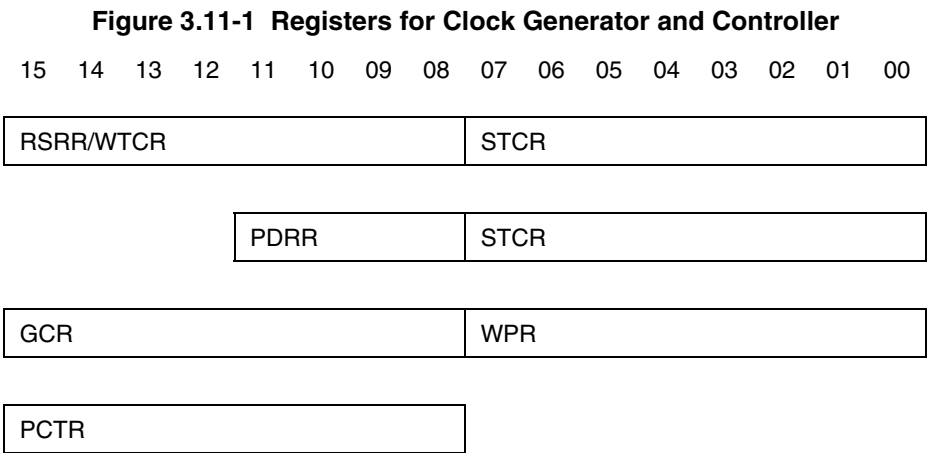
3.11 Clock Generator and Controller

The clock generator module has the following functions:

- CPU clock generation (including gear function)
- Peripheral clock generation (including gear function)
- Resetting and holding reset cause
- Standby function (including hardware standby)
- DMA request suppression
- Built-in PLL (multiplication circuit)

■ Registers for Clock Generator and Controller

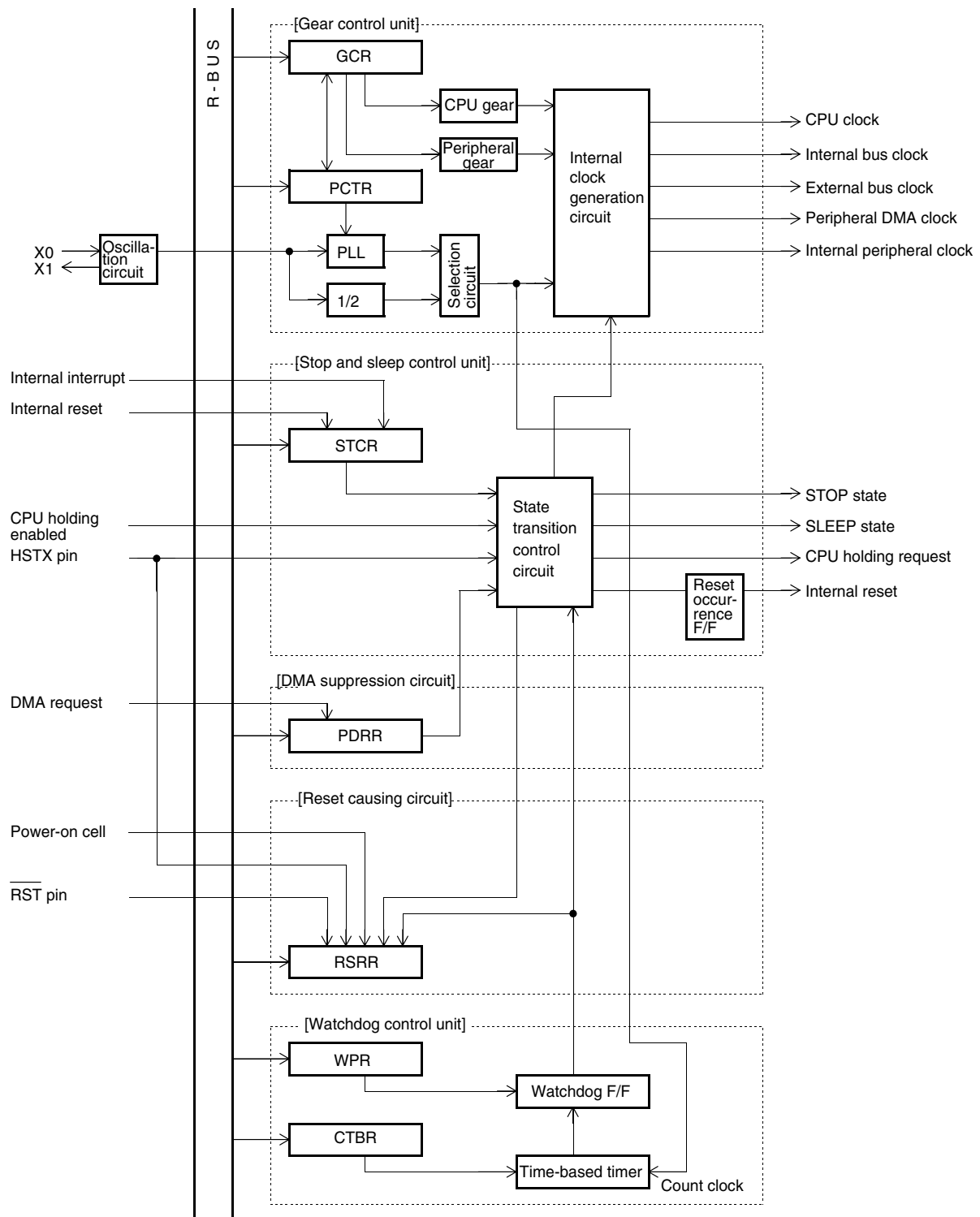
Figure 3.11-1 "Registers for Clock Generator and Controller" shows the clock generator registers.



■ Block Diagram of Clock Generator and Controller

Figure 3.11-2 "Block Diagram of Clock Generator and Controller" shows a block diagram of the clock generator.

Figure 3.11-2 Block Diagram of Clock Generator and Controller"



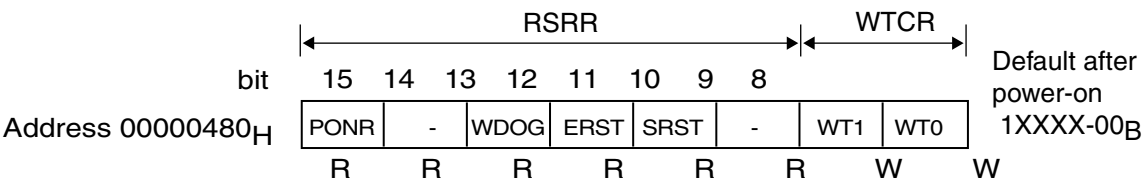
3.11.1 Reset Cause Register (RSRR) and Watchdog Cycle Control Register (WTCR)

The RSRR stores the type of reset that occurred. The WTCR specifies the watchdog timer cycle.

■ RSRR and WTCR configuration

Figure 3.11-3 "Register Configuration of RSRR and WTCR" shows the register configuration of the RSRR and WTCR.

Figure 3.11-3 Register Configuration of RSRR and WTCR



The functions of bits of the RSRR and WTCR are explained below.

[Bit 15] PONR

When this bit is "1", the reset that occurred immediately before is a power-on reset. When this bit is "1", values in bits other than this bit are invalid.

[Bit 14] HSTB

When this bit is "1", the reset that occurred immediately before was caused by hardware standby.

[Bit 13] WDOG

When this bit is "1", the reset that occurred immediately before was a watchdog reset.

[Bit 12] ERST

When this bit is "1", the reset that occurred immediately before was caused by external reset pins.

[Bit 11] SRST

This bit is reserved. The read value is not defined.

[Bit 10] (Reserved)

This bit is reserved. The read value is not defined.

[Bits 09 and 08] WT1 and WT0

These bits specify the watchdog cycle. Table 3.11-1 "Watchdog Interval Specified by WT1 and WT0" shows the relationship between these bits and the cycle selected. These bits are initialized by all reset operations.

Table 3.11-1 Watchdog Interval Specified by WT1 and WT0

WT1	WT0	Minimum writing interval to WPR required for suppressing the generation of watchdog resets	Time between the last time 5A _H was written to WPR and watchdog reset
0	0	$\phi \times 2^{15}$ [Default]	$\phi \times 2^{15}$ to $\phi \times 2^{16}$
0	1	$\phi \times 2^{17}$	$\phi \times 2^{17}$ to $\phi \times 2^{18}$
1	0	$\phi \times 2^{19}$	$\phi \times 2^{19}$ to $\phi \times 2^{20}$
1	1	$\phi \times 2^{21}$	$\phi \times 2^{21}$ to $\phi \times 2^{22}$

When GCR CHC is 1, f is twice X0. When CHC is 0, ϕ is one oscillation cycle of PLL.

3.11.2 Standby Control Register (STCR)

The standby control register (STCR) controls the standby operation and indicates the time to wait for oscillation stabilization.

■ STCR Configuration

Figure 3.11-4 "Register Configuration of STCR" shows the register configuration of the STCR.

Figure 3.11-4 Register Configuration of STCR

bit	7	6	5	4	3	2	1	0	Initial value
Address 00000481 _H	STOP	SLEP	HIZX	SRST	OSC1	OSC0	-	-	000111-- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The functions of bits of the STCR are explained below.

[Bit 07] STOP

When this bit is set, the system switches to stop status where built-in peripheral clocks, internal CPU clocks, and oscillation are stopped.

[Bit 06] SLEP

When this bit is set, the system switches to sleep status where internal CPU clocks are stopped.

If both this bit and bit 07 are set, this bit is ignored and stop status is set.

[Bit 05] HIZX

When stop status is set by setting this bit, the device pin impedance increases.

[Bit 04] SRST

When this bit is cleared, a software reset request is issued.

[Bits 03 and 02] OSC1 and OSC0

These bits indicate the time to wait for oscillation stabilization. Table 3.11-2 "Oscillation Stabilization Wait Times Specified with OSC1 and OSC0" shows which oscillation stabilization wait times (cycles) corresponding to bit combinations.

These bits are initialized by a power-on reset but not affected by any other reset source.

Table 3.11-2 Oscillation Stabilization Wait Times Specified with OSC1 and OSC0

OSC1	OSC0	Oscillation stabilization wait time
0	0	$\phi \times 2^{15}$
0	1	$\phi \times 2^{17}$
1	0	$\phi \times 2^{19}$
1	1	$\phi \times 2^{21}$ [Initial value]

ϕ is a cycle twice as long as X0 if GCR CHC = 1, and it is as long as X0 if GCR CHC = 0.

[Bits 01 and 00] (Reserved)

These bits are reserved. Their values during read operations are undefined.

3.11.3 DMA Request Suppression Register (PDRR)

The PDRR temporarily suppresses DMA requests and operates the CPU.

■ PDRR Configuration

Figure 3.11-5 "Register Configuration of PDRR" shows the register configuration of the PDRR.

Figure 3.11-5 Register Configuration of PDRR

bit	15	14	13	12	11	10	9	8	Initial value
Address 00000482 _H	-	-	-	-	D3	D2	D1	D0	----0000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The functions of bits of the PDRR are explained below.

[Bits 11 to 08] D3 to D0

If a value other than 0 is written to this register, subsequent DMA transfer requests from the DMA to the CPU are suppressed. DMA transfer cannot be performed until 0 is specified in this register.

3.11.4 Time-based Timer Clear Register (CTBR)

CTBR initializes the content of the time-based timer to 0.

■ CTBR Configuration

Figure 3.11-6 "Register Configuration of CTBR" shows the register configuration of the CTBR.

Figure 3.11-6 Register Configuration of CTBR

bit	7	6	5	4	3	2	1	0	Initial value
Address 00000483 _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
	W	W	W	W	W	W	W	W	

The functions of bits of the CTBR are explained below.

[Bits 07 to 00] D7 to D0

When A5_H and 5A_H are written to this register in this order, the time-based timer is cleared to 0 immediately after 5A_H was written. The read value of this register is not defined. There is no restriction on the time between writing A5_H and 5A_H.

Note:

If the time-based timer is cleared using this register, the interval of waiting for oscillation stabilization, watchdog cycle, and cycles of peripheral devices using the time-based timer vary temporarily.

3.11.5 Gear Control Register (GCR)

The GCR controls the gear functions of the CPU and peripheral clocks.

■ GCR Configuration

Figure 3.11-7 "Register Configuration of GCR" shows the register configuration of the GCR.

Figure 3.11-7 Register Configuration of GCR

bit	15	14	13	12	11	10	9	8	Initial value
Address 00000484 _H	CCK1	CCK0	DBLAK	DBLON	PCK1	PCK0	-	CHC	110011-1 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The functions of bits of the GCR are explained below.

[Bits 15 and 14] CCK1 and CCK0

These bits specify the CPU gear cycle. Table 3.11-3 "CPU Machine Clock" shows the relationship between these bits and the cycle selected. The bits are initialized by a reset operation.

Table 3.11-3 CPU Machine Clock

CCK1	CCK0	CHC	CPU machine clock
0	0	0	PLL × 1
0	1	0	PLL × 1/2
1	0	0	PLL × 1/4
1	1	0	PLL × 1/8
0	0	1	Oscillation × 1/2
0	1	1	Oscillation × 1/2 × 1/2
1	0	1	Oscillation × 1/2 × 1/4
1	1	1	Oscillation × 1/2 × 1/8 [Default]

PLL: PLL oscillation frequency

Oscillation: Input frequency from X0

[Bit 13] DBLAK

This bit indicates the operation state of the clock doubler. A value written to this read-only bit is ignored. This bit is initialized by a reset operation.

There is a time lag in switching the bus frequency. This bit is used to confirm that the bus frequency has been switched.

DBLAK	Internal:external operation frequency
0	Operating at 1:1 [default]
1	Operating at 2:1

[Bit 12] DBLON

This bit specifies the operation state of the clock doubler. This bit is initialized by a reset operation.

DBLON	Internal:external operation frequency
0	Operating at 1:1 [default]
1	Operating at 2:1

[Bits 11 and 10] PCK1 and PCK0

These bits specify the peripheral gear cycle. Table 3.11-4 "Peripheral Machine Clock" shows the relationship between these bits and the cycle selected. The bits are initialized by a reset operation.

Table 3.11-4 Peripheral Machine Clock

PCK1	PCK0	CHC	Peripheral machine clock (oscillation: input frequency from X0)
0	0	0	PLL × 1
0	1	0	PLL × 1/2
1	0	0	PLL × 1/4
1	1	0	PLL × 1/8
0	0	1	Oscillation × 1/2
0	1	1	Oscillation × 1/2 × 1/2
1	0	1	Oscillation × 1/2 × 1/4
1	1	1	Oscillation × 1/2 × 1/8 [Default]

PLL: PLL oscillation frequency

Oscillation: Input frequency from X0

When the CPU clock operates at a frequency higher than 25 MHz, set the frequency of the peripheral clock to at least half the frequency of the CPU clock.

The maximum frequency of the peripheral clock is 25 MHz.

Note:

To change the CPU and peripheral gears at the same time, set both gears to the same value, then set each to the desired value.

The CPU and peripheral gears can be set to each desired value at the same time under one of the following conditions: 1) The CPU and peripheral gears before change are set to the same value; 2) Only one gear is changed; 3) Both gears are set to the same value.

When the clock doubler is on, the CPU gear is set to the fixed value regardless of the GCR value. Both gears can also be set to the desired value at the same time.

[Sample program code]

```
ldi    #0x484, r1
ldi    #0x0d,  r0
stb    r0,     @r1      ; CPU: 1/1, Peripheral: 1/8
:
:
ldi    #0x484, r1
ldi    #0xcd,  r0
stb    r0,     @r1      ; CPU: 1/8, Peripheral: 1/8  Temporarily set to same ratio.
ldi    #0xc5,  r0
stb    r0,     @r1      ; CPU: 1/8, Peripheral: Set to desired ratios.
```

[Bit 09] Reserved

Always set this bit to 1.

[Bit 08] CHC

This bit sets the reference clock selection. This bit is initialized by a reset operation. When the VSTP bit of PCTR is 1, writing 0 to this bit is ignored.

CHC	Clock selection
1	Uses half the clock cycle from the oscillation circuit as reference clock [default].
0	Uses the oscillation output from the PLL as reference clock.

Note:

When the VSTR bit of PCTR is 0, PLL stops oscillation when the CPU enters stop mode. However, the VSTP value is still 0. When the CPU returns from a stop that was caused due to an external interrupt or for another cause, PLL oscillation does not become stable for about 100μ seconds plus the time to wait for oscillation stabilization as set in STCR OSC1 and OSC0. Do not set this bit to 0 until this time has elapsed.

For the procedure and information on internal operations to return from stop mode, see Section 3.12.1, "Stop status."

3.11.6 Watchdog Reset Suspending Register (WPR)

The WPR clears the watchdog timer flip-flop. This register is used to suspend a watchdog reset.

■ WPR Configuration

Figure 3.11-8 "Register Configuration of WPR" shows the register configuration of the WPR.

Figure 3.11-8 Register Configuration of WPR

bit	7	6	5	4	3	2	1	0	Initial value
Address 00000485 _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
	W	W	W	W	W	W	W	W	

The functions of bits of the WPR are explained below.

[Bits 07 to 00] D7 to D0

When A5_H and 5A_H are written to this register in this order, the watchdog timer flip-flop is cleared to 0 immediately after 5A_H was written and watchdog reset is suspended.

The read value of this register is not defined. There is no restriction on the time between writing A5_H and 5A_H. If writing both data items is not complete within the cycle in Table 3.11-5 "Watchdog Interval Specified by WT1 and WT0" a watchdog reset occurs.

Table 3.11-5 Watchdog Interval Specified by WT1 and WT0

WT1	WT0	Minimum time for writing to WPR required to suppress watchdog reset	Time between the last time 5A _H was written to WPR and watchdog reset
0	0	$\phi \times 2^{15}$	$\phi \times 2^{15}$ to $\phi \times 2^{16}$
0	1	$\phi \times 2^{17}$	$\phi \times 2^{17}$ to $\phi \times 2^{18}$
1	0	$\phi \times 2^{19}$	$\phi \times 2^{19}$ to $\phi \times 2^{20}$
1	1	$\phi \times 2^{21}$	$\phi \times 2^{21}$ to $\phi \times 2^{22}$

When GCR CHC is 1, ϕ is twice X0. When CHC is 0, ϕ is one oscillation cycle of PLL.

3.11.7 PLL Control Register (PCTR)

The PCTR controls PLL oscillation.

The values set in this register can be changed only when GCR CHC is 1.

■ PCTR Configuration

Figure 3.11-9 "Register Configuration of PCTR" shows the register configuration of the PCTR.

Figure 3.11-9 Register Configuration of PCTR

bit	15	14	13	12	11	10	9	8	Initial value
Address 00000488 _H	SLCT1	SLCT0	-	-	VSTP	-	-	-	00--0--- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The functions of bits of the PCTR are explained below.

[Bits 15 and 14] SLCT1 and SLCT0

These bits control the PLL multiplication ratio and are only initialized at power-on.

The values set in these bits define the internal operation frequencies for GCR CHC=0.

The multiply-by-two setting can be used only if the oscillation frequency is 12.5 MHz or less.
Table 3.11-6 "Internal Operation Frequencies" lists the internal operation frequencies.

Table 3.11-6 Internal Operation Frequencies

SLCT1	SLCT0	Internal operation frequency (for 12.5 MHz oscillation)
0	0	Operated at 6.25 MHz [default]
0	1	Operated at 12.5 MHz (multiply-by-one)
1	0	Cannot be set
1	1	Operated at 25.0 MHz (multiply-by-two)

* Can be set only if the source oscillation frequency is 12.5 MHz or less.

Note

Do not use the multiply-by-two setting if the source oscillation frequency exceeds 12.5 MHz.

[Bits 13 and 12] Reserved

These bits are reserved.

[Bit 11] VSTP

This bit controls PLL oscillation. This bit is initialized at power-on and external reset. Stop PLL after every reset.

VSTP	PLL operation
0	Oscillation [default]
1	Oscillation stop

Note:

When the CPU enters stop mode, PLL stops regardless of the setting in this bit.

[Bits 10, 9, and 8] (Reserved)

These bits are reserved. "0" must always be written to them.

3.11.8 Watchdog Function

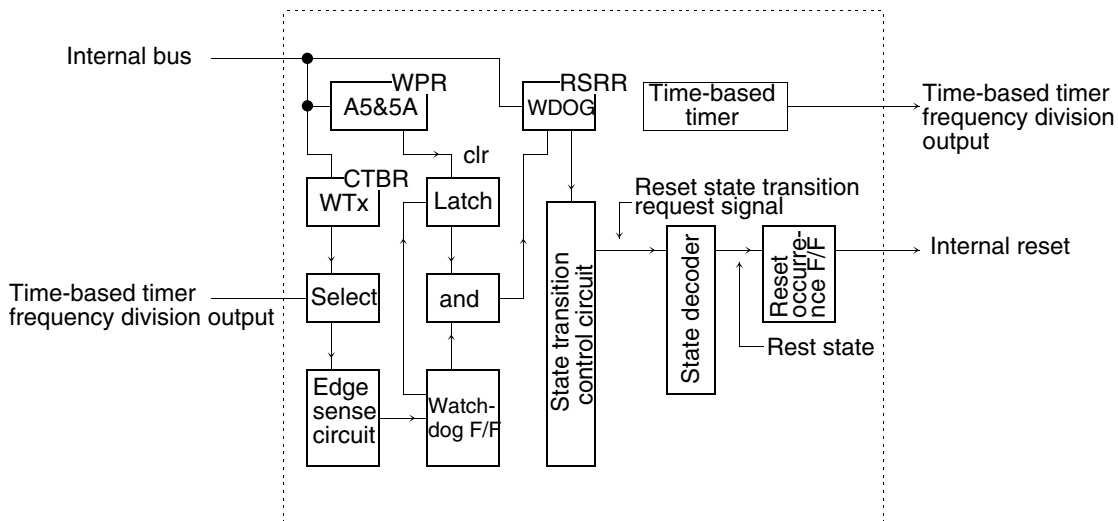
The watchdog function detects a program runaway state.

If $A5_H$ and $5A_H$ are not written to WPR within the defined time because of program runaway, the watchdog timer sends a watchdog reset request.

■ Block Diagram of Watchdog Control Unit

Figure 3.11-10 "Block Diagram of the Watchdog Control Unit" shows a block diagram of the watchdog control unit.

Figure 3.11-10 Block Diagram of the Watchdog Control Unit



■ Starting the Watchdog Timer

The watchdog timer is started by writing to WTCR. Set the watchdog timer interval time using the WT1 and WT0 bits. The interval time is defined the first time it is written, and values set in subsequent writing operations are ignored.

[Example]

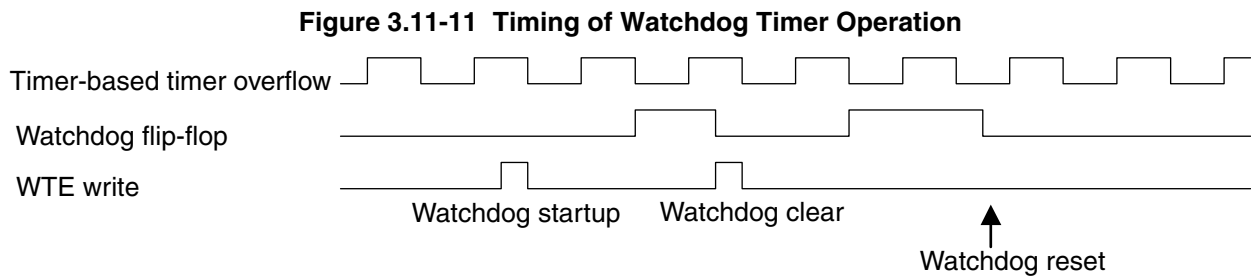
```
LDI:8    #00000010b, R1    ; WT1, 0 = 10
LDI:20    #WTCR, R2
STB      R1, @R2           ; Startup of the watchdog timer
```

■ Reset Suspension

Once the watchdog timer has started, the program must periodically write $A5_H$ and $5A_H$ to the WPR. The watchdog rest flip-flop stores a downward transition of the selected tap of the time-

based timer. If this flip-flop is not cleared at the second downward transition, a reset occurs.

Figure 3.11-11 "Timing of Watchdog Timer Operation" shows the timing of watchdog timer operation.



■ Causes of Reset Delays other than Programs

The following cause the watchdog timer to automatically delay generation of a reset:

1. Stop or sleep state
2. DMA transfer
3. A break occurs when the emulator debugger or the monitor debugger is being used.
4. The INTE instruction is executed.
5. Step trace trap (a break occurs at each instruction by specifying 1 for T in the PS register)

Note:

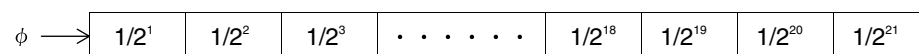
- There is no restriction on the interval between writing the first $A5_H$ and writing $5A_H$. The watchdog can suspend a reset only when the interval between writing the two values is within the time specified by the WT bits or if one $A5_H$ is written in between.
- If $5A_H$ is not written after the first $A5_H$, the first $A5_H$ write operation becomes invalid and $A5_H$ must be written again.

■ Time-based Timer

The time-based timer is used to supply a clock signal to the watchdog timer which acts as a timer to wait for stabilization of oscillation. When GCR CHC is 1, the operation clock ϕ is twice $X0$. When CHC is 0, ϕ is one oscillation cycle of the PLL.

Use the value of this time-based timer in RFCR as counter clock for DRAM refresh.

Figure 3.11-12 Time-based Timer Configuration



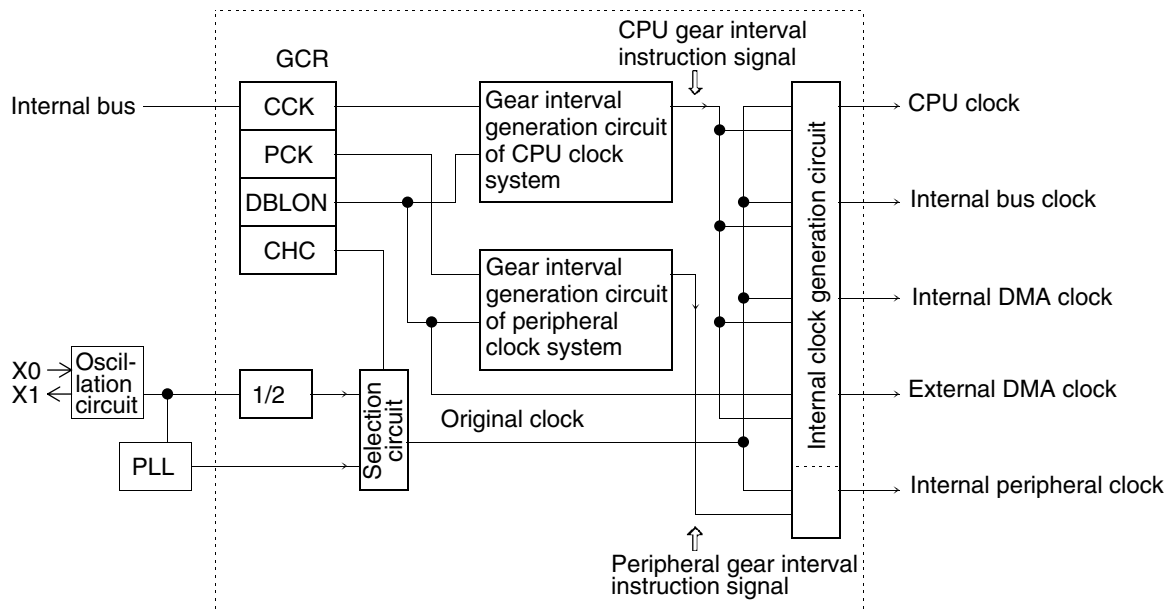
3.11.9 Gear Function

The gear function enables to provide a fractional clock signal. The gear function uses independent circuits for the CPU and the peripheral devices, and data can be exchanged between the CPU and the peripheral devices with different gear ratios. Either a clock signal following the same cycle as that of PLL or the clock through the divide-by-two circuit can be selected as the original clock.

■ Block Diagram of the Gear Control Unit

Figure 3.11-13 "Block Diagram of Gear Control Unit" shows a block diagram of the gear control unit.

Figure 3.11-13 Block Diagram of Gear Control Unit



■ Gear Function Settings

Set the CCK1 and CCK0 bits of GCR to the desired values to control the CPU clock. Set the PCK1 and PCK0 bits of GCR to the desired values to control the clock.

[Example]

```
LDI:20  #GCR, R2
LDI:8   #11111110b, R1      ; CCK=11, PCK=11, CHC=0
STB     R1, @R2              ; CPU clock=1/8f, Periferal clock=1/8f, f=direct
LDI:8   #01111010b, R1      ; CCK=01, PCK=10, CHC=0
STB     R1, @R2              ; CPU clock=1/2f, Periferal clock=1/4f, f=direct
LDI:8   #00111010b, R1      ; CCK=00, PCK=10, CHC=0
STB     R1, @R2              ; CPU clock=f, Periferal clock=1/4f, f=direct
LDI:8   #00110010b, R1      ; CCK=00, PCK=00, CHC=0
STB     R1, @R2              ; CPU clock=f, Periferal clock=f, f=direct
LDI:8   #10110010b, R1      ; CCK=10, PCK=00, CHC=0
STB     R1, @R2              ; CPU clock=1/4f, Periferal clock=f, f=direct
```

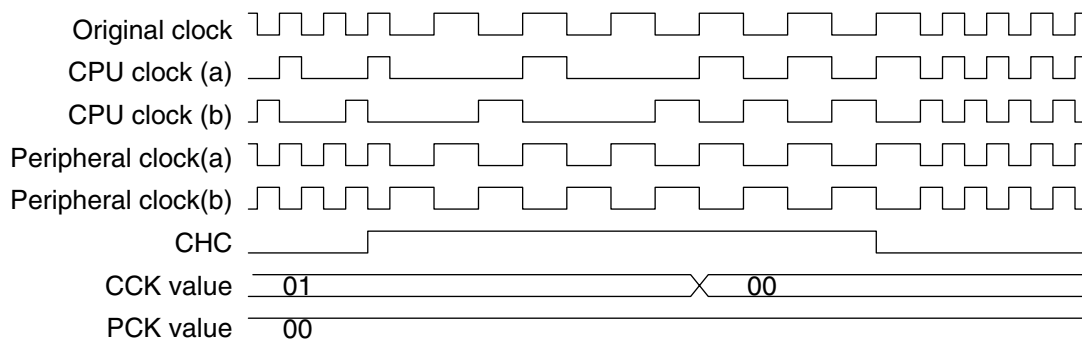
When the CHC bit of GCR is set to "1", the output of the divide-by-two circuit is selected as original clock signal. When this bit is set to "0", the clock with the same cycle as the oscillation circuit is used as the original clock. The CPU and peripheral circuits change concurrently when the original clock signal changes.

[Example]

```
LDI:8   #01110001b, R1      ; CCK=01, PCK=00, CHC=1
LDI:20  #GCR, R2
STB     R1, @R2              ; CPU clock=1/2f, Periferal clock=f, f=1/2xtal
LDI:8   #00110011b, R1      ; CCK=00, PCK=00, CHC=1
STB     R1, @R2              ; CPU clock=f, Periferal clock=f, f=1/2xtal
LDI:8   #00110010b, R1      ; CCK=00, PCK=00, CHC=0
STB     R1, @R2              ; CPU clock=f, Periferal clock=f, f=direct
```

Figure 3.11-14 "Chart of Clock Selection Timing" shows a chart of the clock selection timing

Figure 3.11-14 Chart of Clock Selection Timing



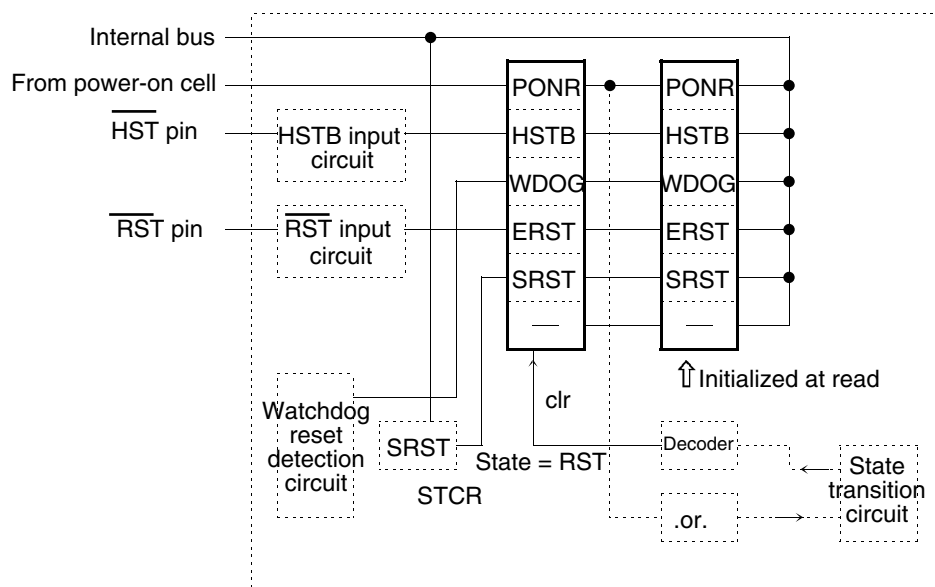
3.11.10 Reset Cause Holding

The reset cause holding register stores a reset cause that occurred immediately before. When this register is read, all flags are set to "0". Once set, the cause flag is not released until this register is read again.

■ Block diagram of reset cause holding circuit

Figure 3.11-15 "Block Diagram of Reset Cause Holding Circuit" shows a block diagram of the reset cause holding circuit.

Figure 3.11-15 Block Diagram of Reset Cause Holding Circuit



■ Reset Cause Holding Settings

No special setting is required when using this function. This function reads the reset cause holding register and branches to the appropriate program at the reset entry address.

[Example]

```

RESET-ENTRY
    LDI:20    #RSRR, R10
    LDI:8     #10000000B, R2
    LDUB     @R10, R1          ; GET RSRR VALUE INTO R1
    MOV      R1, R10           ; R10 USED AS A TEMPORARY REGISTER
    AND      R2, R10           ; WAS PONR RESET?
    BNE      PONR-RESET
    LSR      #1, R2            ; POINT NEXT BIT
    MOV      R1, R10           ; R10 USED AS A TEMPORARY REGISTER
    AND      R2, R10           ; WAS HARDWARE STANDBY RESET?
    BNE      HSTB-RESET
    LSR      #1, R2            ; POINT NEXT BIT
    MOV      R1, R2            ; R10 USED AS A TEMPORARY REGISTER
    AND      R2, R10           ; WAS WATCH DOG RESET?
    BNE      WDOG-RESET
    :
```

Note:

- When the PONR bit is 1, values in other bits are regarded as undefined. To check the reset cause, specify an instruction that checks for power-on reset at startup.
- Instructions that check reset causes other than power-on reset can be specified anywhere in the program. The cause check priority depends on instruction location.

3.11.11 DMA Suppression

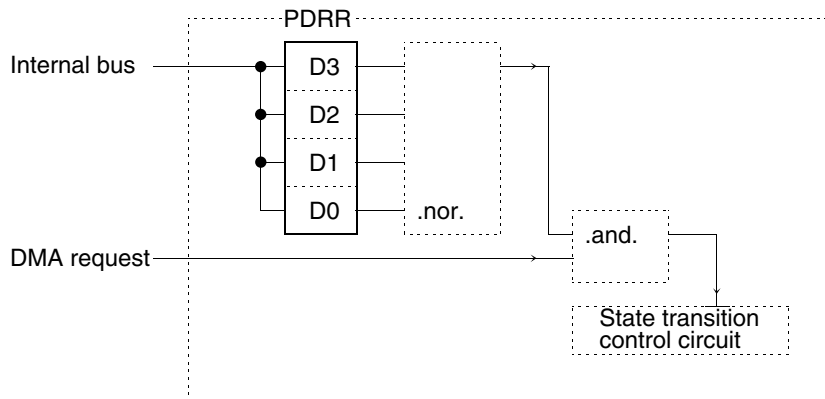
When an interrupt cause with higher priority is detected in the FR family during DMA transfer, the DMA transfer is interrupted and the system branches to the relevant interrupt routine. This feature is enabled when there are interrupt requests. When an interrupt cause is cleared, the suppression feature is disabled and DMA transfer is restarted by the interrupt processing routine.

If an interrupt cause with higher priority occurs and DMA transfer is interrupted, DMA transfer is restarted when the interrupt cause is cleared. To suppress the start of this DMA retransfer in the interrupt cause processing routine, use the DMA suppression function. The DMA suppression function is activated when a value other than 0 is set in the DMA suppression register and is stopped when this register is set to 0.

■ Block Diagram of DMA Suppression Circuit

Figure 3.11-16 "Block Diagram of the DMA Suppression Circuit" shows a block diagram of the DMA suppression circuit.

Figure 3.11-16 Block Diagram of the DMA Suppression Circuit



■ Setting DMA Suppression

As a general rule, use this function only in the interrupt processing routine.

Add 1 to the contents of the DMA suppression register before clearing the interrupt cause in the interrupt processing routine. This increase ensures that no DMA transfer takes place. Apply interrupt processing, then subtract 1 from the value in the DMA suppression register before returning. For multiple interrupts, DMA transfer remains suppressed, since the contents of the DMA suppression register does not change to 0. For a single interrupt, the contents of the DMA suppression register becomes 0 and the DMA request becomes soon effective.

[Example]

INT-ENTRY

```

LDI:20  #PDRR, R10
LD      @R10, R1      ; GET PDRR VALUE INTO R1
ADD     #1, R1
ST      R1, @R10      ; PDRR: =PDRR+1, DMA disabled
LDI:20  #int-REG, R10 ; int occurred with int-REG
LDI:8   #10H, R1      ; example, int-flag=#10h
ST      R1, @R10      ; CLEAR int-REQ, (but still DMA disabled)
:
; interrupt execute routine
:
LDI:20  #PDRR, R10
LD      @R10, R1      ; GET PDRR VALUE INTO R1
ADD2    #-1, R1
ST      R1, @R10      ; PDRR: PDRR-1, DMA may be enabled
RETI

```

Note:

- Because there are four bits in this register, this function cannot be used for multiple interrupts with more than 15 levels. Set the priority for the DMA task 15 levels higher than other interrupts.

3.11.12 Clock doubler function

To ensure that the external bus timing does not become too severe as the internal operation frequency rises, the frequency ratio between the external bus and internal operation can be set to 1:2.

■ Starting the Clock Doubler Function

The clock doubler function is enabled when 1 is set in GCR DBLON. When DBLON is set to 1, this function waits until all C-BUS accesses terminate, then switches the external bus clock. A minor time lag occurs until switching is terminated. The value set in GCR DBLAK is used to confirm the timing of the switching.

At startup of the clock doubler function, the CPU clock gear is set to 1/1 regardless of the GCR setting.

For the FR series, an operation frequency of up to twice as much as the source oscillation frequency can be set. Code the program as shown below to set the clock doubler function to "ON."

[Example]

```
DOUBLER-ON
    LDI:20    #GCR, R0
    BORL     #0001B, @R0      ; Switches to divide-by-two frequency (CHC = 1).
    BORH     #0001B, @R0      ; Clock doubler ON (DBLON = 1).
LOOP
    BTSTH    #0010B, @R0      ; DBLAK is checked.
    BEQ      LOOP             ; Loops until DBLAK becomes 1.
    BANDL    #1110B, @R0      ; Switches to PLL frequency (CHC = 0).
```

■ Stopping the Clock Doubler Function

The clock doubler function is invalid when 0 is set in GCR DBLON. The CPU clock gear setting of 1/1 returns to the CCK bit setting in GCR.

■ Notes on Turning the Clock Doubler Function On or Off

When the clock doubler function is set to on or off, a dead cycle of the internal clock may occur. This leads to an error during time measurement using the timer and during UART transfer.

■ Combinations of Operation Frequencies When the Clock Doubler Function is On or Off

Table 3.11-7 "Combination of Operation Frequencies when the Clock Doubler Function is On or Off" shows the operation frequencies of this device according to combinations between the SLCT1 and SLCT0 bits of the PCTR and the GCR settings. (The following example uses an oscillation of 12.5 MHz.)

Table 3.11-7 Combination of Operation Frequencies when the Clock Doubler Function is On or Off

GCR		PLL oscillation frequency (MHz)	Clock doubler	Internal operation frequency (MHz)	External bus frequency (MHz)	Remarks
CHC	Gear					
Divide-by-two frequency	1/1		OFF	6.25	6.25	
	1/2		OFF	3.13	3.13	
	1/4		OFF	1.56	1.56	
	1/8		OFF	0.78	0.78	Default
	*1		ON	6.25	3.13	
PLL *2	-	50.0	OFF	50.0	50.0	Cannot be set
	1/1	25.0	OFF	25.0	25.0	
	1/2	25.0	OFF	12.5	12.5	
	1/4	25.0	OFF	6.25	6.25	
	1/8	25.0	OFF	3.13	3.13	
	1/1	12.5	OFF	12.5	12.5	
	1/2	12.5	OFF	6.25	6.25	
	1/4	12.5	OFF	3.13	3.13	
	1/8	12.5	OFF	1.56	1.56	
	*1	50.0	ON	50.0	25.0	Must not be set.
	*1	25.0	ON	25.0	12.5	
	*1	12.5	ON	12.5	6.25	

*1: Fixed to 1/1 regardless of setting

*2: To change PLL oscillation frequency, it must be switched to divide-by-two frequency.

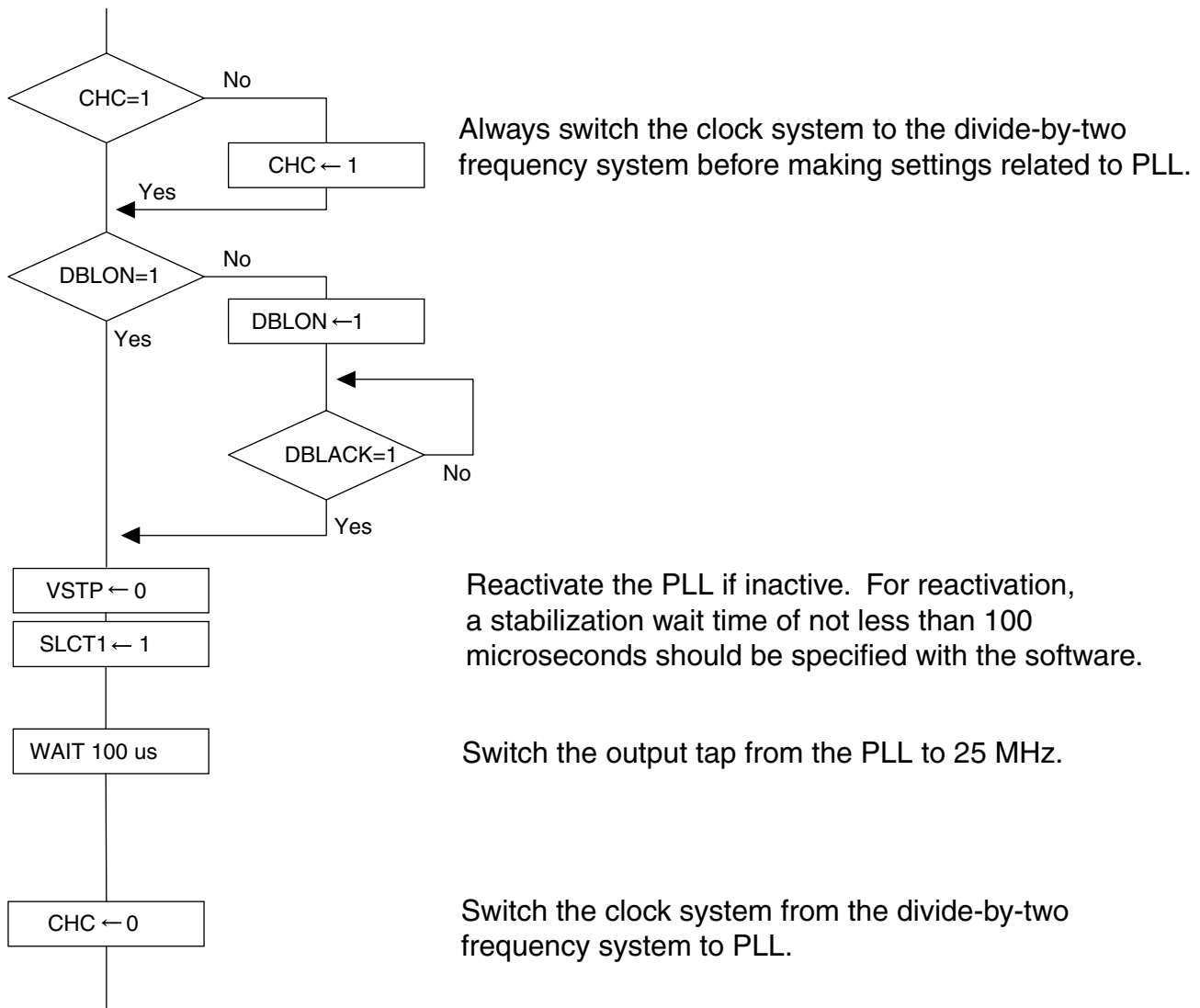
3.11.13 Examples of PLL Clock Settings

This section describes examples of PLL clock settings and the Assembler source code.

■ Example of PLL Clock Settings

Figure 3.11-17 "Example of PLL Clock Settings" shows an example of switching to 25-MHz operation with PLL. (if the source oscillation frequency is 12.5 MHz)

Figure 3.11-17 Example of PLL Clock Settings



Note:

- The order of setting the DBLON, VSTP, and SLCT1 bits is not specified.
- The operation frequency of the peripheral system must not exceed 25 MHz.
- For PLL VCO reactivation, specify via software a stabilization wait time of not less than 100

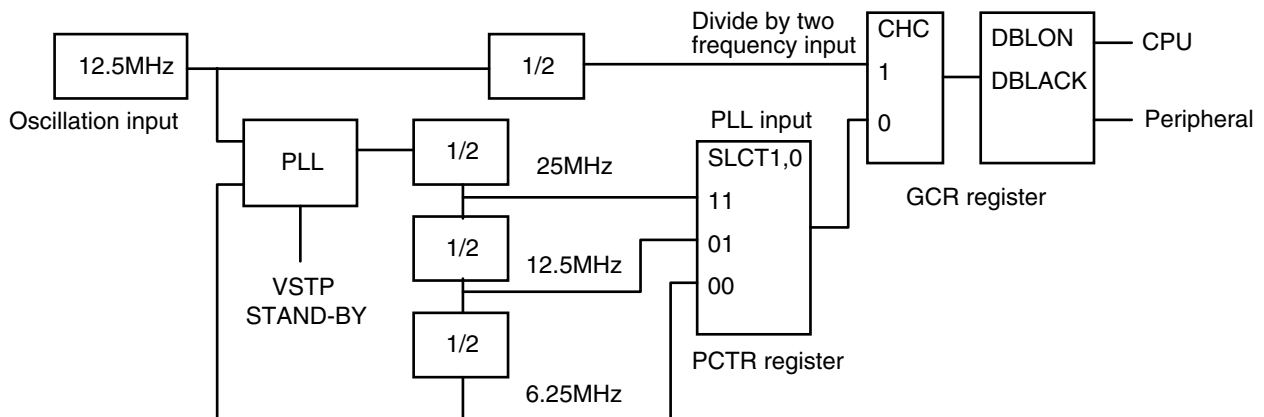
microseconds after determining whether the cache is on or off.

- To rewrite the SLCT bit to change the multiplication ratio, provide in the software a wait time of 100 Éps or more until PLL stabilizes.

■ Clock System Diagram for Reference

Figure 3.11-18 "Clock System Diagram for Reference" shows a clock circuit diagram for reference.

Figure 3.11-18 Clock System Diagram for Reference



■ Sample Assembler Source Code

```

; *****
;
; PLL Sample Program
; *****
; Load Setting Data
    ldi:20    #GCR, R0
    ldi:20    #PCTR, R1
    ldi:8     #GCR_MASK, R2    ; GCR_MASK = 0000 0001 b
    ldi:8     #PCTR_MASK, R3   ; PCTR_MASK = 0000 1000 b
    ldub      @R0, R4          ; read GCR register
    ldub      @R1, R5          ; read PCTR register
    st        PS, @-R15        ; push processor status
    stilm     #0x0             ; disable interrupt
;
    and       R4, R2
    beq       CHC_0
    bra       CHC_1
CHC_0:
    borl      #0001B, @r0      ; to 1/2 clock @r0=GCR register
CHC_1:
    call      VCO_RUN
PLL_SET_END:
    ld        @R15+, PS        ; pop processor status

```

```

, *****
,
,               VCO Setting
, *****
,
VCO_RUN:
    st        R3, @-R15        ; push R3
    ldi:8     #PCTR_MASK, R3    ; PCTR_MASK = 0000 1000 b
    and       R5, R3           ; PTCT->VSTP=1?
    beq       LOOP_100US_END   ; if VSTP=0 return
    bandl     #0111B, @r1      ; set VSTP=0
    st        R2, @-R15        ; push R2 for Loop counter
    ldi:20    #0x15E, R2       ; wait 100  $\mu$ S
WAIT_100US:
    add2      #(-1), R2        ; 100us = 160ns (6.25MHz) * 7 * 100 (2BC) cycle
    bne       WAIT_100US      ; 2BCh/2 = 15Eh (if cache on)
;
LOOP_100US_END:
    ld        @R15+, R2        ; Pop R2
    ld        @R15+, R3        ; Pop R3
    ret

```

3.12 Standby Mode (Low-power Consumption Mechanism)

Stop status and sleep status are available in standby mode.

■ Outline of Stop Status

The stop status refers to either of the statuses listed below. In the stop status, power consumption can be minimized.

- All internal clocks are stopped.
- The oscillation circuit is stopped.

Switch to the stop status as follows:

- Use an instruction to write data to a standby control register (STCR).
- Apply an L-level signal to the $\overline{\text{HST}}$ pin.

Return from the stop status as follows:

- Request an interrupt (only on peripheral systems in which an interrupt request can be issued in the stop status).

Note: INT4 and INT5 cannot be used to return from the stop status.

- Apply an L-level signal to the $\overline{\text{RST}}$ pin.
- Apply an L-level and then H-level signal to the $\overline{\text{HST}}$ pin.

Built-in peripheral circuits are stopped in stop mode unless an interrupt for operation restart can be generated because all internal clocks are stopped.

■ Outline of Sleep Status

The sleep status refers to the status described below. In the sleep status, power consumption can be limited to a certain level when CPU operation is not necessary.

- The CPU clock and internal bus clock are stopped.

The system can be switched to this status as follows:

- Use an instruction to write data to a standby control register (STCR).

Return from the stop status as follows:

- Request an interrupt.
- Generate a reset factor.

Since internal DMA and peripheral clocks operate in sleep status, sleep status can be released with an interrupt from any built-in peripheral system in which internal DMA and peripheral clocks are used.

■ Operations in Standby Mode

Table 3.12-1 "Operations in Standby Mode" lists the operations in the standby mode.

Table 3.12-1 Operations in Standby Mode

Operation status	Shift condition	Oscillator	Internal clock		Peripheral	Pin	Release method
			CPU/ internal bus	DMA/ peripheral			
Run		Y	Y	Y	Y	Operation	
Sleep	STCR SLEP = "1"	Y	N	Y	Y	Operation	
Stop	STCR STOP = "1"	N	N	N	N	*	
Hardware standby	\overline{HST} = "0"	N	N	N	N	Hi-Z	\overline{HST} = "1"

Y: Operation

N: Stop

*: The preceding status is stored with STCR HIZX = "0". Hi-Z is set when HIZX is set.

Note:

Reset: \overline{RST} = "0"

Standby control register SRST bit = "0"

Watchdog reset

Power-on reset

■ Address at Which Program That Can Enter Stop or Sleep Mode Must Be Placed

A program that can enter stop or sleep mode must be placed in the address area of ROM on the C-bus or external memory . It must not be placed in ROM on the C-bus.

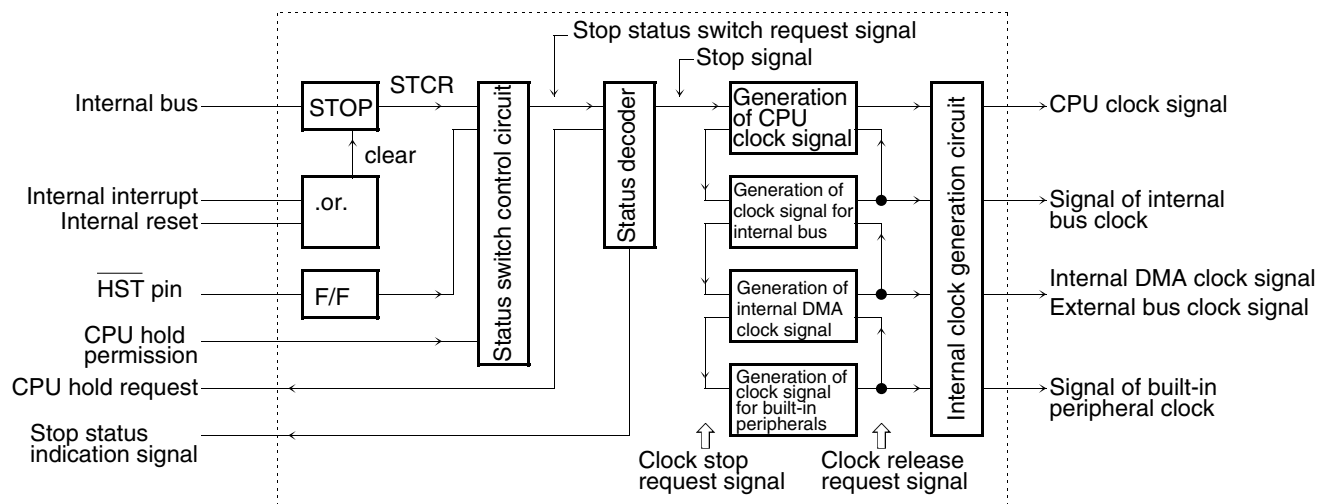
3.12.1 Stop Status

Using the block diagram of the stop control circuit shown in Figure 3.12-1 "Block Diagram of Stop Control Circuit", this section explains switching to the stop status and returning from the stop status.

■ Block Diagram of the Stop Control Circuit

Figure 3.12-1 "Block Diagram of Stop Control Circuit" shows a block diagram of the stop control circuit.

Figure 3.12-1 Block Diagram of Stop Control Circuit



■ How to Switch to Stop Status

○ Using an instruction

Set bit 7 in the STCR register to "1".

When a stop request is issued and the CPU stops using the internal bus, the clocks are stopped in the following order:

CPU clock, internal bus clock, internal DMA clock, built-in peripheral clock.

The oscillation circuit is also stopped when the built-in peripheral clock is stopped.

Note:

Use the routines below to set the stop status using an instruction.

- Before setting the STCR register to "1", set the same combination of values in the CCK1 and CCK0 bits and the PCK1 and PCK0 bits in the GCR register and set the same gear ratio for the CPU system and peripheral system clocks.
- Do not set the stop status while bit CHC in the GCR register is cleared (PLL operation). If

the stop status is to be set under these conditions, set the CHC bit (divide-by-two frequency system is selected) for clock switching.

- At least six continuous NOP instructions are necessary immediately after setting the STCR bit.

Setting method: For the maximum gear speed

```

loop      LDI:20    #GCR, R0
          LDI:8     #00000011b, R1      ; CHC=1, CPU=Peripheral gear ratio
          STB       R1, @R0             ; DBLON=0

          BTSTH     #0010b, @R0         ;
          BNE       loop                ; Wait until DBLAK is cleared.

          LDI:20    #STCR, R0
          LDI:8     #10010000b, R1      ; STOP=1
          STB       R1, @R0
          NOP
          NOP
          NOP
          NOP
          NOP
          NOP
          NOP

```

○ Using the HST pin

Apply an L-level signal to the $\overline{\text{HST}}$ pin.

Then wait until the CPU stops using the internal bus before stopping the clocks in the following order:

CPU clock, internal bus clock, internal DMA clock, built-in peripheral clock

The oscillation circuit is also stopped when the built-in peripheral clock is stopped.

Note:

The stop status is not set by setting the $\overline{\text{HST}}$ pin level to L during reset ($\overline{\text{RST}}=\text{L}$). The stop status is also not set when reset is released by setting the $\overline{\text{HST}}$ pin level to L. The stop status can be set by setting the $\overline{\text{HST}}$ pin level first to H and then to L. If the power is turned on with the $\overline{\text{HST}}$ pin level set to L, the stop status is also not set after power-on reset is released.

■ How to Restart Operation in Stop Status

Operation can be restarted with an interrupt or reset.

○ Using an interrupt

Operation can be resumed with a peripheral interrupt while the interrupt permission bit for the peripheral function is valid.

Proceed as follows:

Generate an interrupt, restart operation of the oscillation circuit, and wait until oscillation has

stabilized. Then restart to supply the built-in peripheral clock signal, the internal DMA clock, the internal bus clock, and the internal CPU clock, in this order.

Start program execution after oscillation has stabilized, as explained below.

- When the level of the generated interrupt is permitted according to the I flag in the ILM register
 - Start with the interrupt handling routine by fetching the interrupt vector after saving the register contents.
- When the level of the generated interrupt is prohibited according to the I flag in the ILM register
 - Restart execution with the next-following instruction after the instruction that was responsible for the stop status.

○ Using the RST pin

Proceed as follows:

Apply an L-level signal to the $\overline{\text{RST}}$ pin, generate an internal reset signal, restart operation of the oscillation circuit, and wait until oscillation has stabilized. Then restart to supply the built-in peripheral clock signal, the internal DMA clock signal, the internal bus clock signal, and the internal CPU clock signal, in this order, before fetching the reset vector and restarting instruction execution with the reset entry address.

Note:

- If an interrupt request has been issued from the peripheral system, the stop status is not set, but write operations are ignored.
- Excepting power-on reset, no internal clock signal is supplied while waiting for oscillation stabilization. In case of power-on reset, all internal clock signals are supplied for initializing the internal status.
- If the CPU switches to the stop status in a C-bus RAM program, use a reset instead of an interrupt to return from the stop status.

○ Using the HST pin

Proceed as follows:

Apply the H-level signal to the $\overline{\text{HST}}$ pin, generate the internal reset signal, restart operation of the oscillation circuit, and wait until oscillation has stabilized. Then restart to supply the built-in peripheral clock signal, the internal DMA clock signal, the internal bus clock signal, and the internal CPU clock signal, in this order, before fetching the reset vector and restarting instruction execution with the reset entry address.

Note:

- If the $\overline{\text{HST}}$ pin level is set to "L" during stop status, the stop status is released immediately. After waiting the specified time for oscillation stabilization, the system status is again switched to stop status if the $\overline{\text{HST}}$ pin level is "L".

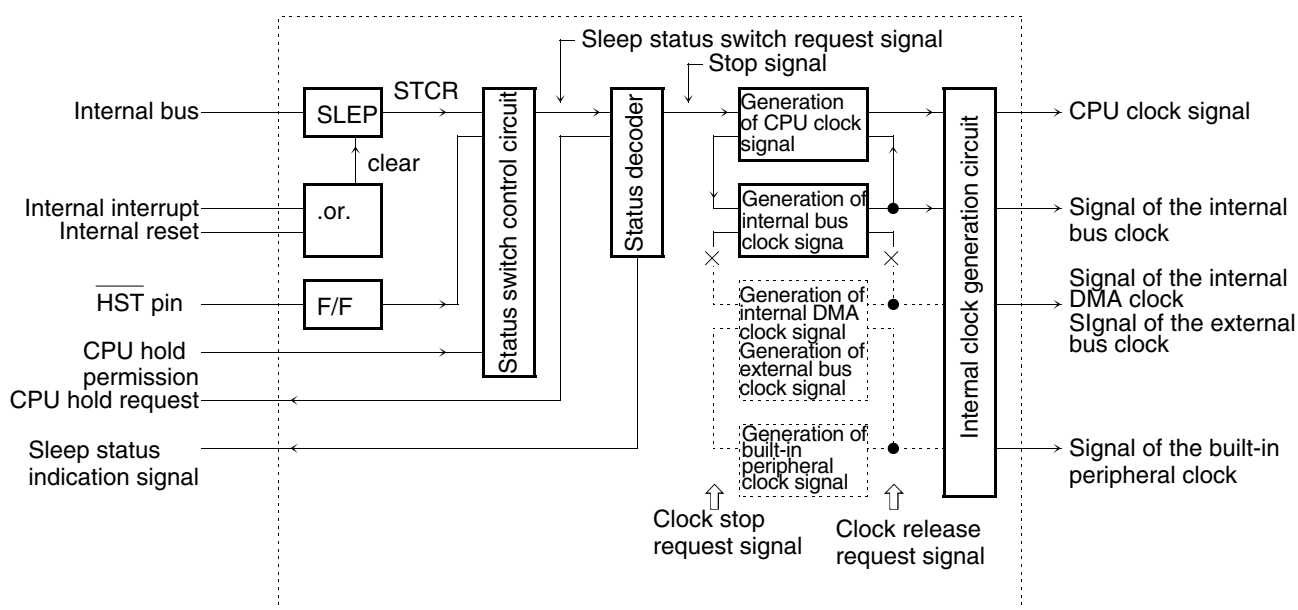
3.12.2 Sleep Status

Using the block diagram of the sleep control circuit shown in Figure 3.12-2 "Block Diagram of the Sleep Control Circuit", this section explains switching to the sleep status and returning from the sleep status.

■ Block Diagram of the Sleep Control Circuit

Figure 3.12-2 "Block Diagram of the Sleep Control Circuit" shows a block diagram of the sleep control circuit.

Figure 3.12-2 Block Diagram of the Sleep Control Circuit



■ How to Switch to Sleep Status

Set bit 7 to "0" and bit 6 to "1" in the STCR register.

When a sleep request is issued and the CPU stops using the internal bus, the clocks are stopped in the following order:

CPU clock, internal bus clock

Note:

Use the routines below to set the stop status.

- Before writing data to the STCR register, set the same combination of values in the CCK1 and CCK0 bits and the PCK1 and PCK0 bits in the GCR register and set the same gear ratio for the CPU system and peripheral system clocks.
- The sleep status can be set regardless of whether the GCR CHC bit is set or cleared.
- At least six continuous NOP instructions are necessary immediately after storing data in the STCR register.

Setting method: For maximum gear speed

```

LDI:20    #GCR, R0
LDI:8     #00000011b, R1      ; CHC=1, CPU=Peripheral gear ratio
STB       R1, @R0             ; When DBLON=0
LDI:20    #STCR, R0
LDI:8     #01010000b, R1      ; SLEP=1
STB       R1, @R0
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

```

■ How to Restart Operation in Sleep Status

System operation can be restarted with an interrupt or reset.

○ Using an interrupt

System operation is restarted with a peripheral interrupt while the interrupt permission bit for the peripheral function is valid.

Proceed as follows:

Generate an interrupt and restart to supply the internal bus clock signal and internal CPU clock signal, in this order.

Execute the program after the clock signals are supplied again, as explained below.

- When the level of the generated interrupt is permitted according to the I flag in the ILM register
 - Restart execution with the next-following instruction after the instruction that was responsible for the sleep status.
- When the level of the generated interrupt is prohibited according to the I flag in the ILM register
 - Restart execution with the next-following instruction after the instruction that was responsible for the sleep status.

○ Restart with a reset

Proceed as follows:

Issue an internal reset request and restart to supply the internal bus clock signal and internal CPU clock signal, in this order, then fetch the reset vector and restart instruction execution at the reset entry address.

○ Using the HST pin

Proceed as follows:

Apply the L-level signal to the $\overline{\text{HST}}$ pin, switch to the hardware standby status, apply the H-level signal to the $\overline{\text{HST}}$ pin, generate the internal reset signal, restart operation of the oscillation circuit, and wait until oscillation has stabilized. Then restart the supply of the signals for the

built-in peripheral clock, the internal DMA clock, the internal bus clock, and the internal CPU clock, in this order, before fetching the reset vector and restarting instruction execution at the reset entry address.

Note:

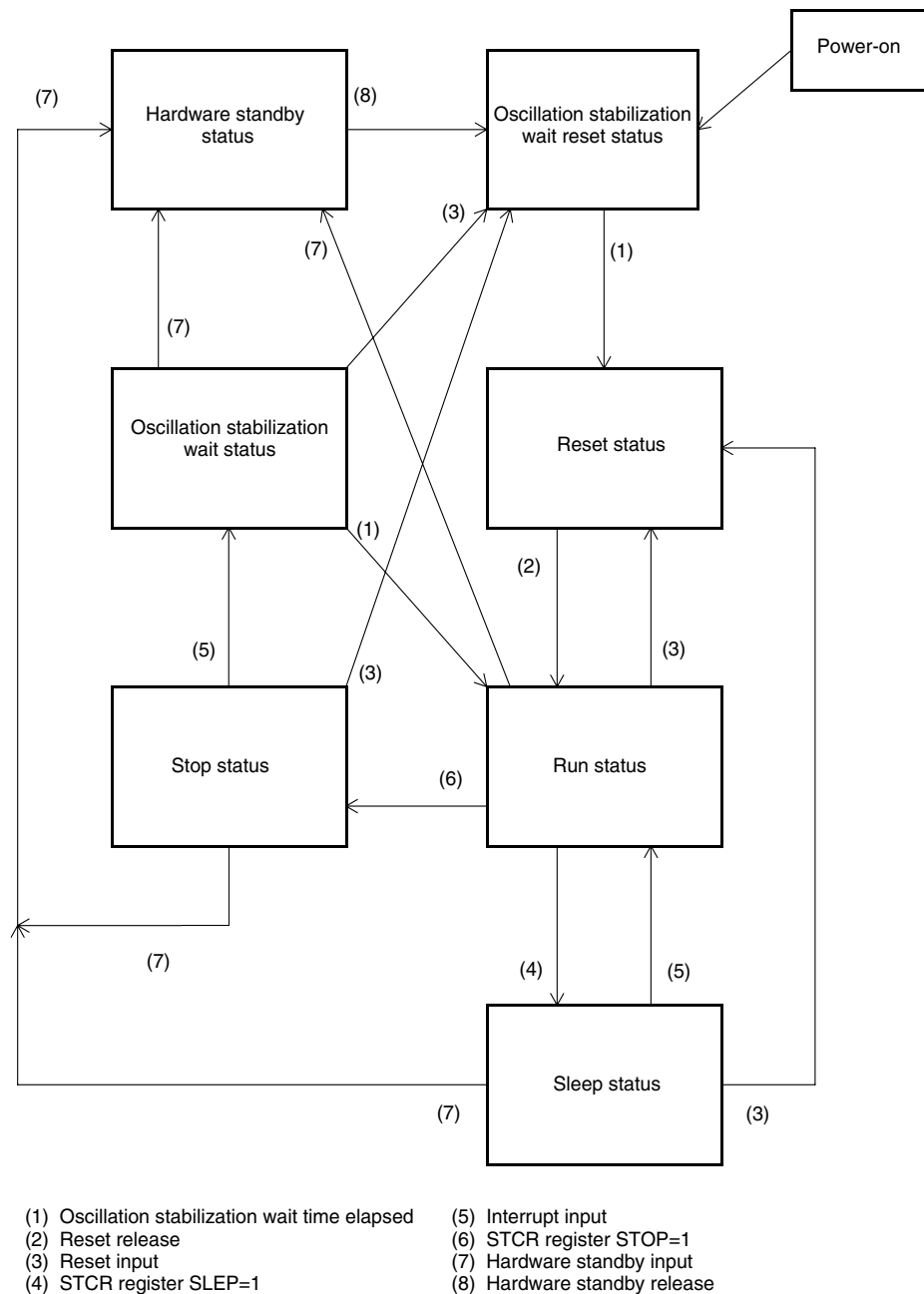
- The DMA transfer operation cannot be used in sleep mode. Be sure to disable the DMA transfer operation before switching to sleep mode.
- If an interrupt request has been issued from the peripheral system, the sleep status is not set.

3.12.3 Status Switch in Standby Mode

Figure 3.12-3 "Status Switch in Standby Mode" illustrates the status switch.

■ Status Switch in Standby Mode

Figure 3.12-3 Status Switch in Standby Mode

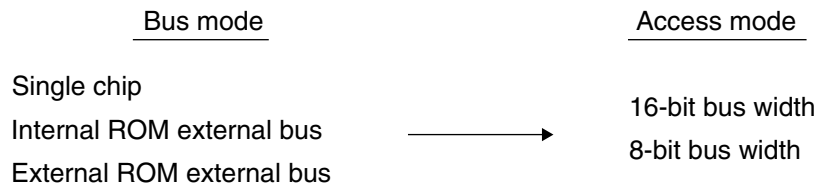


3.13 Operation Mode

The two types of operation mode are bus mode and access mode. Mode pins (MD2, 1, 0) and mode register (MODR) control the operation mode.

■ Operation Mode

The two types of operation mode are bus mode and access mode.



○ Bus mode

In this mode, operation of the internal ROM and of the external access function are controlled. Mode setting pins (MD2, 1, 0) and bits M1 and M0 in a mode register (MODR) are used to specify the operation.

○ Access mode

In this mode, the external data bus width is controlled. Mode setting pins (MD2, 1, 0) and bits BW1 and BW0 in area mode registers (AMD0/AMD1/AMD32/AMD4/AMD5) are used to specify the operation.

■ Mode Pin

The MD2, MD1, and MD0 pins are used to specify the operation as listed in Table 3.13-1 "Mode Pin and Mode".

Table 3.13-1 Mode Pin and Mode

Mode pin			Mode name	Reset vector access area	External data bus width	Remarks
MD2	MD1	MD0				
0	0	0	External vector mode 0	External	8 bit	External ROM external bus mode
0	0	1	External vector mode 1	External	16 bit	External ROM external bus mode
0	1	0	-	-	-	Cannot be set.
0	1	1	Internal vector mode	Internal	(Mode register)	Single chip mode
1	-	-	-	-	-	Must not be used.

■ Mode data

Data that the CPU writes to address "0000 07FF_H" after reset is called mode data.

A mode register (MODR) is located at address "0000 07FF_H". The mode set in this register is valid.

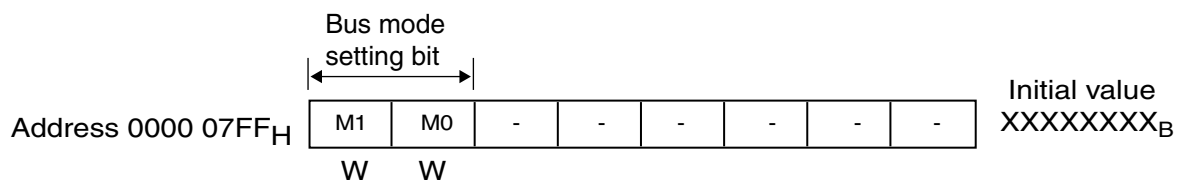
Data can be written to the MODR register only once after reset.

The value in this register is valid immediately after it was written.

■ Mode register (MODR)

Figure 3.13-1 "Configuration of the Mode Register" shows the configuration of the MODR register.

Figure 3.13-1 Configuration of the Mode Register



○ Bus mode setting bit (M1, M0)

These bits are used to specify the bus mode after writing data to the MODR register.

Table 3.13-2 "Bus Mode Setting Bits and Functions" lists the relationship between bit value combinations and functions.

Table 3.13-2 Bus Mode Setting Bits and Functions

M1	M0	Function	Remarks
0	0	Single chip mode	
0	1	Internal ROM external bus mode	
1	0	External ROM external bus mode	
1	1	-	Cannot be set.

Note:

For a model without internal ROM, use only the "1, 0" setting in the above table.

○ Other bits(*)

Always set these bits to "0".

■ Notes on Writing Data to the MODR Register

Always specify AMD0 to AMD5 to determine the bus widths of each chip select (CS) area before writing data to the MODR register.

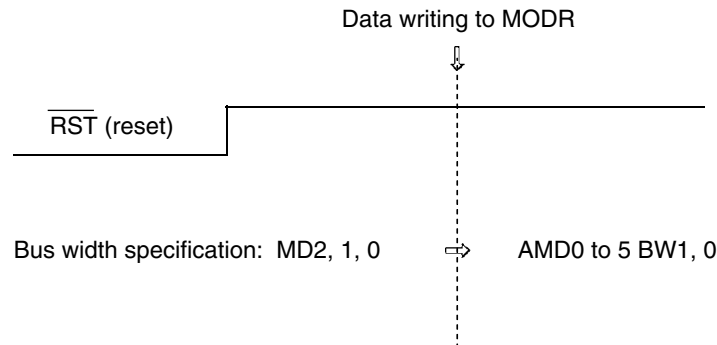
There is no bit in the MODR register to determine the bus width.

For the bus width, values of mode pins MD2 to MD0 are valid before writing data to the MODR register, and BW1 and BW0 values in the AMD0 to AMD5 are valid after writing to the MODR

register.

For example, the external reset vector is generally executed in area 0 (area with CS0X active) with the bus width determined with MD2 to MD0. If a width of 16 bits is set with MD2 to MD0 and data is written to the MODR register without setting a value in the AMD0, area 0 switches to 8-bit bus mode and a bus operation malfunction occurs because the initial bus width in the AMD0 is 8 bits.

This can be prevented by always setting values in AMD0 to AMD5 before writing data to the MODR register.



CHAPTER 4 BUS INTERFACE

This chapter outlines the external bus interface, explains the register configuration, register functions, bus operations, and bus timing, and provides bus operation program examples.

Note that the MB91F127 and 128 do not have pins for DRAM control signals and cannot be used as a DRAM interface.

- 4.1 "Outline of Bus Interface"
- 4.2 "Bus Interface Block Diagram"
- 4.3 "Bus Interface Registers"
- 4.4 "Bus Operations"
- 4.5 "Bus Timings"
- 4.6 "Internal Clock Frequency Multiplier Operations (Clock Doubler)"
- 4.7 "Program Example of External Bus Operations"

4.1 Outline of Bus Interface

The bus interface controls the interfaces to the external memory and external I-O devices.

■ Features

- Address output of 25 bits (32MB)
- Six independent banks are supported via the chip select function
 - Areas in the logical address space can be set freely in units of 64 Kbytes or more using area selection registers (ASR1 to 5) and area mask registers (AMR1 to 5).
 - A total of six 32 Mbytes areas can be set with the address and chip select pins.
- For each chip select area, a bus width of 16 bits or 8 bits can be set.
- A programmable automatic memory wait (up to seven cycles) can be included.
- Unused addresses and data pins can be used as I/O ports.
- Little endian mode support

■ Chip Select Area

6 types of chip select area can be used.

Each area can be allocated freely in a 4 Gbytes space in units of 64 Kbytes or more using area selection registers (ASR1 to 5) and area mask registers (AMR1 to 5).

If external bus access is executed for an area specified with these registers, the corresponding chip select signal ($\overline{CS0}$ to $\overline{CS5}$) is set to "L" (active).

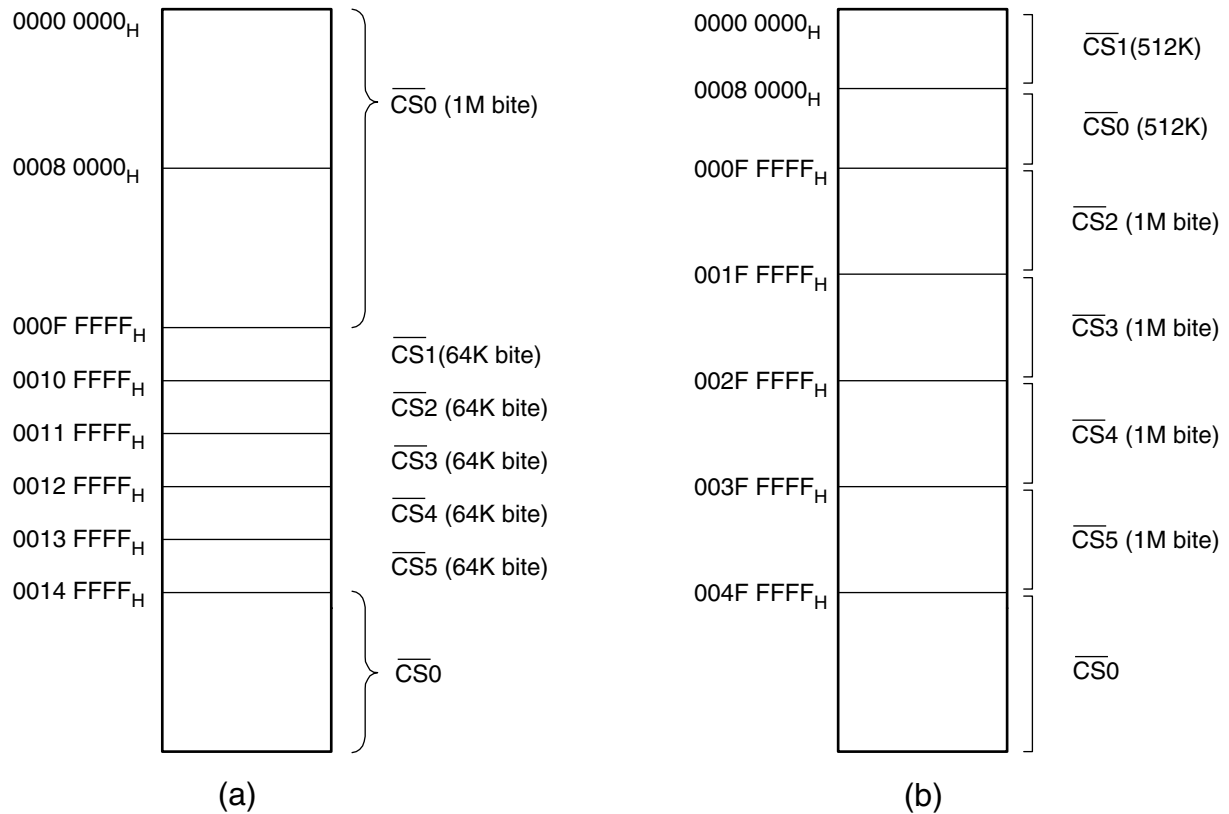
$\overline{CS1}$ to $\overline{CS5}$ are set to "H" (inactive) upon reset.

Note:

Area 0 is allocated in a free area other than that specified with registers ASR1 to ASR5.

An external area in an address range other than 0001 0000_H to 0005 FFFF_H becomes area 0 upon reset.

Figure 4.1-1 "Sample Setting for Chip Select Area" (a) shows the location of areas 1 to 5 in 00100000_H to 0014FFFF_H in 64 Kbytes units. Figure 4.1-1 "Sample Setting for Chip Select Area" (b) shows the location of area 1 in 00000000_H to 0007FFFF_H in 512 Kbytes units and areas 2 to 5 in 00100000_H to 004FFFFF_H in 1 Mbyte units.

Figure 4.1-1 Sample Setting for Chip Select Area

■ Bus interface

The bus interface can have one of the following two types:

- General bus interface
- Address/data time-division I/O interface

These interfaces can be used only in areas determined in advance.

Table 4.1-1 "Relationship between Chip Select Area and Bus Interface" shows the relationship between chip select areas and usable interface functions.

The interface to be used can be selected using an area mode register (AMD).

If no interface is specified, the general bus interface is selected.

Table 4.1-1 Relationship between Chip Select Area and Bus Interface

Area	Bus interface that can be selected		Remarks
	General bus	Time-sharing	
0	O	-	At reset
1	O	O	
2	O	-	
3	O	-	
4	O	-	
5	O	-	

■ Time-division I/O specification

For area 1, addresses and data are input and output on a time-division basis on a bus with the width specified by AMD1. An address latch pulse is output to the ALE pin.

■ Bus size specification

Bus widths can be specified arbitrarily for each area using registers.

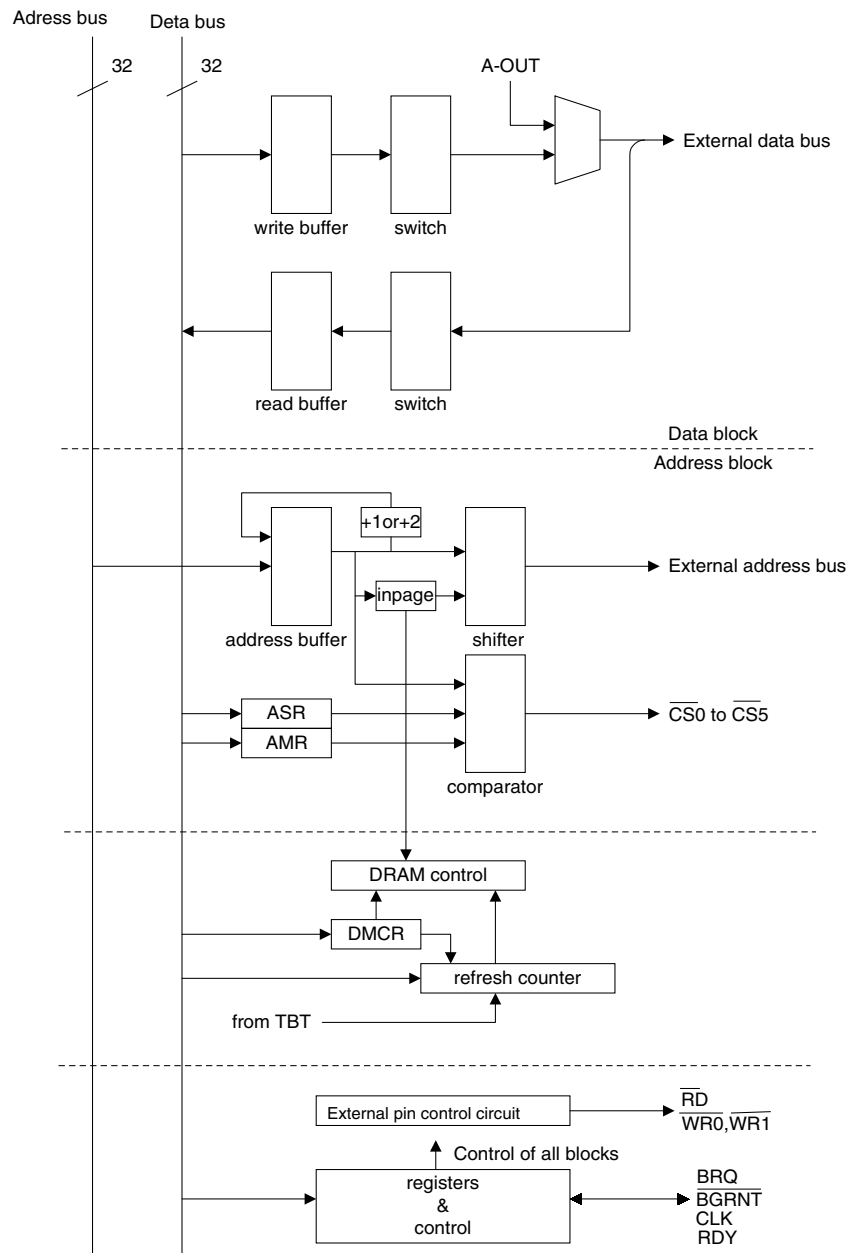
Area 0 obtains at reset the bus width specified by the MD2, MD1, and MD0 pins. The bus size for this area can then be specified according to the AMD0 register value after writing data to the mode register (MODR).

4.2 Bus Interface Block Diagram

Figure 4.2-1 "Bus Interface Block Diagram" shows a block diagram of the bus interface.

■ Block diagram of the Bus Interface

Figure 4.2-1 Bus Interface Block Diagram

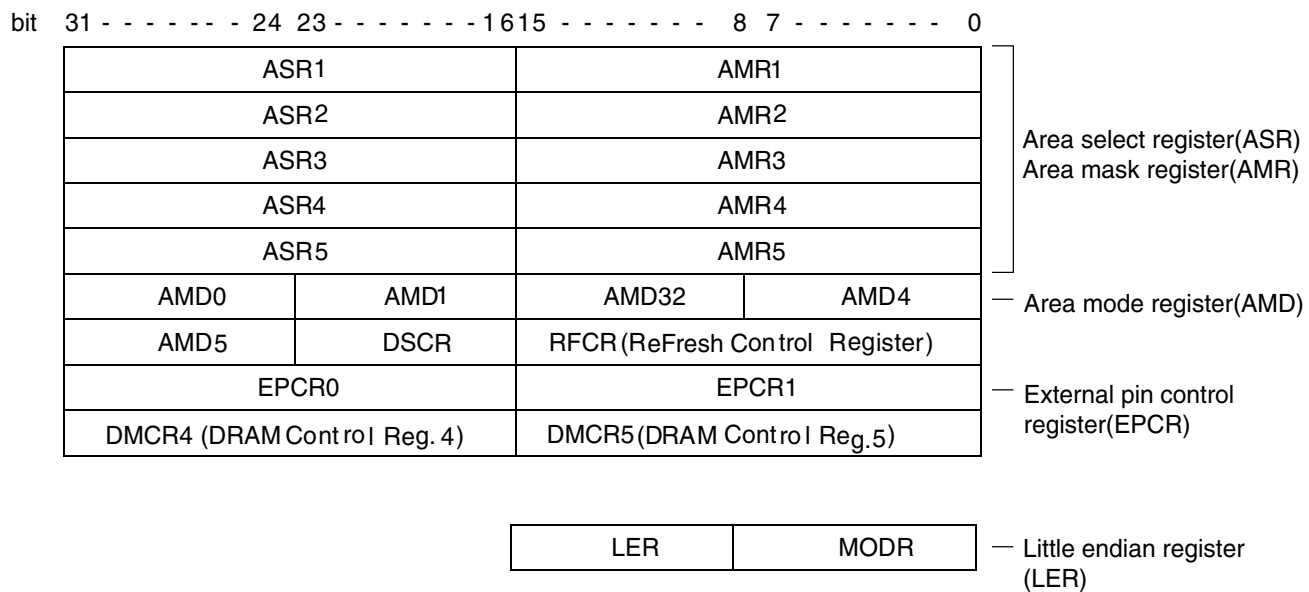


4.3 Bus Interface Registers

Figure 4.3-1 "Bus Interface Registers" shows the bus interface registers.

■ Bus Interface Registers

Figure 4.3-1 Bus Interface Registers



For the mode register (MODR), see Section 3.13 "Memory Access Mode".

4.3.1 Area Selection Register (ASR) and Area Mask Register (AMR)

Area selection registers (ASR1 to 7) and area mask registers (AMR1 to 7) are used to specify address ranges for chip select areas 1 to 7.

■ Area Selection Register (ASR) and Area Mask Register (AMR)

The figure below shows the register configuration.

○ Area selection registers (ASR1 to 7)

Figure 4.3-2 Configuration of ASR

ASR1 bit	15	14	13	12	...	2	1	0	Initial value
Address 0000 060C _H	A31	A30	A29	A18	A17	A16	0001 _H
	W	W	W			W	W	W	
ASR2 bit	15	14	13	12	...	2	1	0	
Address 0000 0610 _H	A31	A30	A29	A18	A17	A16	0002 _H
	W	W	W			W	W	W	
ASR3 bit	15	14	13	12	...	2	1	0	
Address 0000 0614 _H	A31	A30	A29	A18	A17	A16	0003 _H
	W	W	W			W	W	W	
ASR4 bit	15	14	13	12	...	2	1	0	
Address 0000 0618 _H	A31	A30	A29	A18	A17	A16	0004 _H
	W	W	W			W	W	W	
ASR5 bit	15	14	13	12	...	2	1	0	
Address 0000 061C _H	A31	A30	A29	A18	A17	A16	0005 _H
	W	W	W			W	W	W	

○ Area mask registers (AMR1 to 5)

Figure 4.3-3 Configuration of AMR

AMR1 bit	15	14	13	12	...	2	1	0	Initial value
Address 0000 060E _H	A31	A30	A29	A18	A17	A16	0000 _H
	W	W	W			W	W	W	
AMR2 bit	15	14	13	12	...	2	1	0	
Address 0000 0612 _H	A31	A30	A29	A18	A17	A16	0000 _H
	W	W	W			W	W	W	
AMR3 bit	15	14	13	12	...	2	1	0	
Address 0000 0616 _H	A31	A30	A29	A18	A17	A16	0000 _H
	W	W	W			W	W	W	
AMR4 bit	15	14	13	12	...	2	1	0	
Address 0000 061A _H	A31	A30	A29	A18	A17	A16	0000 _H
	W	W	W			W	W	W	
AMR5 bit	15	14	13	12	...	2	1	0	
Address 0000 061E _H	A31	A30	A29	A18	A17	A16	0000 _H
	W	W	W			W	W	W	

Area selection registers (ASR1 to 5) and area mask registers (AMR1 to 5) indicate address ranges for chip select areas 1 to 5.

ASR1 to 5 each indicate the high-order 16 bits (A31 to 16) of an address, and AMR1 to 5 each indicate the mask for corresponding address bits. Value "0" in each bit indicates "care", and value 1 indicates "don't care".

When "care" is set and the corresponding ASR bit is "0", "0" is indicated for the respective address area; similarly, when the ASR bit is "1", "1" is indicated.

When "don't care" is set, "0" or "1" is indicated for the respective address area regardless of the corresponding ASR bit.

Examples of ASR and AMR combinations for specifying each chip select area are shown below.

(Example 1)

ASR1 = 00000000 00000011_B

AMR1 = 00000000 00000000_B

When 00000000 00000011_B is set in ASR1 and 00000000 00000000_B is set in AMR1, the address space for area 1 becomes 64 kilobytes, as shown below. This is because the AMR1 bits corresponding to the ASR1 bits that were set to "1" are "0".

00000000 00000011 00000000 00000000_B (00030000_H)

}

00000000 00000011 11111111 11111111_B (0003FFFF_H)

(Example 2)

ASR2 = 00001111 11111111_B

AMR2 = 00000000 00000011_B

When 00001111 11111111_B is set in ASR2 and 00000000 00000011_B is set in AMR2, the address space for area 2 becomes 256 kilobytes, as shown below. This is because "care" is valid for the values of the ASR2 bits corresponding to the AMR2 bits set to "0", while "do not care" is valid for the values in the ASR2 bits corresponding to the AMR2 bits set to "1."

00001111 11111100 00000000 00000000_B (0FFC0000_H)

}

00001111 11111111 11111111 11111111_B (0FFFFFFF_H)

Areas 1 to 5 can be allocated arbitrarily in a 4 Gbytes space in units of 64 Kbytes or more using ASR1 to ASR5. If bus access is performed for these areas, the output level of the corresponding chip select pins ($\overline{CS0}$ to $\overline{CS5}$) becomes "L".

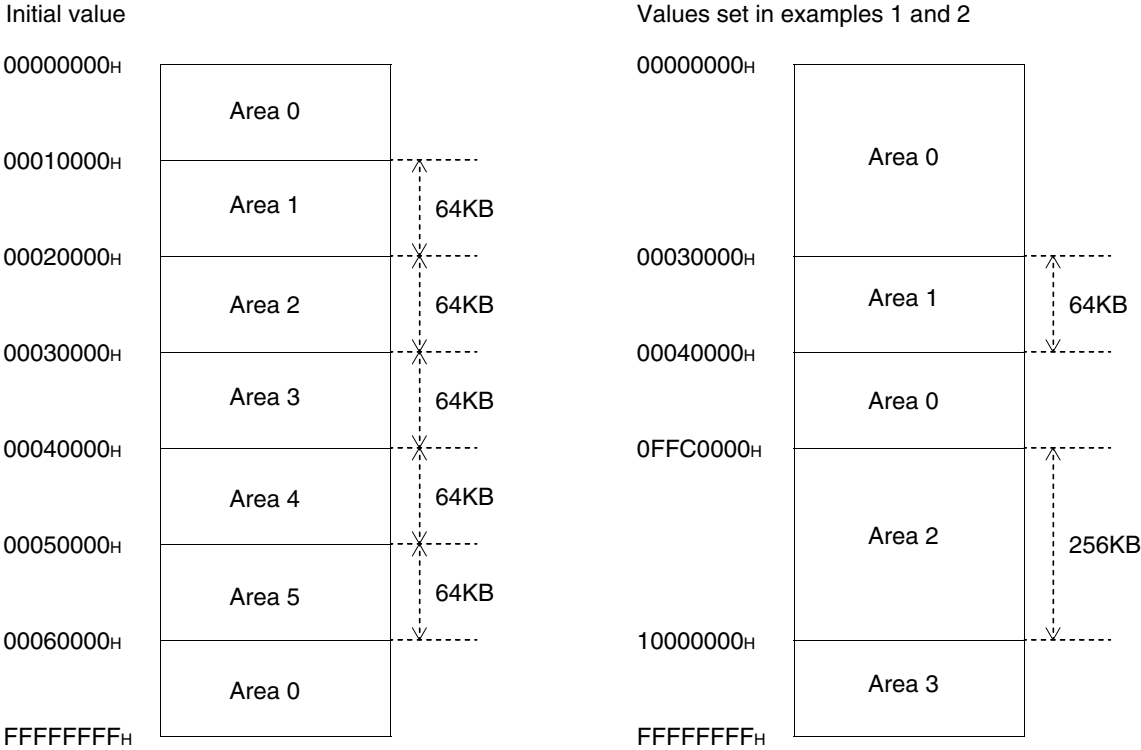
Area 0 is allocated in a space other than the spaces set with ASR1 to 5 and AMR1 to 5. An area in an address range other than 0001000_H to 0005FFFF_H is allocated as area 0 according to the initial values of ASR1 to 5 and AMR1 to 5 upon reset.

Note:

Chip select areas should not overlap.

Figure 4.3-4 "Sample Map for a Chip Select Area Setting" shows a map of chip select areas set in 64 Kbytes units together with the initial values upon reset and a map of areas set as shown in examples 1 and 2 above.

Figure 4.3-4 Sample Map for a Chip Select Area Setting



4.3.2 Area Mode Register 0 (AMD0)

Area mode register 0 (AMD0) indicates the operation mode of chip select area 0 (area other than areas specified with ASR1 to 5 and AMR1 to 5). Area 0 is selected upon reset.

■ AMD0 Configuration

Figure 4.3-5 "Register Configuration of AMD0" shows the register configuration of AMD0.

Figure 4.3-5 Register Configuration of AMD0

	bit	7	6	5	4	3	2	1	0	Initial value
Address	00000620 _H	-	-	-	BW1	BW0	WTC2	WTC1	WTC0	---00111 _B
		W	W	W	W	W	W	W	W	

■ Bit Functions of AMD0

The functions of bits of AMD0 are explained below.

[Bits 4, 3] BW1, 0 (Bus Width bits)

BW1 and BW0 indicate the bus width for area 0.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	Cannot be set.
1	1	Cannot be set.

Note

Both BW1 and BW0 have an initial value of "0". However, the pin levels of MD1 and MD0 instead of the register values are read until the MODR is written to during reading.

[Bits 2 to 0] WTC2 to 0 (Wait Cycle bits)

WTC2 to 0 indicate how often wait is inserted automatically for the general bus interface.

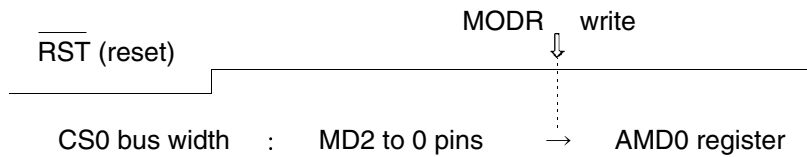
WTC2	WTC1	WTC0	Number of wait cycles
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The AMD0 bits WTC2 to WTC0 are set to "111" upon reset. A seven-cycle wait is inserted automatically in bus access immediately after reset release.

Note:

Always set the same bus width as that set with the MD2 to 0 pins in the AMD0 BW1 and 0 bits before writing data to the mode register (MODR).

The bus width of area 0 is set with the MD2 to 0 pins upon reset. A bus width set with the AMD0 is valid after writing data to the MODR register.



If a bus width of 16 bits is set for area 0 with the pins MD2, 1, and 0, and data is written to the MODR register without setting a bus width in the AMD0 register, a malfunction occurs. This is because the initial value of AMD0 for BW1 and 0 is "00", and the system consequently switches to a bus width of 8-bits.

4.3.3 Area Mode Register 1 (AMD1)

Area mode register 1 (AMD1) indicates the operation mode of chip select area 1 (area specified with ASR1 and AMR1 registers).

In area 1, a time-division I/O interface for address/data input-output can be specified. The time-division interface is used to output an address and input and output data to the data bus. Only the 8-bit bus width and the 16-bit bus width are supported as described below.

- 8-bit bus width: A7 to A0 are multiplexed on D31 to D24
- 16-bit bus width: A15 to A0 are multiplexed on D31 to D16

■ AMD1 Configuration

Figure 4.3-6 "Register Configuration of AMD1" shows the register configuration of AMD1.

Figure 4.3-6 Register Configuration of AMD1

	bit	7	6	5	4	3	2	1	0	Initial value
Address	00000621 _H	MPX	-	-	BW1	BW0	WTC2	WTC1	WTC0	0--00000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Functions of AMD1

The functions of bits of AMD1 are explained below.

[Bit 7] MPX (MultiPleX bit)

The MPX bit controls the time-sharing I-O interface for address and data input-output.

MPX	Function
0	Normal bus interface
1	Time-division I/O interface

[Bits 4, 3] BW1, 0 (Bus Width bit)

BW1 and 0 indicate the bus width for area 1.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	Cannot be set.
1	1	Reserved

[Bits 2 to 0] WTC2 to 0 (Wait Cycle bit)

These bits indicate the number of wait cycles to be inserted automatically for the general bus interface. The bits can be used in the same way as the AMD0 bit WTC2, 1, and 0, and are initialized to "000" to indicate wait count "0" upon reset.

4.3.4 Area Mode Register 32 (AMD32)

Area mode register 32 (AMD32) controls the operation mode of chip select area 2 (area specified with ASR2 and AMR2 registers) and that of chip select area 3 (area specified with ASR3 and AMR3 registers).

Only general bus access is valid for these chip select areas, and time-division I/O interfaces cannot be used.

Bits BW1 and 0 indicate the same bus width for these areas. Different automatic wait counts can be specified.

■ AMD32 Configuration

Figure 4.3-7 "Register Configuration of AMD32" shows the register configuration of AMD32.

Figure 4.3-7 Register Configuration of AMD32

	bit	7	6	5	4	3	2	1	0	Initial value
Address	00000622 _H	BW1	BW0	WT32	WT31	WT30	WT22	WT21	WT20	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Functions of AMD32

The functions of bits of AMD32 are explained below.

[Bits 7, 6] BW1, 0 (Bus Width bit)

BW1 and 0 indicate the bus width for areas 2 and 3.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	Cannot be set.
1	1	Reserved

[Bits 5 to 3] WT32 to 30 (Wait Cycle bit)

These bits indicate the number of wait cycles to be inserted automatically in memory access to area 3.

These bits can be used in the same way as the AMD0 bits WTC2, 1, and 0.

The WT32 to 30 bits are initialized to "000" to indicate wait count "0" upon reset.

[Bits 2 to 0] WT22 to 20 (Wait Cycle bit)

Bits WT22 to 20 specify the value of the wait cycle inserted automatically at memory accesses to area 2.

These bits can be operated in the same way as the AMD0 bits WTC2, 1, and 0.

The bits WT22 to 20 are initialized to "000" to indicate wait count "0" upon reset.

4.3.5 Area Mode Register 4 (AMD4)

Area mode register 4 (AMD4) indicates the operation mode of chip select area 4 (area specified with ASR4 and AMR4 registers).

The DRAM interface can be used with area 4.

■ AMD4 Configuration

Figure 4.3-8 "Register Configuration of AMD4" shows the register configuration of AMD4.

Figure 4.3-8 Register Configuration of AMD4

		bit	7	6	5	4	3	2	1	0	Initial value
Address	00000623 _H		DRME	-	-	BW1	BW0	WTC2	WTC1	WTC0	0--00000 _B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Functions of AMD4

The functions of bits of AMD4 are explained below.

[Bit 7] DRME (DRaM Enable bit)

This bit selects the general bus interface or DRAM interface for area 4.

DRME	Operation
0	Normal bus interface
1	Must not be set.

[Bits 4, 3] BW1, 0 (Bus Width bit)

BW1 and 0 indicate the bus width for area 4. These bits have the same function as the BW bits in other AMD registers.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	Cannot be set.
1	1	Reserved

[Bits 2 to 0] WTC2 to 0 (Wait Cycle bit)

These bits indicate the number of wait cycles to be inserted automatically for memory access to area 4.

These bits have the same function as the WTC bits in other AMD registers, and they are initialized to "000" at reset to indicate an insert wait cycle count of "0".

4.3.6 Area Mode Register 5 (AMD5)

Area mode register 5 (AMD5) indicates the operation mode of chip select area 5 (area specified with ASR5 and AMR5 registers).

■ AMD5 Configuration

Figure 4.3-9 "Register Configuration of AMD5" shows the register configuration of AMD5.

Figure 4.3-9 Register Configuration of AMD5

	bit	7	6	5	4	3	2	1	0	Initial value
Address	00000624 _H	DRME	-	-	BW1	BW0	WTC2	WTC1	WTC0	0--00000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Functions of AMD5

The functions of bits of AMD5 are explained below.

[Bit 7] DRME (DRaM Enable bit)

This bit selects the general bus interface for area 5.

DRME	Operation
0	Normal bus interface
1	Must not be set.

[Bits 4, 3] BW1, 0 (Bus Width bit)

BW1 and 0 indicate the bus width for area 5 and have the same function as the BW bits in other AMD registers.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	Cannot be set.
1	1	Reserved

[Bits 2 to 0] WTC2 to 0 (Wait Cycle bit)

These bits indicate the number of wait cycles to be inserted automatically for memory access to area 5.

These bits have the same function as the WTC bits in other AMD registers and are initialized to "000" to indicate wait count "0" upon reset.

4.3.7 External pin control register 0 (EPCR0)

External pin control register 0 (EPCR0) controls the output of each signal. When output is allowed, the specified timing is output in bus mode. When input is valid, external input signals are accepted. When output is inhibited or input is invalid, the corresponding pin can be used as an I/O port.

■ EPCR0 Configuration

Figure 4.3-10 "Register Configuration of EPCR0" shows the register configuration of EPCR0.

Figure 4.3-10 Register Configuration of EPCR0

bit		15	14	13	12	11	10	9	8	Initial value
Address	00000628 _H	-	-	ALEE	-	WRE	RDXE	RDYE	BRE	--1-1100 _B
		W	W	W	W	W	W	W	W	
bit		7	6	5	4	3	2	1	0	
		-	CKE	COE5	COE4	COE3	COE2	COE1	COE0	-1111111 _B
		W	W	W	W	W	W	W	W	

■ Bit Functions of EPCR0

The functions of bits of EPCR0 are explained below.

[Bit 13] ALEE (ALE output Enable bit)

This bit indicates a selection on whether to allow ALE output.

Output is allowed while this bit is set at "1".

WRE	Operation
0	Output is prohibited.
1	Output is allowed (initial value).

ALE is used as the address strobe of an external peripheral during use of area 1 in time-division mode.

[Bit 11] WRE (WRite pulse output Enable bit)

This bit indicates whether write pulse $\overline{WR0}$ to $\overline{WR1}$ is to be output.

Output is allowed upon reset.

WRE	Operation
0	Output is prohibited.
1	Output is allowed (initial value).

Because the I/O port of the $\overline{WR0}$ to $\overline{WR1}$ pin is not controlled by this bit in MB91F127/128, always set this bit to "1".

For the write pulse, an I/O port pulse set according to the bus width specified in the AMD register can be used even though this bit is set to "1" (for example, $\overline{WR1}$ is not output in 8-bit mode, but the corresponding pin can be used as an I/O port).

[Bit 10] RDXE (ReaDX pulse output Enable bit)

This bit indicates whether read pulse \overline{RD} is to be output.

Output is allowed upon reset.

WRE	Function
0	Output is prohibited (cannot be set).
1	Output is allowed (initial value).

Because this product does not control the \overline{RD} pin I/O port by using the RDXE bit in external bus mode, always set this bit to "1".

[Bit 9] RDYE (ReaDY input Enable bit)

This bit controls the RDY input as follows.

Input becomes disabled reset.

RDYE	Function
0	RDY input is disabled (initial value).
1	RDY input is enabled.

[Bit 8] BRE (Bus Request Enable bit)

This bit controls the BRQ and \overline{BGRNT} as follows.

The BRQ input becomes disabled and \overline{BGRNT} output is inhibited upon reset.

BRE	Function
0	BRQ input is disabled, and \overline{BGRNT} output is prohibited (the pin functions as an I/O port) (initial value).
1	BRQ input is enabled, and \overline{BGRNT} output is allowed.

[Bit 6] CKE (Clock output Enable bit)

This bit controls the CLK (external bus operation clock waveform) output.

CKE	Function
0	Output is prohibited.
1	Output is allowed (initial value).

This bit is initialized to "1" during reset, enabling CLK.

[Bit 5] COE5 (Chip select Output Enable 5)

This bit controls the $\overline{CS5}$ output. Output is enabled upon reset.

COE5	Function
0	Output is prohibited.
1	Output is allowed (initial value).

[Bit 4] COE4 (Chip select Output Enable 4)

This bit controls the $\overline{CS4}$ output. Output is enabled upon reset.

COE4	Function
0	Output is prohibited.
1	Output is allowed (initial value).

[Bit 3] COE3 (Chip select Output Enable 3)

This bit controls the $\overline{CS3}$ output. Output is enabled upon reset.

COE3	Function
0	Output is prohibited.
1	Output is allowed (initial value).

[Bit 2] COE2 (Chip select Output Enable 2)

This bit controls the $\overline{CS2}$ output. Output is enabled upon reset.

COE2	Function
0	Output is prohibited.
1	Output is allowed (initial value).

[Bit 1] COE1 (Chip select Output Enable 1)

This bit controls the $\overline{CS1}$ output. Output is enabled upon reset.

COE1	Function
0	Output is prohibited.
1	Output is allowed (initial value).

[Bit 0] COE0 (Chip select Output Enable 0)

This bit controls the $\overline{CS0}$ output. Output is enabled upon reset.

COE0	Function
0	Output is prohibited.
1	Output is allowed (initial value).

Because this product does not use this bit for $\overline{CS0}$ pin I/O port control, always set this bit to "1".

4.3.8 External Pin Control Register 1 (EPCR1)

External pin control register 1 (EPCR1) controls the address signal output.

EPCR1 Configuration

Figure 4.3-11 "Register Configuration of EPCR1" shows the register configuration of EPCR1.

Figure 4.3-11 Register Configuration of EPCR1

bit	15	14	13	12	11	10	9	8	Initial value
Address 0000062A _H	-	-	-	-	-	-	-	AE24	-----1 _B
	W	W	W	W	W	W	W	W	
bit	7	6	5	4	3	2	1	0	
	AE23	AE22	AE21	AE20	AE19	AE18	AE17	AE16	11111111 _B
	W	W	W	W	W	W	W	W	

Bit Functions of EPCR1

The functions of bits of EPCR1 are explained below.

[Bits 8 to 0] AE24 to AE16 (Address output Enable 24 to 16)

These bits indicate whether the corresponding address signals are to be output.

The corresponding pins can be used as I/O ports if output is inhibited.

AE24 to AE16	Function
0	Output is prohibited.
1	Output is allowed (initial value).

These bits are initialized to "1FF_H" upon reset.

4.3.9 Little Endian Register (LER)

The MB91F127/128 bus access is generally executed using the big endian for all areas, but one of the areas 1 to 5 can be handled as a little endian area according to the data of this register.

The little endian area is supported for all types of bus mode independent of whether the general, time-sharing, or DRAM interface is used. Area 0 cannot be handled as a little endian area.

■ LER Configuration

Figure 4.3-12 "Register Configuration of LER" shows the register configuration of the LER.

Figure 4.3-12 Register Configuration of LER

bit	7	6	5	4	3	2	1	0	Initial value
Address 000007FE _H	-	-	-	-	-	LE2	LE1	LE0	-----000 _B
	W	W	W	W	W	W	W	W	

■ Bit Functions of LER

The functions of bits of the LER are explained below.

[Bits 2 to 0] LE2 to 0

The combinations of these bits indicate a little endian area as listed in Table 4.3-1 "Mode Corresponding to Bit (LE2 to 0) Combinations".

Table 4.3-1 Mode Corresponding to Bit (LE2 to 0) Combinations

LE2	LE1	LE0	Mode
0	0	0	Initial value after reset No little endian area
0	0	1	Area 1 is a little endian area. Areas 0 and 2 to 5 are big endian areas.
0	1	0	Area 2 is a little endian area. Areas 0, 1, and 3 to 5 are big endian areas.
0	1	1	Area 3 is a little endian area. Areas 0 to 2, 4, and 5 are big endian areas.
1	0	0	Area 4 is a little endian area. Areas 0 to 3 and 5 are big endian areas.
1	0	1	Area 5 is a little endian area. Areas 0 to 4 are big endian areas.

Note

The little endian register (LER) can be written to only once after a reset.

4.4 Bus Operations

This section describes the following basic items of bus operations:

- Relationship between the data bus widths and the control signals
 - Big endian bus access
 - Little endian bus access
 - Comparison of external access operations
-

■ Relationship Between Data Bus Widths and Control Signals

This section describes the relationship between the data bus widths and the control signals for the following interfaces:

- Normal bus interface
- Time-division I/O bus interface

■ Big Endian Bus Access

This section describes the following items related to external access operations:

- Data Formats
- Data Bus Widths
- External Bus Access
- Sample External Device Connection

■ Little Endian Bus Access

This section describes the following items related to external access operations:

- Outline of Little Endian Order
- Data Formats
- Data Bus Widths
- Examples for External Device Connection

■ Comparison of External Access Operations of Big Endian and Little Endian Type

This section compares the external access operations of big endian and little endian type for word access, halfword access, and byte access in relationship to the bus widths.

4.4.1 Relationship Between Data Bus Widths and Control Signals

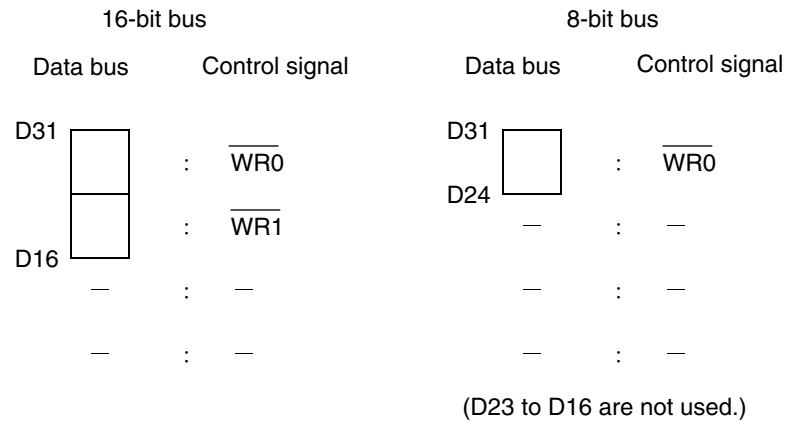
The data bus control signals ($\overline{WR0}$, $\overline{WR1}$, $\overline{CS0H}$, $\overline{CS1L}$, $\overline{CS1H}$, $\overline{DW0X}$, and $\overline{DW1X}$) are always in a one-to-one relationship with the data bus byte positions regardless of big endian or little endian access, or the data bus width.

■ Relationship Between Data Bus Widths and Control Signals

This section shows for each bus mode the byte positions and corresponding control signals of the data buses used for the specified data bus widths.

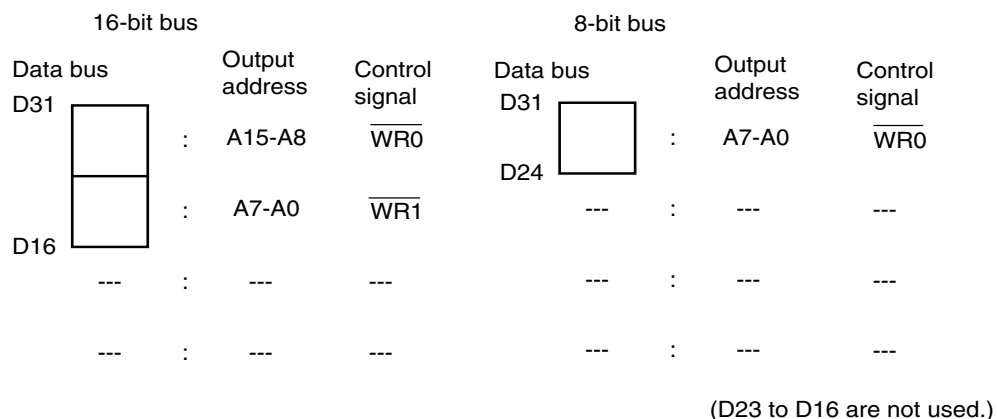
○ Data Bus Widths and Control Signals of the Normal Bus Interface

Figure 4.4-1 Data Bus Widths and Control Signals of the Normal Bus Interface



○ Data Bus Widths and Control Signals of Time-division I/O Bus Interface

Figure 4.4-2 Data Bus Widths and Control Signals of Time-division I/O Bus Interface



4.4.2 Big Endian Bus Access

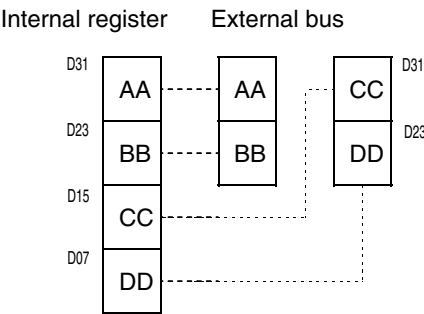
Big endian order is used for external bus access of areas for which the little endian register (LER) has not been set.
The FR series uses normal big endian order.

■ Data Formats

The figures below show the relationship between the internal registers and external data buses for each data format.

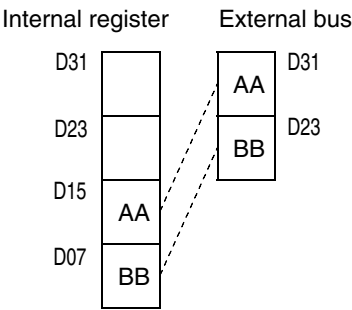
○ Word access (during execution of LD or ST instruction)

Figure 4.4-3 Relationship between Internal Registers and External Data Buses for Word Access



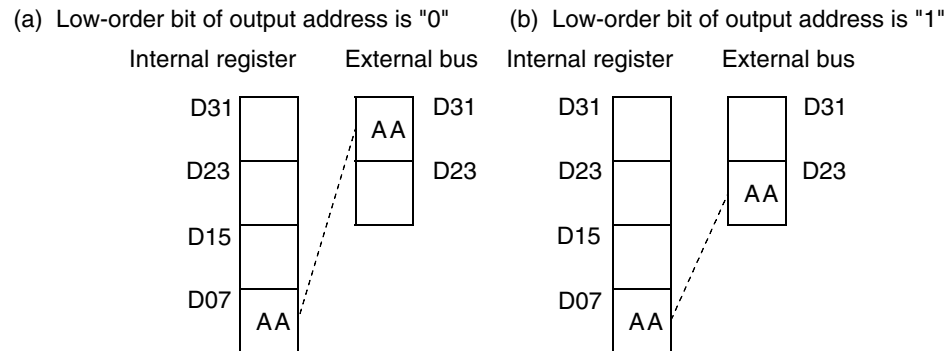
○ Halfword access (at LDUH or STH instruction execution)

Figure 4.4-4 Relationship between Internal Registers and External Data Buses for Halfword Access



○ **Byte access (when executing an LDUB or STB instruction)**

Figure 4.4-5 Relationship between Internal Registers and External Data Buses for Byte Access

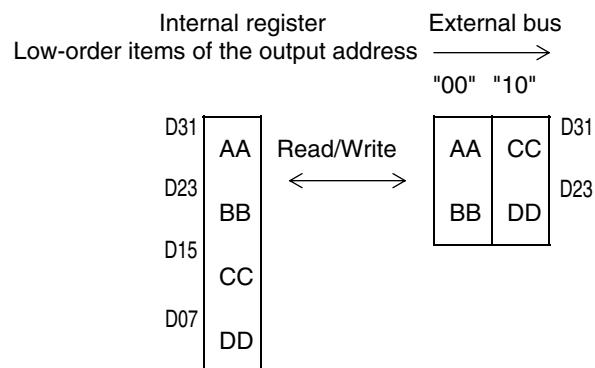


■ **Data Bus Widths**

The figures below show the relationship between the internal registers and external data buses for each data bus width.

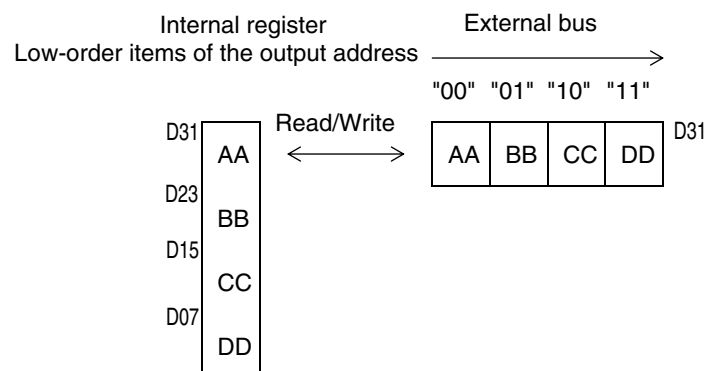
○ **16-bit bus**

Figure 4.4-6 Relationship between Internal Registers and External Data Buses for 16-bit Bus Width



○ **8-bit bus**




Figure 4.4-7 Relationship between Internal Registers and External Data Buses for 8-bit Bus Width



■ External Bus Access

Figure 4.4-8 "External Bus Access for 16-bit Bus Width" shows external bus access (16-bit bus width) for word, halfword, and byte access. Figure 4.4-9 "External Bus Access for 8-bit Bus Width" shows external bus access (8-bit bus width) for word, halfword, and byte access. Moreover, Figure 4.4-8 "External Bus Access for 16-bit Bus Width" and Figure 4.4-9 "External Bus Access for 8-bit Bus Width" the following items:

- Access byte positions
- Program and output addresses
- 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
	:	Leading byte position of the output address
 + 	:	Data byte position accessed
① to ④	:	Bus access count

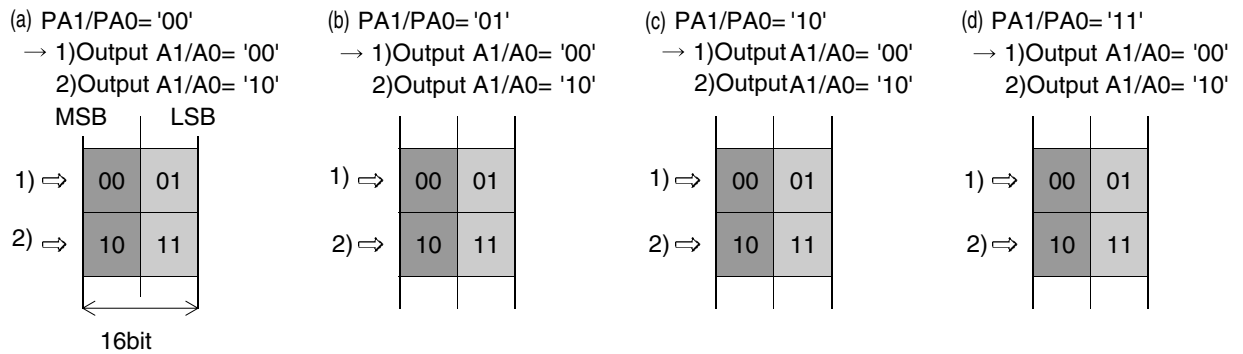
Note

The MB91F127/128 does not detect misalignment errors. Therefore, for word access, the two low-order bits of the output addresses will be set to "00" regardless of whether the two low-order bits of addresses specified by the program are "00", "01", "10", or "11". For halfword access, "00" will be set for "00" or "01" and "10" will be set for "10" or "11".

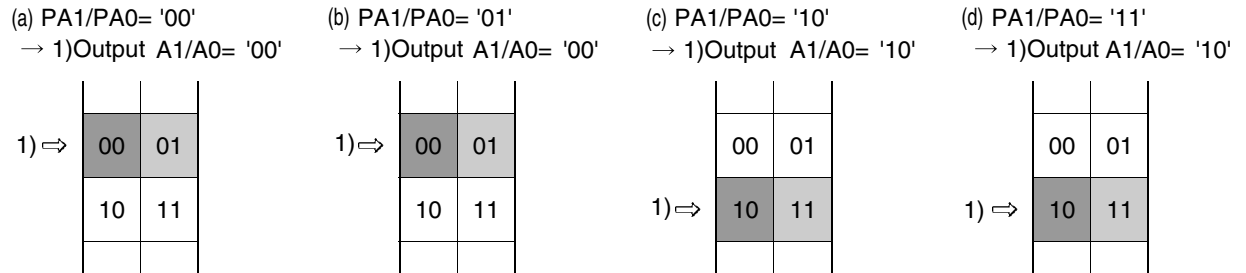
○ 16-bit bus

Figure 4.4-8 External Bus Access for 16-bit Bus Width

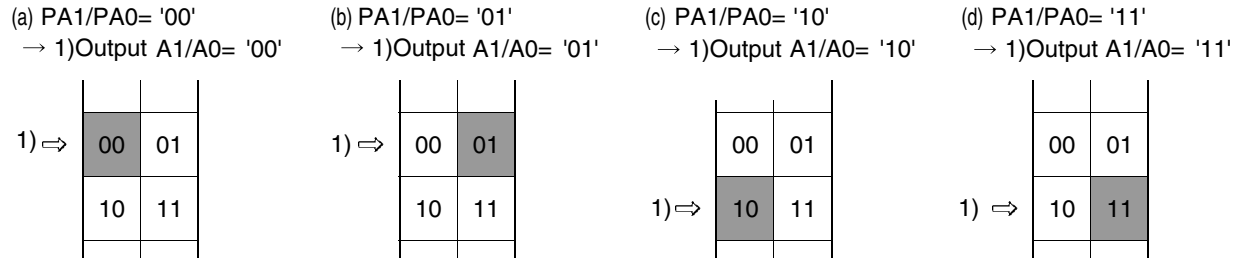
(A) Word access



(B) Halfword access



(C) Byte access



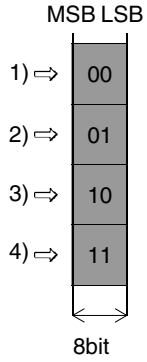
○ 8-bit bus

Figure 4.4-9 External Bus Access for 8-bit Bus Width

(A) Word access

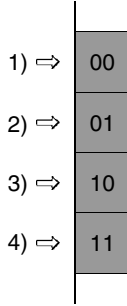
(a) PA1/PA0= '00'

- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'
- 3) Output A1/A0= '10'
- 4) Output A1/A0= '11'



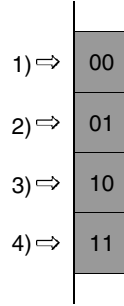
(b) PA1/PA0= '01'

- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'
- 3) Output A1/A0= '10'
- 4) Output A1/A0= '11'



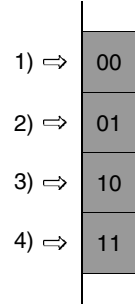
(c) PA1/PA0= '10'

- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'
- 3) Output A1/A0= '10'
- 4) Output A1/A0= '11'



(d) PA1/PA0= '11'

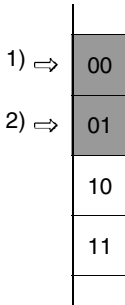
- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'
- 3) Output A1/A0= '10'
- 4) Output A1/A0= '11'



(B) Halfword access

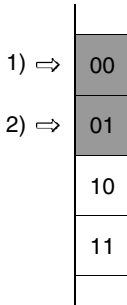
(a) PA1/PA0= '00'

- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'



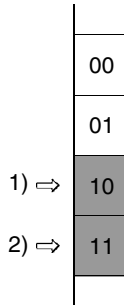
(b) PA1/PA0= '01'

- 1) Output A1/A0= '00'
- 2) Output A1/A0= '01'



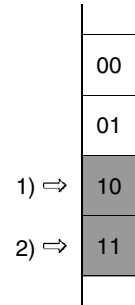
(c) PA1/PA0= '10'

- 1) Output A1/A0= '10'
- 2) Output A1/A0= '11'



(d) PA1/PA0= '11'

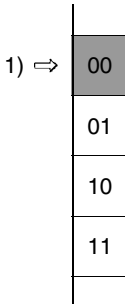
- 1) Output A1/A0= '10'
- 2) Output A1/A0= '11'



(C) Byte access

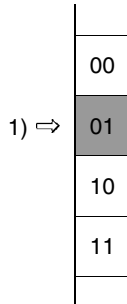
(a) PA1/PA0= '00'

- 1) Output A1/A0= '00'



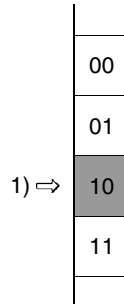
(b) PA1/PA0= '01'

- 1) Output A1/A0= '01'



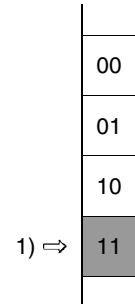
(c) PA1/PA0= '10'

- 1) Output A1/A0= '10'



(d) PA1/PA0= '11'

- 1) Output A1/A0= '11'



PA1/PA0 : Two low-order bits of the address specified by the program

Output A1/A0 : Two low-order bits of the output address

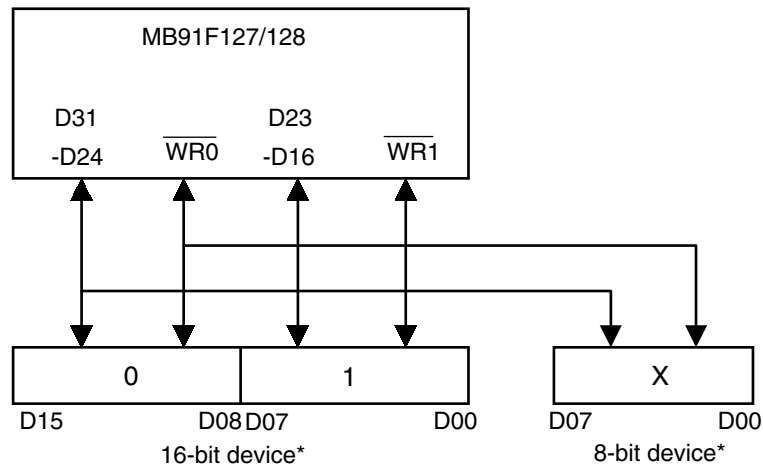
■ : Leading byte position of the output address

1) to 4) : Bus access count

■ Sample External Device Connection

Figure 4.4-10 "Sample Connection between the MB91F127/128 and External Devices" shows a sample connection between the MB91F127/128 and external devices.

Figure 4.4-10 Sample Connection between the MB91107/MB91108 and External Devices



("0"/"1" one low-order bit of the address. For "X", the one low-order bit of the address can be "0" or "1".)

*: For a 16-bit or 8-bit device, the data bus on the MSB side of the MB91F127/128 is used.

4.4.3 Little Endian Bus Access

Little endian order is used for external bus access of areas for which the little endian register (LER) has been set.

■ Outline of Little Endian Order

Little endian bus access of the MB91F127/128 uses the same operations as for big endian bus access. There is basically no difference between little endian and big endian order for as far as output addresses and output of control signals are concerned. Little endian and big endian order are supported by swapping the data bus byte positions according to the bus widths.

Note that the big endian areas and little endian areas must be physically separated during connection.

- The order of the output addresses are the same regardless of whether big endian or little endian order is applied.
- Word access: The byte data on the MSB side corresponding to address 00 of the big endian area becomes the byte data on the LSB side for the little endian area.

For word access, all byte positions of the four bytes in the word are reversed.

"00" --> "11", "01" --> "10", "10" --> "01", "11" --> "00"

- Word access: The byte data on the MSB side corresponding to address 00 of the big endian area becomes the byte data on the LSB side for the little endian area.

For word access, all byte positions of the four bytes in the word are reversed.

"00" --> "11", "01" --> "10", "10" --> "01", "11" --> "00"

- Halfword access: The byte data on the MSB side corresponding to address "0" of the big endian area becomes the byte data on the LSB side for the little endian area.

For halfword access, the byte positions of the two bytes in the halfword are reversed.

"0" --> "1", "1" --> "0"

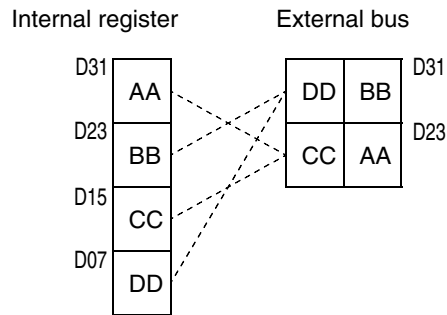
- Byte access: For byte access, operations are the same for big endian and little endian order.
- The data bus control signals used for the 16-bit and 8-bit bus width are the same regardless of whether the big endian or little endian order is applied.

■ Data Formats

The figures below show the relationship between the internal registers and external data buses for each data format.

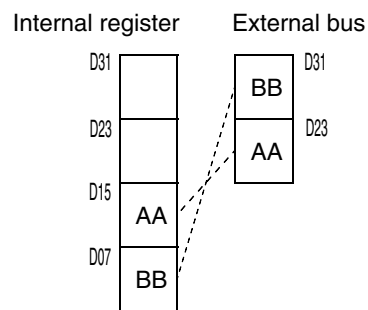
○ Word access (at LD or ST instruction execution)

Figure 4.4-11 Relationship between Internal Registers and External Data Buses for Word Access



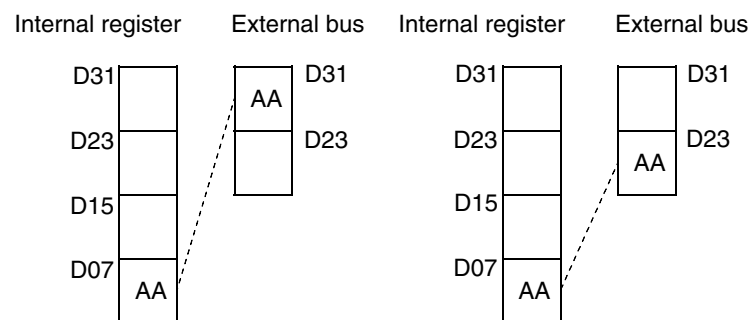
○ Halfword access (when executing an LDUH or STH instruction)

Figure 4.4-12 Relationship between Internal Registers and External Data Buses for Halfword Access



○ Byte access (when executing an LDUB or STB instruction)

Figure 4.4-13 Relationship between Internal Registers and External Data Buses for Byte Access

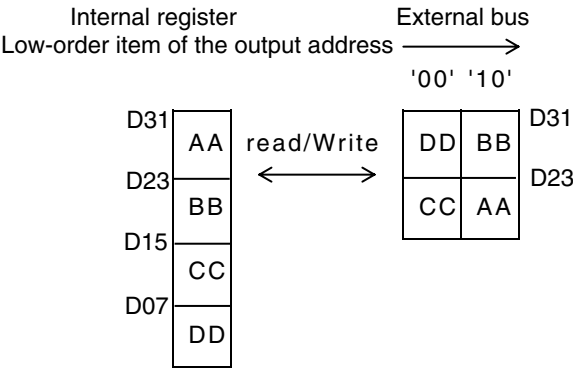


■ Data Bus Widths

The figures below show the relationship between the internal registers and external data buses for each data bus width.

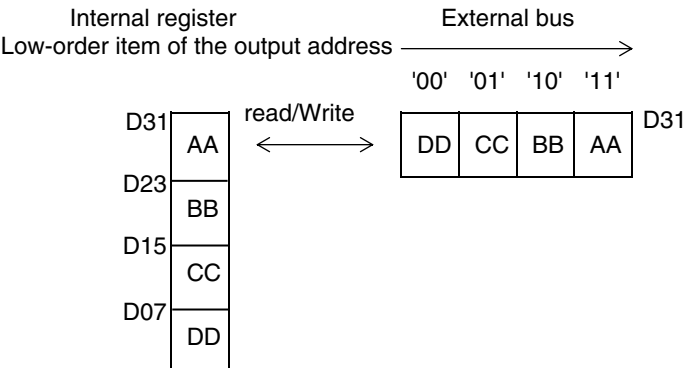
○ 16-bit bus

Figure 4.4-14 Relationship between Internal Registers and External Data Buses for 16-bit Bus



○ 8-bit bus

Figure 4.4-15 Relationship between Internal Registers and External Data Buses for 8-bit Bus

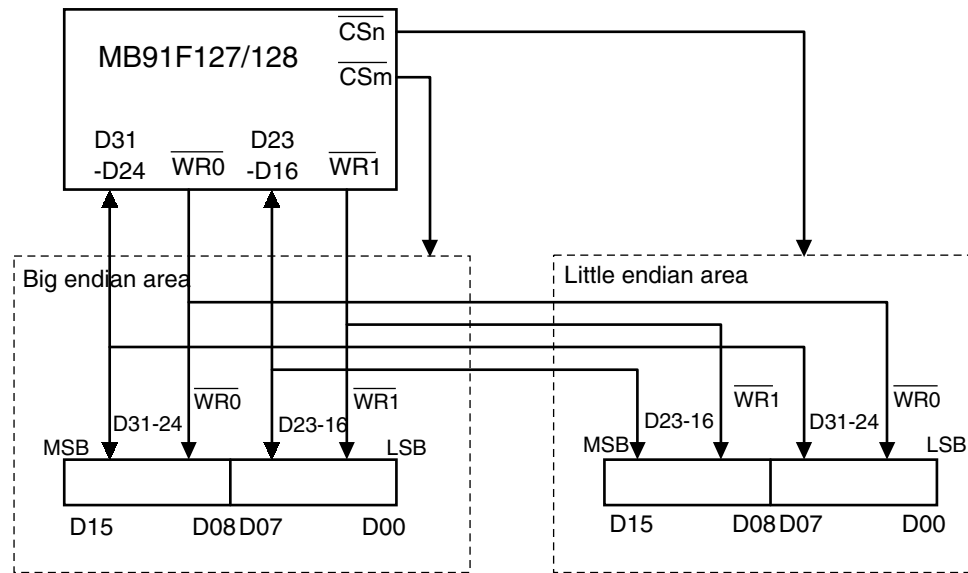


■ Examples for External Device Connection

Figure 4.4-16 "Example for the Connection of the MB91F127/128 with External Devices (for 16-bit Bus)" and Figure 4.4-17 "Example for the Connection of the MB91F127/127 with External Devices (for 8-bit Bus)" show examples for the connection of the MB91F127/128 with external devices.

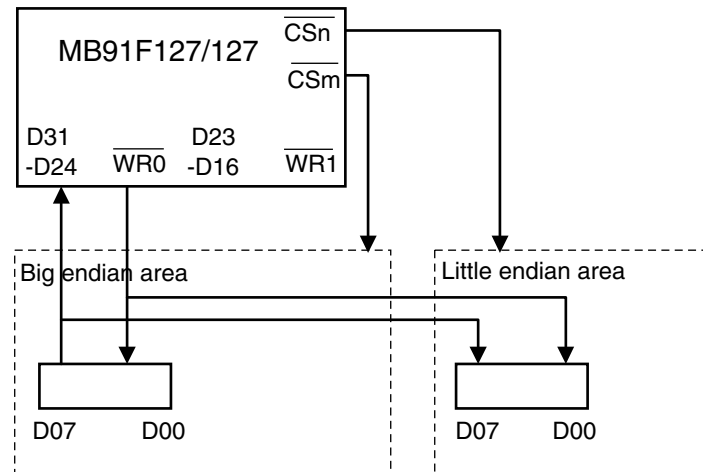
○ 16-bit bus

Figure 4.4-16 Example for the Connection of the MB91F127/128 with External Devices (for 16-bit Bus)



○ 8-bit bus

Figure 4.4-17 Example for the Connection of the MB91F127/127 with External Devices (for 8-bit Bus)



4.4.4 Comparison of External Access Operations of Big Endian and Little Endian Type

This section compares the external access operations of big endian and little endian type for word access, halfword access, and byte access in relationship to the bus widths.

■ Word Access

Bus width	Big endian mode	Little endian mode
16-bit bus	<div><div>Internal register</div><div>External pin</div><div>Control pin</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><div><div>D16</div></div><div><div>WR0 CASL WEL</div><div>WR1 CASH WEH</div><div>— — —</div><div>— — —</div></div><div><div>① ②</div></div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</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>BB</div><div>CC</div><div>AA</div></div><div><div>D16</div></div><div><div>WR0 CASL WEL</div><div>WR1 CASH WEH</div><div>— — —</div><div>— — —</div></div><div><div>① ②</div></div></div>
8-bit bus	<div><div>Internal register</div><div>External pin</div><div>Control pin</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><div><div>D24</div></div><div><div>WR0 CASL WEL</div><div>— — —</div><div>— — —</div><div>— — —</div></div><div><div>① ② ③ ④</div></div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</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><div><div>D24</div></div><div><div>WR0 CASL WEL</div><div>— — —</div><div>— — —</div><div>— — —</div></div><div><div>① ② ③ ④</div></div></div>

Note: MB91F127/128 dose not have CASL, CASH, CAS, WEL, WEH and WE pins that are used for the cntrol pin.

■ Halfword Access

Bus width	Big endian mode	Little endian mode
16-bit bus	<p>Internal register External pin Control pin</p> <p>address: "0"</p> <p>①</p>	<p>Internal register External pin Control pin</p> <p>address: "0"</p> <p>①</p>
	<p>Internal register External pin Control pin</p> <p>address: "2"</p> <p>①</p>	<p>Internal register External pin Control pin</p> <p>address: "2"</p> <p>①</p>

Bus width	Big endian mode	Little endian mode
8-bit bus	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address: "0" "1"</div><div><div>D31</div><div>D24</div><div>AA</div><div>BB</div><div>D00</div></div><div><div>AA</div><div>BB</div></div><div><div>WR0</div><div>CAS</div><div>WEL</div></div><div><div>①</div><div>②</div></div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address: "0" "1"</div><div><div>D31</div><div>D24</div><div>BB</div><div>AA</div><div>D00</div></div><div><div>BB</div><div>AA</div></div><div><div>WR0</div><div>CAS</div><div>WEL</div></div><div><div>①</div><div>②</div></div></div>
	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address: "2" "3"</div><div><div>D31</div><div>D24</div><div>CC</div><div>DD</div><div>D00</div></div><div><div>CC</div><div>DD</div></div><div><div>WR0</div><div>CAS</div><div>WEL</div></div><div><div>①</div><div>②</div></div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address: "2" "3"</div><div><div>D31</div><div>D24</div><div>DD</div><div>CC</div><div>D00</div></div><div><div>DD</div><div>CC</div></div><div><div>WR0</div><div>CAS</div><div>WEL</div></div><div><div>①</div><div>②</div></div></div>

■ Byte Access

Bus width	Big endian mode	Little endian mode
16-bit bus	<p>Internal register External pin Control pin</p> <p>address : "0"</p> <p>①</p>	<p>Internal register External pin Control pin</p> <p>address : "0"</p> <p>①</p>
	<p>Internal register External pin Control pin</p> <p>address : "1"</p> <p>①</p>	<p>Internal register External pin Control pin</p> <p>address : "1"</p> <p>①</p>
	<p>Internal register External pin Control pin</p> <p>address : "2"</p> <p>①</p>	<p>Internal register External pin Control pin</p> <p>address : "2"</p> <p>①</p>

CHAPTER 4 BUS INTERFACE

Bus width	Big endian mode	Little endian mode
	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "3"</div><div><div>D31</div><div>D31</div><div>D16</div><div>D00</div><div>DD</div></div><div><div>DD</div></div><div><div>WR1</div><div>CASH</div><div>WEH</div></div><div>①</div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "3"</div><div><div>D31</div><div>D31</div><div>D16</div><div>D00</div><div>DD</div></div><div><div>DD</div></div><div><div>WR1</div><div>CASH</div><div>WEH</div></div><div>①</div></div>
8-bit bus	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "0"</div><div><div>D31</div><div>D31</div><div>D24</div><div>D00</div><div>AA</div></div><div><div>AA</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div>①</div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "0"</div><div><div>D31</div><div>D31</div><div>D24</div><div>D00</div><div>AA</div></div><div><div>AA</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div>①</div></div>
	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "1"</div><div><div>D31</div><div>D31</div><div>D24</div><div>D00</div><div>BB</div></div><div><div>BB</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div>①</div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "1"</div><div><div>D31</div><div>D31</div><div>D24</div><div>D00</div><div>BB</div></div><div><div>BB</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div>①</div></div>

Bus width	Big endian mode	Little endian mode
	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "2"</div><div><div>D31</div><div>D24</div><div>CC</div><div>D00</div></div><div><div>CC</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div>①</div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "2"</div><div><div>D31</div><div>D24</div><div>CC</div><div>D00</div></div><div><div>CC</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div>①</div></div>
	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "3"</div><div><div>D31</div><div>D24</div><div>DD</div><div>D00</div></div><div><div>DD</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div>①</div></div>	<div><div>Internal register</div><div>External pin</div><div>Control pin</div><div>address : "3"</div><div><div>D31</div><div>D24</div><div>DD</div><div>D00</div></div><div><div>DD</div></div><div><div>WR0</div><div>CAS</div><div>WE</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div><div>—</div><div>—</div><div>—</div></div><div>①</div></div>

4.5 Bus Timings

This section provides bus access timing diagrams for each mode and describes operations for the following items:

- Normal bus access
 - Wait cycle
 - Time-division I/O interface
 - External bus requests
-

■ Normal Bus Access

For the normal bus interface, the basic bus cycle is "two clock cycles" that are used for both the read and write cycles. In this manual, these two cycles are represented as "BA1" and "BA2".

- Basic read cycle
- Basic write cycle
- Read cycle for each mode
- Write cycle for each mode
- Read/write mixed cycles

■ Wait Cycle

There are two types of wait cycles: Automatic wait cycles that use the WTC bit of the AMD register, and external wait cycles, that use the RDY pin.

Wait cycle is a mode where the previous cycle is continued. The "BA1" cycle is repeated until wait is released.

- Automatic wait cycle
- External wait cycle

■ Time-division I/O Interface

In area 1, the time-division I/O interface for addresses and data is supported. Time-division input-output is performed with the bus width specified by BW1 and BW0.

For the time-division I/O interface, a total of four clock cycles (two clocks of the address output cycle and two clocks of the data access cycle) are the basic bus cycle. In the address output cycle, the ALE pin is asserted as the output address latch signal.

■ External Bus Requests

- Bus authority release
- Bus authority acquisition

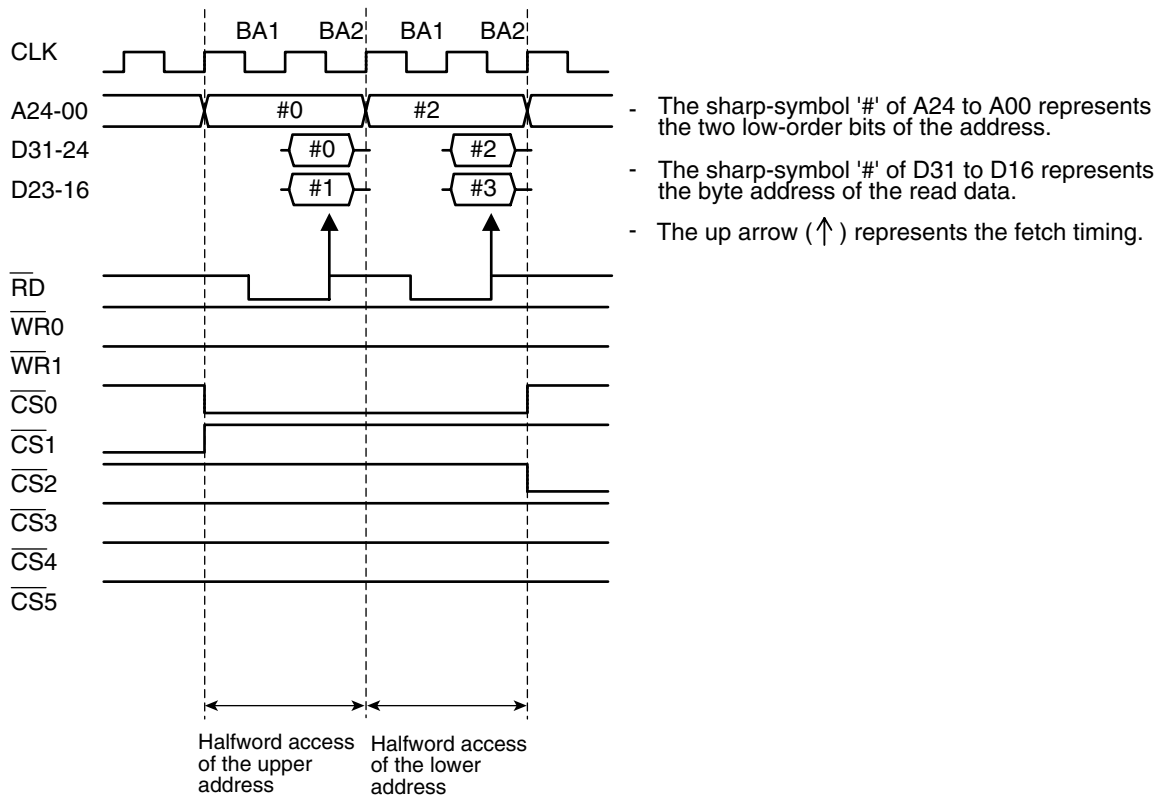
4.5.1 Basic Read Cycle

This section presents the operation timings of the basic read cycle.

■ Basic Read Cycle Timing

- Bus width: 16 bits, access: CS0 area access

Figure 4.5-1 Timing Example of the Basic Read Cycle



[Explanation of the operation]

- For CLK, the operating clock frequency of the external bus is output.
When the gear catches, the CLK frequency is also reduced based on the gear ratio.
- For A24 to A00 (addresses 24 to 00), the addresses of the leading byte positions for word, halfword, or byte access of the read cycle are output from the start of the bus cycle (BA1). In Figure 4.5-1 "Timing Example of the Basic Read Cycle", word access using a 16-bit width is executed. The address (two low-order bits "0") of the 16 high-order bits for word access is output at the first bus cycle. The address (two low-order bits "2") of the 16 low-order bits is output at the second bus cycle.
- D31 to D16 (data 31 to 16) represent the read data from external memory and I/O. In a read cycle, D31 to D16 are fetched at the rise of \overline{RD} regardless of the bus width and of whether word, halfword, or byte access is performed. The decision on whether the fetched data is

CHAPTER 4 BUS INTERFACE

valid is made internally on the chip.

- \overline{RD} is a read strobe signal of the external data bus. \overline{RD} is asserted at the fall of BA1 and negated at the fall of BA2.
- In a read cycle, $\overline{WR0}$ and $\overline{WR1}$ are in "negate" state.
- The output of the $\overline{CS0}$ to $\overline{CS5}$ (area chip select) signals are asserted from the start of the bus cycle (BA1) using the same timings as those of A24 to A00. $\overline{CS0}$ to $\overline{CS5}$ are created by decoding the address output. $\overline{CS0}$ to $\overline{CS5}$ remain unchanged until the address output is changed and the chip select areas set using ASR and AMR are changed. In addition, at least one of $\overline{CS0}$ to $\overline{CS5}$ is always asserted.

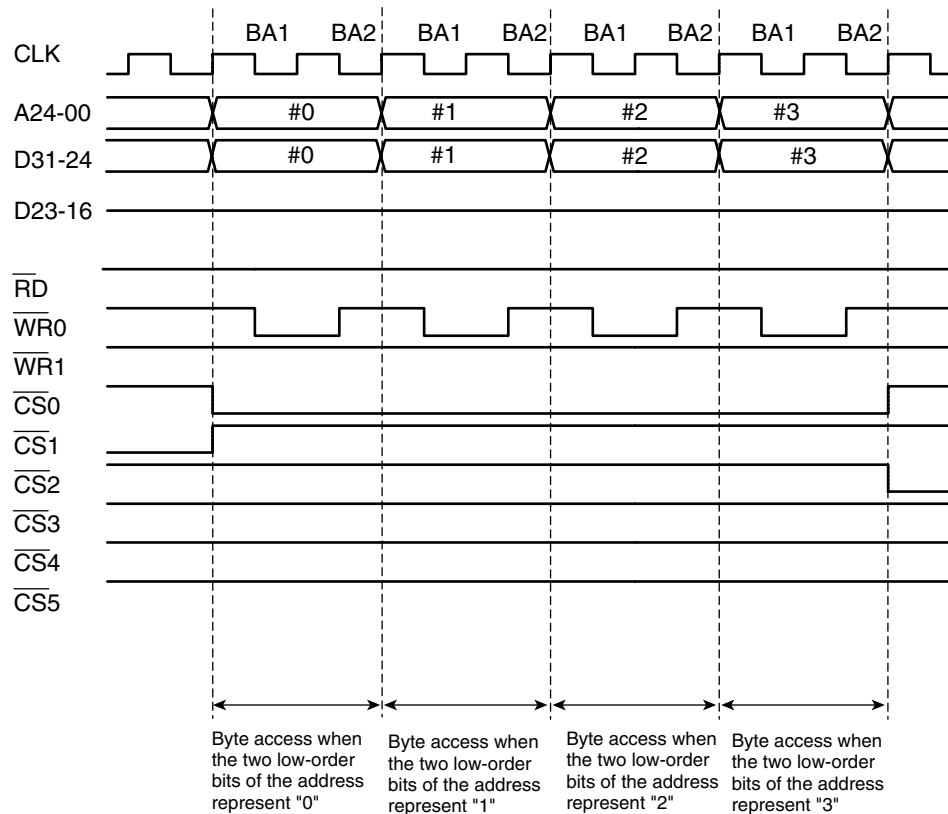
4.5.2 Basic Write Cycle

This section provides the operation timings of the basic write cycle.

■ Basic Write Cycle Timing

- Bus width: 8 bits, access: CS0 area access

Figure 4.5-2 Timing Example of the Basic Write Cycle



[Explanation of operation]

- For A24 to A00 (addresses 24 to 00), the addresses of the leading byte positions for half, halfword, or byte access of the write cycle are output from the start of the bus cycle (BA1). In Figure 4.5-2 "Timing Example of the Basic Write Cycle", word access using a width of 8 bits is executed. The leading byte (address lower "0") of word access is first output. The address of +1 ("1") from the leading byte, address of +2 ("2"), and address of +3 ("3") are then output sequentially.
- D31 to D16 (data 31 to 16) represent the data to be written to external memory and I/O. For a write cycle, the data to be written is output from the start of the bus cycle (BA1). When the bus cycle ends (end of BA2), High-Z is set. In Figure 4.5-2 "Timing Example of the Basic Write Cycle", the data to be written is output to D31 to D24 because a data-bus width of 8 bits is used.

CHAPTER 4 BUS INTERFACE

- During a write cycle, \overline{RD} is in "negate" state.
- $\overline{WR0}$ and $\overline{WR1}$ are write read strobe signals of the external data bus. $\overline{WR0}$ and $\overline{WR1}$ are asserted at the fall of BA1 and negated at the fall of BA2.

D31 to D24 comprise $\overline{WR0}$ and D23 to D16 comprise $\overline{WR1}$. $\overline{WR0}$ and $\overline{WR1}$ are asserted based on the corresponding data bus. In Figure 4.5-2 "Timing Example of the Basic Write Cycle", only $\overline{WR0}$ is asserted because an 8-bit data bus is used.

- If the maximum bus width of check select areas 0 to 5 is eight bits, that is, the entire set area is eight bits, D23 to D16 and $\overline{WR1}$ will automatically be used as I/O ports and High-Z will be set.

Figure 4.5-2 "Timing Example of the Basic Write Cycle" shows the case that D23 to D16 and $\overline{WR1}$ are set as I/O ports.

Note that D23 to D16 and $\overline{WR1}$ cannot be used as I/O ports if the bus width of even one of the check select areas 0 to 5 is 16 bits.

Maximum bus width	Pin	
	D31-24 $\overline{WR0}$	D23-16 $\overline{WR1}$
16 bits	D31-24 $\overline{WR0}$	D23-16 $\overline{WR1}$
8 bits	D31-24 $\overline{WR0}$	I/O port

- DACK0 to DACK2 and EOP0 to EOP2 are output during the external bus cycle of the DMA. Whether to output DACK0 to DACK2 and EOP0 to EOP2 is decided based on the DMAC register settings. The output timings are the same as those of $\overline{WR0}$ and $\overline{WR1}$.

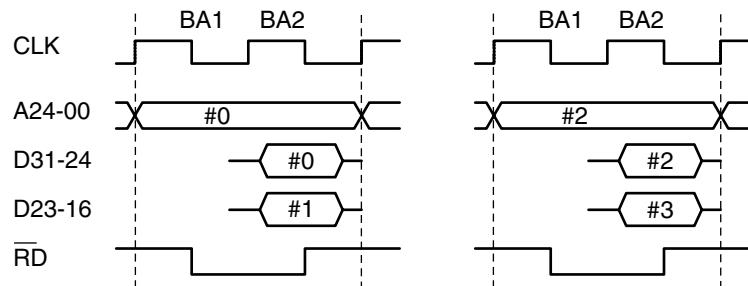
4.5.3 Read Cycle of Each Mode

This section presents the operating timings of the read cycle for each mode.

■ Read Cycle Timings

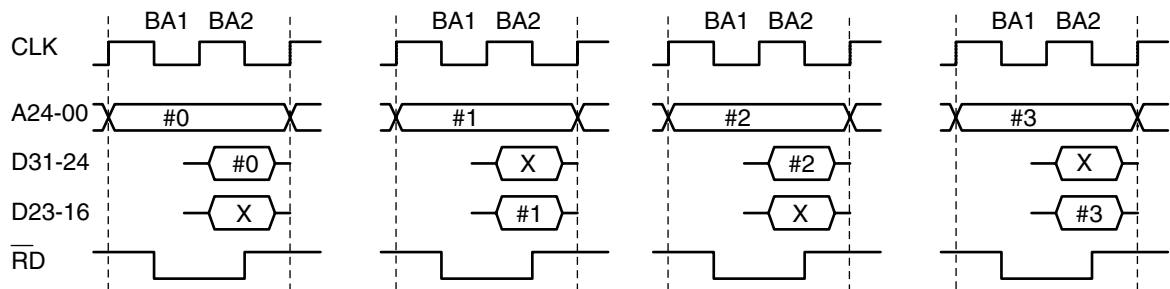
- Bus width: 16 bits, access: Halfword

Figure 4.5-3 Read Cycle Timing Example 1



- Bus width: 16 bits, access: Byte

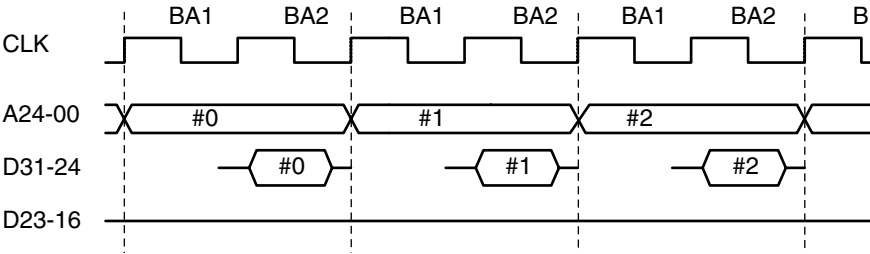
Figure 4.5-4 Read Cycle Timing Example 2



X: Invalid data input

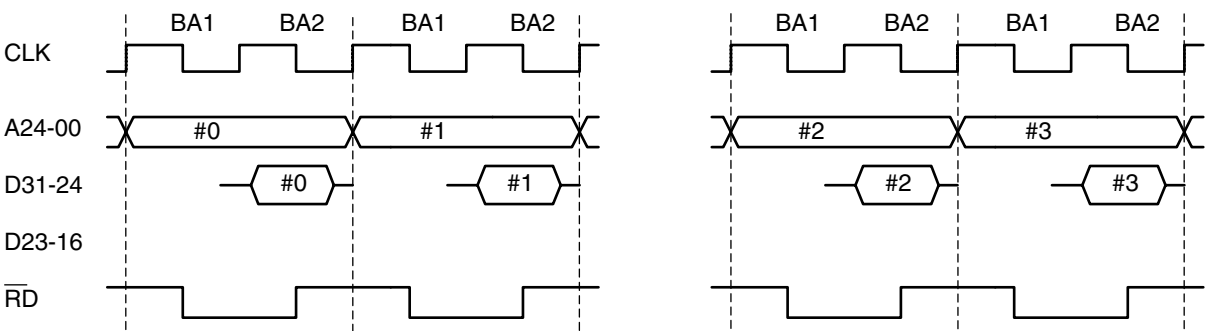
- Bus width: 8 bits, access: Word

Figure 4.5-5 Read Cycle Timing Example 3



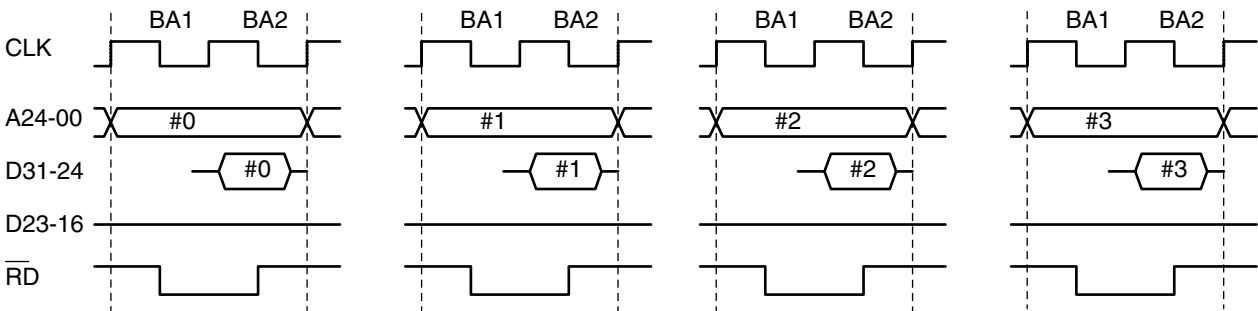
- Bus width: 8 bits, access: Halfword

Figure 4.5-6 Read Cycle Timing Example 4



- Bus width: 8 bits, access: Byte

Figure 4.5-7 Read Cycle Timing Example 5



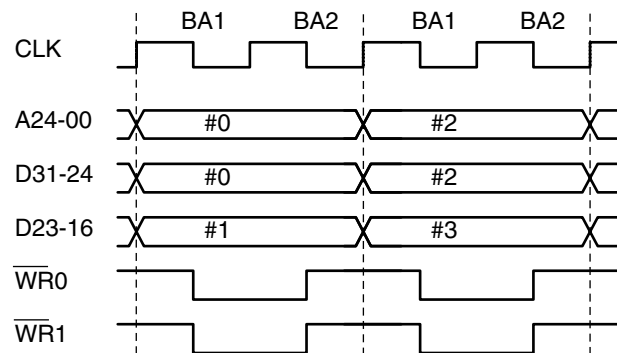
4.5.4 Write Cycle of Each Mode

This section presents the operating timings of the write cycle for each mode.

■ Write Cycle Timings

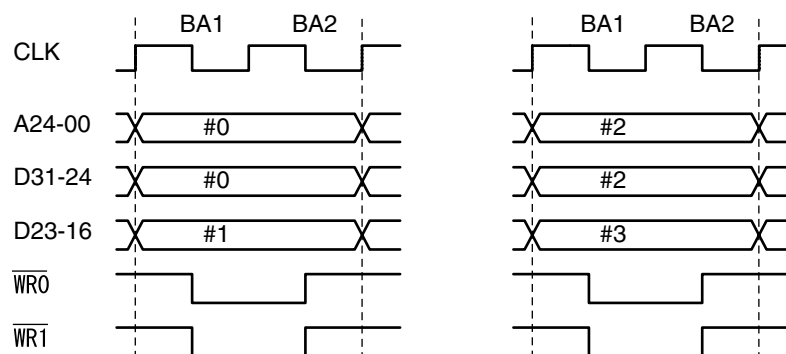
- Bus width: 16 bits, access: Word

Figure 4.5-8 Write cycle timing example 1



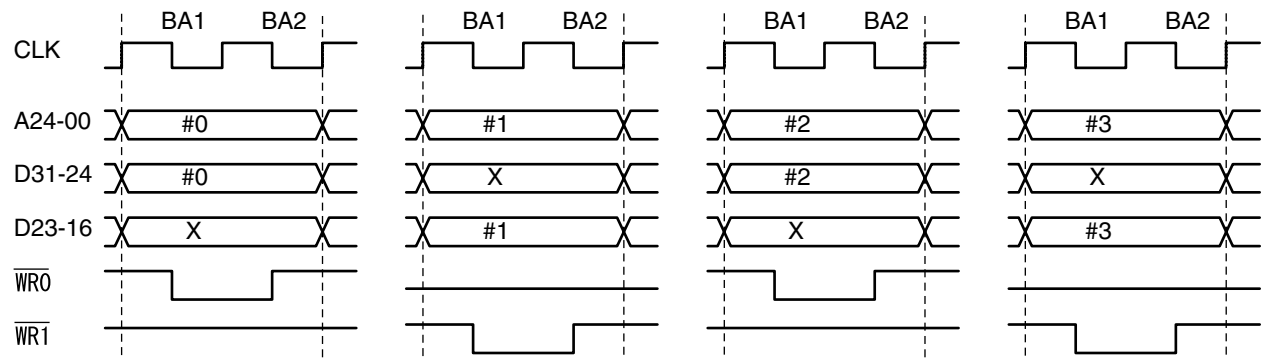
- Bus width: 16 bits, access: Halfword

Figure 4.5-9 Write cycle timing example 2



- Bus width: 16 bits, access: Byte

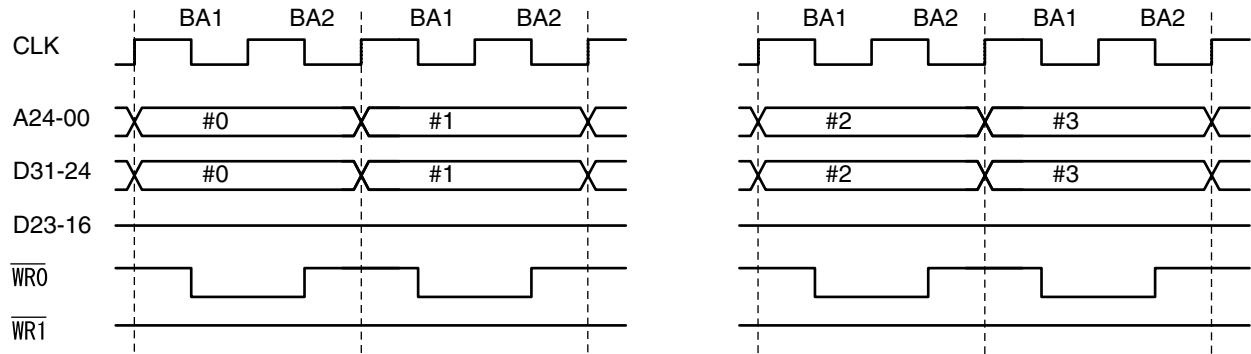
Figure 4.5-10 Write cycle timing example 3



X: Invalid data output

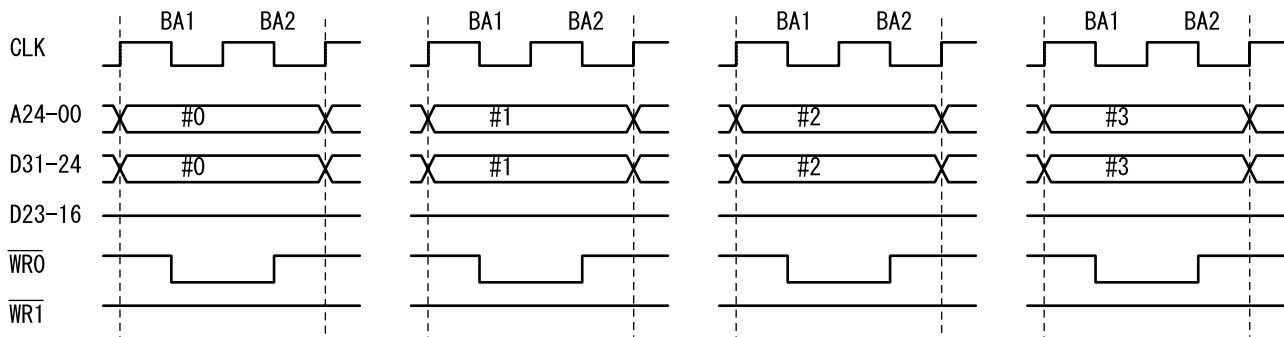
- Bus width: 8 bits, access: Halfword

Figure 4.5-11 Write cycle timing example 4



- Bus width: 8 bits, access: Byte

Figure 4.5-12 Write cycle timing example 4



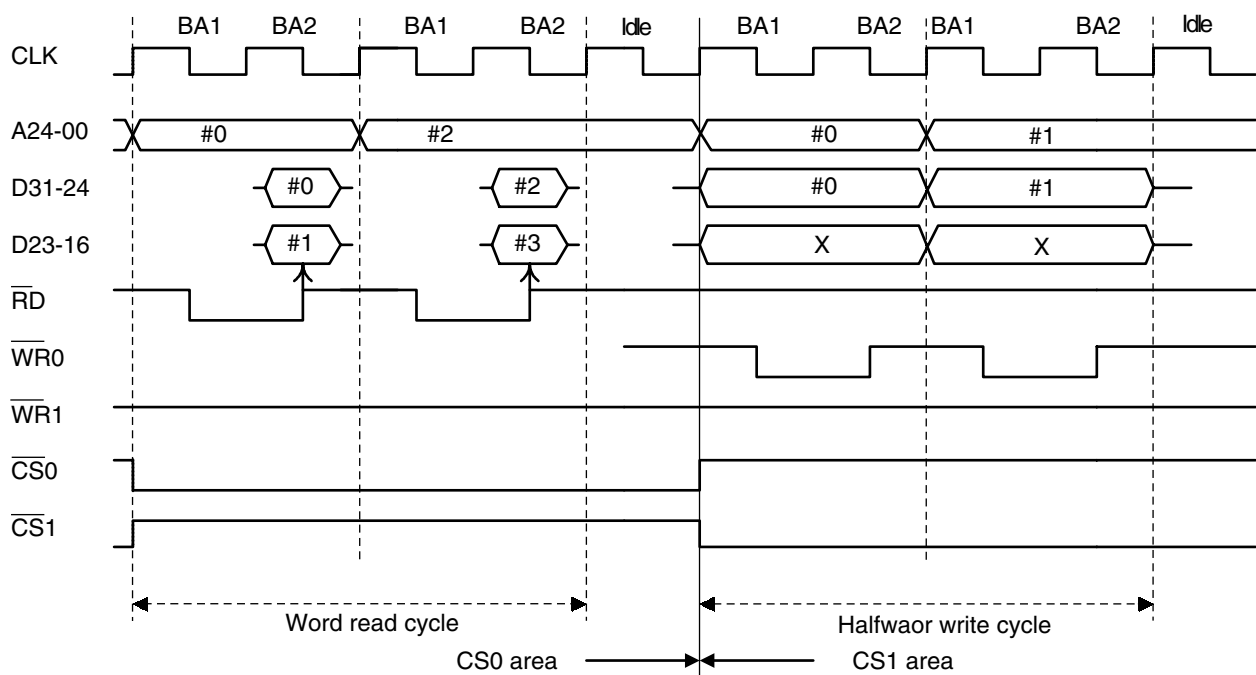
4.5.5 Read/Write Mixed Cycles

This section presents the operating timings of read/write mixed cycles.

■ Read/Write Mixed Cycle Timings

- CS0 area: Bus width 16 bits, word read
- CS1 area: Bus width 8 bits, halfword write

Figure 4.5-13 Sample Read/Write Mixed Cycle Timing



[Explanation of operation]

- The Figure 4.5-13 "Sample Read/Write Mixed Cycle Timing" shows insertion of an idle cycle (cycle where no bus cycle is executed) when the chip select areas are switched. When an idle cycle is inserted between bus cycles, the address of the previous bus cycle is continuously output until the next bus cycle starts. Accordingly, $\overline{CS0}$ to $\overline{CS5}$ that correspond to the output address are also continuously asserted.
- In the Figure 4.5-13 "Sample Read/Write Mixed Cycle Timing", the use of 16-bit and 8-bit buses is mixed.

Because the maximum bus width is 16 bits, D23 to D16 and $\overline{WR1}$ will not be used as I/O ports even for the 8-bit access area (CS1 area). Undefined data is output for D23 to D16 and $\overline{WR1}$ is set to "negate" state.

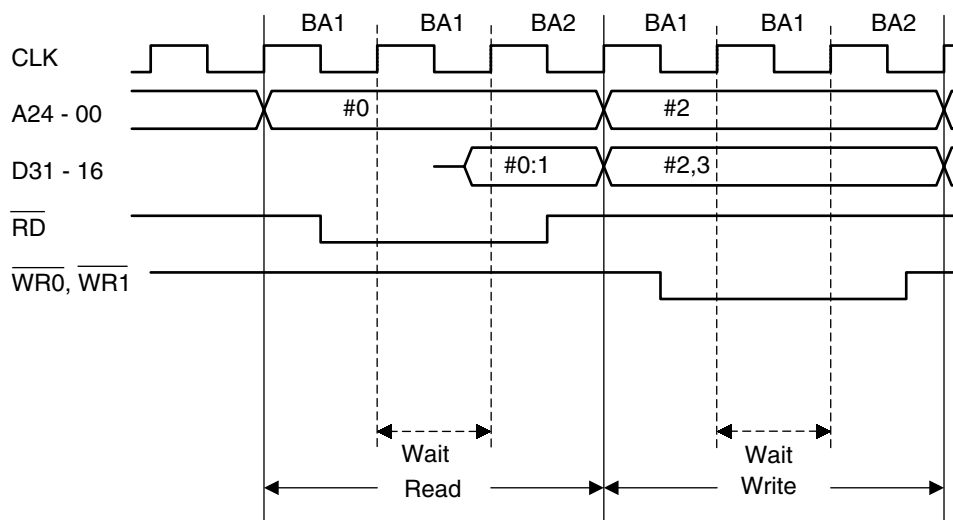
4.5.6 Automatic Wait Cycle

This section presents the operating timings of the automatic wait cycle.

■ Automatic Wait Cycle Timings

- Bus width: 16 bits, access: Halfword read/write

Figure 4.5-14 Sample Automatic Wait Cycle Timing"



[Explanation of operation]

- Automatic insertion of a wait cycle can be initiated by setting the WTC bit of the respective chip select area.
- In the Figure 4.5-14 "Sample Automatic Wait Cycle Timing", the WTC bit is set to "001" and a one-wait bus cycle is inserted in the normal bus cycle. In this case, a three-clock bus cycle is thus obtained by adding the two clock pulses of the normal bus cycle and the one clock pulse of the wait cycle.

Automatic wait of up to seven clock cycles can be set (in this case, the normal bus cycle is nine clock cycles).

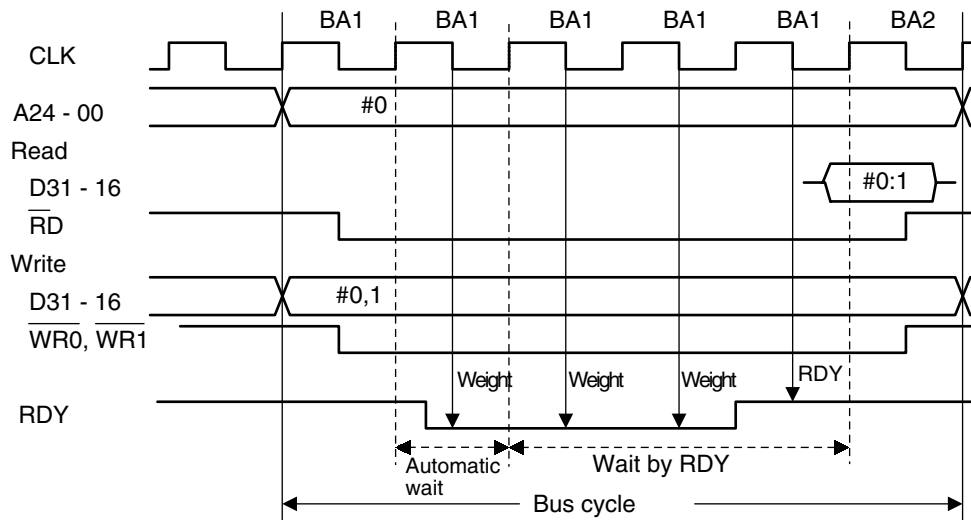
4.5.7 External Wait Cycle

This section presents the operating timings of the external wait cycle.

■ External Wait Cycle Timings

- Bus width: 16 bits, access: Halfword

Figure 4.5-15 Sample External Wait Cycle Timing



[Explanation of operation]

- The external wait cycle is enabled by setting the EPCR0 RDYE bit to "1" and enabling external RDY pin input.
- When using external RDY, always set an automatic wait cycle of more than one clock signal. That is, set the AMD register WTC bit to a value greater than "001". RDY is detected from the last cycle of automatic wait.
- In addition, synchronize input of external RDY with the fall of CLK pin output. If external RDY is at the "L" level when CLK falls, a wait cycle will be set and the same BA1 cycle repeated. At the "H" level, it is determined that the wait cycle has ended and the cycle then transits to the BA2 cycle.

4.5.8 Time-division I/O Interface

This section presents the operating timings of the time-division I/O interface.

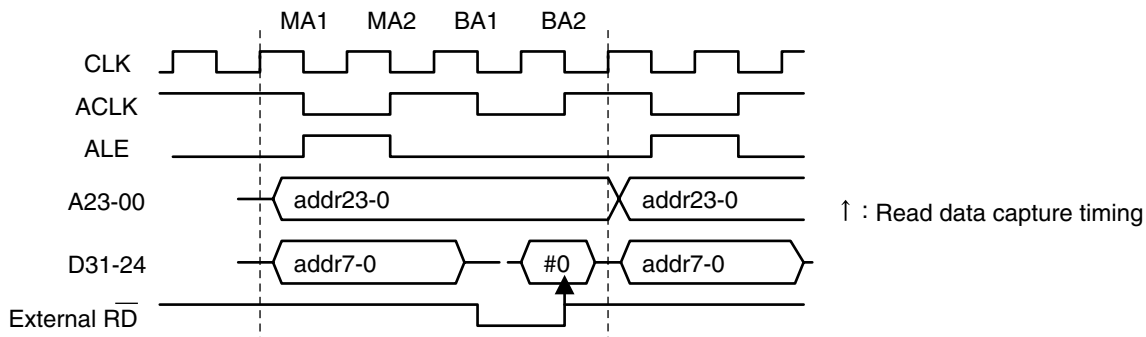
■ Sample Time-division I/O Interface Timing

○ 8-bit bus width read

Figure 4.5-16 "Sample 8-bit Bus Width Read Cycle Timing" shows the sample operating timing for the following conditions:

- AMD1: MPX=1, BW=00B, WTC=000B,E PCR0: ALEE=1
- Access: 8-bit data read

Figure 4.5-16 Sample 8-bit Bus Width Read Cycle Timing

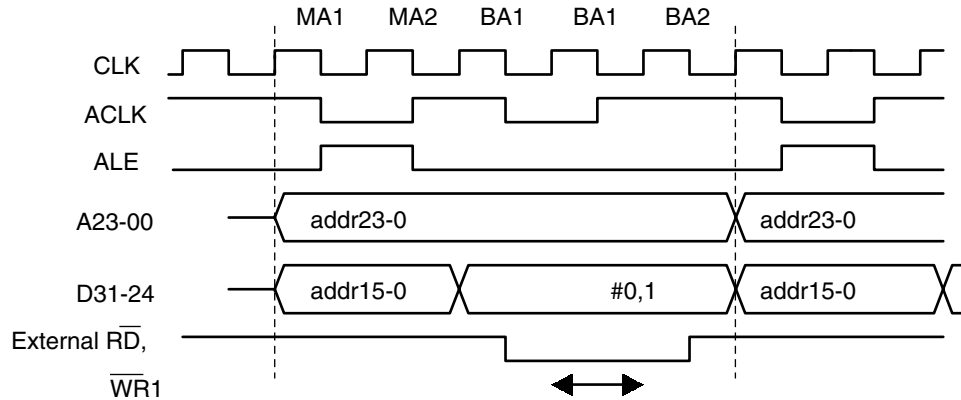


For the 8-bit bus width, addresses A07 to 00 are output to D31 to 24.

○ Automatic wait cycle in time-division mode (16-bit bus width write)

Figure 4.5-17 "Sample Automatic Wait Cycle Timing in Time-division Mode" shows the sample operating timing for the following conditions:

- - AMD1: MPX=1, BW=01B, WTC=001B, EPCR0: ALEE=1
- Access: 16-bit data read

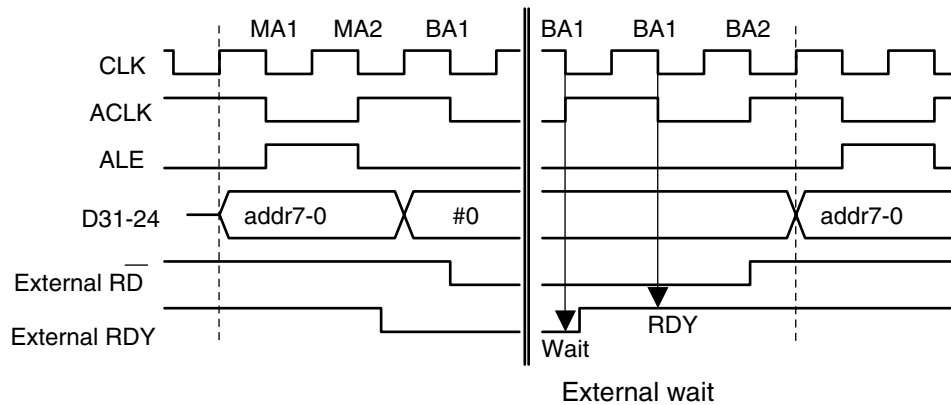
Figure 4.5-17 Sample Automatic Wait Cycle Timing in Time-division Mode

For the 16-bit bus width, addresses A15 to 00 are output to D31 to 16.

○ External wait cycle in time-division mode

Figure 4.5-18 "Sample External Wait Cycle Timing in Time-division Mode" shows the sample operating timing for the following conditions:

- AMD1: MPX=1, BW=00B, WTC=010B, EPCR0: ALEE=1
- Access: 8-bit data read

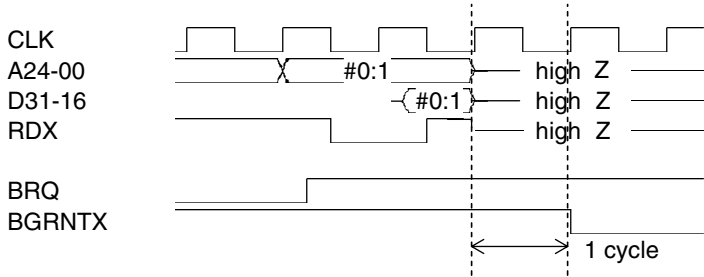
Figure 4.5-18 Sample External Wait Cycle Timing in Time-division Mode

4.5.9 External Bus Requests

This section presents the operating timings of external bus requests.

■ Bus Authority Release

Figure 4.5-19 Example for Bus Authority Release Timing

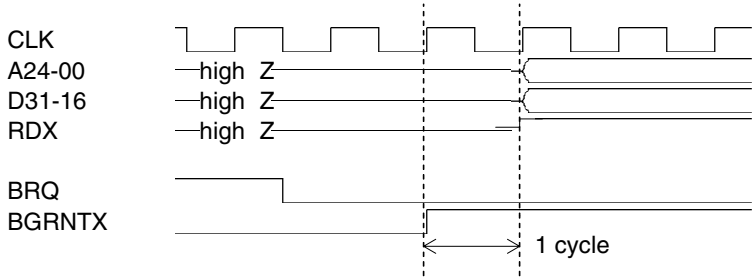


[Explanation of operation]

- Bus arbitration using BRQ and $\overline{\text{BGRNT}}$ can be performed by setting the EPCR0 BRE bit to "1".
- When the bus authority is released, $\overline{\text{BGRNT}}$ is asserted one cycle after the pins are set to High-Z.

■ Bus Authority Acquisition

Figure 4.5-20 Sample Timing for Bus Authority Acquisition



[Explanation of operation]

- Bus arbitration using BRQ and $\overline{\text{BGRNT}}$ can be executed by setting the EPCR0 BRE bit to "1".
- When the bus authority is obtained, each pin is asserted one cycle after $\overline{\text{BGRNT}}$ is negated.

4.6 Internal Clock Frequency Multiplier Operations (Clock Doubler)

The MB91F127/128 has a clock frequency multiplier circuit. The CPU can operate internally using a frequency that is single or double that of the bus interface. Even when either clock frequency is selected, the bus interface operates in synch with the CLK output pin. When an external access request from the CPU is detected, external access is started after waiting for the rise of CLK output.

■ Clock Frequency Selection Method

See Section 3.11.12 "Clock doubler function" for details on selecting double or single clock frequency.

The clock frequency selection can be changed at any time during chip operation. The bus operations are temporarily suppressed while the clock frequency is being switched. A reset automatically sets the clock frequency selection to single.

Figure 4.6-1 "Sample Timing for the Case of Double Clock Frequency (BW: 16 Bits, Access: Word Read)" shows a timing example of double clock frequency. Figure 4.6-2 "Sample Timing for the Case of Single Clock Frequency (BW: 16 Bits, Access: Word Read)" shows a timing example of single clock frequency.

Figure 4.6-1 Sample Timing for the Case of Double Clock Frequency (BW: 16 Bits, Access: Word Read)

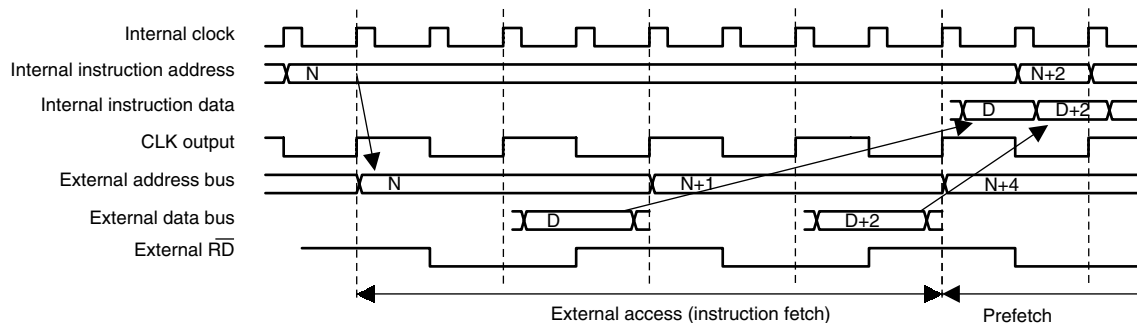
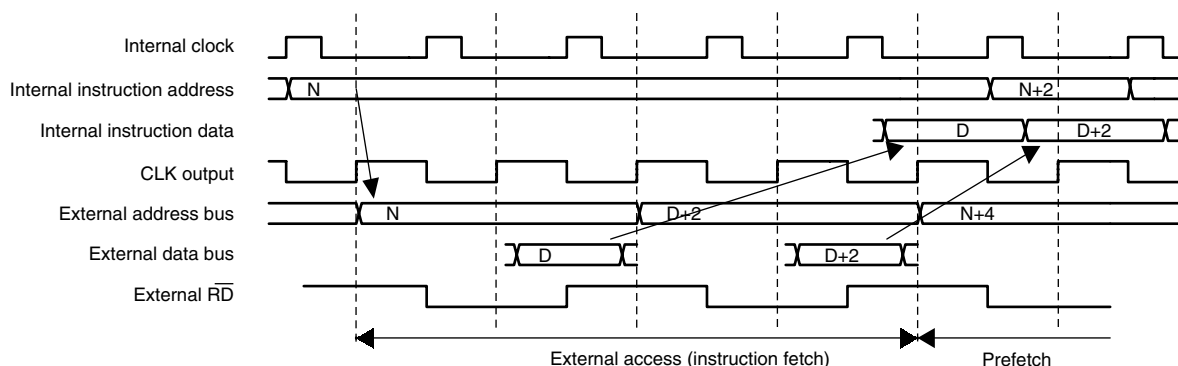


Figure 4.6-2 Sample Timing for the Case of Single Clock Frequency (BW: 16 Bits, Access: Word Read)



4.7 Program Example of External Bus Operations

This section presents a simple program example for operating the external buses.

■ Program Specifications for External Bus Operations

The register settings are as follows:

○ Areas

- Area 0 (AMD0): 16 bits, normal bus, and automatic wait 0
- Area 1 (AMD1): 16 bits, normal bus, and automatic wait 2
- Area 2 (AMD32): 16 bits, normal bus, and automatic wait 1
- Area 3 (AMD32): 16 bits, normal bus, and automatic wait 1
- Area 4 (AMD4): 16 bits, normal bus, and automatic wait 4
- Area 5 (AMD5): 16 bits, normal bus, and automatic wait 3

○ Other buses

- Refresh (RFCR): No wait and 1/8 setting
- External pin (EPCR0): External RDY acceptance and BRQ and BGRNTX arbitration
- External pin (DSCR): DRAM pin setting
- Little endian (LER): Area 2

Note also the following points:

- MD2, MD1, and MD0 are 001 and the external vector is 16-bit mode.
- After setting area 0 to the same bus width, set the mode register (MODR).
- Set areas 1 to 5 so that no overlap occurs.

■ Program Example of External Bus Operations

In the explanation of this program, the byte register is referred to as "byte" and the halfword register is referred to as "halfword".

* * * * * Program example * * * * *

// Settings of each register

init_epcr	ldi:20	#0xffff, r0	// External pin setting
			// External RDY wait, BRQ and BGRNTX bus arbitration
	ldi:20	#0x628, r1	// epcr0 register address setting
	sth	r0, @r1	// epcr0 register write
init_dscr	ldi:8	#0x0f, r0	//
init_amd0	ldi:8	#0x08, r0	// 16-bit bus, 0 wait
	ldi:20	#0x620, r1	// amd0 register address setting
	stb	r0, @r1	// amd0 register write
init_amd1	ldi:8	#0x0a, r0	// 16-bit bus, 2 wait
	ldi:20	#0x621, r1	// amd1 register address setting
	stb	r0, @r1	// amd1 register write
init_amd32	ldi:8	#0x49, r0	// Normal, 16-bit bus, 1 wait
	ldi:20	#0x622, r1	// amd32 register address setting
	stb	r0, @r1	// amd32 register write
init_amd4	ldi:8	#0x0c, r0	// 16-bit bus, 4 wait
	ldi:20	#0x623, r1	// amd4 register address setting
	stb	r0, @r1	// amd4 register write
init_amd5	ldi:8	#0x0b, r0	// 16-bit bus, 3 wait
	ldi:20	#0x624, r1	// amd5 register address setting
	stb	r0, @r1	// amd5 register write
init_rfcr	ldi:20	#0x0205, r0	// REL=2, R1W/R3W no wait, refresh, 1/8
	ldi:20	#0x626, r1	// rfcr register address setting
	sth	r0, @r1	// rfcr register write

```

init_asr      ldi:32  #0x0013001, r0 // asr1, amr1 register setting value
              ldi:32  #0x0015001, r1 // asr2, amr2 register setting value
              ldi:32  #0x0017001, r2 // asr3, amr3 register setting value
              ldi:32  #0x0019001, r3 // asr4, amr4 register setting value
              ldi:32  #0x001b001, r4 // asr5, amr5 register setting value
              ldi:20  #0x60c, r5      // asr1, amr1 register address setting
              ldi:20  #0x610, r6      // asr2, amr2 register address setting
              ldi:20  #0x614, r7      // asr3, amr3 register address setting
              ldi:20  #0x618, r8      // asr4, amr4 register address setting
              ldi:20  #0x61C, r9      // asr5, amr5 register address setting
              st       r0, @r5        // asr1, amr1 register write
              st       r1, @r6        // asr2, amr2 register write
              st       r2, @r7        // asr3, amr3 register write
              st       r3, @r8        // asr4, amr4 register write
              st       r4, @r9        // asr5, amr5 register write

init_ler      ldi:8    #0x02, r0      // CS2 little endian
              ldi:20  #0x7fe, r1      // ler register address setting
              stb      r0, @r1        // ler register write

init_modr     ldi:8    #0x80, r0      // External ROM external bus
              ldi:20  #0x7ff, r1      // modr register address setting
              stb      r0, @r1        // modr register write

// External bus access

adr_set       ldi:32  #0x00136da0, r0 // CS1 address
              ldi:32  #0x00151300, r1 // CS2 address
              ldi:32  #0x00196434, r2 // CS4 address (within a page)
              ldi:32  #0x0019657c, r3 // CS4 address (within a page)
              ldi:32  #0x00196600, r4 // CS4 address (outside a page)
              ldi:32  #0x001a6818, r5 // CS5 address (within a page)
              ldi:32  #0x001a6b8c, r6 // CS5 address (within a page)
              ldi:32  #0x001a6c00, r7 // CS5 address (outside a page)

bus_acc       ld       @r0, r8        // CS1 data word load
              ldub     @r1, r9        // CS2 data half word load
              ld       @r2, r10       // CS4 data word load
              ldub     @r3, r11       // CS4 data byte load
              st       r8, @r4        // CS4 data word store
              sth      r9, @r5        // CS5 data half word store
              st       r10, @r6       // CS5 data word store
              stb      r11, @r7       // CS5 data byte store

```

CHAPTER 5 I/O PORTS

This chapter outlines the I/O ports and describes the register configuration, and describes the conditions for using external pins as I/O ports.

5.1 "Outline of I/O Ports"

5.2 "Port Data Registers (PDR)"

5.3 "Data Direction Registers (DDR)"

5.4 "Using External Pins as I/O Ports"

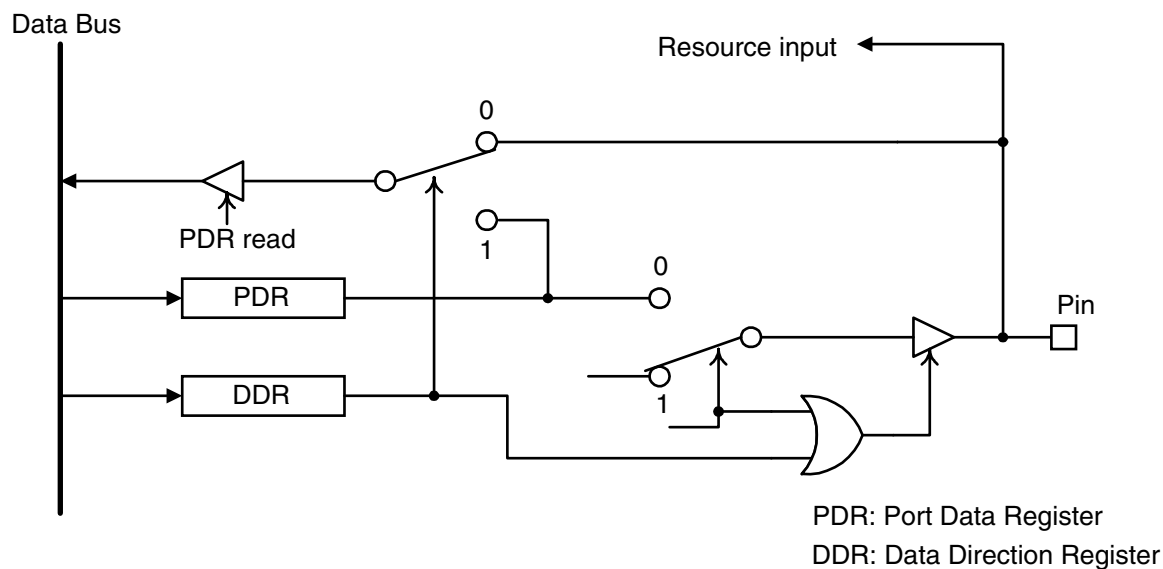
5.1 Outline of I/O Ports

MB91F127/128 can use pins as I/O ports that have not been specified to be used for resource input-output.

■ Basic Block Diagram of I/O Ports

Figure 5.1-1 "Basic Configuration of I/O Ports" shows the basic configuration of the I/O ports.

Figure 5.1-1 Basic Configuration of I/O Ports



■ I/O Port Registers

The I/O ports consist of a port data register (PDR) and data direction register (DDR).

- **During input mode (DDR = "0")**

- At PDR read: The level of the corresponding external pin is read.
- At PDR write: The setting value is written to the PDR.

- **During output mode (DDR = "1")**

- At PDR read: The PDR value is read.
- At PDR write: The PDR setting is output to the corresponding external pin.

5.2 Port Data Registers (PDR)

The port data registers (PDR2 and PDRJ) are I/O data registers of I/O ports. The corresponding data direction registers (DDR2 and DDRJ) are used to control input-output.

■ Port Data Registers (PDR)

Figure 5.2-1 "Register Configuration of PDR" shows the configuration of PDR.

Figure 5.2-1 Register Configuration of PDR

	PDR2 bit	7	6	5	4	3	2	1	0	Initial value
Address	000001 _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 bit	7	6	5	4	3	2	1	0	
Address	000000 _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	
	PDR4 bit	7	6	5	4	3	2	1	0	
Address	000007 _H	P47	P46	P45	P44	P43	P42	P41	P40	XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDR5 bit	7	6	5	4	3	2	1	0	
Address	000006 _H	P57	P56	P55	P54	P53	P52	P51	P50	XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDR6 bit	7	6	5	4	3	2	1	0	
Address	000005 _H	P67	P66	P65	P64	P63	P62	P61	P60	XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDR7 bit	7	6	5	4	3	2	1	0	
Address	000004 _H	-	-	-	-	-	-	-	P70	-----X _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDR8 bit	7	6	5	4	3	2	1	0	
Address	00000B _H	-	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	
	PDRA bit	7	6	5	4	3	2	1	0	
Address	000009 _H	-	PA6	PA5	PA4	PA3	PA2	PA1	PA0	-XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDRE bit	7	6	5	4	3	2	1	0	
Address	000012 _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 bit	7	6	5	4	3	2	1	0	
Address	000013 _H	-	-	-	-	PF3	PF2	PF1	PF0	----XXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDRG bit	7	6	5	4	3	2	1	0	
Address	000014 _H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	PDRJ bit	7	6	5	4	3	2	1	0	
Address	000017 _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	

5.3 Data Direction Registers (DDR)

The data direction registers (DDR2 and DDRJ) are used to control input-output of the corresponding data direction registers in units of bits.

■ Data Direction Registers (DDR)

Figure 5.3-1 "Register Configuration of DDR" shows the register configuration of the DDR.

Figure 5.3-1 Register Configuration of DDR

	DDR bit	7	6	5	4	3	2	1	0	Initial value
Address	000601 _H	P27	P26	P25	P24	P23	P22	P21	P20	00000000 _B
		W	W	W	W	W	W	W	W	
Address	000600 _H	P37	P36	P35	P34	P33	P32	P31	P30	00000000 _B
		W	W	W	W	W	W	W	W	
Address	000607 _H	P47	P46	P45	P44	P43	P42	P41	P40	00000000 _B
		W	W	W	W	W	W	W	W	
Address	000606 _H	P57	P56	P55	P54	P53	P52	P51	P50	00000000 _B
		W	W	W	W	W	W	W	W	
Address	000605 _H	P67	P66	P65	P64	P63	P62	P61	P60	00000000 _B
		W	W	W	W	W	W	W	W	
Address	000604 _H	-	-	-	-	-	-	-	P70	-----0 _B
		W	W	W	W	W	W	W	W	
Address	00060B _H	-	P86	P85	P84	P83	P82	P81	P80	-0000000 _B
		W	W	W	W	W	W	W	W	
Address	000609 _H	-	PA6	PA5	PA4	PA3	PA2	PA1	PA0	-0000000 _B
		W	W	W	W	W	W	W	W	
Address	0000D2 _H	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	00000000 _B
		W	W	W	W	W	W	W	W	
Address	0000D3 _H	-	-	-	-	PF3	PF2	PF1	PF0	----0000 _B
		W	W	W	W	W	W	W	W	
Address	0000D8 _H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	00000000 _B
		W	W	W	W	W	W	W	W	
Address	0000DB _H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	00000000 _B
		W	W	W	W	W	W	W	W	

5.4 Using External Pins as I/O Ports

This section provides the initial values for the external pins and describes their relationship with registers that indicate whether to switch between their use as I/O ports or as control pins.

■ Selecting the Function (I/O Ports or Control Pins) of the External Pins

Table 5.4-1 "Function Selection of External Pins" lists the initial values for the external pins and their relationship with registers that indicate whether to switch between their use as I/O ports or as control pins.

In Table 5.4-1 "Function Selection of External Pins", the "Single chip:" and "External bus:" items indicate that the functions depend on the operating mode, and the "8 bits:" and "16 bits:" items indicate that the functions depend on the external bus width.

Table 5.4-1 Function Selection of External Pins

Pin number	Pin symbol	Initial value (single chip mode)	Switching register
25 to 32	P20 to P27	P20 to P27	Switched automatically based on the mode selected in MD0 to 2, AMD0 to 5, M0, and M1. Single chip: P20 to P27 8 bits: P20 to P27 16 bits: D16 to D23
	D16 to D23		
33 to 39, 41	P30 to P37	P30 to P37	Switched automatically based on the mode selected in MD0 to 2, M0, and M1. Single chip: P30 to P37 External bus: D24 to D31
	D24 to D31		
42, 44 to 58	P40 to P47 P50 to P57	P40 to P47 P50 to P57	Switched automatically based on the mode selected in MD0 to 2, M0, and M1. Single chip: P40 to P47, P50 to P57 External bus: A00 to A15
	A00 to A15		
59 to 64	P60 to P65	P60 to P65	EPCR1 (AE16 to AE21 bits) 0: P60 to P65 1: A16 to A21
	A16 to A21		
66, 67	P66, P67	P66, P67	EPCR1 (AE22, AE23 bits) 0: P66, P67 1: A22, A23 The pin value is always input to IN2 and IN3 (except during stop).
	A22, A23		
	IN2, IN3		
68	P70	P70	EPCR1 (AE24 bit) 0: P70 1: A24 The pin value is always input to FRCK and TC12 (except during stop).
	A24		
	FRCK		
	TC12		

Table 5.4-1 Function Selection of External Pins (Continued)

Pin number	Pin symbol	Initial value (single chip mode)	Switching register
19	P80	P80	EPCR0 (RDYE bit) 0: P80 1: RDY
	RDY		
20	P81	P81	EPCR0 (BRE bit) 0: P81 1: $\overline{\text{BGRNT}}$ The pin value is always input to IN0 (except during stop).
	$\overline{\text{BGRNT}}$		
	IN0		
21	P82	P82	EPCR0 (BRE bit) 0: P82 1: BRQ The pin value is always input to IN1 (except during stop).
	BRQ		
	IN1		
22	P83	P83	Switched automatically based on the mode selected in MD0 to 2, M0, and M1. Single chip: P83 External Bus: $\overline{\text{RD}}$
	$\overline{\text{RD}}$		
23, 24	P84, P85	P84, P85	Switched automatically based on the mode selected in MD0 to 2, AMD0 to 5, M0, and M1. Single chip: P84, P85 8 bits: $\overline{\text{WR0}}$, P85 16 bits: $\overline{\text{WR0}}$, $\overline{\text{WR1}}$
	$\overline{\text{WR0}}$, $\overline{\text{WR1}}$		
11 to 9	PA0 to PA2	PA0 to PA2	EPCR0 (C0E0 to C0E2 bits) 0: PA0 to PA2 1: $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$
	$\overline{\text{CS0}}$ to $\overline{\text{CS2}}$		
8	PA3	PA3	EPCR0 (C0E3 bit) and SMR (S0E bit) C0E3, S0E 00: PA3 10: $\overline{\text{CS3}}$ Other: S01
	$\overline{\text{CS3}}$		
	S01		
7, 6	PA4, PA5	PA4, PA5	EPCR0 (C0E4, C0E5 bits) 0: PA4, PA5 1: $\overline{\text{CS4}}$, $\overline{\text{CS5}}$ The pin value is always input to SI1 and SC1 (except during stop).
	$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		
	SI1, SC1		
5	PA6	PA6	EPCR0 (C0E6 bit) 0: PA6 1: CLK
	CLK		
96 to 99	PG0 to PG3	PG0 to PG3	PCNL (P0EN bit) 0: PG0 to PG3 1: OCPA0 to OCPA3
	OCPA0 to OCPA3		
100, 1 to 3	PG4 to PG7	PG4 to PG7	OCS0 to 3 (0TE bit) 0: PG4 to PG7 1: OC0 to OC3
	OC0 to OC3		
16 to 18	MD0 to MD2	MD0 to MD2	-

Table 5.4-1 Function Selection of External Pins (Continued)

Pin number	Pin symbol	Initial value (single chip mode)	Switching register
12	P86	P86	EPCR0 (ALEE bit) 0: P86 1: ALE
	ALE		
72	PJ0 to PJ7	PJ0 to PJ7	AIC (AI bit), DDRJ (PJ bit) AI, PJ 00: AN Other: PJ
	AN0 to AN7		
95 to 94	PE0, PE1	PE0/INT0, PE1/INT1	The pin value is always input to INT0 and INT1.
	INT0, INT1		
89, 88	PE2, PE3	PE2/INT2, PE3/INT3	The pin value is always input to INT2 and INT3.
	INT2, INT3		
87	PE4	PE4/INT4/TCI1	The pin value is always input to INT4 and TCI1 (except during stop).
	INT4		
	TCI1		
86	PE5	PE5/INT5/SC0	The pin value is always input to INT5 and SC0 (except during stop).
	INT5		
	SC0		
85	PE6	PE6/SI0	The pin value is always input to SI0 (except during stop).
	SI0		
84	PE7	PE7	SMR (SOE bit) 0: PE7 1: SO0
	SO0		
80	PF0	PF0/TCI0	The pin value is always input to TCI0 (except during stop).
	TCI0		
81	PF1	PF1/SI2	The pin value is always input to SI2 (except during stop).
	SI2		
82	PF2	PF2	SMR (SOE bit) 0: PF2 1: SO2
	SO2		
83	PF3	PF3/SC2/ $\overline{\text{ATG}}$	The pin value is always input to SC2 and $\overline{\text{ATG}}$ (except during stop).
	SC2		
	$\overline{\text{ATG}}$		
69	AV _{CC}	AV _{CC}	-
70	AVRH	AVRH	-
71	AV _{SS} (AVRL)	AV _{SS} (AVRL)	-

Table 5.4-1 Function Selection of External Pins (Continued)

Pin number	Pin symbol	Initial value (single chip mode)	Switching register
14	$\overline{\text{RST}}$	$\overline{\text{RST}}$	-
13	$\overline{\text{HST}}$	$\overline{\text{HST}}$	-
92	X0	X0	-
91	X1	X1	-
4, 93, 43	V _{CC}	V _{CC}	-
15, 40, 65, 90	V _{SS}	V _{SS}	-

CHAPTER 6 16-BIT RELOAD TIMER

This chapter outlines the 16-bit reload timer and describes the register configuration/ functions and operations of the 16-bit reload timer.

6.1 "Outline of the 16-Bit Reload Timer"

6.2 "16-Bit Reload Timer Registers"

6.3 "16-Bit Reload Timer Operations"

6.4 "Counter Operation States"

6.1 Outline of the 16-Bit Reload Timer

The 16-bit reload timer is configured from a 16-bit down counter, 16-bit reload register, a prescaler unit for generating internal count clock pulses, and control register.

■ Outline of the 16-bit Reload Timer

Three types of internal clocks (machine clock divided by 2/8/32) can be selected as the input clock.

DMA transfer can be activated using an interrupt.

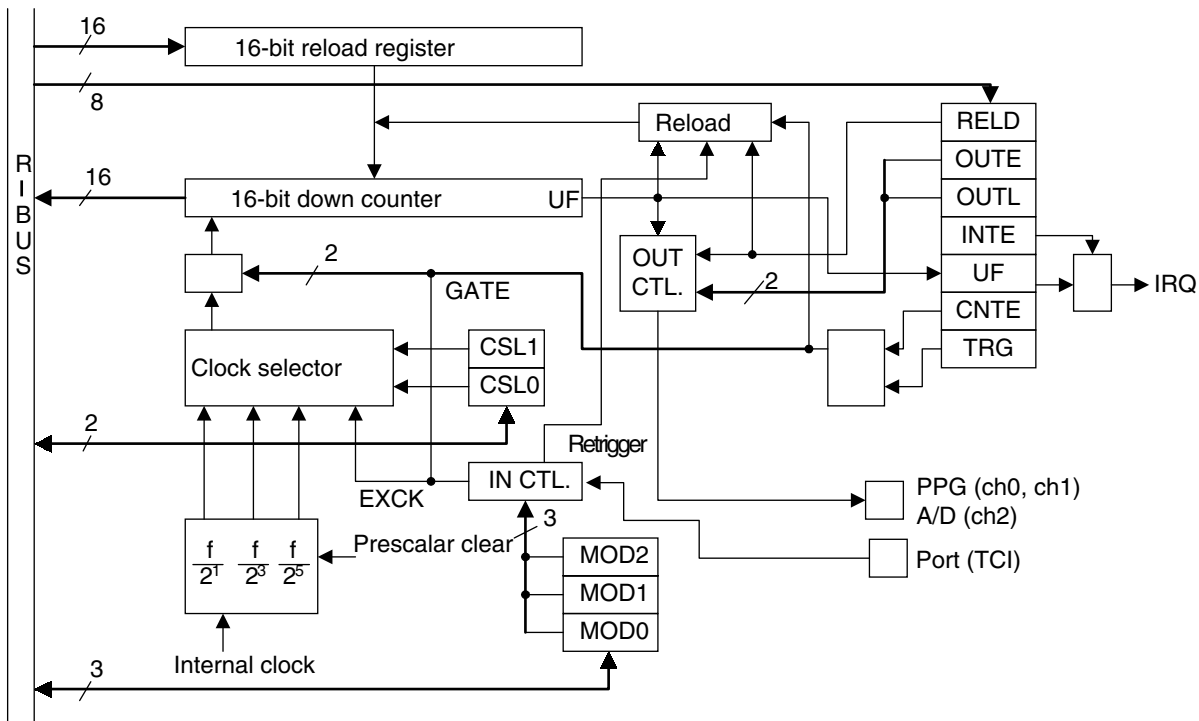
The MB91F127/128 has three channels built in for this timer.

The TO output of channel 2 of the reload timer is connected to the A/D converter internally in the LSI MODULE. As a result, A/D conversion can be activated using the cycle set in the reload register.

■ Block Diagram

Figure 6.1-1 "Block Diagram of the 16-bit Reload Timer" shows a block diagram of the 16-bit reload timer.

Figure 6.1-1 Block Diagram of the 16-bit Reload Timer

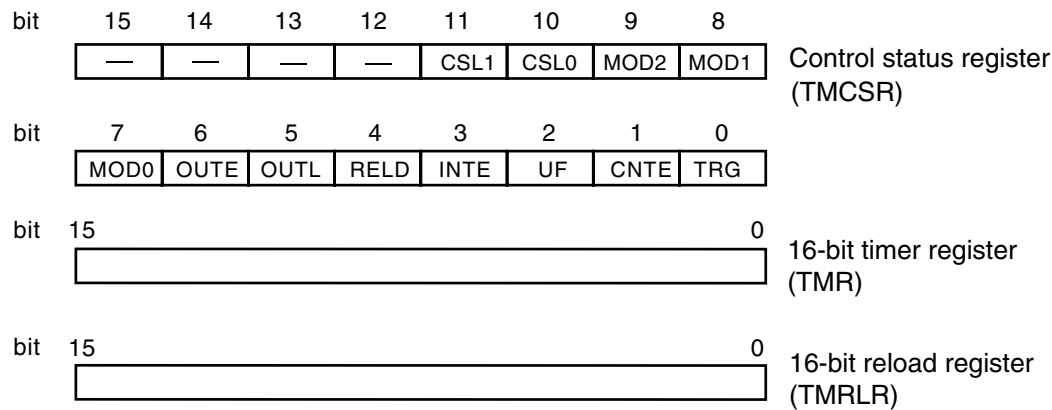


6.2 16-Bit Reload Timer Registers

This section explains the configuration and functions of the registers to be used in the 16-bit reload timer.

■ 16-Bit Reload Timer Registers

Figure 6.2-1 16-bit Reload Timer Registers

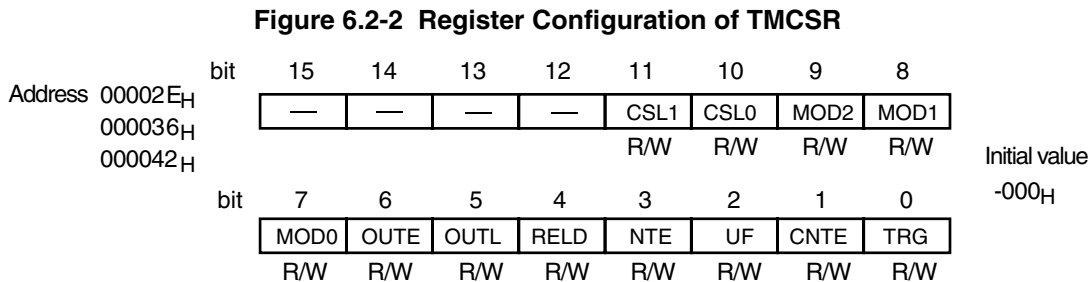


6.2.1 Control Status Register (TMCSR)

The control status register (TMCSR) controls the operating mode and interrupts of the 16-bit timer.

■ TMCSR Configuration

Figure 6.2-2 "Register Configuration of TMCSR" shows the register configuration of the TMCSR.



With the exception of the UF, CNTE, and TRG bits, ensure through the settings that all bits are rewritten when CNTE = 0.

Simultaneous writing is enabled.

■ Bit Functions of TMCSR

The functions of bits of the TMCSR are explained below.

[Bit 11 and 10] CSL1 and CSL0 (Count clock Select)

These bits are the count clock select bits.

Table 6.2-1 "CSL Bit Setting Clock Sources" lists the clock sources that can be selected.

Table 6.2-1 CSL Bit Setting Clock Sources

CSL1	CSL0	Clock source (f : Machine clock)
0	0	$f/2^1$
0	1	$f/2^3$
1	0	$f/2^5$
1	1	External clock

[Bits 9, 8, and 7] MOD2, MOD1, and MOD0 (MODE)

These bits specify the operating mode and the function of the input-output pins.

The MOD2 bit is used to select the function of the input pin.

If this bit is set to "0", the input pin becomes a trigger input pin. When a valid edge is input, the contents of the reload register are loaded into the counter, and the count operation continues.

If this bit is set to "1", the gate count mode is set and the input pin becomes a gate input pin. The counting continues only while a valid level is input.

The MOD1 and MOD0 bits specify the pin function in each mode.

Table 6.2-2 "Settings of MOD2, 1, and 0 Bits (1) Internal clock mode (CSL0, 1 = 00, 01, or 10)" and Table 6.2-3 "Settings of MOD2, 1, and 0 Bits (2) Event count mode (CSL0, 1 = 11)" list the settings of MOD2, 1, and 0.

Table 6.2-2 Settings of MOD2, 1, and 0 Bits (1) Internal clock mode (CSL0, 1 = 00, 01, or 10)

MOD2	MOD1	MOD0	Input pin function	Valid edge, level
0	0	0	Trigger prohibited	-
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edges
1	×	0	Gate input	L level
1	×	1		H level

Table 6.2-3 Settings of MOD2, 1, and 0 Bits (2) Event count mode (CSL0, 1 = 11)

MOD2	MOD1	MOD0	Input pin function	Valid edge, level
×	0	0	Event input	-
	0	1		Rising edge
	1	0		Falling edge
	1	1		Both edges

[Bit 6] OUTE (OUTput Enable)

Always set this bit to "0".

[Bit 5] OUTL

Always set this bit to "0".

[Bit 4] RELD

This bit is the reload enable bit. Reload mode is set when this bit is "1". The count operation is continued by loading the reload register contents to the counter at the same time that the counter value underflows from 0000_H to FFFF_H. When this bit is "0", the count operation is stopped when the counter value underflows from 0000_H to FFFF_H.

[Bit 3] INTE

This bit is used to enable interrupt requests. When this bit is "1", setting the UF bit to "1" will generate an interrupt request. When the INTE bit is "0", no interrupt requests will be generated.

[Bit 2] UF

This bit is the timer interrupt request flag. This bit is set to "1" when the counter value underflows from 0000_H to FFFF_H. Writing 0 to this bit clears the bit.

Writing "1" to this bit has no effect.

When reading with read-modify-write instructions, "1" will always be returned for this bit.

[Bit 1] CNTE

This bit is the timer count enable bit. Setting this bit to "1" sets activation trigger wait state. Setting it to "0" stops the count operation.

[Bit 0] TRG

This bit is the software trigger bit. Setting this bit to "1" activates the software trigger, loads the reload register contents to the counter, and starts the count operation.

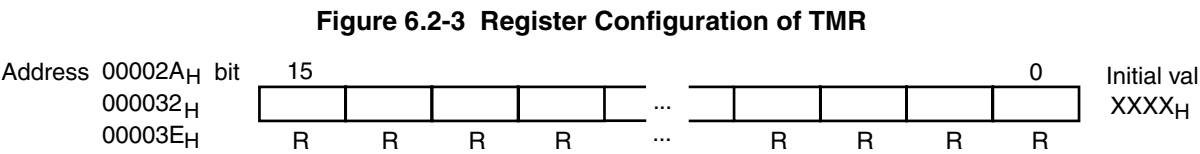
Writing "0" to this bit has no effect. During reading operations, zero is always returned for this bit. Trigger input using this register is enabled only when CNTE = "1". No operation is performed when CNTE = "0".

6.2.2 16-Bit Timer Register (TMR)

The 16-bit timer register (TMR) can be used to read the count value of the 16-bit timer. The initial value in this register is undefined. Always use a 16-bit data transfer instruction to read this register.

■ TMR Configuration

Figure 6.2-3 "Register Configuration of TMR" shows the register configuration of the TMR.

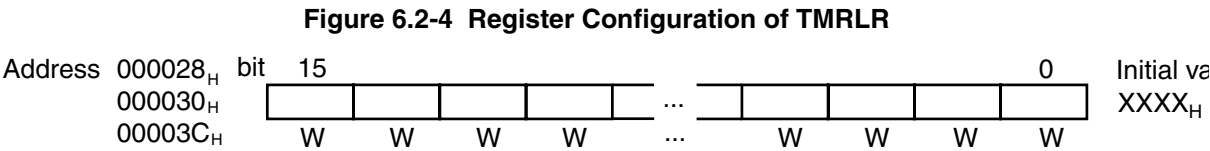


6.2.3 16-Bit Reload Register (TMRLR)

The 16-bit reload register (TMRLR) is used to save the initial count value. The initial value of this register is undefined. Always use a 16-bit data transfer instruction to write to this register.

■ TMRLR Configuration

Figure 6.2-4 "Register Configuration of TMRLR" shows the register configuration of the TMRLR.



6.3 16-Bit Reload Timer Operations

This section explains the operations of the 16-bit reload timer.

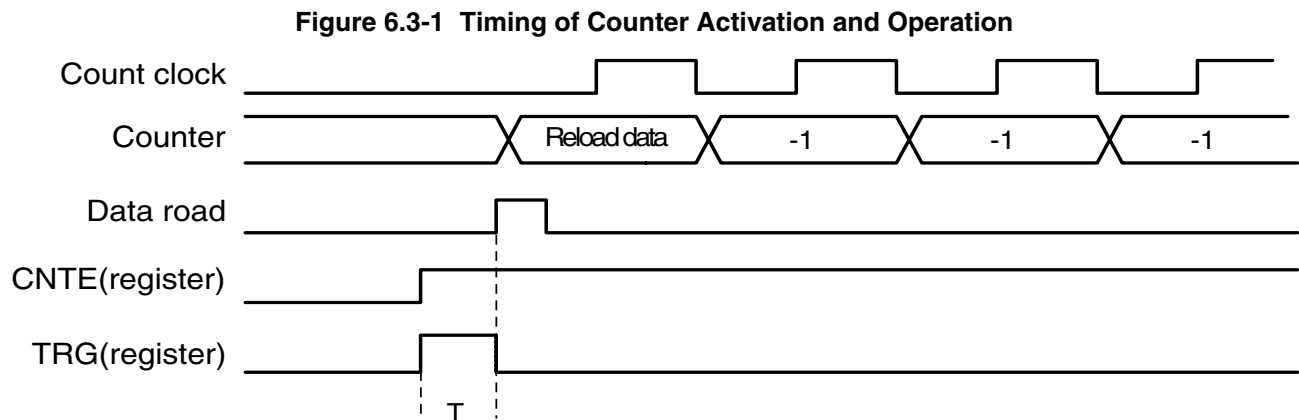
■ Internal Clock Operation

To operate the timer with a fractional frequency of the internal clock, a clock frequency of 1/2, 1/8, and 1/32 of the machine clock can be selected as the count source.

To enable counting and starting the count operation, set the CNTE and TRG bits of the control status register to "1". When the timer is in active state (CNTE = "1"), trigger input using the TRG bit is enabled regardless of the operating mode.

Figure 6.3-1 "Timing of Counter Activation and Operation" shows the timing for activation and operation of the counter.

The time T (T: Peripheral clock machine cycle) will be required from when the count start trigger is input to when TMRLR is loaded to the counter.



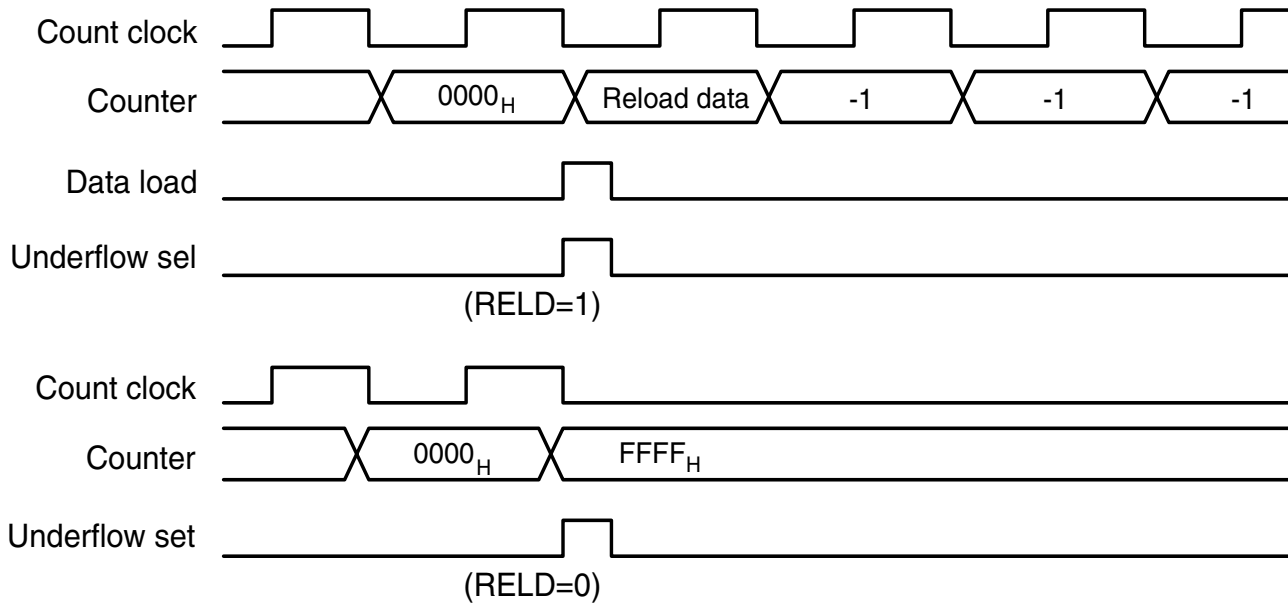
■ Underflow Operation

Underflow occurs when the counter value changes from 0000_H to FFFF_H. In other words, underflow occurs at the count of (reload register setting value + 1).

When underflow occurs, TMRLR contents are loaded to the counter and the count operation continues if the RELD bit TMCSR is "1". If the RELD bit is "0", the count operation will stop at FFFF_H.

When TMCSR UF bit is set by an underflow, an interrupt request will be generated if the INTE bit is "1".

Figure 6.3-2 "Timings of Underflow Operations" shows the underflow-related chip operation.

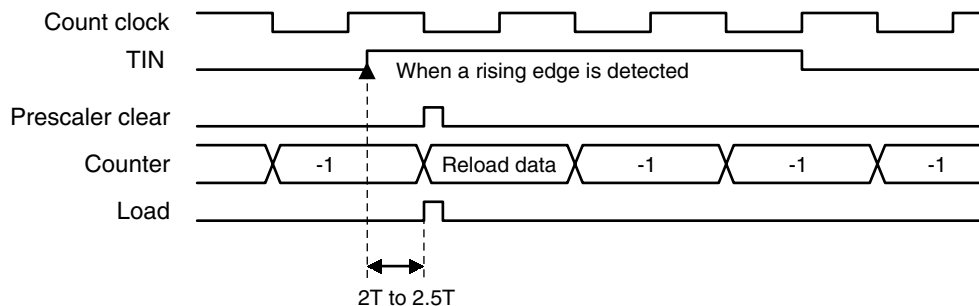
Figure 6.3-2 Timings of Underflow Operations

■ Input Pin Function

If the internal clock is selected as the clock source, the TCI pin can be used either as a trigger input or gate input pin. In trigger input mode, when a valid edge is input, the contents of the 16-bit reload register (TMRLR) are loaded into the counter, the internal prescaler is cleared, and then the count operation is started.

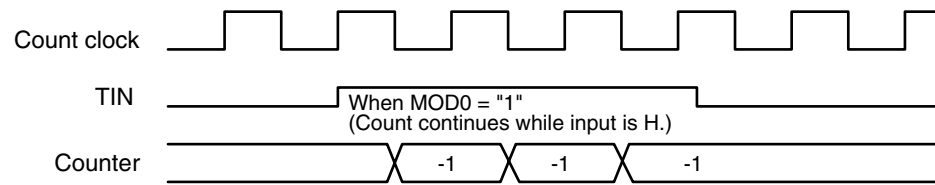
To TCI, input a pulse longer than $2 \times T$ (where T is the peripheral clock machine cycle).

Figure 6.3-3 "Trigger input operation" shows the trigger input operation.

Figure 6.3-3 Trigger input operation

In gate input mode, the counting continues only while a valid level specified by the MOD0 bit of the control status register (TMCSR) is input from the TCI pin. At this time, the count clock continues to run without stopping. In gate mode, a software trigger can be used regardless of the gate level. The pulse width of the TCI pin must be longer than $2 \times T$ (where T is the peripheral clock machine cycle).

Figure 6.3-4 "Gate input operation" shows the gate input operation.

Figure 6.3-4 Gate input operation

■ External Event Count

If the external clock is selected, the TCI pin becomes an external event input pin, which is used to count valid edges specified by the register. The pulse width of the TCI pin must be longer than $2 \times T$ (where T is the peripheral clock machine cycle).

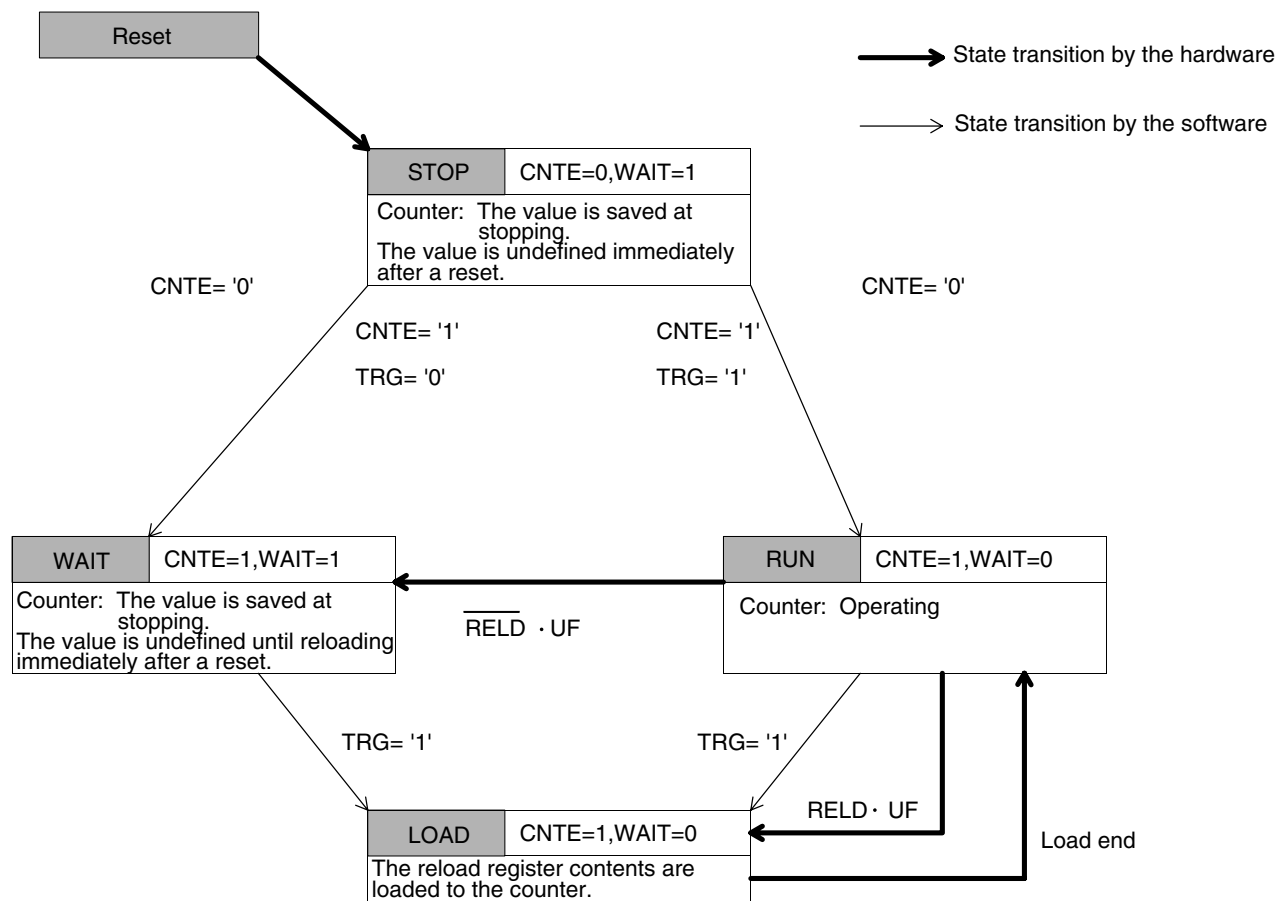
6.4 Counter Operation States

The counter operation states are determined based on TMCSR CNTE bit and WAIT signal of the internal signals. The states that can be set are CNTE = "0" and stop state of WAIT = "1" (STOP state), CNTE = "1" and activation trigger wait state of WAIT = "1" (WAIT state), and CNTE = "1" and operating state of WAIT = "0" (RUN state).

■ Counter Operation States

Figure 6.4-1 "Counter State Transition" shows the state transition.

Figure 6.4-1 Counter Sstate Transition



CHAPTER 7 PROGRAMMABLE PULSE GENERATOR (PPG) TIMER

This chapter gives an outline of the PPG timer and explains the register configuration and functions and the timer operations.

- 7.1 "Outline of the PPG Timer"
- 7.2 "Block Diagram of the PPG Timer"
- 7.3 "Registers of the PPG Timer"
- 7.4 "PPG Mode"
- 7.5 "One-shot Mode"
- 7.6 "Interrupts"
- 7.7 "PPG Output of All-L and All-H"
- 7.8 "Activation of Multiple Channels of PPG Timer"

7.1 Outline of the PPG Timer

**The PPG timer can efficiently output highly accurate PPG waveforms.
The MB91F127 and 128 have four channels of the PPG timer.**

■ Characteristics of PPG Timer

- Each channel consists of a 16-bit down counter, 16-bit data register with a cycle setting buffer, 16-bit compare register with a duty setting buffer, and pin control block.
- One of the four count clocks can be selected for the 16-bit down counter:
 - Internal clocks: ϕ , $\phi/4$, $\phi/16$, and $\phi/64$
- A reset or counter borrow can initialize the counter value to "FFFF_H".
- Each channel has PPG output.
- Outline of registers
 - Cycle setting register: Data register for reload with buffer
 - Duty setting register: Compare register with buffer
 - Transfers from the buffer are performed with counter borrow.
- Pin control
 - Set to "1" when the duty matches (priority)
 - Reset to "0" when a counter borrow occurs.
 - Output-value fixed mode is available to facilitate output of all-L (or H).
 - The polarity can be specified.
- An interrupt request can be generated as one of the following combinations:
 - Activation of PPG timer
 - Generation of counter borrow (cycle match)
 - Generation of duty match
 - Generation of counter borrow (cycle match) or duty match

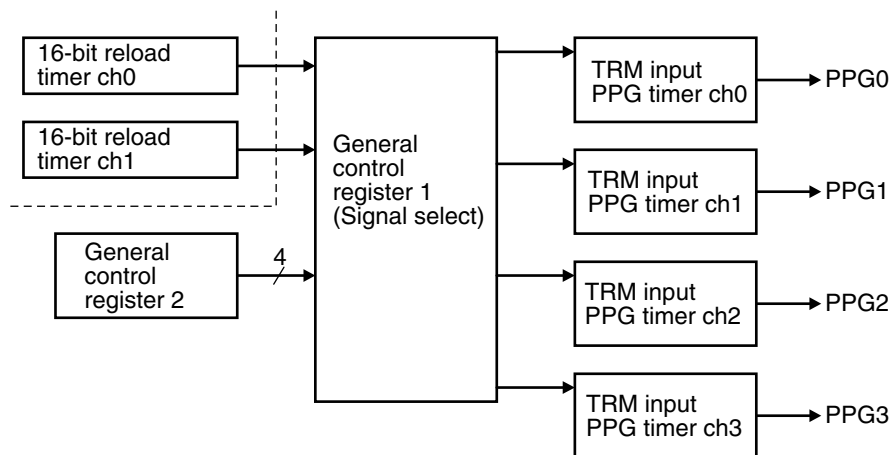
Either of the above interrupt requests can start a DMA transfer.
- Multiple channels can be activated at one time by using software or other interval timers.
Restart during operation can also be set.

7.2 Block Diagram of the PPG Timer

Figure 7.2-1 "Overall Block Diagram of PPG Timer" shows an overall block diagram of the PPG timer. Figure 7.2-2 "Block Diagram for One Channel of PPG Timer" shows the block diagram for one channel of the PPG timer.

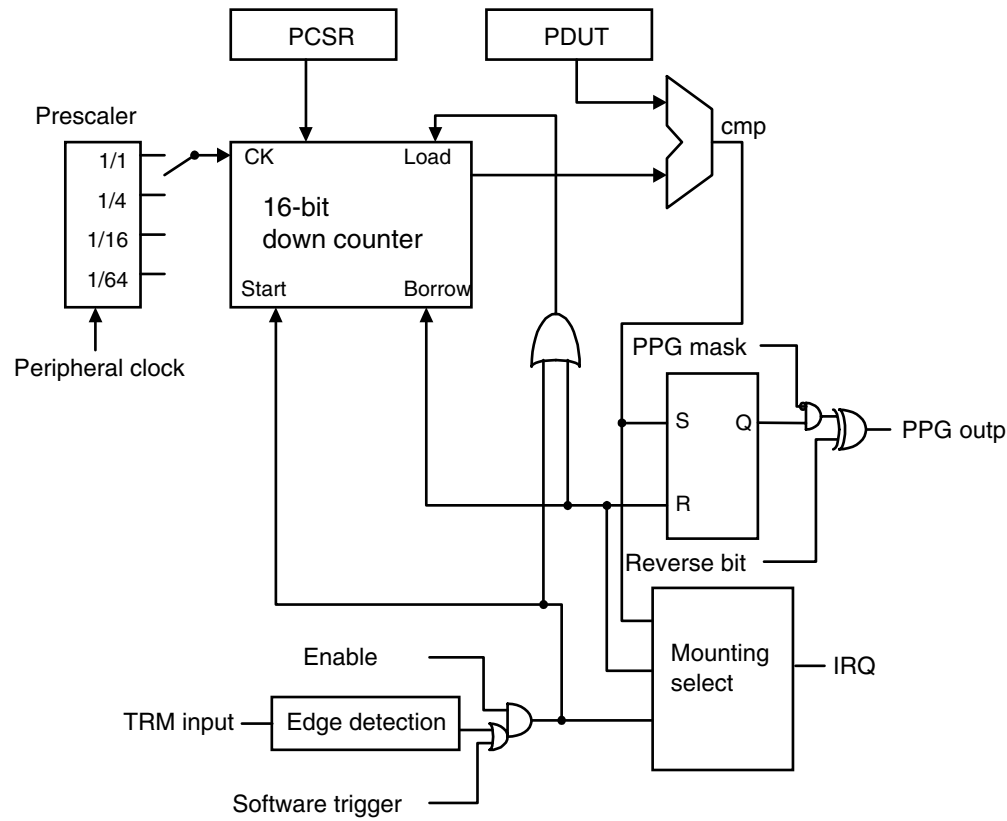
■ Overall Block Diagram of PPG Timer

Figure 7.2-1 Overall Block Diagram of PPG Timer



■ Block Diagram for One Channel of PPG Timer

Figure 7.2-2 Block Diagram for One Channel of PPG Timer

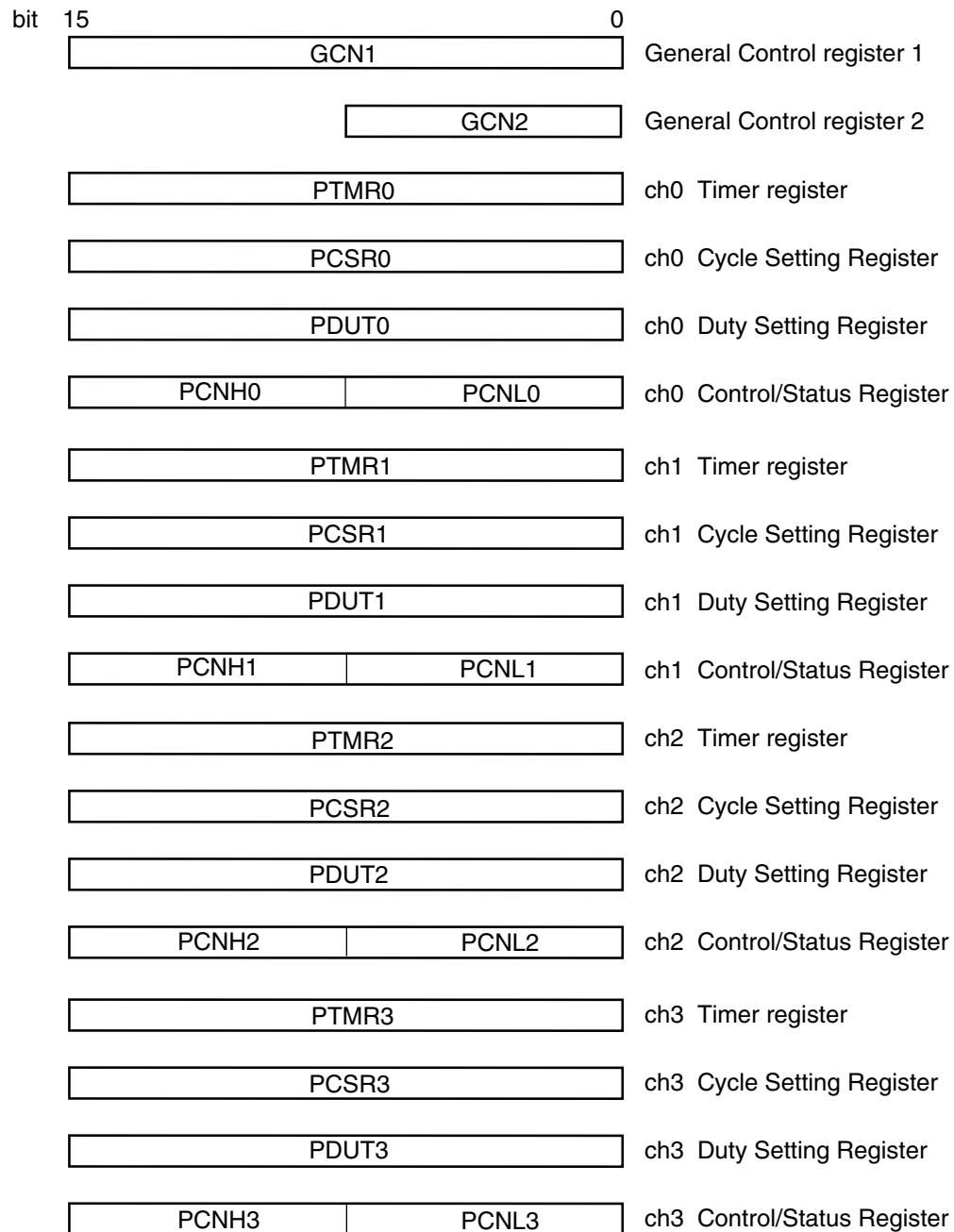


7.3 Registers of the PPG Timer

Figure 7.3-1 "Registers of the PPG Timer" shows the registers of the PPG timer.

■ Registers of the PPG Timer

Figure 7.3-1 Registers of the PPG Timer



7.3.1 Control Status Register (PCNH, PCNL)

The control status registers (PCNH and PCNL) are used to control and display the status of the PPG timer. Note that certain bits cannot be rewritten while the PPG timer is running.

■ Register Configurations of Control Status Registers (PCNH and PCNL)

Figure 7.3-2 "Register Configurations of PCNH and PCNL" shows the register configuration of the control status registers (PCNH and PCNL).

Figure 7.3-2 Register Configurations of PCNH and PCNL

PCNH		bit	15	14	13	12	11	10	9	8	Initial value 00000000-00000000 _B ← Rewrite during operation
Address	ch0 0000E6 _H		CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	—	
	ch1 0000EE _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	
	ch2 0000F6 _H		○	○	×	×	×	×	○	—	
	ch3 0000FE _H										
PCNL		bit	7	6	5	4	3	2	1	0	← Rewrite during operation
Address	ch0 0000E7 _H		EGS1	EGS0	IREN	IRQF	IRS1	RS0	POEN	OSEL	
	ch1 0000EF _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	ch2 0000F7 _H		×	×	○	○	×	×	×	×	
	ch3 0000FF _H										

■ Bit Functions of PCNH and PCNL

The functions of bits of the PCNH and PCNL are explained below.

Bit 15: CNTE (Timer Enable)

This bit enables operation of the 16-bit down counter.

CNTE	Function
0	Disabled (initial value)
1	Enabled

Bit 14: STGR (Software Trigger)

Writing 1 into this bit applies software trigger.

The read value is always 0.

Bit 13: MDSE (Mode Select)

This bit is used to select either the PPG mode in which continuous pulses are output or the

one-shot mode in which a single pulse is output.

MDSE	Function
0	PPG mode (initial value)
1	One-shot mode

Bit 12: RTRG (Retrigger Select) (only ch0 to ch2 valid)

This bit enables a restart due to a software trigger.

RTRG	Function
0	Restart disabled (initial value)
1	Restart enabled

Bits 11 and 10: CKS1 and CKS0 (Counter Clock Select)

These bits are used to select the count clock of the 16-bit down counter.

CKS1	CKS0	Cycle
0	0	$\phi/2$ (initial value)
0	1	$\phi/4$
1	0	$\phi/16$
1	1	$\phi/64$

ϕ : Machine clock

Bit 9: PGMS (PPG Output Mask Select)

Writing 0 into this bit allows PPG output to be masked to 0 or 1, regardless of mode, cycle, and duty settings.

Polarity	PPG output
Ordinary polarity	L output
Reverse polarity	H output

For all-H output in ordinary polarity mode and all-L output in reverse polarity mode, specify the same value in the cycle setting register and duty setting register in order to output the above mask value with the polarity reversed.

[Bit 8] (reserved)

This bit is reserved.

Bits 7 and 6: EGS1 and EGS0 (Trigger Input Edge Select Bit)

These bits are used to select an effective edge for the activation cause selected in general control register 1.

Regardless of the mode that is selected, writing "1" to the bit of a software trigger enables the software trigger.

EGS1	EGS0	Edge selection
0	0	Not effective
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Bit 5: IREN (PPG Interrupt Request Enable)

This bit enables an interrupt request.

IREN	Function
0	Disabled (initial value)
1	Enabled

Bit 4: IRQF (PPG Interrupt Request Flag)

If bit 5, the IREN bit, is enabled and an interrupt source selected in bits 3 and 2, the IRS1 and IRS0 bits, occurs then this bit is set and an interrupt request is generated and issued to the CPU. If DMA transfer start is selected, a DMA transfer is started.

This bit is cleared if "0" is written to it and a clear signal from DMAC is received.

This bit remains unchanged if "1" is written to it.

The read value by a read-modify-write instruction is always 1, regardless of the bit value.

Bits 3 and 2: IRS1, IRS0 (Interrupt Resource Select)

These bits are used to select a source that sets bit 4, the IRQF bit.

IRS1	IRS0	Interrupt resource
0	0	Input of a software trigger (initial value)
0	1	Occurrence of a counter borrow (cycle match)
1	0	Occurrence of a duty match
1	1	Occurrence of a counter borrow (cycle match) or duty match

[Bit 1] POEN: PPG Output Enable Bit

By setting this bit to 1, PPG outputs from the output pin.

POEN	Function
0	General-purpose I/O port (initial value)
1	PPG output pin

[Bit 0] OSEL: PPG Output Polarity Specification Bit

This bit sets the polarity of the PPG output.

Table 7.3-1 "Combination of PPG Output Polarity Specifications" shows the PPG output polarity that is determined based on the combination of this bit and bit 9, the PGMS bit.

Table 7.3-1 Combination of PPG Output Polarity Specifications

PGMS	OSEL	PPG output
0	0	Ordinary polarity (initial value)
0	1	Reverse polarity
1	0	Output fixed to L
1	1	Output fixed to H

Table 7.3-2 PPG Output Polarity Specification

Polarity	After reset	Duty match	Counter borrow
Ordinary polarity	L output	Rising edge	Rising edge
Reverse polarity	H output	Falling edge	Falling edge

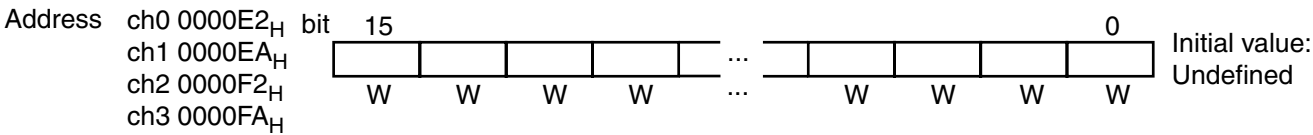
7.3.2 PPG Cycle Setting Register (PCSR)

The PPG cycle setting register (PCSR) is a register with a buffer for setting a cycle. Transfers from the buffer are performed with counter borrow.

■ Bit Configuration of PPG Cycle Setting Register (PCSR)

Figure 7.3-3 "Bit Configuration of PPG Cycle Setting Register (PCSR)" shows the bit configuration of the PPG cycle setting register (PCSR).

Figure 7.3-3 Bit Configuration of PPG Cycle Setting Register (PCSR)



When initializing or rewriting the cycle setting register, be sure to write to the duty setting register after the writing of the cycle setting register.

This register must be accessed using 16-bit data.

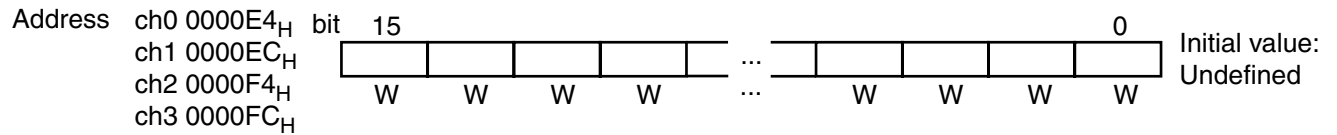
7.3.3 PPG Duty Setting Register (PDUT)

The PPG duty setting register (PDUT) is a register with buffer for setting a duty. Transfers from the buffer are performed with counter borrow.

■ Bit Configuration of PPG Duty Setting Register (PDUT)

Figure 7.3-4 "Bit Configuration of PPG Duty Setting Register (PDUT)" shows the bit configuration of the PPG duty setting register (PDUT).

Figure 7.3-4 Bit Configuration of PPG Duty Setting Register (PDUT)



When the same value is set in the cycle setting register and the duty setting register, all-H is output in ordinary polarity mode and all-L is output in reverse polarity mode.

Do not specify a smaller value in PCSR than that in PDUT. Otherwise, PPG output becomes undefined.

This register must be accessed using 16-bit data.

■ Bit Configuration of PPG Timer Register (PTMR)

Figure 7.3-5 Bit Configuration of PPG Timer Register (PTMR)



7.3.5 General Control Register 1 (GCN1)

General control register 1 (GCN1) selects a trigger input cause for the PPG timer.

■ Bit Configuration of General Control Register 1 (GCN1)

Figure 7.3-6 "Bit Configuration of General Control Register 1 (GCN1)" shows the bit configuration of general control register 1 (GCN1).

Figure 7.3-6 Bit Configuration of General Control Register 1 (GCN1)

Address 0000DC _H	bit 15	14	13	12	11	10	9	8	Initial value
	TSEL33 to 30				TSEL23 to 20				00110010
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00010000 _B
	bit 7	6	5	4	3	2	1	0	
	TSEL13 to 10				TSEL03 to 00				
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Details of General Control Register 1 (GCN1)

[Bits 15 to 12] TSEL33-30: ch3 trigger input select bits

These bits are the ch3 trigger input select bits.

TSEL 33 to 30				ch3 trigger input
15	14	13	12	
0	0	0	0	EN0 bit of GCN2
0	0	0	1	EN1 bit of GCN2
0	0	1	0	EN2 bit of GCN2
0	0	1	1	EN3 bit of GCN2 (initial value)
0	1	0	0	16-bit reload timer ch0
0	1	0	1	16-bit reload timer ch1
0	1	1	×	Setting disabled
1	0	0	0	Setting disabled
1	0	0	1	Setting disabled
1	0	1	0	Setting disabled
1	0	1	1	Setting disabled
1	1	×	×	Setting disabled

CHAPTER 7 PROGRAMMABLE PULSE GENERATOR (PPG) TIMER

[Bits 11 to 8] TSEL23-20: ch2 trigger input select bits

These bits are the ch2 trigger input select bits.

TSEL 23 to 20				ch2 trigger input
11	10	9	8	
0	0	0	0	EN0 bit of GCN2
0	0	0	1	EN1 bit of GCN2
0	0	1	0	EN2 bit of GCN2 (initial value)
0	0	1	1	EN3 bit of GCN2
0	1	0	0	16-bit reload timer ch0
0	1	0	1	16-bit reload timer ch1
0	1	1	×	Setting disabled
1	0	0	0	Setting disabled
1	0	0	1	Setting disabled
1	0	1	0	Setting disabled
1	0	1	1	Setting disabled
1	1	×	×	Setting disabled

[Bits 7 to 4] TSEL13-10: ch1 trigger input select bits

These bits are the ch1 trigger input select bits.

TSEL 13 to 10				ch1 trigger input
7	6	5	4	
0	0	0	0	EN0 bit of GCN2
0	0	0	1	EN1 bit of GCN2 (initial value)
0	0	1	0	EN2 bit of GCN2
0	0	1	1	EN3 bit of GCN2
0	1	0	0	16-bit reload timer ch0
0	1	0	1	16-bit reload timer ch1
0	1	1	×	Setting disabled
1	0	0	0	Setting disabled
1	0	0	1	Setting disabled
1	0	1	0	Setting disabled
1	0	1	1	Setting disabled
1	1	×	×	Setting disabled

[Bits 3 to 0] TSEL03-00: ch0 trigger input select bits

These bits are the ch0 trigger input select bits.

TSEL 00 to 03				ch0 trigger input
3	4	1	0	
0	0	0	0	EN0 bit of GCN2 (initial value)
0	0	0	1	EN1 bit of GCN2
0	0	1	0	EN2 bit of GCN2
0	0	1	1	EN3 bit of GCN2
0	1	0	0	16-bit reload timer ch0
0	1	0	1	16-bit reload timer ch0
0	1	1	×	Setting disabled
1	0	0	0	Setting disabled
1	0	0	1	Setting disabled
1	0	1	0	Setting disabled
1	0	1	1	Setting disabled
1	1	×	×	Setting disabled

7.3.6 General Control Register 2 (GCN2)

General control register 2 (GCN2) is used to generate an activation trigger using software.

■ Bit Configuration of General Control Register 2 (GCN2)

Figure 7.3-7 "Bit Configuration of General Control Register 2 (GCN2)" shows the bit configuration of general control register 2 (GCN2).

Figure 7.3-7 Bit Configuration of General Control Register 2 (GCN2)

	bit	7	6	5	4	3	2	1	0	Initial value
Address	0000DF _H	—	—	—	—	EN3	EN2	EN1	EN0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

If the EN bit of this register is selected in general control register 1 (GCN1), the value of the register is conveyed without change to the trigger input of the PPG timer.

If software generates an edge selected in bits EGS1 and EGS0 of the control status register (PCN), multiple channels of the PPG timer can be activated simultaneously.

Be sure to write 0 to bits 7 to 4 of this register.

7.4 PPG Mode

In PPG mode, continuous pulses are output.

■ PPG Mode

In PPG mode, the PPG timer can output pulses continuously after a trigger signal is detected.

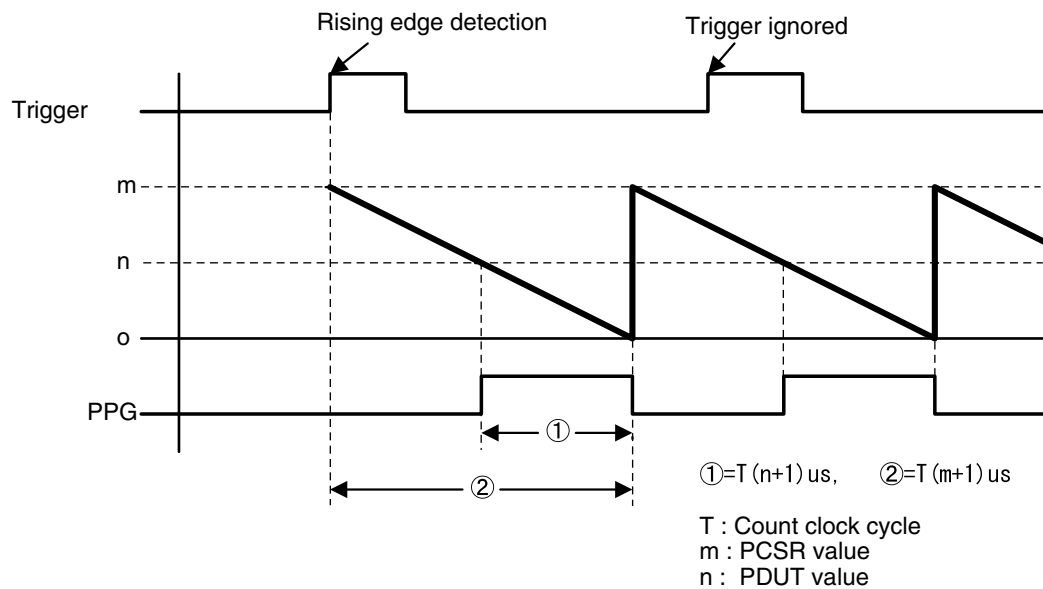
The output pulse cycle can be controlled with the PCSR value and the duty ratio with the PDUT value.

After data is written to PCSR, be sure to write to PDUT.

Figure 7.4-1 "PPG Mode Timing Chart (Trigger Reactivation Disabled)" shows the PPG mode timing chart when trigger reactivation is disabled. Figure 7.4-2 "PWM Mode Timing Chart (Retrigger enabled)" shows the PPG mode timing chart when trigger reactivation is enabled.

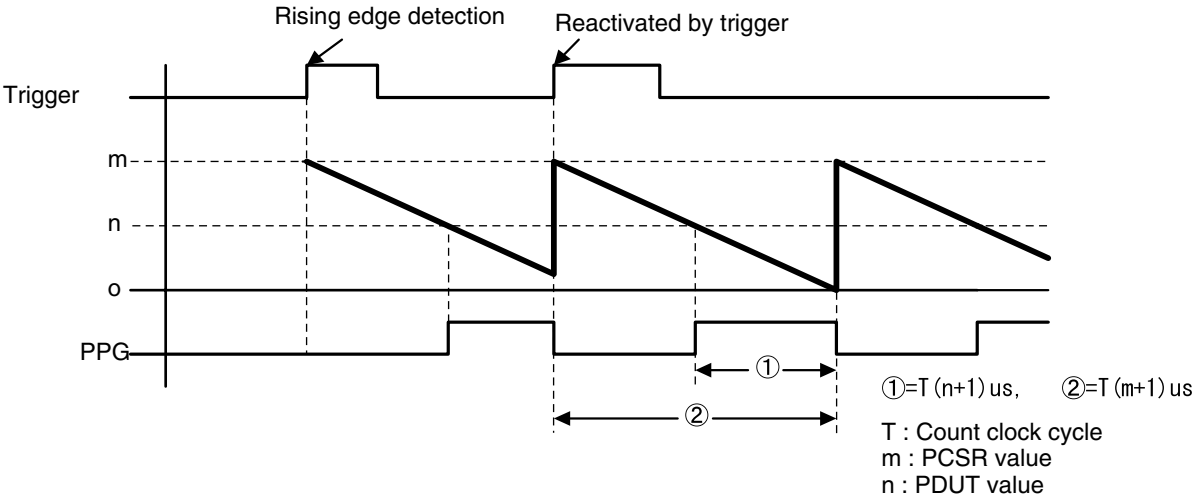
○ When reactivation is disabled

Figure 7.4-1 PPG Mode Timing Chart (Trigger Reactivation Disabled)



○ When reactivation is enabled

Figure 7.4-2 PWM Mode Timing Chart (Retrigger enabled)



7.5 One-shot Mode

In one-shot mode, a single pulse is output.

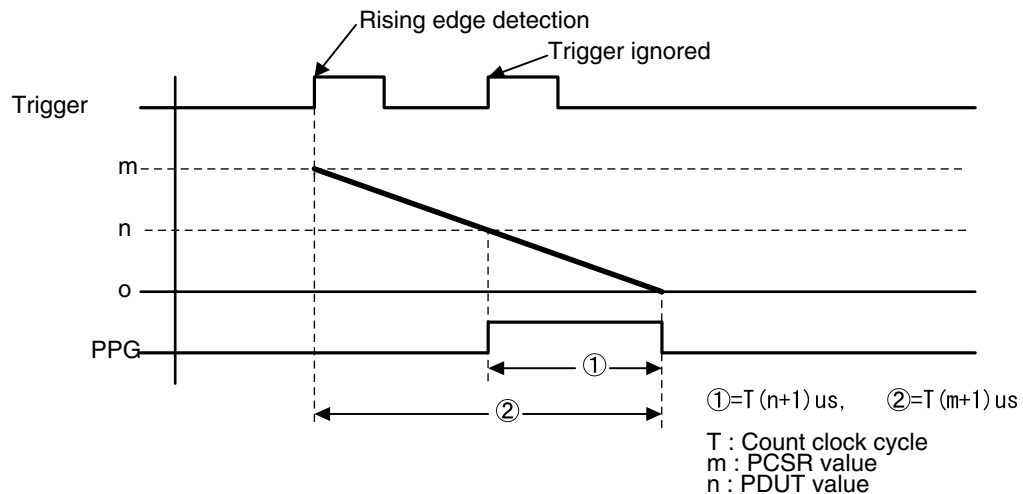
■ One-shot Mode

In one-shot mode, the PPG timer can output a single pulse of an arbitrary width when triggered. When reactivation is enabled, the PPG timer reloads the counter value after an edge is detected during operation.

Figure 7.5-1 "One-shot Mode Timing Chart(Trigger Reactivation Disabled)" shows the one-shot mode timing chart when trigger reactivation is disabled. Figure 7.5-2 "One-shot Mode Timing Chart (Trigger Reactivation Enabled)" shows the one-shot mode timing chart when trigger reactivation is enabled.

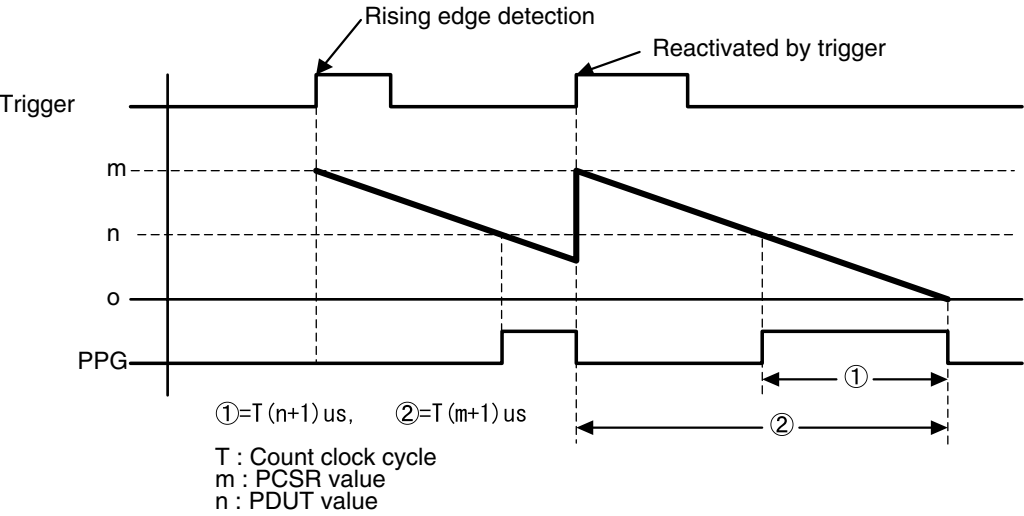
○ When reactivation is disabled

Figure 7.5-1 One-shot Mode Timing Chart(Trigger Reactivation Disabled)



○ When reactivation is enabled

Figure 7.5-2 One-shot Mode Timing Chart (Trigger Reactivation Enabled)



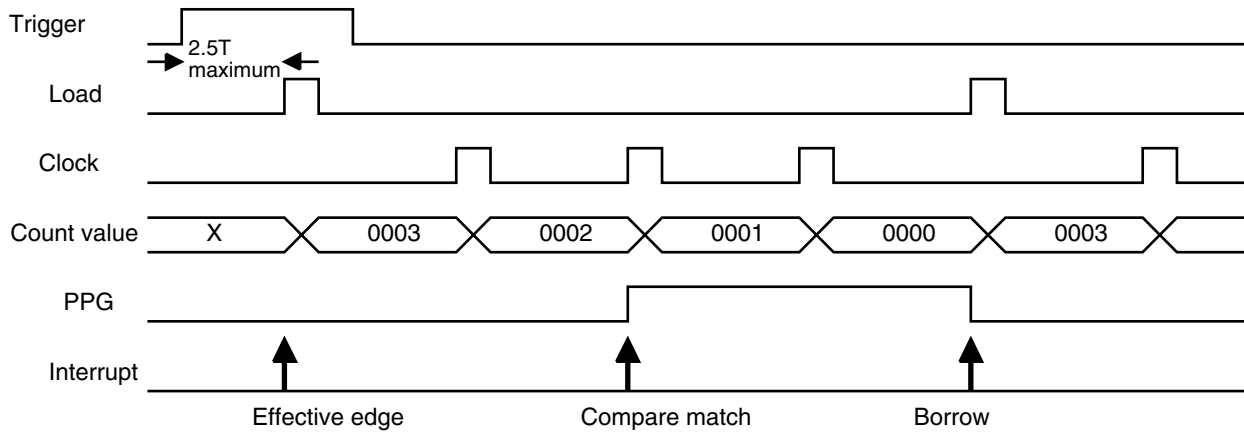
7.6 Interrupts

This section describes interrupt sources and shows timing charts.

■ Interrupts

Figure 7.6-1 "Interrupt Resources and Timing Chart (PPG Output: Ordinary Polarity)" shows the interrupt resources and timing chart.

Figure 7.6-1 Interrupt Resources and Timing Chart (PPG Output: Ordinary Polarity)



* : A value of up to 2.5T is required for a period after the retrigger is enabled until the count value is loaded.

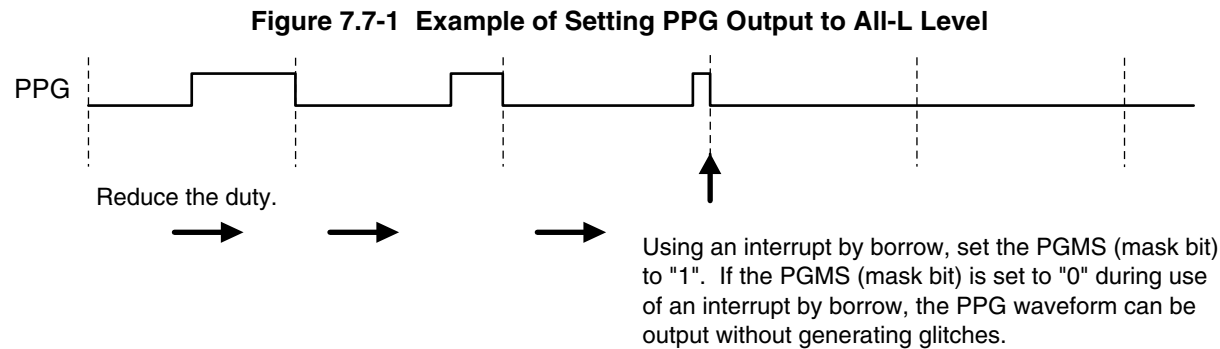
7.7 PPG Output of All-L and All-H

This section describes PPG output of all-L and all-H.

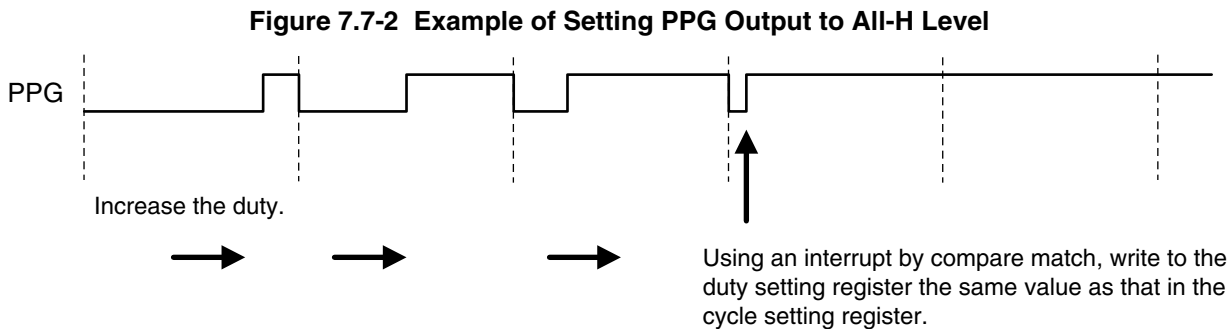
■ PPG Output All-L and All-H

Figure 7.7-1 "Example of Setting PPG Output to All-L Level" shows the output method that sets PPG output to all-L. Figure 7.7-2 "Example of Setting PPG Output to All-H Level" shows the output method that sets PPG output to all-H.

○ Example of setting PPG output to all-L



○ Example of setting PPG output to all-H



7.8 Activation of Multiple Channels of PPG Timer

Multiple channels of the PPG timer can be activated by using general control registers 1 and 2 (GCN1 and GCN2). If activation trigger is selected in general control register 1 (GCN1), multiple channels of the PPG timer can be activated simultaneously. This section provides examples of software activation using general control register 2 (GCN2) and using the 16-bit reload timer.

■ Activation of Multiple Channels of PPG Timer by Software

[Setting procedure]

1. Set a cycle with PCSR
2. Set a duty with PDUT. (Note: Make sure to the PCSR setting is prior to the PDUT setting.)
3. For GCN1, select trigger input causes for channels to be activated. Because GCN2 is used in this example, leave the initial settings. (ch0 --> EN0, ch1 --> EN1, ch2 --> EN2, and ch3 --> EN3)
4. Set the control status register settings for the channels to be activated.
 - - CNTE: 1 --> Enables the PPG timer.
 - - STGR: 0 --> Not activated because it is activated in GCN2
 - - MDSE: 0 --> PPG operation
 - - RTRG: 0 --> Disables reactivation.
 - - CSK1,0: 00 --> Count clock = ϕ
 - - PGMS: 0 --> PPG output is not masked.
 - - (Bit 8: 0 --> This bit is not used. Any value can be specified.)
 - - EGS1,0:01 --> Activates a rising edge.
 - - IREN: 1 --> Enables an interrupt request.
 - - IRQF: 0 --> Clears the interrupt cause.
 - - IRS1,0: 01 --> A counter borrow occurs and an interrupt request occurs.
 - - POEN: 1 --> Enables PPG output.
 - - OSEL: 0 --> Ordinary polarity
5. Writing data in GCN2 generates an activation trigger.

To activate ch0 and ch1 simultaneously with the above settings, write 1 in both EN0 and EN1 of GCN2. A rising edge occurs, and pulses are output from PPG0 and PPG1.

■ Activation Using the 16-bit Reload Timer

To activate channels using the 16-bit reload timer, select the 16-bit reload timer as the trigger input cause for GCN1 in step 3 at the above procedure. In step 5, activate the 16-bit reload timer instead of GCN2. Then, set the control status register settings as follows:

- - RTRG: 1 --> Enables reactivation.
- - EGS1,0: 11 --> Activates both edges.

CHAPTER 7 PROGRAMMABLE PULSE GENERATOR (PPG) TIMER

When the 16-bit reload timer output is set to toggle output, the PPG timer can be activated at certain intervals.

CHAPTER 8 MULTIFUNCTIONAL TIMER

This chapter provides a general outline of the multifunctional timer, its register configuration and functions, and its operation.

8.1 "Overview of the Multifunctional Timer"

8.2 "Block Diagram of the Multifunctional Timer Unit"

8.3 "Registers of the Multifunctional Timer Unit"

8.4 "Operation of the Multifunctional Timer Unit"

8.1 Overview of the Multifunctional Timer

The multifunctional timer unit consists of one 16-bit free-running timer, four 16-bit output compare units, and four 16-bit input capture units.

■ Configuration of the multifunctional timer

○ 16-bit free-running timer (×1)

The 16-bit free-running timer consists of a 16-bit up-counter, a control register, a 16-bit compare clear register, and a prescaler. The output value of the counter is used as the time base of the output compare units and input capture units.

- The operating clock of the counter can be selected from eight types, as follows:

Eight types of internal clock (ϕ , $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/128$)

ϕ : Machine clock

- An interrupt can be caused by an overflow of the counter value or compare matching with the compare clear register (compare matching requires mode setting).
- The counter value can be initialized to "0000_H" by a reset operation or software, or during compare matching with the compare clear register.

○ Output compare unit (×4)

The output compare unit consists of four 16-bit compare registers, a latch for compare output, and a control register. If the values of the 16-bit free-running timer and compare register match, the output level can be reversed and an interrupt can be caused.

- The four compare registers can operate independently. The output compare unit has an output pin and interrupt flag corresponding to each compare register.
- The output pin can be controlled by using two compare registers as a pair. It is also possible to reverse the output level by using two compare registers.
- The initial value of each output pin can be set.
- An interrupt can be caused by compare matching.

○ Input capture unit (×4)

The input capture unit consists of the capture registers corresponding to four independent external input pins and the control register. The detection of any edge of a signal input from an external input pin can be used to retain the value of the 16-bit free-running timer in the capture register and also cause an interrupt simultaneously.

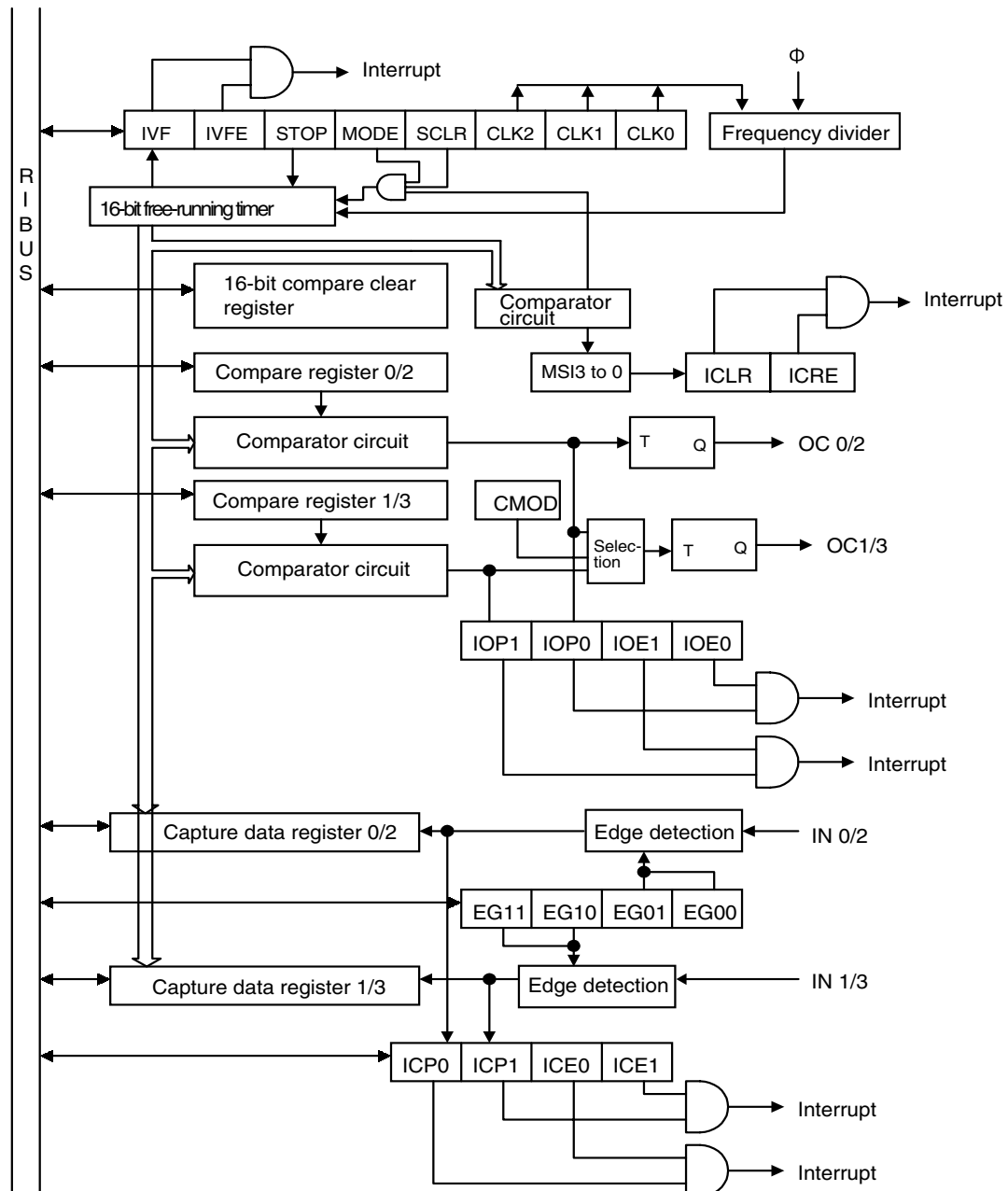
- Valid edges (rising edge, falling edge, or both) of the external input signal can be selected.
- The four input capture units can operate independently.
- An interrupt can be caused by a valid edge of the external input signal.

8.2 Block Diagram of the Multifunctional Timer Unit

Figure 8.2-1 "Block Diagram of the Multifunctional Timer Unit" is a block diagram of the multifunctional timer unit.

■ Block diagram of the Multifunctional Timer Unit

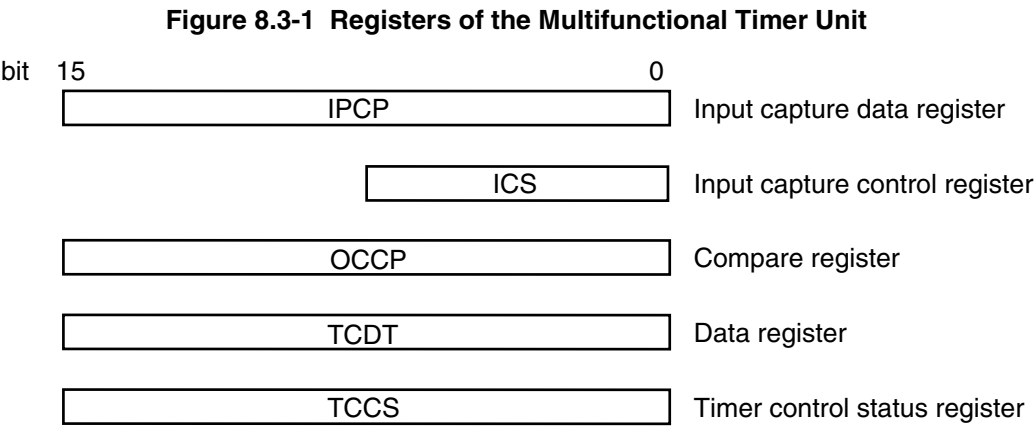
Figure 8.2-1 Block Diagram of the Multifunctional Timer Unit



8.3 Registers of the Multifunctional Timer Unit

Figure 8.3-1 "Registers of the Multifunctional Timer Unit" shows the registers of the multifunctional timer unit.

■ Registers of the Multifunctional Timer Unit



8.3.1 Register of the 16-bit free-running timer

The 16-bit free-running timer has three types of registers, as follows:

- Data register (TCDT)
- Compare clear register
- Timer control status register (TCCS)

■ Configuration of the Data Register (TCDT)

Figure 8.3-2 "Bit Configuration of the Data Register (TCDT)" shows the bit configuration of the data register (TCDT).

Figure 8.3-2 Bit Configuration of the Data Register (TCDT)

Data register high-order address	bit	15	14	13	12	11	10	9	8	Initial value
	006C _H	T15	T14	T13	T12	T11	T10	T09	T08	00000000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00000000 _B
Data register low-order address	bit	7	6	5	4	3	2	1	0	
	006D _H	T07	T06	T05	T04	T03	T02	T01	T00	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

This register can read the counter value of the 16-bit free-running timer. The counter value is cleared to "0000_H" by a reset operation. The timer value is set by writing to the register. However, the timer must be stopped (STOP=1) before writing to the register. The register is accessed in words.

Initialization of this register can be triggered by any of the following events:

- Initialization by a reset
- Initialization by clearing (SCLR) the timer control status register (TCCS)
- Initialization by matching of the values of the compare clear register (ch2 compare register) and the timer counter (requires mode setting).

■ Compare Register

A 16-bit compare register to be compared with the 16-bit free-running timer. The compare register of ch2 of the output compare unit is used. If the values of this register and the 16-bit free-running timer match, the 16-bit free-running timer is initialized to "0000_H" and the compare clear interrupt flag is set.

If interrupts are enabled, an interrupt request is sent to the CPU.

■ Configuration of the Timer Control Status Register (TCCS)

Figure 8.3-3 "Bit configuration of the Timer Control Status Register (TCCS)" shows the bit configuration of the timer control status register (TCCS).

Figure 8.3-3 Bit configuration of the Timer Control Status Register (TCCS)

Timer control status register high-order address	bit	15	14	13	12	11	10	9	8	Initial value
006E _H		ECLK	—	—	—	—	—	—	—	0-----
		R/W	—	—	—	—	—	—	—	00000000 _B
Timer control status register low-order address	bit	7	6	5	4	3	2	1	0	
006F _H		IVF	IVFE	STOP	MODE	SCLR	CLK2	CLK1	CLK0	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The bit functions of the timer control status register (TCCS) are explained below.

[Bit 15]: ECLK

This bit is used to select whether the count clock source of the 16-bit free-running timer is internal or external. Since the clock is changed immediately after writing to this bit, only change this bit while the output compare units and input capture units are stopped.

ECLK	Function
0	Select the internal clock (initial value)
1	Enter the clock from the external pin (FRCK)

Notes:

If the internal clock is selected, set the count clock to bits 2 to 0 (CLK2 - CLK0). This count clock becomes the base clock. If a clock is input from FRCK, set the corresponding DDR bit to "0".

[Bits 14 to 8]: (reserved)

These bits are not used.

[Bit 7]: IVF

Interrupt request flag of the 16-bit free-running timer. If an overflow of the 16-bit free-running timer occurs, this bit is set to "1". If the interrupt request enable bit (bit 6: IVFE) is set, an interrupt occurs. This bit is cleared by writing "0". Writing "1" has no meaning. "1" is always read in the read modify write instructions.

IVF	Function
0	No interrupt request (initial value)
1	Interrupt request exists

[Bit 6]: IVFE

Interrupt enable bit of the 16-bit free-running timer. If the interrupt flag (bit 7: IVF) is set to

"1" when this bit is "1", an interrupt occurs.

IVFE	Function
0	Disable interrupts (initial value)
1	Enable interrupts

[Bit 5]: STOP

Bit used to stop the count of the 16-bit free-running timer. The timer stops counting when "1" is written to this bit. The timer starts counting when "0" is written.

STOP	Function
0	Enable counts (operation) (initial value)
1	Disable counts (stop)

Notes:

If the 16-bit free-running timer stops, the output compare operation also stops.

[Bit 4]: MODE

This bit is used to set the initialization condition for the 16-bit free-running timer. If this bit is "0", the counter value can be initialized by a reset or the clear bit (bit 3: SCLR). If this bit is "1", the counter value can be initialized by, in addition to a reset and the clear bit (bit 3: SCLR), matching the values of the compare register 2 with the output compare unit.

MODE	Function
0	Initialize by reset or clear bit (initial value)
1	Initialize by reset, clear bit, or compare register 2

Notes:

The counter value is initialized at the change point of the count value.

[Bit 3]: SCLR

This bit is used to initialize the value of the 16-bit free-running timer in operation to "0000_H". The counter value is initialized to "0000_H" when "1" is written to this bit. Writing "0" to this bit has no meaning. "0" is always read from this bit. The counter value is initialized whenever the count value is changed.

SCLR	Meaning of flag
0	No meaning (initial value)
1	Initialize the counter value to "0000 _H ".

Notes:

For initialization while the timer is stopped, write "0000_H" into the data register.

[Bits 2, 1, and 0]: CLK2 to CLK0

Bits used to select the count clock of the 16-bit free-running timer. Since the clock is changed immediately after writing to these bits, change them while the output compare units and input capture units are stopped.

CLK2	CLK1	CLK0	Count clock	$\phi=25\text{MHz}$	$\phi=16\text{MHz}$	$\phi=12.5\text{MHz}$	$\phi=8\text{MHz}$
0	0	0	ϕ	40ns	62.5ns	80ns	0.125 μs
0	0	1	$\phi/2$	80ns	0.125 μs	0.16 μs	0.25 μs
0	1	0	$\phi/4$	0.16 μs	0.25 μs	0.32 μs	0.5 μs
0	1	1	$\phi/8$	0.32 μs	0.5 μs	0.64 μs	1 μs
1	0	0	$\phi/16$	0.64 μs	1 μs	1.28 μs	2 μs
1	0	1	$\phi/32$	1.28 μs	2 μs	2.56 μs	4 μs
1	1	0	$\phi/64$	2.56 μs	4 μs	5.12 μs	8 μs
1	1	1	$\phi/128$	5.12 μs	8 μs	10.24 μs	16 μs

ϕ =Machine clock

8.3.2 Register of the Output Compare Unit

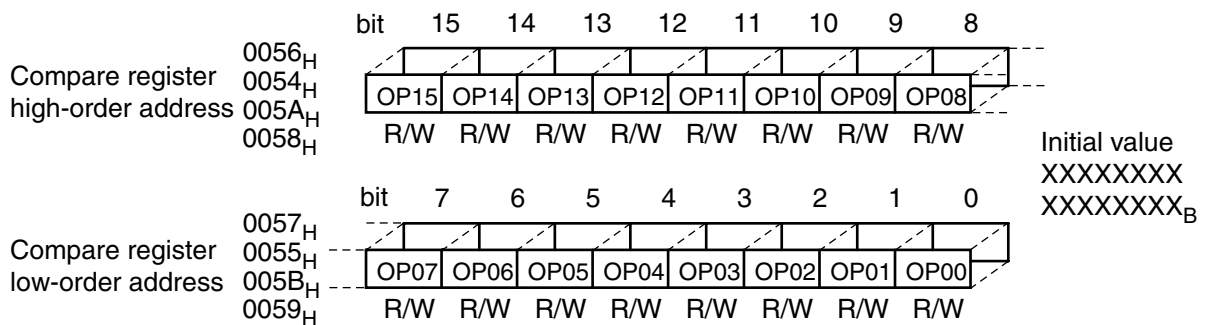
The output compare unit has the two types of registers, as follows:

- Compare registers (0CCP0 to 0CCP3)
- Output control registers (0CS0 to 0CCP3)

■ Configuration of the Compare Registers (0CCP0 to 0CCP3)

Figure 8.3-4 "Bit configuration of the Compare Registers (0CCP0 to 0CCP3)" shows the bit configuration of the compare registers (0CCP0 to 0CCP3).

Figure 8.3-4 Bit configuration of the Compare Registers (0CCP0 to 0CCP3)



The 16-bit compare register to be compared with the 16-bit free-running timer. Since the initial value of this register is undefined, start it after setting an initial value. Access this register in words. If the values of this register and the 16-bit free-running timer match, a compare signal is generated to set the output compare interrupt flag. If output is enabled, the output level corresponding to the compare register is reversed.

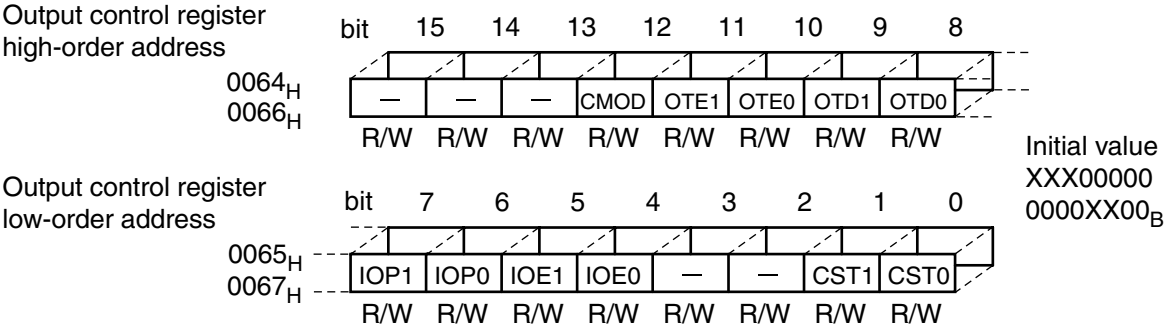
Notes:

Rewrite the compare register within a compare interrupt routine or after disabling the compare so that compare matching and write do not occur simultaneously.

■ Configuration of the Output Control Registers (0CS0 to 0CS3)

Figure 8.3-5 "Bit Configuration of the Output Control Registers (0CS0 to 0CS3)" shows the bit configuration of the output control register (0CS0 to 0CS3).

Figure 8.3-5 Bit Configuration of the Output Control Registers (0CS0 to 0CS3)



The explanation below refers to ch0 and ch1. For reading, replace ch2 and ch3 with ch0 and ch1, respectively.

The following bit functions of the output control registers (OCS0 to OCS3) are explained below.

[Bit 12]: CMOD

If pin output is enabled (OTE1=0 or OTE0=1), the pin output level reversal mode is switched when compare matching occurs.

- If CMOD=0 (initial value), the output level of the pin corresponding to the compare register is reversed.
 - OC0: Reverse the level if a match with compare register 0 occurs.
 - OC1: Reverse the level if a match with compare register 1 occurs.
- If CMOD=1, compare register 0 reverses the output level to CMOD=0. The output level of the pin (OC1) corresponding to compare register 1 is only reversed if both compare register 0 and compare register 1 match. If compare register 0 and compare register 1 have the same value, the action is the same as that with only one compare register.
 - OC0: Reverse the level if a match with compare register 0 occurs.
 - OC1: Reverse the level if a match with both compare register 0 and compare register 1 occurs.

[Bits 11 and 10]: OTE1 and OTE0

Bit used to enable pin output of the output compare unit

OTE	Function
0	Operate as a general-purpose port (initial value)
1	Enable the output compare pin output

OTE1: Corresponding to output compare 1

OTE0: Corresponding to output compare 0

[Bits 9 and 8]: ODT1 and ODT0

Bits used for changing the pin output level if pin output of the output compare register is enabled. The initial value of the compare pin output is "0". Write to these bits after stopping the compare operation. The value of the output compare pin can be read from these bits.

ODT	Function
0	Set the compare pin output to "0" (initial value)
1	Set the compare pin output to "1"

ODT1: Corresponding to output compare 1

ODT0: Corresponding to output compare 0

[Bits 7 and 6]: IOP1 and IOP0

Interrupt flag of the output compare unit. If the values of the compare register and the 16-bit free-running timer match, these bits are set to "1". Setting the bits to "1" when an interrupt

CHAPTER 8 MULTIFUNCTIONAL TIMER

request bits (IOE1 and IOE0) are enabled causes an output compare interrupt. Writing "0" to these bits clears them, but writing "1" has no meaning. "1" can be read by a read modify write instruction.

IOP	Function
0	No output compare matching (initial value)
1	Output compare matched

IOP1: Corresponding to output compare 1

IOP0: Corresponding to output compare 0

[Bits 5 and 4]: IOE1 and IOE0

Bits used to enable an interrupt of the output compare unit. Setting the interrupt flags (IOP1 and IOP0) to "1" when this bit is "1" causes an output compare interrupt.

IOE	Function
0	Disable output compare interrupts (initial value)
1	Enable output compare interrupts

IOE1: Corresponding to output compare 1

IOE0: Corresponding to output compare 0

[Bits 3 and 2]: (reserved)

These bits are not used.

[Bits 1 and 0]: CST1 and CST0

These bits are used to enable a match operation with the 16-bit free-run timer. Before allowing a compare operation, be sure to set a compare register value and an output control register value.

CST	Function
0	Disable compare operation (initial value)
1	Enable compare operation

CST1: Corresponding to output compare 1

CST0: Corresponding to output compare 0

Notes:

Because the output compare unit is synchronized with the 16-bit free-running timer, the compare operation is stopped if the 16-bit free-running timer is stopped.

8.3.3 Register of the Input Capture Unit

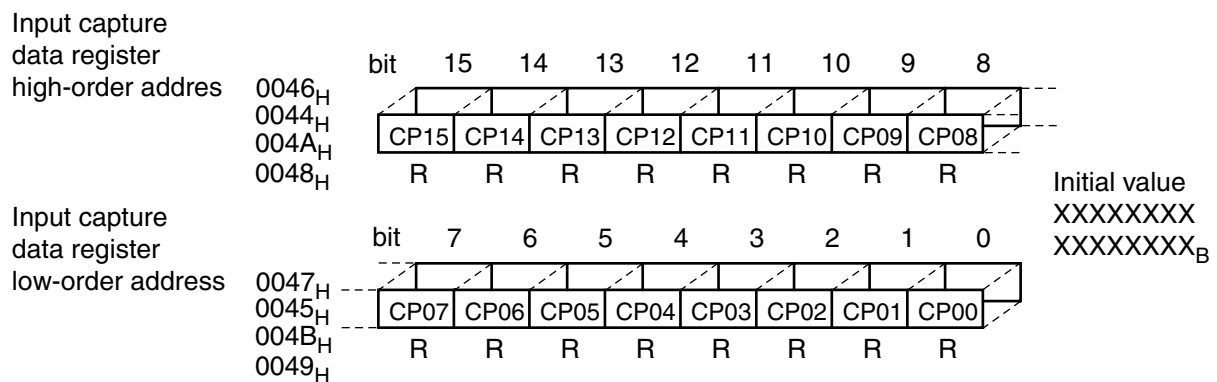
The input capture unit has two types of registers, as follows:

- Input capture data registers (IPCP0 to IPCP3)
- Input capture control registers (ICS01 and ICS23)

■ Configuration of the Input Capture Data Registers (IPCP0 to IPCP3)

Figure 8.3-6 "Bit Configuration of the Input Capture Data Registers (IPCP0 to IPCP3)" shows the bit configuration of the input capture data registers (IPCP0 to IPCP3).

Figure 8.3-6 Bit Configuration of the Input Capture Data Registers (IPCP0 to IPCP3)

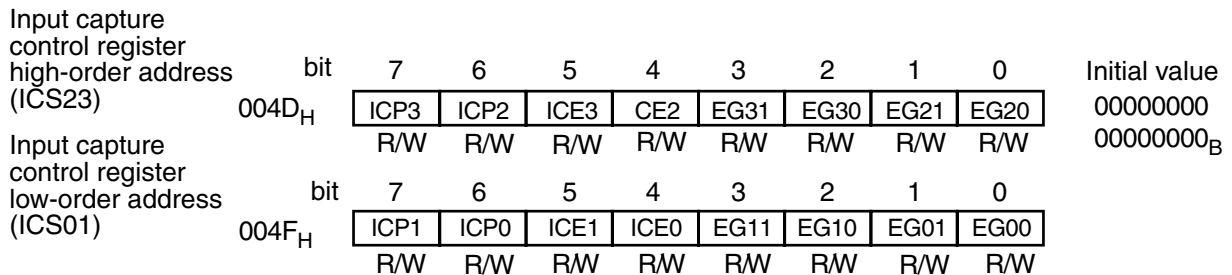


This register holds the value of the 16-bit free-running timer if a valid edge of input waveforms of the corresponding external pin is detected. Access the register in words. Writing to this register is not allowed.

■ Configuration of the Input Capture Control Registers (ICS01 and ICS23)

Figure 8.3-7 "Bit Configuration of the Input Capture Control Registers (ICS01 and ICS23)" shows the bit configuration of the input capture control registers (ICS01 and ICS23).

Figure 8.3-7 Bit Configuration of the Input Capture Control Registers (ICS01 and ICS23)



The bit functions of the input capture control registers (ICS01 and ICS23) are explained below.

[Bits 7 and 6]: ICP3, ICP2, ICP1, and ICP0

These bits are used as input capture interrupt flags. If a valid edge of the external input pin is detected, the relevant bit is set to "1". If an interrupt enable bit (ICE3, ICE2, ICE1, or ICE0) is set, detection of a valid edge causes an interrupt. This bit can be cleared by writing "0" to it, but writing "1" has no meaning. "1" can be read by a read modify write instruction.

ICP	Function
0	No valid edge detected (initial value)
1	Valid edge detected

ICPn: The number n corresponds to the channel number of the input capture unit.

[Bits 5 and 4]: ICE3, ICE2, ICE1, and ICE0

Input capture interrupt enable bit. If an interrupt flag (ICP3, ICP2, ICP1, or ICP0) is set to "1" setting this bit is "1" causes an input capture interrupt.

ICE	Function
0	Disable interrupts (initial value)
1	Enable interrupts

ICE_n: The number n corresponds to the channel number of the input capture unit.

[Bits 3 to 0]: EG31/30, EG21/20, EG11/10, and EG01/00

These bits are used to select the valid edge polarity of the external input. These bits are also used to enable input capture operations.

EG31	EG30	Edge detection polarity
0	0	No edge detected (stopped) (initial value)
0	1	Rising edge detected
1	0	Falling edge detected
1	1	Both edges detected

EG_{n1}/EG_{n0}: The number n corresponds to the channel number of the input capture unit.

8.4 Operation of the Multifunctional Timer Unit

This section explains the operation of the multifunctional timer unit.

■ Explanation of the operation of the multifunctional timer unit

○ 16-bit free-running timer

After reset is released, the 16-bit free-running timer begins counting at "0000_H". This counter value becomes the time base of the 16-bit output compare unit and the 16-bit input capture unit.

○ 16-bit output compare unit

If the values of the setup compare register and the 16-bit free-running timer match, the 16-bit output compare unit can set the interrupt flag and reverse the output level.

○ 16-bit input capture unit

If a setup valid edge is detected, the 16-bit input capture unit can capture the value of the 16-bit free-running timer in the capture register to cause an interrupt.

8.4.1 Operation of the 16-bit Free-running Timer Unit

After the reset is released, the 16-bit free-running timer unit begins counting at "0000_H". This counter value becomes the time base of the 16-bit output compare unit and the 16-bit input capture unit.

■ Explanation of Operation of the 16-bit Free-Running Timer

The counter value is cleared when one of the following conditions is met:

- An overflow occurs
- Match with the compare clear register (compare register of the output compare ch2) (requires mode setting)
- "1" is written to the SCLR bit of the timer control status register (TCCS) during operation
- "0000_H" is written to TCDT while the timer is stopped

An interrupt can be caused if an overflow occurs and the counter is cleared after compare matching with the compare clear register value (the compare match interrupt requires mode setting).

Figure 8.4-1 Timing for Counter Startup Caused by an Overflow and Operation

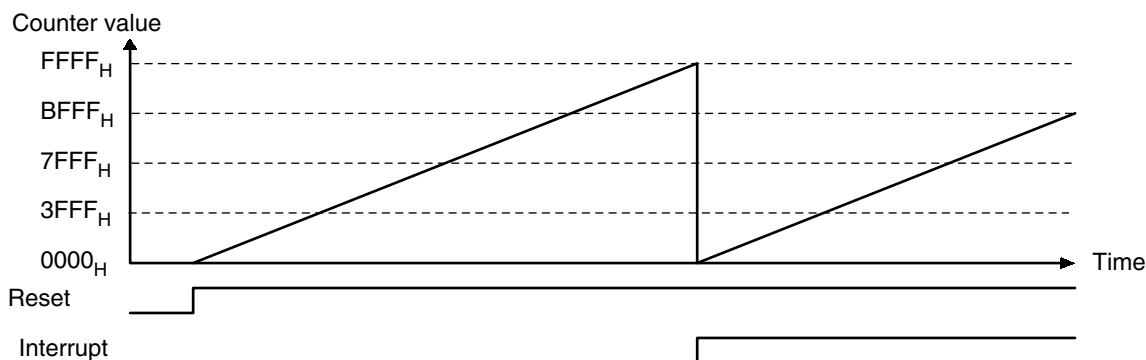
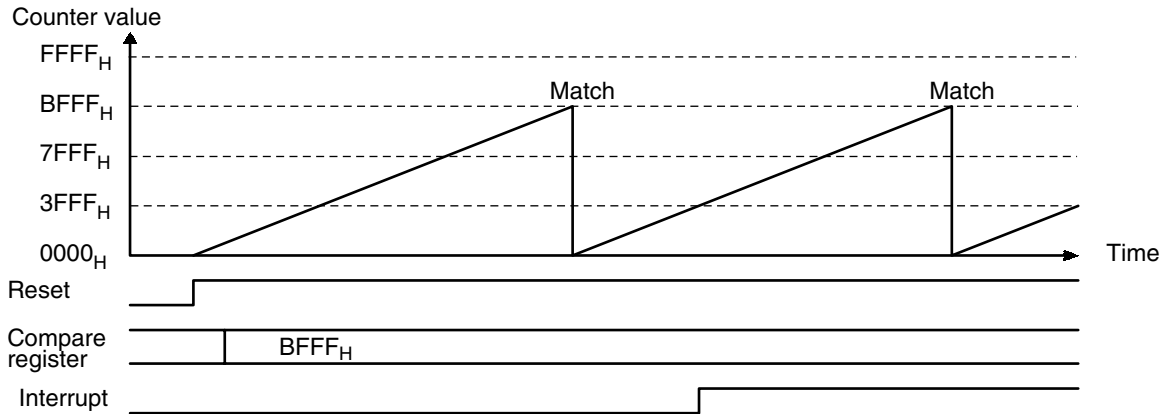
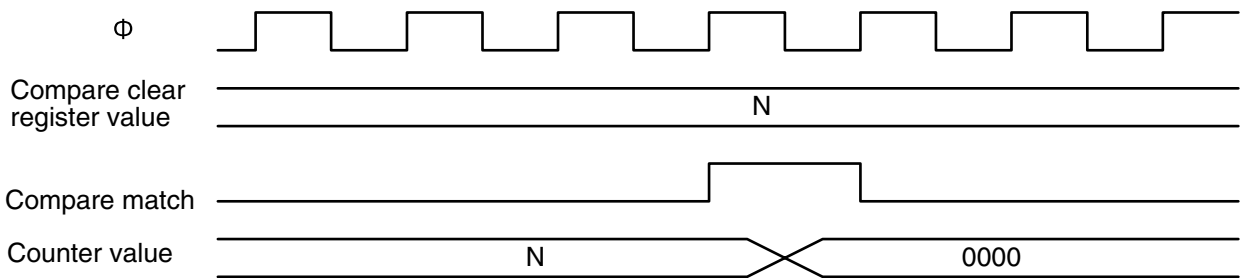


Figure 8.4-2 Counter Clearing when Matched with the Compare Clear Register Value

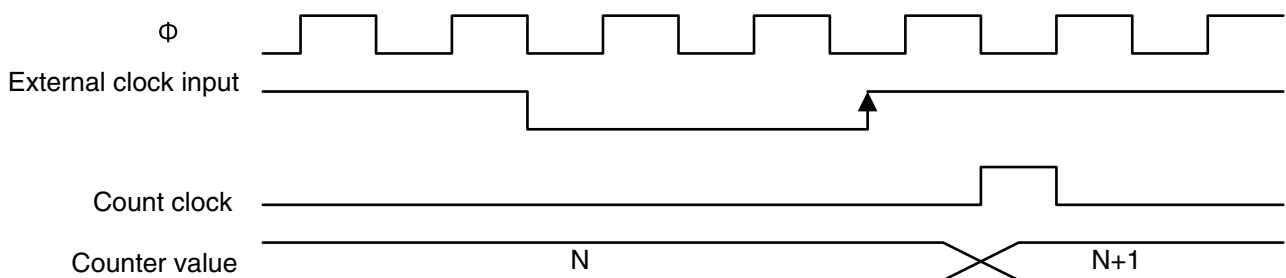
■ Clear Timing of the 16-bit Free-running Timer

The counter is cleared by a reset operation or software, or during compare matching with the compare clear register. The counter is cleared by a reset operation or software immediately after the occurrence of a clear instruction. However, performance of counter clearing when matched with the compare clear register is synchronized with the count timing.

Figure 8.4-3 Clear Timing of the Free-running Timer

■ Count Timing of the 16-bit Free-running Timer

The 16-bit free-running timer counts up with the input clock (internal or external). If an external clock is selected, counting occurs at the rising edge.

Figure 8.4-4 Count Timing of the 16-bit Free-Running Timer

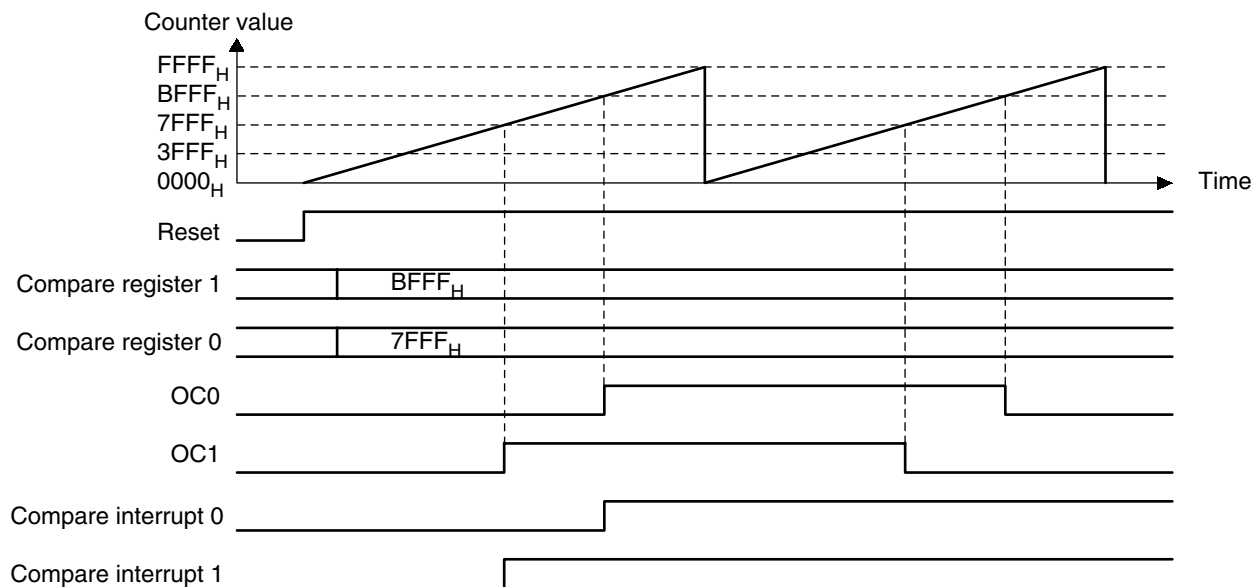
8.4.2 Operation of the 16-bit Output Compare Unit

If the values of the setup compare register and the 16-bit free-running timer are compared and matched, the 16-bit output compare unit can set the interrupt flag and reverse the output level.

■ Operation Explanation of the 16-bit Output Compare Unit

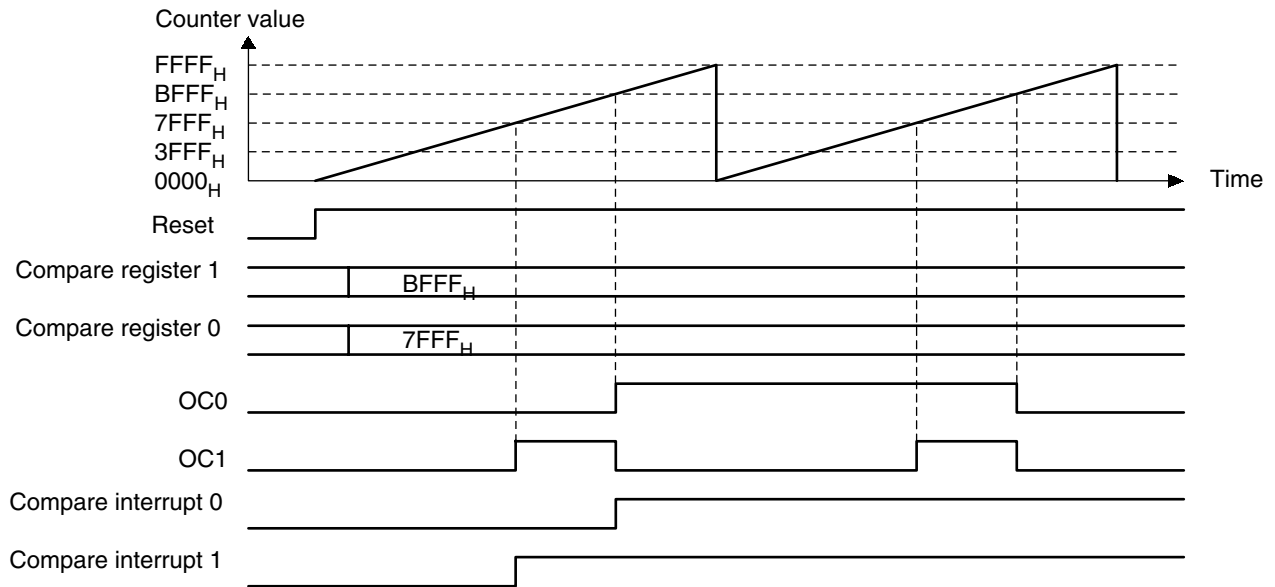
- Compare operations can be performed independently by each channel (CMOD=0).

Figure 8.4-5 Example of Output Waveforms when Compare Registers 0 and 1 are Used (Initial Value of Output = "0")



- Two pairs of compare registers (CMOD=1) can be used to change the output level.

Figure 8.4-6 Example of Output Waveforms when Compare Registers 0 and 1 are Used (Initial Value of Output = "0")

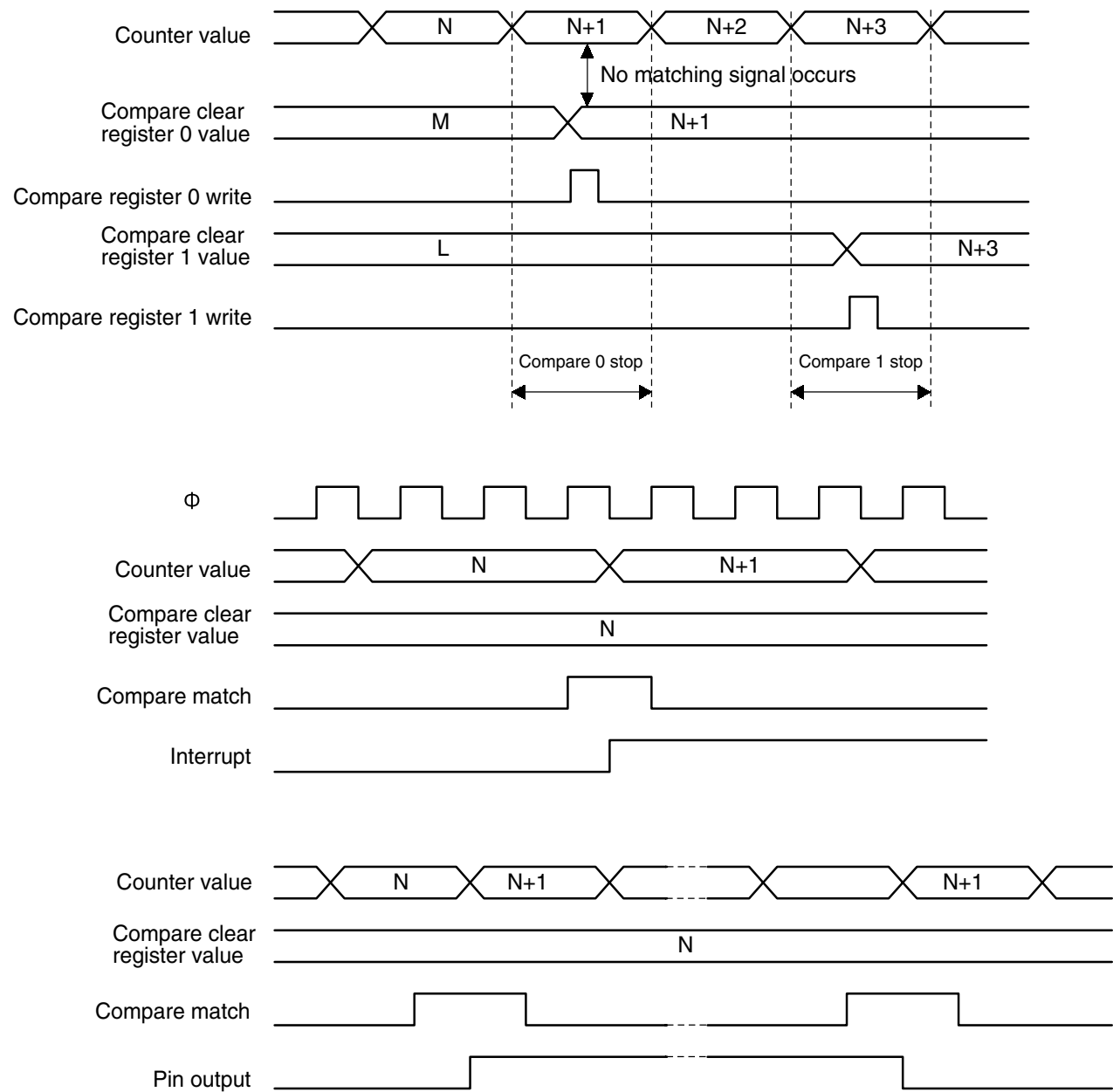


■ Timing of the 16-bit Output Compare Unit

- Two pairs of compare registers (CMOD=1) can be used to change the output level.

If the values of the free-running timer and the setup compare register match, the output compare unit can reverse the output by generating the compare match signal and also cause an interrupt. Output reversal timing when the compare matches is synchronized with the count timing. During rewriting of the compare register, the counter value is not compared.

Figure 8.4-7 Timing of the 16-bit Output Compare Unit

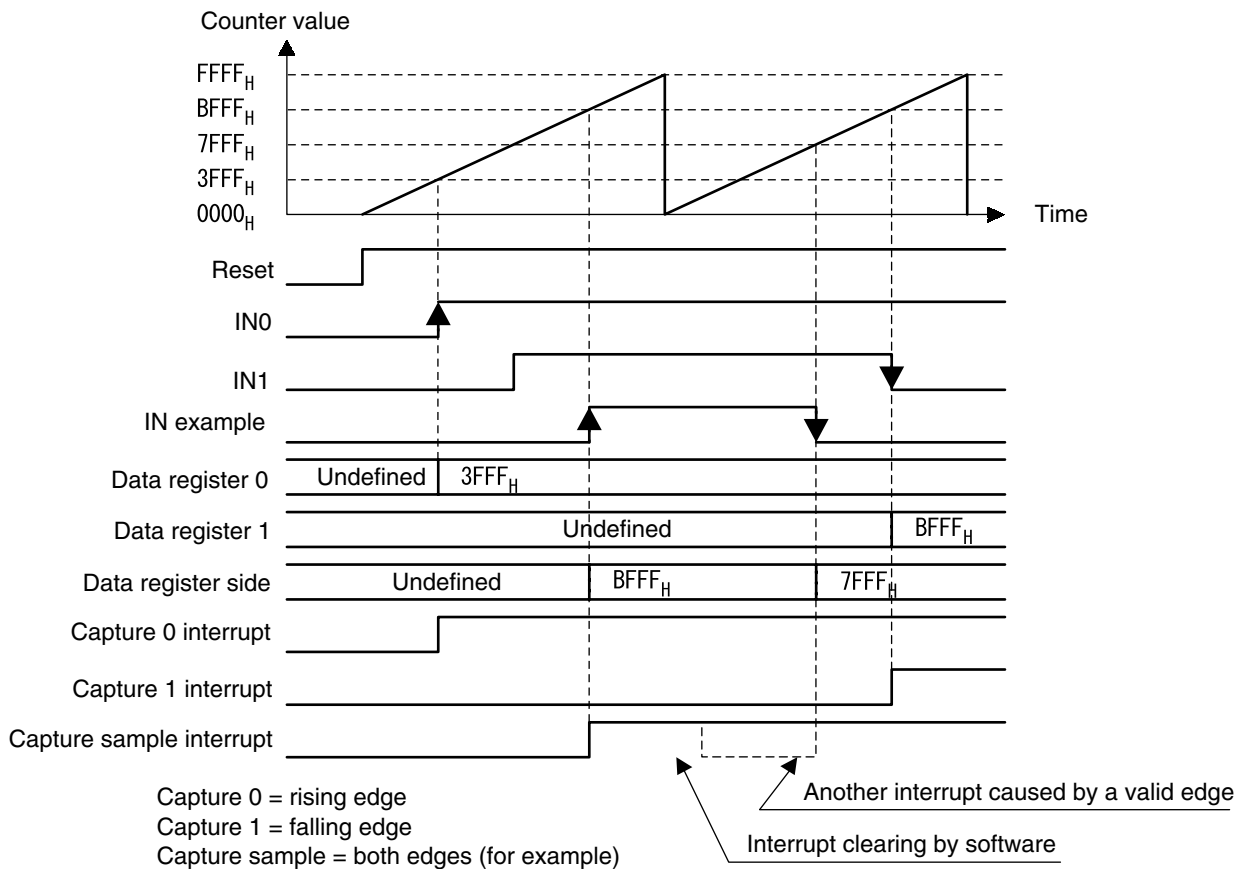


8.4.3 Operation of the 16-bit Input Capture Unit

If a setup valid edge is detected, the 16-bit input capture unit can capture the 16-bit free-running timer value in the capture register and cause an interrupt.

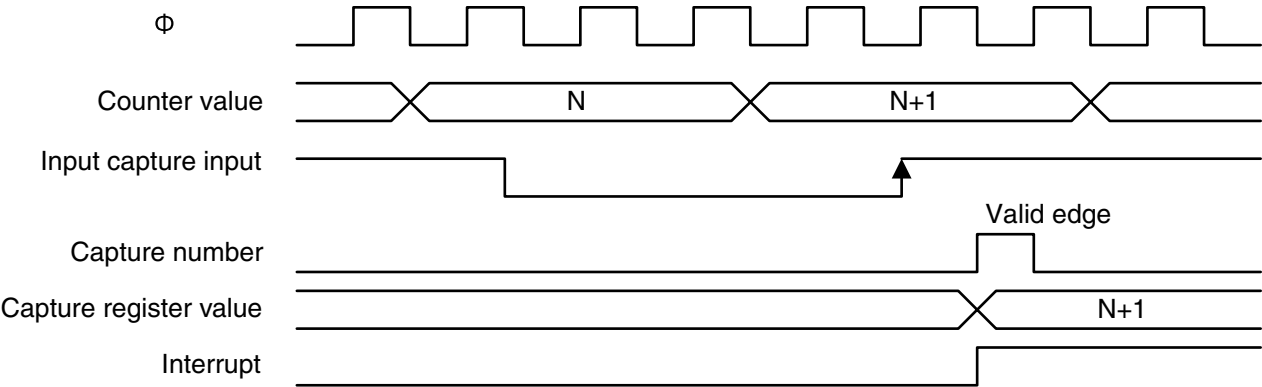
■ Operation of the 16-bit Input Capture Unit

Figure 8.4-8 Example of Capture Timing of the Input Capture Unit



■ Timing of the 16-bit Input Capture Unit

Figure 8.4-9 Timing of the 16-bit Input Capture Unit



CHAPTER 9 U-TIMER

This chapter outlines U-TIMER, describes the configuration and functions of the registers, and explains U-TIMER operation.

9.1 "Outline of U-TIMER"

9.2 "U-TIMER Registers"

9.3 "U-TIMER Operation"

9.1 Outline of U-TIMER

This section provides an outline and shows block diagrams of U-TIMER (16-bit timer for UART baud rate generation).

■ Outline of U-TIMER

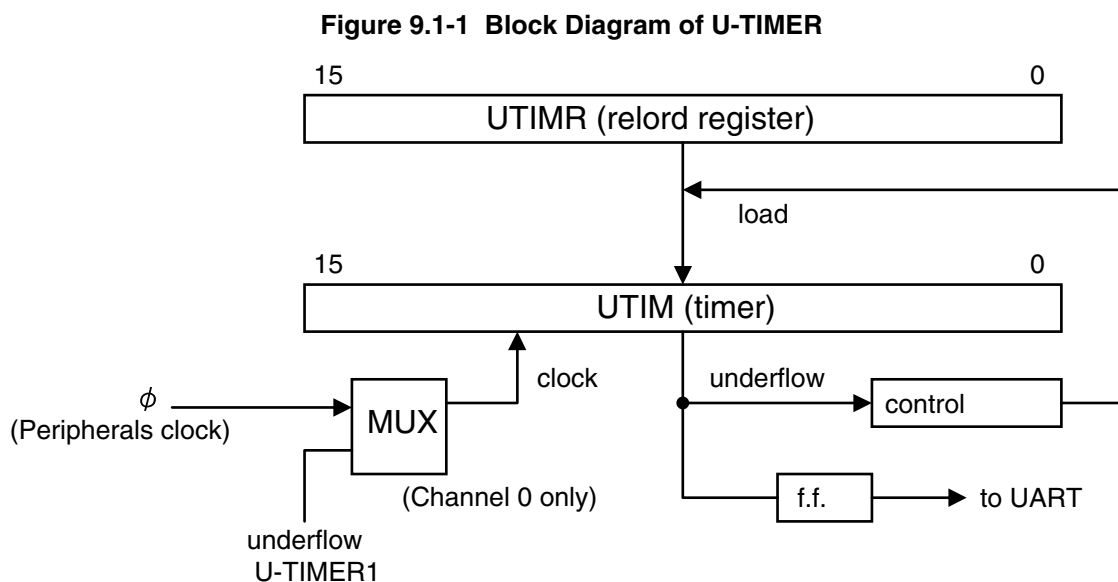
U-TIMER is a 16-bit timer for generating the UART baud rate. Any baud rate can be set by a combination of a chip operation frequency and U-TIMER reload value.

U-TIMER can be used as an interval timer because it generates an interrupt when the count underflows.

The MB91F127/128 has three internal channels for this type of interval timer. When U-TIMER is used as an interval timer, two U-TIMERS (channel 0 and 1) are cascaded, providing up to 2^{32} x ϕ intervals that can be counted.

■ Block Diagram

Figure 9.1-1 "Block Diagram of U-TIMER" shows a block diagram of U-TIMER.



9.2 U-TIMER Registers

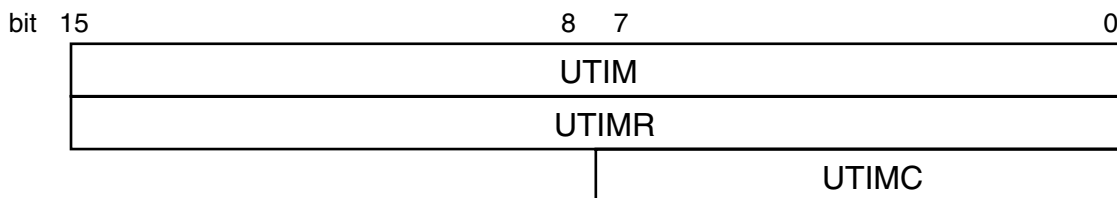
U-TIMER has the following three registers:

- U-TIMER value register (UTIM)
- Reload register (UTIMR)
- U-TIMER control register (UTIMC)

■ List of Registers of U-TIMER

Figure 9.2-1 "Register Configuration of U-TIMER" shows a list of registers of U-TIMER.

Figure 9.2-1 Register Configuration of U-TIMER



■ U-TIMER Value Register (UTIM)

Figure 9.2-2 "Bit Configuration of U-TIMER Value Register (UTIM)" shows the bit configuration of the U-TIMER value register (UTIM).

Figure 9.2-2 Bit Configuration of U-TIMER Value Register (UTIM)

Address	bit	15	14	2	1	0	Initial value
0000 0078 _H (ch.0)		b15	b14	b2	b1	b0	00000000 00000000 _B
0000 007C _H (ch.1)							
0000 0080 _H (ch.2)		R	R	R	R	R	

UTIM indicates the timer value. Use a 16-bit transfer instruction to access the time value.

■ Reload Register (UTIMR)

Figure 9.2-3 "Bit Configuration of Reload Register (UTIMR)" shows the bit configuration of the reload register (UTIMR).

Figure 9.2-3 Bit Configuration of Reload Register (UTIMR)

Address	bit	15	14	2	1	0	Initial value
0000 0078 _H (ch.0)		b15	b14	b2	b1	b0	00000000 00000000 _B
0000 007C _H (ch.1)							
0000 0080 _H (ch.2)		W	W	W	W	W	

The value reloaded into UTIM when UTIM underflows is stored in the UTIMR register.

Use a 16-bit transfer instruction to access the time value.

■ U-TIMER Control Register (UTIMC)

Figure 9.2-4 "Bit Configuration of U-TIMER Control Register (UTIMC)" shows the bit configuration of the U-TIMER control register (UTIMC).

Figure 9.2-4 Bit Configuration of U-TIMER Control Register (UTIMC)

Address	bit	7	6	5	4	3	2	1	0	Initial value
0000 007B _H (ch.0)		UCC1	—	—	UTIE	UNDR	CLKS	UTST	UTCR	0--00001 _B
0000 007F _H (ch.1)		R/W	—	—	R/W	R/W	R/W	R/W	R/W	
0000 0083 _H (ch.2)										

UTIMC controls U-TIMER operation.

■ Bit Details of U-TIMER Control Register (UTIMC)

The function of each bit of the U-TIMER control register (UTIMC) is explained below.

[bit 7] UCC1 (U-TIMER count control 1)

The UCC1 bit controls U-TIMER counting.

UCC1	Function
0	Normal operation $\alpha=2n+2$ (Initial value)
1	+1 mode $\alpha=2n+3$

n: UTIMER setting value (decimal number)

α : Clock cycle output to the UART

In U-TIMER, a regular clock whose cycle is $2(n+1)$ can be set for the UART. In addition to this, a clock whose frequency is divided by an odd number can also be set. When UCC1 is set to 1, a clock whose cycle is $2n+3$ is generated.

Examples of setting:

1. UTIMR = 5, UCC1 = 0 --> Generation cycle = $2n+2 = 12$ cycles
2. UTIMR = 25, UCC1 = 1 --> Generation cycle = $2n+3 = 53$ cycles
3. UTIMR = 60, UCC1 = 0 --> Generation cycle = $2n+2 = 122$ cycles

To use U-TIMER as an interval timer, set the UCC1 value to 0.

[bits 6 and 5] (reserved)

These bits are reserved.

[bit 4] UTIE (U-TIMER interrupt enable)

UTIE is an interrupt-enable bit for U-TIMER underflow.

UTIE	Function
0	Interrupts not allowed (initial value)
1	Interrupts not allowed

[bit 3] UNDR (underflow flag)

UNDR is a flag that indicates that an underflow has been generated.

If bit 4, UTIE, is set to 1 and UNDR is set, an underflow interrupt is generated. UNDR is cleared by a reset or by writing 0.

When a Read Modify Write instruction performs a read operation, 1 is always read.

Writing 1 to UNDR is invalid.

[bit 2] CLKS (clock select)

CLKS is a bit that specifies cascading for U-TIMER channels 0 and 1.

CLKS	Function
0	The clock source is the peripherals clock (ϕ). (initial value)
1	The channel 1 underflow signal is used as the source clock timing for U-TIMER channel 0

(f.f. in the block diagram.)

CLKS is valid only for channel 0. Always set CLKS to 0 for channel 1.

[bit 1] UTST (U-TIMER start)

UTST is the bit that enables U-TIMER operation.

UTST	Function
0	Stopped. Operation stops when 0 is written during operation. (initial value)
1	Active. Operation continues even though 1 is written during operation.

[bit 0] UTCR (U-TIMER clear)

When 0 is written to UTCR, U-TIMER is cleared to 0000_H. (f.f. is cleared to 0 at this point.)

A 1 is always read.

■ Notes on Using U-TIMER Control Register (UTIMC)

- When the UTST start bit is asserted (started) the stop state, automatic reloading starts.
- When the UTCR clear bit and UTST start bit are asserted simultaneously in the stop state, an underflow is generated during decrementing of the counter immediately after the counter is cleared to 0.
- When the UTCR clear bit is asserted during operation, the counter is cleared to 0. Therefore, a short pulse may be output in the output waveform, causing the UART or upper-level U-TIMER in cascade mode to operate incorrectly. The counter must not be cleared by this clear bit during operation when an output clock is used.
- When 0 or 1 is set in the lower-level UTIMER (reload register) in cascade mode, counting is not performed properly.

9.3 U-TIMER Operation

This section explains the U-TIMER baud rate calculation and cascade mode timings.

■ Baud Rate Calculation

The UART uses the underflow flip-flop (f.f. in the block diagram) of the corresponding U-TIMER (U-TIMER0 --> UART0, U-TIMER1 --> UART1, U-TIMER2 --> UART2) as the baud rate clock source.

○ Asynchronous (Start-Stop) mode

The UART divides the U-TIMER output into 16 cycles, which it then uses.

- When UCC1 = 0

$$\text{bps} = \frac{f}{(2n+2) \times 16}$$

- When UCC1 = 1

$$\text{bps} = \frac{f}{(2n+3) \times 16}$$

n: UTIMR (reload value)

f: Peripherals machine clock frequency (changes with the gear)

○ CLK Synchronous mode

- When UCC1 = 0

$$\text{bps} = \frac{f}{(2n+2)}$$

- When UCC1 = 1

$$\text{bps} = \frac{f}{(2n+3)}$$

n: UTIMR (reload value)

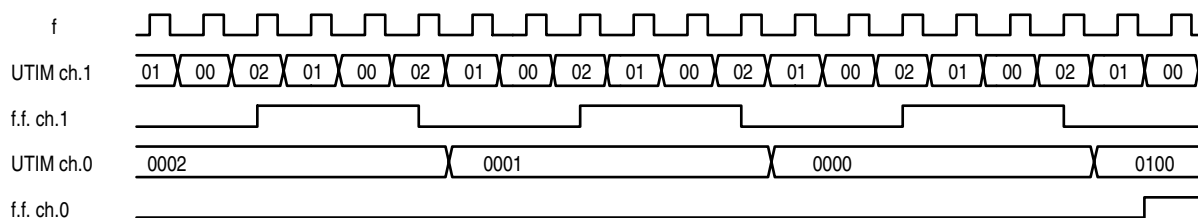
f: Peripherals machine clock frequency (changes with the gear)

■ Cascade Mode

U-TIMER channels 0 and 1 can be used in cascade mode.

Figure 9.3-1 "Cascade mode timing chart" shows sample timing charts in which UTIMER ch0 is set to "0100H" and UTIMER ch1 is set to "0002H".

Figure 9.3-1 Cascade mode timing chart



CHAPTER 10 EXTERNAL INTERRUPT

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

10.1 "Overview of the External Interrupt"

10.2 "External Interrupt Controller Registers"

10.3 "Operation of the External Interrupt"

10.1 Overview of the External Interrupt

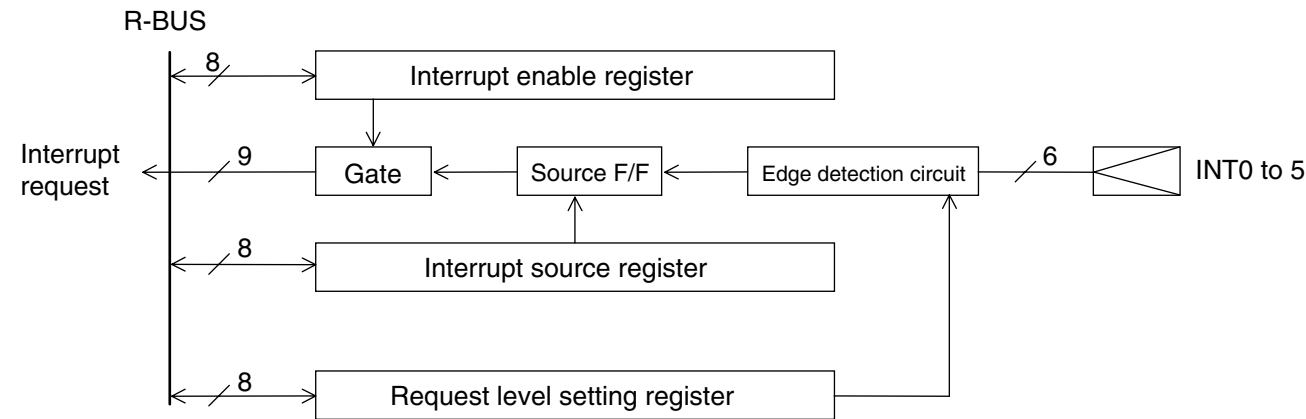
The external interrupt controller is a block that controls external interrupt requests input input TO-5.

H level, L level, rising edge, or falling edge can be selected as the level of a request to be detected.

■ Block Diagram of the External Interrupt

Figure 10.1-1 "Block Diagram of the External Interrupt" is a block diagram of the external interrupt.

Figure 10.1-1 Block Diagram of the External Interrupt



10.2 External Interrupt Controller Registers

This section describes the configuration and functions of the registers used by the external interrupt controller.

■ External Interrupt Controller Registers

Figure 10.2-1 "External Interrupt Register List" shows the registers used by the external interrupt controller.

○ External interrupt enable register (ENIR)

Figure 10.2-1 External Interrupt Register List

bit	7	6	5	4	3	2	1	0	
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	External interrupt enable register (ENIR)
bit	15	14	13	12	11	10	9	8	
	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	External interrupt source register (EIRR)
bit	7	6	5	4	3	2	1	0	
	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	Request level setting register (ELVR)
bit	15	14	13	12	11	10	9	8	
	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	Request level setting register (EHVR)

10.2.1 Interrupt Enable Register (ENIR)

The interrupt enable register (ENIR) performs mask control for external interrupt request output.

■ ENIR Configuration

Figure 10.2-2 "Bit Configuration of the Interrupt Enable Register (ENIR)" shows the bit configuration of the interrupt enable register (ENIR: ENable Interrupt Request Register)

Figure 10.2-2 Bit Configuration of the Interrupt Enable Register (ENIR)

		bit	7	6	5	4	3	2	1	0	Initial value 00000000 _B
Address	000095 _H		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The interrupt enable register (ENIR) controls masks for 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), after which 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.

For this device, writing to bits EN6 and EN7 has no meaning.

Write "0" to bits EN6 and EN7.

10.2.2 External Interrupt Source Register (EIRR)

This section describes the bit configuration and functions of the external interrupt source register.

■ EIRR Configuration

Figure 10.2-3 "Bit Configuration of the External Interrupt Source Register (EIRR)" shows the bit configuration of the external interrupt source register (EIRR: External Interrupt Request Register).

Figure 10.2-3 Bit Configuration of the External Interrupt Source Register (EIRR)

	bit	15	14	13	12	11	10	9	8	Initial value
Address	000094 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	0000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The EIRR register, when it is read, indicates that a corresponding external interrupt request exists. When it is written to, the contents of the flip-flop that indicates this request are cleared. If 1 is read from the EIRR register, an external interrupt request exists 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 has no effect.

For a read by a read modify write instruction, 1 is read.

10.2.3 External Interrupt Request Level Setting Register (ELVR, EHVR)

This section describes the bit configuration and functions of the external interrupt request level setting register (ELVR, EHVR).

■ External Interrupt Request Level Setting Register (ELVR, EHVR)

Figure 10.2-4 "Bit Configuration of the External Interrupt Request Level Setting Register (ELVR, EHVR)" Interrupt Request Level Setting Register (ELVR, EHVR)" shows the bit configuration of the external interrupt request level setting register (ELVR, EHVR).

Figure 10.2-4 Bit Configuration of the External Interrupt Request Level Setting Register (ELVR, EHVR)

ELVR	bit	7	6	5	4	3	2	1	0	Initial value
Address 000099 _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	
EHVR	bit	15	14	13	12	11	10	9	8	Initial value
Address 000098 _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	

The external interrupt request level setting register (ELVR, EHVR) selects how a request is detected. Two bits are assigned to each of INT0 to 5, which results in the settings shown in Table 10.2-1 "External Interrupt Request Level Assignment". Even though the bits of the EIRR are cleared while the request input is a level, the pertinent bits are set again as long as the input is an active level.

Table 10.2-1 External Interrupt Request Level Assignment

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.

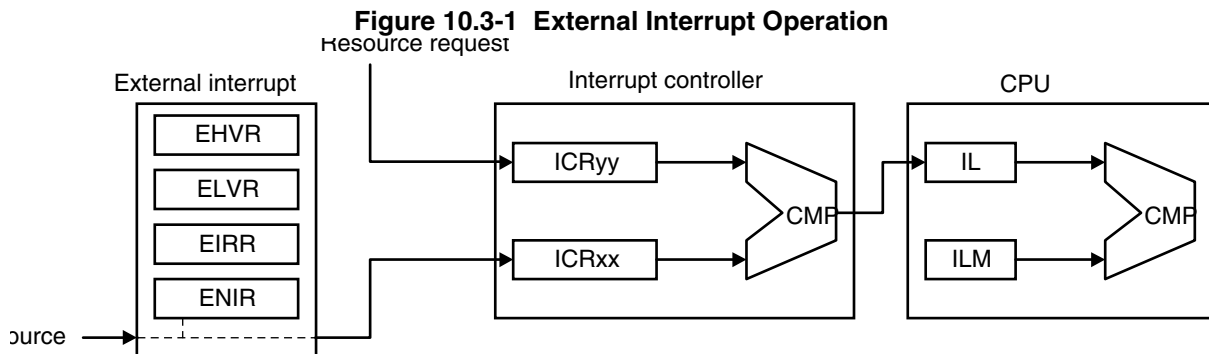
10.3 Operation of the External Interrupt Controller

After a request level and an enable register are defined, if a request defined in the external interrupt request level setting registers (ELVR and EHVR) is input to the corresponding pin, this module generates an interrupt request signal to the interrupt controller.

■ Operation of an External Interrupt

For simultaneous interrupt requests from resources, the interrupt controller determines the interrupt request with the highest priority and generates an interrupt for it.

Figure 10.3-1 "External Interrupt Operation" shows external interrupt operation.



■ Return from the Stopped State

To use an external interrupt to return from the stopped state in clock stop mode, use an H-level request as the input request. If you use an L-level request, a malfunction may occur.

If you use an edge request, the device does not return from the stop state in clock stop mode.

■ Operating Procedure for an External Interrupt

Set up a register located inside the external interrupt controller as follows:

1. Disable the target bit in the enable register.
2. Set the target bit in the request level setting register.
3. Clear the target bit in the interrupt source register.
4. Enable the target bit in the enable register.

Simultaneous writing of 16-bit data is supported for steps 3) and 4).

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 three machine cycles (peripheral clock machine cycles) is required to detect an edge.

If the request level is a level setting and request input arrives from outside and is then cancelled, the request to the interrupt controller remains active because a source holding circuit exists internally.

The interrupt source register must be cleared to cancel a request to the interrupt controller.

Figure 10.3-2 "Clearing the Source Holding Circuit when a Level is Set" shows clearing of the source holding circuit when a level is set. Figure 10.3-3 "Interrupt Source and Interrupt Request to Interrupt Controller when Interrupts are Enabled" shows an interrupt source and an interrupt request to the interrupt controller when interrupts are enabled.

Figure 10.3-2 Clearing the Source Holding Circuit when a Level is Set

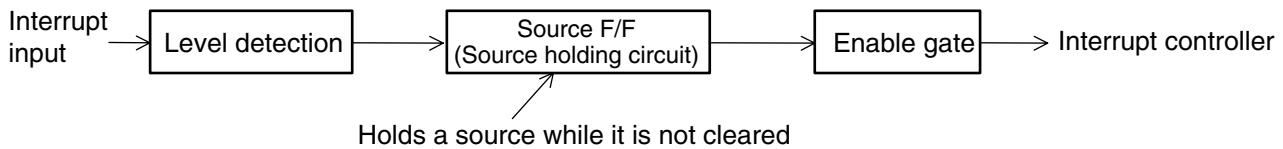
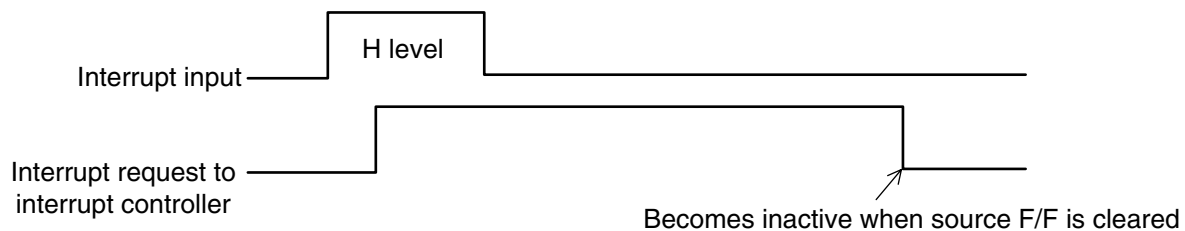


Figure 10.3-3 Interrupt Source and Interrupt Request to Interrupt Controller when Interrupts are Enabled



CHAPTER 11 DELAYED INTERRUPT MODULE

This chapter outlines the delayed interrupt module and describes the register configuration/functions and operations of the delayed interrupt module.

11.1 "Outline of the Delayed Interrupt Module"

11.2 "Delayed Interrupt Module Register (DICR)"

11.3 "Delayed Interrupt Module Operation"

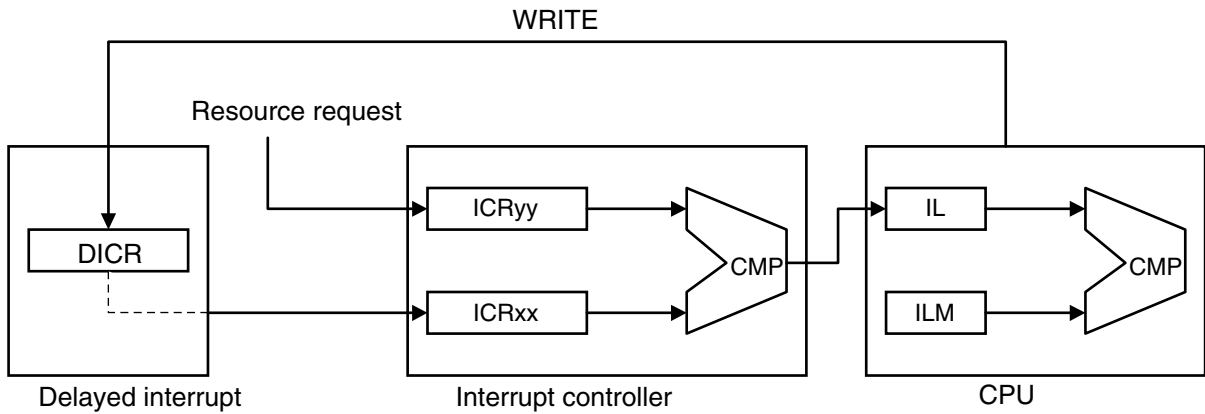
11.1 Outline of the Delayed Interrupt Module

The delayed interrupt module generates interrupts for switching tasks. This module enables using the software to generate and cancel interrupt requests for the CPU.

■ Block Diagram of the Delayed Interrupt Module

Figure 11.1-1 "Block Diagram of the Delayed Interrupt Module" shows a block diagram of the delayed interrupt module.

Figure 11.1-1 Block Diagram of the Delayed Interrupt Module



11.2 Delayed Interrupt Module Register

This section explains the configuration and functions of the register used in the delayed interrupt module.

■ Delayed Interrupt Module Registers

The delayed interrupt module contains the delayed interrupt control register (DICR).

Figure 11.1-1 "Delayed Interrupt Module Registers" lists the registers of the delayed interrupt module.

Figure 11.2-1 Delayed Interrupt Module Registers

bit	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	DLY1	Delayed interrupt control register (DICR)

■ DICR Configuration

The delayed interrupt control register (DICR: Delayed Interrupt Control Register) controls delayed interrupts.

Figure 11.2-2 "Bit Configuration of Delayed Interrupt Control Register (DICR)" shows the bit configuration of the delayed interrupt control register (DICR).

Figure 11.2-2 Bit Configuration of Delayed Interrupt Control Register (DICR)

	bit	7	6	5	4	3	2	1	0	
Address 00000430 _H		—	—	—	—	—	—	—	DLY1	Initial value -----0 _B
										R/W

The functions of bits of the DICR are explained below.

[Bit 0] DLY1

DLY1	Function
0	Release delayed interrupt source/No request (initial value)
1	Generate delayed interrupt source

11.3 Delayed Interrupt Module Operation

Delayed interrupts are generated for switching tasks. This function enables generating and cancelling interrupt requests for the CPU by the software.

■ Interrupt Number

A delayed interrupt is assigned to the interrupt source corresponding to the highest interrupt number.

For the MB91F127 and 128, a delayed interrupt is assigned to interrupt number 63 (3F_H).

■ DICR DLYI Bit

Setting this bit to "1" generates a delayed interrupt source. Setting it to "0" releases the delayed interrupt source.

This bit is the same as the interrupt source flag for general interrupts. Clear this bit via the interrupt routine and switch tasks.

CHAPTER 12 INTERRUPT CONTROLLER

This chapter describes the interrupt controller, the configuration and functions of registers, and interrupt controller operation. It also presents an example of using the hold request cancellation request function.

12.1 "Overview of the Interrupt Controller"

12.2 "Interrupt Controller Registers"

12.3 "Interrupt Controller Operation"

12.4 "Example of Using the Hold Request Cancellation Request Function (HRCR)"

12.1 Overview of the Interrupt Controller

The interrupt controller controls interrupt acceptance and arbitration processing.

■ Hardware Configuration of the Interrupt Controller

The interrupt controller consists of the following components:

- Interrupt control register (ICR)
- Interrupt priority decision circuit
- Interrupt level and interrupt number (vector) generator
- HOLD request cancellation request generator

■ Major Functions

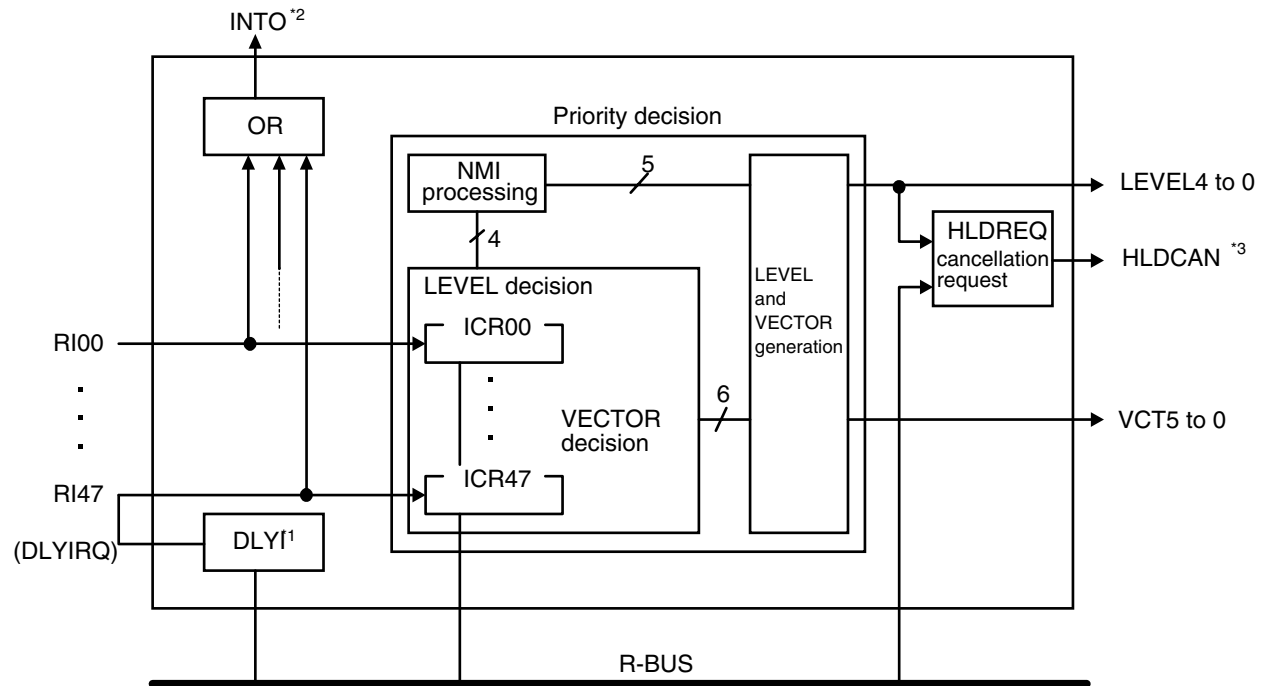
The interrupt controller has the following major functions:

- Detecting interrupt requests
- Deciding priority (using a level or number)
- Passing to the CPU an interrupt level based on the decision result to provide information about the interrupt source
- Passing to the CPU an interrupt number based on the decision result to provide information about the interrupt source
- Instruction to return from stop mode due to the occurrence of an interrupt
- Generating a HOLD request cancellation request for the bus master

■ Block Diagram

Figure 12.1-1 "Block Diagram of the Interrupt Controller" is a block diagram of the interrupt controller.

Figure 12.1-1 Block Diagram of the Interrupt Controller



*1: DLYI is the delayed interrupt section. (For details, see Chapter 9, "Delayed Interrupt Module.")

*2: INTO is a wakeup signal to the clock controller in the sleep or stop state.

*3: HLD CAN is a bus transfer request signal to a bus master other than the CPU bus master.

12.2 Interrupt Controller Registers

This section describes the configuration and functions of the registers used by the interrupt controller.

■ Interrupt Controller Registers

Figure 12.2-1 "Interrupt Controller Registers (Continued on next page)" shows the registers used by the interrupt controller.

Figure 12.2-1 Interrupt Controller Registers (Continued on next page)

bit	7	6	5	4	3	2	1	0	Register name
Address 00000400 _H				ICR4	ICR3	ICR2	ICR1	ICR0	ICR00
Address 00000401 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01
Address 00000402 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02
Address 00000403 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03
Address 00000404 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04
Address 00000405 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05
Address 00000406 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06
Address 00000407 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07
Address 00000408 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR08
Address 00000409 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR09
Address 0000040A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR10
Address 0000040B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR11
Address 0000040C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR12
Address 0000040D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR13
Address 0000040E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR14
Address 0000040F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR15
Address 00000410 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR16
Address 00000411 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR17
Address 00000412 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR18
Address 00000413 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR19
Address 00000414 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR20
Address 00000415 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR21
Address 00000416 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR22
Address 00000417 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR23
Address 00000418 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR24
Address 00000419 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR25
Address 0000041A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR26
Address 0000041B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR27
Address 0000041C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR28
Address 0000041D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR29
Address 0000041E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR30
Address 0000041F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR31

bit	7	6	5	4	3	2	1	0	Register name
Address 00000420 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR32
Address 00000421 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR33
Address 00000422 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR34
Address 00000423 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR35
Address 00000424 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR36
Address 00000425 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR37
Address 00000426 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38
Address 00000427 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR39
Address 00000428 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR40
Address 00000429 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR41
Address 0000042A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR42
Address 0000042B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR43
Address 0000042C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR44
Address 0000042D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR45
Address 0000042E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR46
Address 0000042F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR47
Address 00000431 _H	-	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	HRCL

12.2.1 Interrupt Control Register (ICR)

An interrupt control register is provided for each of the interrupt input and sets the interrupt level of the corresponding interrupt request.

■ ICR Configuration

Figure 12.2-2 "Bit Configuration of the Interrupt Control Register (ICR)" shows the bit configuration of the interrupt control register (ICR: Interrupt Control Register).

Figure 12.2-2 Bit Configuration of the Interrupt Control Register (ICR)

bit	7	6	5	4	3	2	1	0	Initial value
	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---11111 _B
				R	R/W	R/W	R/W	R/W	

The following describes the functions of the interrupt control register (ICR) bits.

[Bits 4 to 0] ICR4 to 0



These bits, which are the interrupt level setting bits, specify the interrupt level of the corresponding interrupt request.

If an interrupt request has an interrupt level defined in this register that exceeds the level mask value defined in the ILM register of the CPU, it is masked by the CPU.

These bits are initialized to 11111_B by a reset.

Table 12.2-1 "Correspondence Between Possible Interrupt Level Setting Bits and Interrupt Levels" shows the correspondence between possible interrupt level setting bits and interrupt levels.

Table 12.2-1 Correspondence Between Possible Interrupt Level Setting Bits and Interrupt Levels

ICR4	ICR3	ICR2	ICR1	ICR0	Interrupt level		
0	0	0	0	0	0	 Reserved for system	
0	1	1	1	0	14		
0	1	1	1	1	15		
1	0	0	0	0	16	Maximum level that can be set	
1	0	0	0	1	17	 (High) <	

ICR4 is fixed to "1". "0" cannot be written to ICR4.

12.2.2 Hold Request Cancellation Request Level Setting Register (HRCL)

The hold request cancellation request level setting register (HRCL) is a level setting register used to generate a hold request cancellation request.

■ Hold Request Cancellation Request Level Setting Register (HRCL)

Figure 12.2-3 "Bit Configuration of the Hold Request Cancellation Request Level Setting Register (HRCL)" shows the bit configuration of the hold request cancellation request level setting register (HRCL).

Figure 12.2-3 Bit Configuration of the Hold Request Cancellation Request Level Setting Register (HRCL)

bit	7	6	5	4	3	2	1	0	Initial value
Address 00000431 _H	-	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	---11111 _B
				R	R/W	R/W	R/W	R/W	

The following describes the functions of the hold request cancellation request level setting register (HRCL) bits.

[Bits 4 to 0] LVL4 to 0

This bit sets the interrupt level used to issue a hold request cancellation request to the bus master.

If an interrupt request with a higher level than the level defined in the HRCL register occurs, issue a hold request cancellation request to the bus master.

The LVL4 bit is always 1; 0 cannot be written to it.

12.3 Interrupt Controller Operation

This section describes the following items regarding operation of the interrupt controller:

- Priority decision
- Return from standby mode (stop/sleep)
- Hold request cancellation request

■ Priority Decision

The interrupt controller selects the interrupt source with the highest priority from among those that exist simultaneously and outputs the interrupt level and the interrupt number of this source to the CPU.

The following shows the priority decision criteria for interrupt sources:

- Source that meets the following conditions:
 - Source with a value other than 31 as the interrupt level (31 means interrupts disabled)
 - Source with the smallest value for the interrupt level
 - Source with the smallest interrupt number that satisfies the both conditions above

Table 12.3-1 "Relationship Between Interrupt Sources, Interrupt Numbers, and Interrupt Levels" shows the relationship between interrupt sources, interrupt numbers

Table 12.3-1 Relationship Between Interrupt Sources, Interrupt Numbers, and Interrupt Levels

Interrupt source	Interrupt number		Interrupt level	Offset	Default address of TBR
	Decimal	Hexadecimal			
Reserved for system	15	0F	15(F _H) Fixed	3C0 _H	000FFFC0 _H
External interrupt 0	16	10	ICR00	3BC _H	000FFFBC _H
External interrupt 1	17	11	ICR01	3B8 _H	000FFFB8 _H
External interrupt 2	18	12	ICR02	3B4 _H	000FFFB4 _H
External interrupt 3	19	13	ICR03	3B0 _H	000FFFB0 _H
UART0 reception completed	20	14	ICR04	3AC _H	000FFFAC _H
UART1 reception completed	21	15	ICR05	3A8 _H	000FFFA8 _H
UART2 reception completed	22	16	ICR06	3A4 _H	000FFFA4 _H
UART0 transmission completed	23	17	ICR07	3A0 _H	000FFFA0 _H
UART1 transmission completed	24	18	ICR08	39C _H	000FFF9C _H

CHAPTER 12 INTERRUPT CONTROLLER

Table 12.3-1 Relationship Between Interrupt Sources, Interrupt Numbers, and Interrupt Levels

Interrupt source	Interrupt number		Interrupt level	Offset	Default address of TBR
	Decimal	Hexadecimal			
UART2 transmission completed	25	19	ICR09	398 _H	000FFF98 _H
DMAC0 (end, error)	26	1A	ICR10	394 _H	000FFF94 _H
DMAC1 (end, error)	27	1B	ICR11	390 _H	000FFF90 _H
DMAC2 (end, error)	28	1C	ICR12	38C _H	000FFF8C _H
DMAC3 (end, error)	29	1D	ICR13	388 _H	000FFF88 _H
DMAC4 (end, error)	30	1E	ICR14	384 _H	000FFF84 _H
DMAC5 (end, error)	31	1F	ICR15	380 _H	000FFF80 _H
DMAC6 (end, error)	32	20	ICR16	37C _H	000FFF7C _H
DMAC7 (end, error)	33	21	ICR17	378 _H	000FFF78 _H
A/D	34	22	ICR18	374 _H	000FFF74 _H
Reload timer 0	35	23	ICR19	370 _H	000FFF70 _H
Reload timer 1	36	24	ICR20	36C _H	000FFF6C _H
Reload timer 2	37	25	ICR21	368 _H	000FFF68 _H
External interrupt 4	38	26	ICR22	364 _H	000FFF64 _H
External interrupt 5	39	27	ICR23	360 _H	000FFF60 _H
Reserved for system	40	28	ICR24	35C _H	000FFF5C _H
Reserved for system	41	29	ICR25	358 _H	000FFF58 _H
U-TIMER 0	42	2A	ICR26	354 _H	000FFF54 _H
U-TIMER 1	43	2B	ICR27	350 _H	000FFF50 _H
U-TIMER 2	44	2C	ICR28	34C _H	000FFF4C _H
Flash memory	45	2D	ICR29	348 _H	000FFF48 _H
Reserved for system	46	2E	ICR30	344 _H	000FFF44 _H
Reserved for system	47	2F	ICR31	340 _H	000FFF40 _H
PPG0	48	30	ICR32	33C _H	000FFF3C _H
PPG1	49	31	ICR33	338 _H	000FFF38 _H
PPG2	50	32	ICR34	334 _H	000FFF34 _H
PPG3	51	33	ICR35	330 _H	000FFF30 _H
ICU0 (capture)	52	34	ICR36	32C _H	000FFF2C _H
ICU1 (capture)	53	35	ICR37	328 _H	000FFF28 _H
ICU2 (capture)	54	36	ICR38	324 _H	000FFF24 _H

Table 12.3-1 Relationship Between Interrupt Sources, Interrupt Numbers, and Interrupt Levels

Interrupt source	Interrupt number		Interrupt level	Offset	Default address of TBR
	Decimal	Hexadecimal			
ICU3 (capture)	55	37	ICR39	320 _H	000FFF20 _H
OCU0 (match)	56	38	ICR40	31C _H	000FFF1C _H
OCU1 (match)	57	39	ICR41	318 _H	000FFF18 _H
OCU2 (match)	58	3A	ICR42	314 _H	000FFF14 _H
OCU3 (match)	59	3B	ICR43	310 _H	000FFF10 _H
Reserved for system	60	3C	ICR44	30C _H	000FFF0C _H
16-bit free-run timer	61	3D	ICR45	308 _H	000FFF08 _H
Reserved for system	62	3E	ICR46	304 _H	000FFF04 _H
Delayed interrupt source bit	63	3F	ICR47	300 _H	000FFF00 _H

■ Clearing an Interrupt Source

In the interrupt routine, a restriction applies to coding between the instruction that clears an interrupt source and the RETI instruction.

For more information, see Section 3.9 "Exception, Interrupt, and Trap (EIT)."

■ Return from Standby Mode (Sleep/Stop)

This module implements a function that causes a return from stop mode if an interrupt request occurs. If even one interrupt request occurs from the periphery, a return request from stop mode is generated for the clock controller.

Since the priority decision unit restarts operation when a clock is supplied after returning from stop, the CPU executes instructions until the result of the priority decision unit is obtained.

The same operation occurs after a return from the sleep state. The registers in this module can be accessed by DMAC or other means even in the sleep status.

Note:

For an interrupt source that you do not want to return from the stop or sleep state, prohibit interrupt request output in the corresponding peripheral control register. Since a request signal for return from standby is simply a logical OR output for all interrupt sources, content about the interrupt level set in ICR is not considered.

To perform a DMA transfer during sleep mode, configure the DMA setting so that an interrupt request is not conveyed to this module and an unintended return from sleep mode is not caused.

■ Hold Request Cancellation Request (HRCR)

For an interrupt with a higher priority to be processed during CPU hold, the device that has generated the hold request must cancel the request. Set in the HRCL register the interrupt level to be used as the criterion of generating a cancellation request.

○ Generation criteria

If an interrupt source with a higher interrupt level than the level defined in the HRCL register occurs, a hold request cancellation request is generated.

- If the interrupt level of the HRCL register is greater than the interrupt level after a priority decision, a cancellation request occurs.
- If the interrupt level of the HRCL register is equal to or less than the interrupt level after a priority decision, no cancellation request occurs.

Because the cancellation request remains valid, no DMA transfer occurs unless the interrupt source that has caused the cancellation request is cleared. Be sure to clear the corresponding interrupt source.

○ Possible levels

Values that can be set in the HRCL register range from 10000_B to 11111_B, which is the same range as for the ICR.

If this register is set to 11111_B, a cancellation request is issued for all interrupt levels.

Table 12.3-2 "Settings of Interrupt Levels at which Hold Request Cancellation Request Occurs"

shows the settings of interrupt levels at which a hold request cancellation request occurs.

Table 12.3-2 Settings of Interrupt Levels at which Hold Request Cancellation Request Occurs

HRCL register	Interrupt levels at which a cancellation request occurs
16	
17	Interrupt level 16
18	Interrupt levels 16 and 17
–	–
31	Interrupt levels 16 to 30 [initial value]

After a reset, the hold request cancellation state is set for all interrupt levels.

12.4 Example of Using the Hold Request Cancellation Request Function (HRCR)

To allow the CPU to perform high-priority processing during DMA transfer, cancel a hold request for DMA and clear the hold state. In this example, an interrupt is used to cancel a hold request to the DMA, allowing the CPU to perform priority operations.

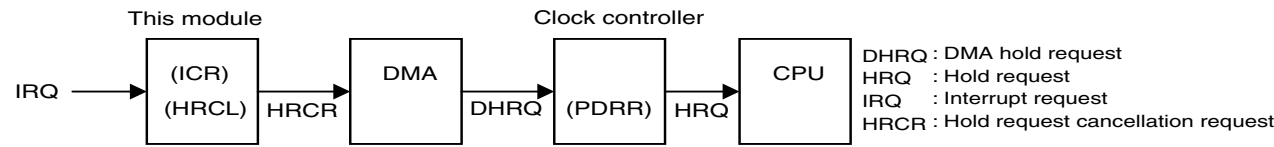
■ Control Registers

- HRCL (hold request cancellation level setting register): If an interrupt with a higher interrupt level than the level in the HRCL register occurs, a hold request cancellation request is generated for DMA. This register sets the level to be used as the criterion for this purpose.
- ICR (interrupt control register): This register sets a level higher than the level in the HRCL register for the ICR corresponding to the interrupt source that will be used.
- DMA request suppress register (PDRR): clock controller:
This register temporarily suppresses a hold request from DMA. This register clears an interrupt source to prevent a return to the hold state again. A hold request from DMA is conveyed to the CPU only while this register is set to 0000_B. To use this register, increment the contents of this register at the entrance of an interrupt routine and decrement it at the exit of the routine.

■ Hardware Configuration

The flow of signals is as follows.

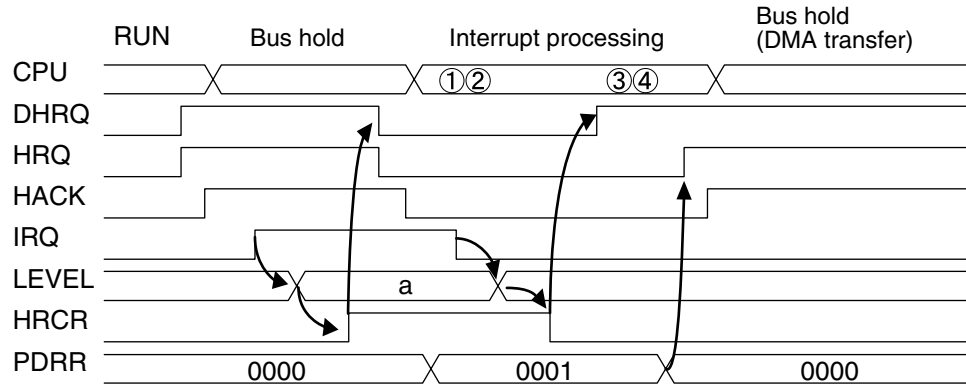
Figure 12.4-1 Sample Hardware Configuration in Which Hardware Hold Request Cancellation Request Function Is Used



■ Hold Request Cancellation Request Sequence

○ Sample interrupt routine

Figure 12.4-2 "Sample Timing Chart of Hold Request Cancellation Request Sequence" shows a sample timing chart showing the sequence of a hold request cancellation request (Write level: HRCL>a).

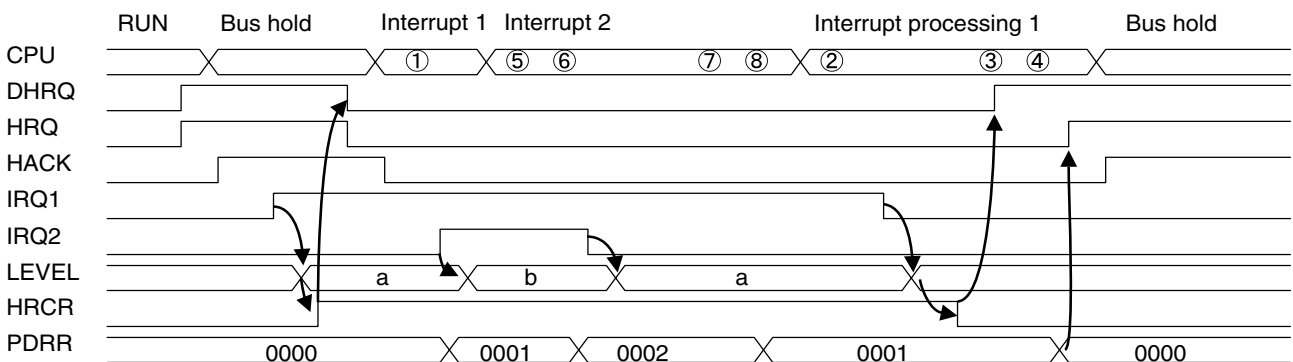
Figure 12.4-2 Sample Timing Chart of Hold Request Cancellation Request Sequence

- ① PDDR increment
- ② Interrupt source clear
- ...
- ③ PDDR decrement
- ④ RETI

If an interrupt request occurs, the interrupt level changes. If the interrupt level is higher than the level in the HRCL register, HRCR is started for DMA. This causes DMA to cancel a hold request and the CPU to return from the hold state to perform the interrupt processing. In an interrupt routine, (1) PDDR is incremented and (2) an interrupt source is cleared. This changes the interrupt level to make HRCR inactive and cause DMA to issue a hold request again. Since PDDR is not 0, this hold request is blocked. The hold request is conveyed to the CPU and DMA transfer begins again only when (3) PDDR is decremented.

○ Sample interrupt routine

Figure 12.4-3 "Sample Timing Chart of Hold Request Cancellation Request Sequence" shows a sample timing chart of the sequence of a hold request cancellation request in a multiple interrupt state (Interrupt level: $HRCL > a > b$).

Figure 12.4-3 Sample Timing Chart of Hold Request Cancellation Request Sequence

- ①, ⑤ PDDR increment
- ②, ⑥ Interrupt source clear
- ③, ⑦ PDDR decrement
- ④, ⑧ RETI

In the Figure 12.4-3 "Sample Timing Chart of Hold Request Cancellation Request Sequence", while Interrupt Routine I is being executed, an interrupt with a higher priority occurs.

As shown in this example too, reception of a hold request can be prevented if PDRR is incremented at the entrance of an interrupt routine and decremented at the exit of the routine.

Note:

- Be sure to increment and decrement PDRR at the entrance and exit, respectively, of an interrupt routine that is not to be processed during a DMA transfer (during CPU hold). Otherwise, the DMA transfer occurs again in the middle of an interrupt routine.
- Conversely, do not increment and decrement PDRR for a normal interrupt routine. Otherwise, DMA transfers cannot occur while the interrupt routine is executed, and performance deteriorates.
- Be especially careful about the relationship between interrupt levels defined in the HRCL register and ICR.

CHAPTER 13 A/D CONVERTER (SUCCESSIVE APPROXIMATION TYPE)

This chapter outlines the A/D converter, describes the register configuration, and the functions and operations of the A/D converter.

13.1 "Outline of the A/D Converter"

13.2 "A/D Converter Registers"

13.3 "A/D Converter Operation"

13.4 "Conversion Data Protection Function"

13.5 "Notes on Using the A/D Converter"

13.1 Outline of the A/D Converter

The A/D converter uses the successive approximation conversion method to convert input voltage from analog to digital values.

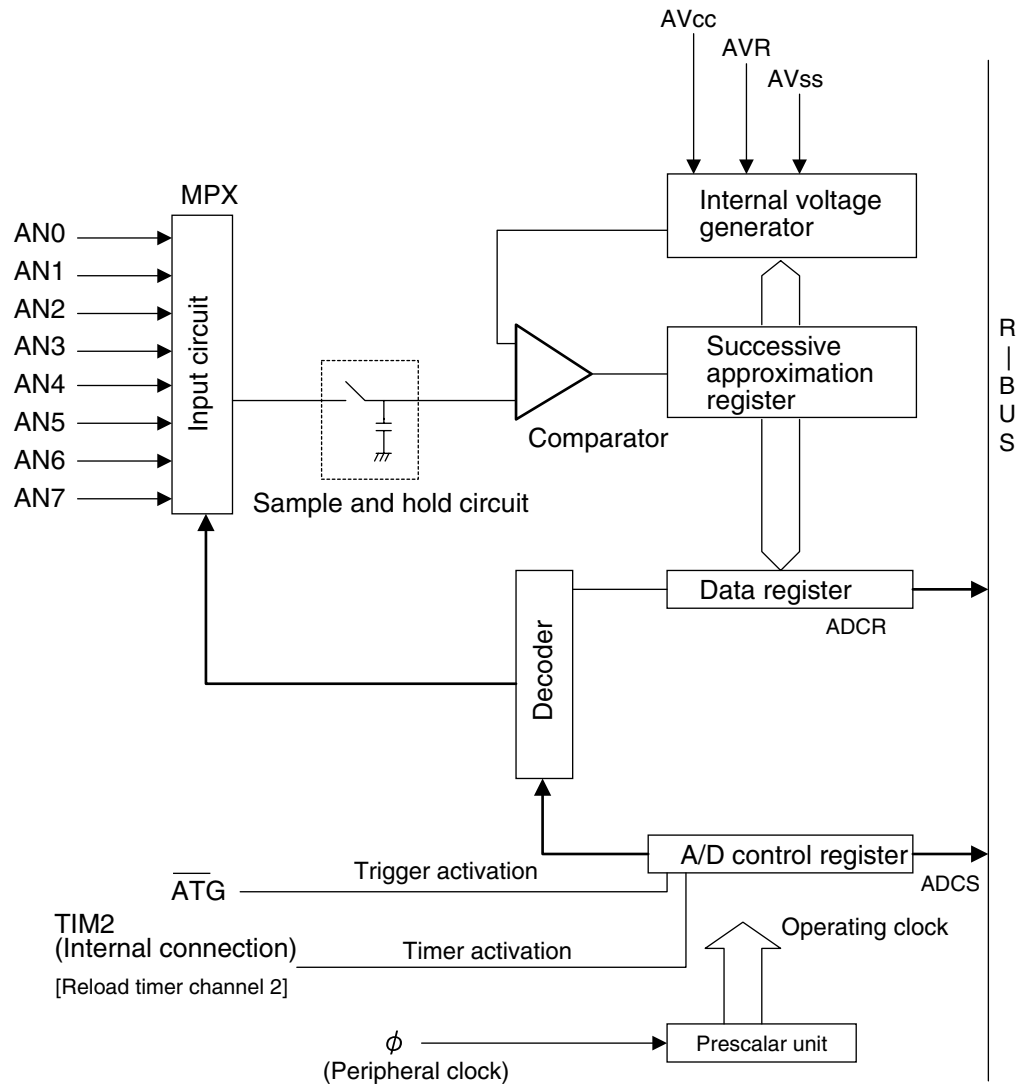
■ A/D Converter Features

- Minimum conversion time: 5.2 μ s per channel (at a system clock frequency of 25 MHz)
- Built-in sample and hold circuit
- Resolution: 10 bits
- Analog input can be selected from four channels using a program.
 - Single conversion mode: One channel is selected and converted.
 - Scan conversion mode: Multiple channels are continuously converted. Up to four channels can be programmed.
 - Continuous conversion mode: The specified channels are converted repeatedly.
 - Stop conversion mode: Standby function in which conversion stops temporarily after one channel is converted. The program then waits for the next activation. (The start of conversion can be synchronized with this mode.)
- DMA transfer can be activated by interrupt.
- A software interrupt, external trigger (falling edge), or reload timer (rising edge) can be selected as the interrupt source.

■ Block diagram

Figure 13.1-1 "A/D Converter Block Diagram" shows the A/D converter block diagram.

Figure 13.1-1 A/D Converter Block Diagram



13.2 A/D Converter Registers

This section explains the configuration and functions of the registers used in the A/D converter.

■ A/D Converter Register Configuration

Figure 13.2-1 "A/D Converter Registers" shows a list of registers of the A/D converter. Figure 13.2-2 "Register Configuration of A/D Converter" shows the register configuration of the A/D converter.

Figure 13.2-1 A/D Converter Registers

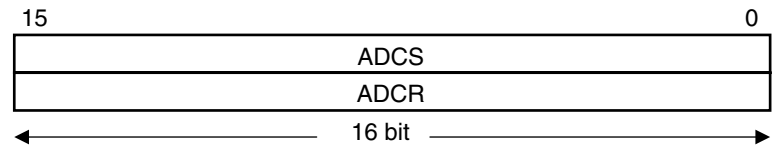


Figure 13.2-2 Register Configuration of A/D Converter

bit	15	14	13	12	11	10	9	8	Control status register (ADCS)
	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	-	
bit	7	6	5	4	3	2	1	0	Data register(ADCR)
	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	
bit	15	14	13	12	11	10	9	8	
	S10	ST1	ST0	CT1	CT0	-	9	8	
bit	7	6	5	4	3	2	1	0	
	7	6	5	4	3	2	1	0	

13.2.1 Control status register (ADCS)

The control status register (ADCS) controls the A/D converter and indicates the A/D converter status.

■ ADCS Configuration

Figure 13.2-3 "Bit Configuration of ADCS" shows the bit configuration of the ADCS.

Figure 13.2-3 Bit Configuration of ADCS

bit	15	14	13	12	11	10	9	8	Initial value
Address 000 0052 _H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	-	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
bit	7	6	5	4	3	2	1	0	Initial value
	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note

Do not rewrite the control status register (ADCS) while A/D conversion is in progress. Furthermore, do not use a read-modify-write instruction to access it.

■ Bit Details of ADCS

The functions of bits of the ADCS are explained below.

[Bit 15] BUSY (BUSY flag and stop)

This bit has different functions during read and write operations.

- During read operations:

This bit is used to indicate A/D converter operation. This bit is set when A/D conversion starts and cleared when A/D conversion ends.

- During write operations:

Setting this bit to "0" during A/D operation forcibly stops the operation. This bit is used to forcibly stop operation in continuous and stop conversion modes.

The bit for indicating A/D operation cannot be set to "1". However, RMW instructions always read "1". In single mode, the bit is cleared after the termination of A/D conversion. In continuous and stop modes, the bit is not cleared before it is set to "0" to stop operation.

A reset initializes the bit to "0".

Do not execute forced stop and software activation simultaneously (BUSY = 0 and STRT = 1).

[Bit 14] INT (INTerrupt)

This bit is used to indicate data. It is set when conversion data is written to the ADCR.

When INTE (bit 13) is "1", setting the INT bit will generate an interrupt request. In addition,

CHAPTER 13 A/D CONVERTER (SUCCESSIVE APPROXIMATION TYPE)

the DMA will be activated if activation of DMA transfer has been selected. Setting this bit to "1" has no effect.

The clear signal from the DMAC is used when this bit is set to "0" during a clear operation.

Note:

Set this bit to "0" to clear it while the A/D converter is stopped. If this bit is set to "1", it is initialized to "0". A read-modify-write instruction reads "1" from this bit.

[Bit 13] INTE (INTerrupt Enable)

This bit is used to enable or disable interrupts at the end of conversion.

INTE	Function
0	Interrupts prohibited (Initial value)
1	Interrupts enabled

Set this bit when DMA transfer is to be started by an interrupt request. A reset initializes the bit to "0".

[Bit 12] PAUS (A/D converter PAUSE)

This bit is set when A/D conversion stops temporarily.

Because there is only one register to store the A/D conversion results, the conversion results must be transferred by the DMA when continuous mode is used, otherwise the previous data item will be overwritten.

To protect the previous data item, the next conversion data item is not stored until the data register contents are transferred by the DMA. A/D conversion is suspended during this time.

A/D conversion restarts when DMA transfer ends.

This bit is valid only when the DMA is used.

For details, refer to 13.4 "Conversion Data Protection Function".

A reset initializes the bit to "0".

[Bits 11 and 10] STS1 and STS0 (STart Source select)

A reset initializes these bits to "00".

These bits are used to select the A/D converter start source.

Table 13.2-1 A/D Converter Start Source Selection

STS1	STS0	Function
0	0	Software activation
0	1	External pin trigger activation or software activation
1	0	Timer activation or software activation
1	1	External pin trigger activation, software activation, or timer activation

For modes that have multiple start sources, the A/D converter is activated by the first of

these sources.

Because these bits are overwritten when the start source changes, exercise caution when changing the start source during A/D converter operation.

Note:

- For the external pin trigger, the falling edge is detected. If the external trigger input level is the "L" level, the A/D converter can be activated when this bit is rewritten to set external trigger activation.
- Selecting the timer selects channel 2 of the reload timer. If the reload timer output level is the "H" level, the A/D converter can be activated when this bit is rewritten to set timer activation.

[Bit 9] STRT (STaRT)

Setting this bit to "1" activates the A/D converter.

To restart the A/D converter, set this bit to "1" again.

In stop mode, the operating function cannot be restarted.

A reset initializes this bit to "0".

Do not execute forced stop and software activation simultaneously (BUSY = 0 and STRT = 1).

Reading operations using read-modify-write instructions always return "0".

[Bit 8] (Reserved)

This is a test bit. When writing, always set it to "0".

[Bits 7 and 6] MD1 and MD0 (A/D converter MoDe set)

These bits are used to select the operation mode.

Table 13.2-2 "A/D Converter Operation Mode Selection" shows settings for selecting the A/D converter operation mode.

Table 13.2-2 A/D Converter Operation Mode Selection

MD1	MD0	Operation mode
0	0	Single mode. All types of restarts are enabled during operation.
0	1	Single mode. Restarts are disabled during operation.
1	0	Continuous mode. Restarts are disabled during operation.
1	1	Stop mode. Restarts are disabled during operation.

- Single mode: A/D conversion is performed continuously for the channels set by ANS2 to ANS0 up to the channels set by ANE2 to ANE0, but operation stops in between conversion operations.
- Continuous mode: A/D conversion is performed repeatedly and continuously for the channels set by ANS2 to ANS0 up to the channels set by ANE2 to ANE0 without interruption.
- Stop mode: A/D conversion is performed for the channels set by ANS2 to ANS0 up to the channels set by ANE2 to ANE0, but operation stops temporarily for each channel.

A reset initializes these bits to "00_B".

Note:

When A/D conversion is activated in continuous and stop modes, the conversion operation continues until being stopped by the BUSY bit.

To stop the conversion operation, set the BUSY bit to "0".

A/D conversion cannot be restarted in single, continuous, or stop mode when the timer, external trigger, or a software interrupt have been selected for activating the A/D converter.

[Bits 5, 4, and 3] ANS2, ANS1, and ANS0 (ANalog Start channel set)

This bit group is used to set the start channel of A/D conversion.

When the A/D converter is activated, A/D conversion starts with the channel selected by these bits.

Table 13.2-3 "Setting A/D Conversion Start Channels" shows settings for specifying the start channel of A/D conversion.

Table 13.2-3 Setting A/D Conversion Start Channels

ANS2	ANS1	ANS0	Start channel
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4

Table 13.2-3 Setting A/D Conversion Start Channels (Continued)

ANS2	ANS1	ANS0	Start channel
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

During read operations, the conversion channels selected by this bit group are read for during A/D conversion.

During stops in stop mode, the conversion channels that were previously selected by this bit group are read.

A reset initializes these bits to "000_B".

[Bits 2, 1, and 0] ANE2, ANE1, and ANE0 (ANalog End channel set)

This bit group is used to set the end channels of A/D conversion.

Table 13.2-4 "Setting A/D Conversion End Channels" shows settings for specifying the end channel of A/D conversion.

Table 13.2-4 Setting A/D Conversion End Channels

ANE2	ANE1	ANE0	Start channel
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Setting the same channels as those of ANS2 to ANS0 results in one-channel conversion (single conversion).

When continuous or stop mode has been set, processing returns to the start channels set by ANS2 to ANS0 as soon as conversion of the channels set by ANE2 to ANE0 ends.

When setting the channels, always ensure that ANS is less than or equal to ANE.

A reset initializes these bits to "000_B".

[Example]

Operating in single mode using channel settings ANS = 1 channel, ANE = 3 channels

Channel conversion: 1 channel --> 2 channels --> 3 channels

13.2.2 Data register (ADCR)

The data register (ADCR) is a conversion storage register for storing the digital value that is the result of conversion. ADCR also has a function for selecting the resolution of A/D conversion.

■ ADCR Conversion

Figure 13.2-4 "Bit Configuration of ADCR" shows the bit configuration of the ADCR.

Figure 13.2-4 Bit Configuration of ADCR

bit	15	14	13	12	11	10	9	8	Initial value
Address 0000 0050 _H	S10	ST1	ST0	CT1	CT0	-	9	8	000000XX _B
	R	R	R	R	R	R	R	R	
bit	7	6	5	4	3	2	1	0	Initial value
	7	6	5	4	3	2	1	0	XXXXXXXX _B
	R	R	R	R	R	R	R	R	

The value of this register is updated each time conversion ends, and the final conversion value is stored.

The contents of this register after a reset is undefined. (except for high-order bits 15 to 10).

The functions of bits of the data register (ADCR) are explained below.

[Bit 15] S10 (Select 10-bit or 8-bit resolution)

This bit is used to select the resolution of A/D conversion.

S10	Function
0	10-bit resolution (D9 to D0) (initial value)
1	8-bit resolution (D7 to D0)

Depending on the resolution, different data bits are used.

[Bits 14 and 13] ST1, ST0 (Sampling Time)

These bits are used to select the sampling time for A/D conversion.

When A/D is activated, analog input is captured during the time specified in these bits.

Table 13.2-5 "Sampling Time Setting of A/D Converter" shows the sampling time setting of the A/D converter.

Table 13.2-5 Sampling Time Setting of A/D Converter

ST1	ST0	Sampling time	
0	0	11 * ϕ	1.4 μ s @8MHz
0	1	23 * ϕ	1.4 μ s @16MHz

Table 13.2-5 Sampling Time Setting of A/D Converter (Continued)

ST1	ST0	Sampling time	
1	0	33 * ϕ	1.3 μ s @25MHz
1	1	45 * ϕ	

[Bits 12 and 11] CT1 and CT0 (Conversion Time)

These bits are used to select the compare time during A/D conversion.

After analog input is captured (the sampling time has passed), conversion results data obtains a conclusion during the time specified in these bits and then stored into bits 9 to 0 of this register.

Table 13.2-6 "Compare Time Setting for A/D Converter" shows the compare time setting for the A/D converter.

Table 13.2-6 Compare Time Setting for A/D Converter

CT1	CT0	Compare time	
0	0	34 * ϕ	4.3 μ s @8MHz
0	1	67 * ϕ	4.2 μ s @16MHz
1	0	100 * ϕ	4.0 μ s @25MHz
1	1	122 * ϕ	

[Bit 10] Vacant bit**[Bits 9 to 0] D9 to D0 (Data register)**

The results of A/D conversion are stored in these bits. The register is rewritten every time conversion ends.

Normally, the final conversion value is stored. The initial value of the register is undefined.

A conversion data protection function is provided.

Do not write data to these bits during A/D conversion.

13.2.3 Analog Input Control Register (AIC)

The analog input control register (AIC) is used to select analog input.

■ Configuration of the Analog Input Control Register (AIC)

Figure 13.2-5 "Bit Configuration of Analog Input Control Register (AIC)" shows the bit configuration of the analog input control register (AIC).

Figure 13.2-5 Bit Configuration of Analog Input Control Register (AIC)

bit	7	6	5	4	3	2	1	0	Initial value
Address 0000D5 _H	AI7	AI6	AI5	AI4	AI3	AI2	AI1	AI0	1111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The functions of bits of the analog input control register (AIC) are explained below.

[Bits 0 to 7] AI0 to AI7 (Analog Input control register)

These bits control the pins of the corresponding I/O port as follows:

A10	Function
0	Analog input mode
1	Port mode (initial value)

Note

A leakage current occurs if an intermediate level of voltage is input while port input mode is set. To use analog input, be sure to specify the setting of analog input.

13.3 A/D Converter Operation

The A/D converter employs the successive approximation method and has a resolution of 10 bits.

Because the A/D converter has only one (16-bit) register for storing the conversion results, the conversion data register (ADCR) will be updated when one conversion operation ends.

For continuous conversion, DMA transfer can be used.

The A/D converter operates in one of three modes: single conversion mode, continuous conversion mode, and stop conversion mode. The operations in each mode are explained below.

■ One Conversion Mode

In the one conversion mode, the analog input defined by the ADCS ANS and ANE bits is converted sequentially. The A/D converter stops when conversion up to the end channel set by the ANE bit ends.

If the start and end channels are the same (ANS = ANE), one-channel conversion will be performed.

Examples:

- For ANS = 000_B and ANE = 011_B
Start --> AN0 --> AN1 --> AN2 --> AN3 --> End
- For ANS = 010_B and ANE = 010_B
Start --> AN2 --> End

■ Continuous Conversion Mode

In the continuous conversion mode, the analog input set by the ADCS ANS and ANE bits is converted sequentially. When conversion up to the end channel set by the ANE bit ends, processing returns to analog input of ANS, and A/D conversion continues.

If the start and end channels are the same (ANS = ANE), operation continues with one-channel conversion.

Examples:

- For ANS = 000_B and ANE = 011_B
Start --> AN0 --> AN1 --> AN2 --> AN3 --> AN0 ... --> Repeated
- For ANS = 010_B and ANE = 010_B
Start --> AN2 --> AN2 --> AN2 ... --> Repeated

For conversion in continuous mode, conversion will be repeated until the BUSY bit is set to "0". Setting the BUSY bit to "0" forcibly stops the operation.

Forcibly stopping the operation will stop the currently performed conversion of the analog input.

When the operation is forcibly stopped, the data before completion of conversion will be stored in the conversion register.

■ Stop Conversion Mode

In stop mode, the analog input set by the ADCS ANS and ANE bits is converted sequentially, but conversion stops temporarily between conversions for a single channel. To release the temporary stop, activate A/D conversion again.

When conversion up to the end channel set by the ANE bit ends, processing returns to analog input of ANS, and A/D conversion continues.

If the start and end channels are the same (ANS = ANE), one-channel conversion will be performed.

Examples:

- For ANS = 000_B and ANE = 011_B
 Start --> AN0 --> Stop --> Start --> AN1 --> Stop --> Start --> AN2 --> Stop -->
 Start --> AN3 --> Stop --> Start --> AN0 ... --> Repeated
- For ANS = 010_B and ANE = 010_B
 Start --> AN2 --> Stop --> Start --> AN2 --> Stop --> Start --> AN2 ... --> Repeated

Only the start source set by the STS1 and STS0 bits are used at this time.

Using stop mode enables synchronizing the start of conversion.

13.4 Conversion Data Protection Function

The A/D converter has a conversion data protection function. This function uses the DMAC to enable continuous conversion and saving of multiple data items.

■ Conversion Data Protection Function

Because there is only one conversion data register, continuous A/D conversion can result in the loss of the previously stored data item when the new conversion result is stored at the end of a conversion operation. To protect the previous data, the A/D converter has a function for temporarily stopping A/D conversion without storing the conversion data to the register. A/D conversion is stopped until the previous data item has been transferred to memory using the DMAC even if the A/D converter has completed conversion.

The temporary stop is released after the previously stored data item has been transferred to memory by DMA transfer.

If the previously stored data item has been transferred, the A/D converter can continue conversion without performing a temporary stop.

Note:

This function is related to the ADCS INT and INTE bits.

The data protection function operates only in interrupt enable state (INTE = 1).

The data protection function will not operate in interrupt prohibited state (INTE = 0). If continuous A/D conversion is performed in this state, the conversion data will be stored continuously to the register and earlier stored data items will be lost.

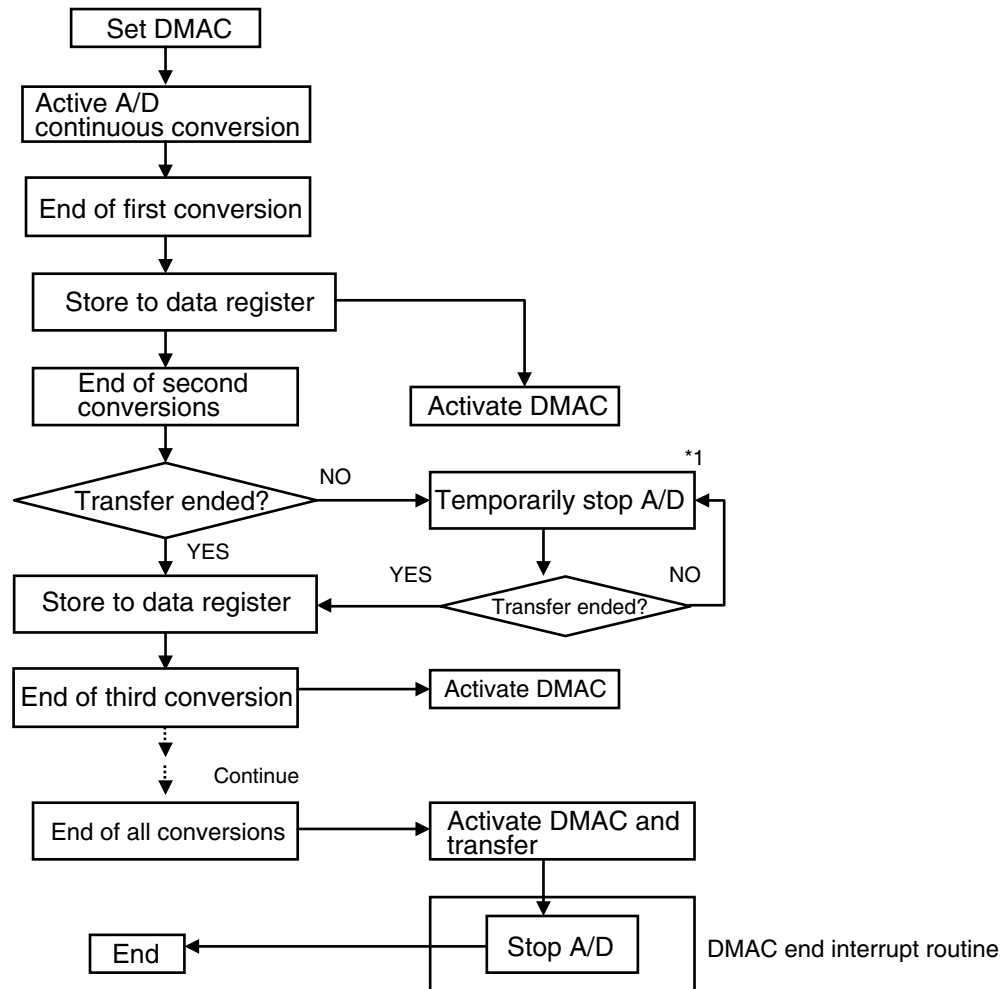
In addition, if DMA transfer is not used in interrupt enable state (INTE = 1), the INT bit will not be cleared and the data protection function will set A/D conversion to temporary stop state. In this case, clearing the INT bit in the interrupt sequence will release the stop state.

During DMA operation, A/D converter operations can change the contents of the conversion data register to change before data is transferred. This applies if interrupts are prohibited when the A/D converter is stopped temporarily.

In addition, the previously stored data item will be overwritten if the A/D converter is restarted during a temporary stop.

Figure 13.4-1 "Flow of Operation of the Data Protection Function when DMA Transfer is Used" shows the flow of operation of the data protection function when DMA transfer is used.

Figure 13.4-1 Flow of Operation of the Data Protection Function when DMA Transfer is Used



* 1: The standby conversion data item will be overwritten if the A/D converter is restarted during temporary stopping.

13.5 Notes on Using the A/D Converter

This section provides notes on using the A/D converter.

■ Notes On Using the A/D Converter

○ Using an external trigger or internal timer to activate the A/D converter

The A/D start source bits (STS1 and STS0) of the ADCS register are used to store the settings for an external trigger or internal timer to activate the A/D converter. To prevent a malfunction during using the A/D converter, set the input values of the external trigger and internal timer. For the STS1 and STS0 settings, ensure the status $\overline{ATG} = 1$ input and reload timer (channel 2) = 0 output. Set the timer to the inactivate side.

■ Other Notes On Using the A/D Converter

If the internal impedance exceeds the specified value, the analog input values will not be sampled within the prescribed sampling time and the correct conversion results will not be obtained.

CHAPTER 14 UART

This chapter outlines the UART and describes the register configuration/functions and operations of the UART.

14.1 "Outline of the UART"

14.2 "UART Registers"

14.3 "UART Operation"

14.4 "Example of Using UART"

14.5 "Sample Settings for Baud Rate and U-TIMER Reload Value"

14.1 Outline of the UART

The UART is a serial I/O port for asynchronous communication and CLK synchronous communication.

The MB91F127/128 has three built-in UART channels.

■ UART Features

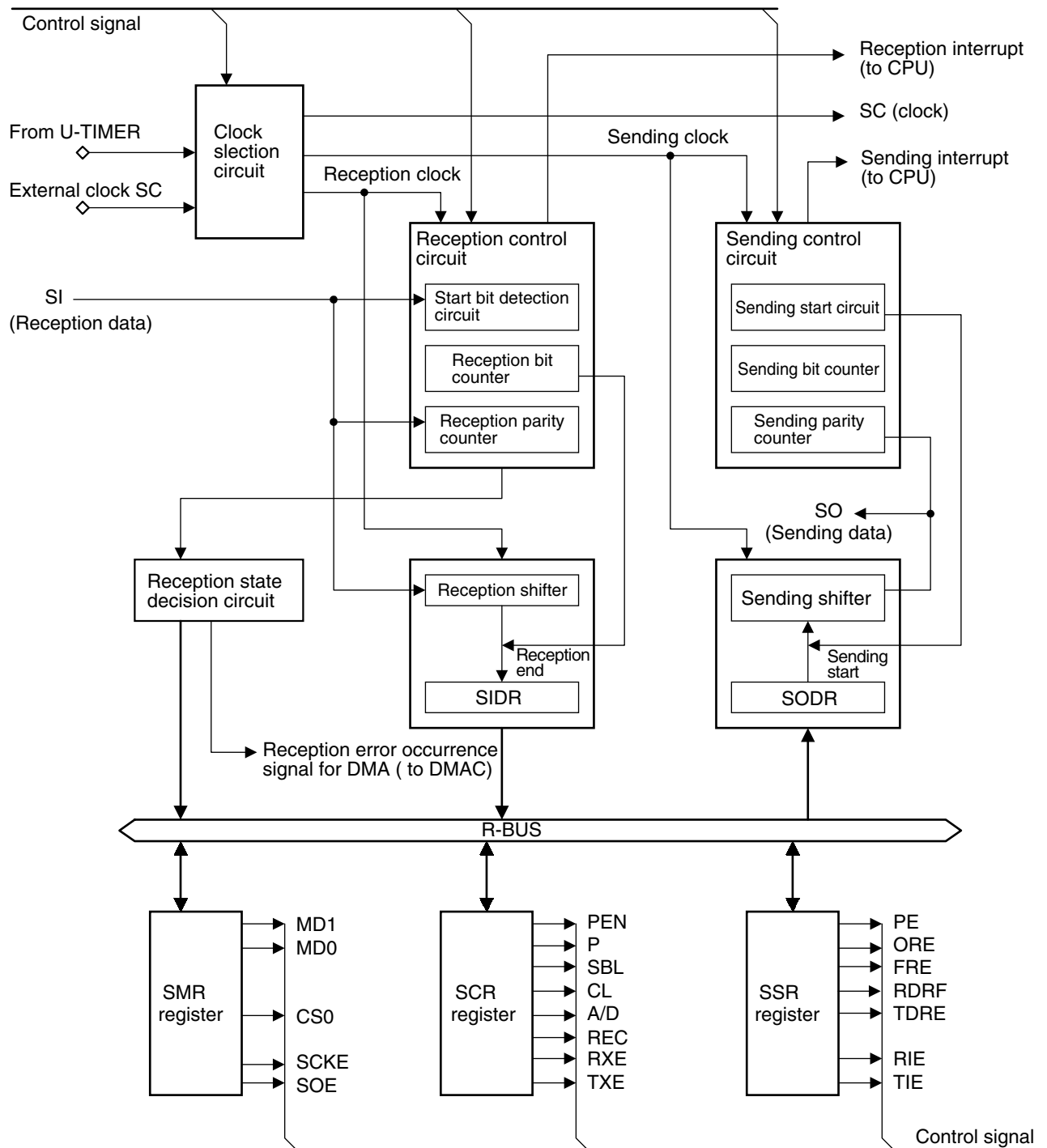
The UART has the following characteristics:

- Full-duplex double buffer
- Support of asynchronous communication and CLK synchronous communication
- Support of multiprocessor mode
- Completely programmable baud rate
 - An optional baud rate can be set using an internal timer. (See Section 9.3 "U-TIMR Operation")
- The baud rate can be freely set using an external clock.
- Error detection function (parity, framing, and overrun)
- NRZ-signed transfer signals
- DMA transfer can be activated by interrupts.

■ Block Diagram

Figure 14.1-1 "UART Block Diagram" shows a block diagram of the UART.

Figure 14.1-1 UART Block Diagram



14.2 UART Registers

This section explains the configuration and functions of registers used for the UART.

■ UART Register Configuration

Figure 14.2-1 "UART Registers" shows the register configuration of the UART. Figure 14.2-2 "Register Configuration of UART" shows a list of registers of the UART.

Figure 14.2-1 UART Registers

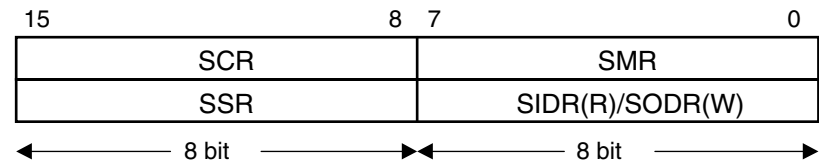


Figure 14.2-2 Register Configuration of UART

bit	7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	Serial input register and serial output register (SIDR/SODR)

bit	15	14	13	12	11	10	9	8	
	PE	ORE	FRE	RDRF	TDRE	-	RIE	TIE	Serial status register (SSR)

bit	7	6	5	4	3	2	1	0	
	MD1	MD0	-	-	CS0	-	SCKE	SOE	Serial mode register (SMR)

bit	15	14	13	12	11	10	9	8	
	PEN	P	SBL	CL	A/D	REC	RXE	TXE	Serial control register (SCR)

14.2.1 Serial Mode Register (SMR)

The serial mode register (SMR) specifies the operation mode of the UART.
Set the operation mode while operation is stopped. Do not write to this register during operation.

■ SMR Configuration

Figure 14.2-3 "Bit Configuration of SMR" shows the bit configuration of the SMR.

Figure 14.2-3 Bit Configuration of SMR

Address	bit	7	6	5	4	3	2	1	0	Initial value
0000 001F _H		MD1	MD0	—	—	CS0	—	SCKE	SOE	00--0-00 _B
0000 0023 _H		R/W	R/W			W		R/W	R/W	
0000 0027 _H										

■ Bit Functions of SMR

The functions of bits of the SMR are explained below.

[Bits 7 and 6] MD1 and MD0 (MoDe select)

These bits are used to select the UART operation mode.

Table 14.2-1 "UART Operation Mode Selection" shows settings for selecting the UART operation mode.

Table 14.2-1 UART Operation Mode Selection

Mode	MD1	MD0	Operation mode
0	0	0	Asynchronous normal mode (initial value)
1	0	1	Asynchronous multiprocessor mode
2	1	0	CLK synchronous mode
-	1	1	Setting prohibited

Note:

CLK asynchronous mode of mode 1 (multiprocessor) is used when multiple slave CPUs are connected to one host CPU. Because the data format of the reception data cannot be identified for this resource, only the master in multiprocessor mode is supported.

Set the PEN bit of the SCR register to "0" because a parity check function cannot be used in this context.

[Bits 5 and 4] (reserved)

These bits are unused.

Always set these bits to "1".

[Bit 3] CS0 (Clock Select)

This bit is used to select the UART operation clock.

CS0	Function
0	Internal timer (U-TIMER) (initial value)
1	External clock

[Bit 2] (reserved)

This bit is unused.

Always set this bit to "0".

[Bit 1] SCKE (SCLK Enable)

For communication in CLK synchronous mode (mode 2), this bit is used to specify whether the SC pin is to be used as a clock input pin or a clock output pin.

SCKE	Function
0	The SC pin functions as a clock input pin (initial value).
1	The SC pin functions as a clock output pin.

Note:

To use the SC pin as a clock input pin, the CS0 bit must be set to "1" to select an external clock.

[Bit 0] SOE (Serial Output Enable)

This bit is used to specify whether the external pin (S0) used as a general-purpose I/O port pin is to be used as a serial output pin or as an I/O port pin.

SOE	Function
0	0: The pin functions as a general-purpose I/O port pin (initial value).
1	1: The pin functions as a serial output pin (SO).

14.2.2 Serial Control Register (SCR)

The serial control register (SCR) controls the transfer protocol for serial communication.

■ SCR Configuration

Figure 14.2-4 "Bit Configuration of SCR" shows the bit configuration of the SCR.

Figure 14.2-4 Bit Configuration of SCR

Address	bit	15	14	13	12	11	10	9	8	Initial value
0000 001E _H		PEN	P	SBL	CL	A/D	REC	RXE	TXE	00000100 _B
0000 0022 _H										
0000 0026 _H		R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	

■ Bit Functions of SCR

The functions of bits of the SCR are explained below.

[Bit 7] PEN (Parity ENable)

This bit indicates a selection whether to add (send) and detect (receive) parity bits for serial data communication.

PEN	Function
0	No parity (initial value)
1	With parity

Note:

Parity bits cannot be used for serial data communication when mode 1 or 2 is selected. In such cases, this bit must always be set to 0.

[Bit 6] P (Parity)

This bit is used to specify odd or even parity for data communication with parity.

p	Function
0	Odd parity (initial value)
1	Even parity

[Bit 5] SBL (Stop Bit Length)

This bit is used to specify the bit length of the stop bit used as the frame end mark for

asynchronous communication.

SBL	Function
0	1 stop bit (initial value)
1	2 stop bits

[Bit 4] CL (Character Length)

This bit is used to specify the data length of a single frame for sending and receiving.

CL	Function
0	7-bit data (initial value)
1	8-bit data

Note:

Only normal mode (mode 0) of asynchronous communication supports handling of 7-bit data. For multiprocessor mode (mode 1) and CLK synchronous mode (mode 2), set 8-bit data.

[Bit 3] A/D (Address/Data)

This bit is used to specify the data format of the transmission frame for multiprocessor mode (mode 1) of asynchronous communication.

A/D	Function
0	Data frame (initial value)
1	Address frame

[Bit 2] REC (Receiver Error Clear)

Setting this bit to "0" clears the error flags (PE, ORE, and FRE) of the SSR register.

Setting this bit to "1" is invalid. Reading operations always return "1" for this bit.

[Bit 1] RXE (Receiver Enable)

This bit is used to control UART reception.

RXE	Function
0	Receiving prohibited (initial value)
1	Receiving enabled

Note:

If receiving becomes prohibited during reception (data is input to the reception shift register), reception stops when frame reception is completed and the reception data is stored to the

reception data buffer SDR register.

[Bit 0] TXE (Transmitter Enable)

This bit is used to control UART sending.

TXE	Function
0	Sending prohibited (initial value)
1	Sending enabled

Note:

If sending becomes prohibited during sending (data is output from the sending register), sending stops when there is no more send data in the sending data buffer SODR register. Write "0" while leaving a certain period of time after writing data to SODR. A "certain period of time" refers to 1/16 as long as the baud rate in clock asynchronous transfer mode or a time as long as the baud rate in clock synchronous transfer mode.

14.2.3 Serial Input Data Register (SIDR) and Serial Output Data Register (SODR)

The serial input data register (SIDR) and serial output data register (SODR) are data buffer registers used for receiving and sending.

■ Serial Input Data Register (SIDR) and Serial Output Data Register (SODR)

Figure 14.2-5 "Bit Configuration of SIDR and SODR" shows the bit configuration of the SIDR and SODR.

Figure 14.2-5 Bit Configuration of SIDR and SODR

SIDR	bit	7	6	5	4	3	2	1	0	Initial value
Address 0000 001D _H										Undefined
0000 0021 _H		D7	D6	D5	D4	D3	D2	D1	D0	
0000 0025 _H		R	R	R	R	R	R	R	R	
SODR	bit	7	6	5	4	3	2	1	0	Initial value
Address: same as above										Undefined
		D7	D6	D5	D4	D3	D2	D1	D0	
		W	W	W	W	W	W	W	W	

When the data length is seven bits, bit 7 (D7) will become invalid. Write to the SODR register when the TDRE bit of the SSR register is "1".

Note:

Writing to this address means writing to the SODR register. Reading means reading the contents of the SIDR register.

14.2.4 Serial Status Register (SSR)

The serial status register (SSR) is configured from flags that indicate the operating state of the UART.

■ SSR Configuration

Figure 14.2-6 "Bit Configuration of SSR" shows the bit configuration of the SSR.

Figure 14.2-6 Bit Configuration of SSR

Address	bit	15	14	13	12	11	10	9	8	Initial value
0000 001C _H		PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	00001-00B
0000 0020 _H		R	R	R	R	R		R/W	R/W	
0000 0024 _H										

■ Bit Functions of SSR

The functions of bits of the SSR are explained below.

[Bit 7] PE (Parity Error)

This bit is an interrupt request flag and is set if a parity error occurs during reception.

To clear this flag once it is set, set the SCR register REC bit (bit 10) to "0".

When the PE bit is set, the SDR data will be invalid.

PE	Function
0	No parity error (initial value)
1	Parity error

[Bit 6] ORE (OverRun Error)

This bit is an interrupt request flag and is set if an overrun error occurs during reception.

To clear this flag once it is set, set the SCR register REC bit to "0".

When the ORE bit is set, the SDR data will be invalid.

ORE	Function
0	No overrun error (initial value)
1	Overrun error

[Bit 5] FRE (FRaming Error)

This bit is an interrupt request flag and is set if a framing error occurs during reception.

To clear this flag once it is set, set the SCR register REC bit to "0".

When the FRE bit is set, the SIDR data will be invalid.

FRE	Function
0	No framing error (initial value)
1	Framing error

[Bit 4] RDRF (Receiver Data Register Full)

This bit is an interrupt request flag indicating that the SIDR register contains reception data.

This bit is set when reception data is loaded to the SIDR register and is cleared automatically when the SIDR register is read.

RDRF	Function
0	No receive data (initial value)
1	Receive data

[Bit 3] TDRE (Transmitter Data Register Empty)

This bit is an interrupt request flag indicating that sending data is to be written to the SODR register.

This bit is cleared when the sending data is written to the SODR register and is set again when the written data is loaded to the sending shifter and data transfer starts. The bit then indicates that the next item of sending data is to be written.

TDRE	Function
0	Writing of sending data prohibited
1	Writing of sending data enabled (initial value)

[Bit 2] (reserved)

This bit is unused.

[Bit 1] RIE (Receiver Interrupt Enable)

This bit is used to control receiver interrupts.

RIE	Function
0	Interrupts prohibited (initial value)
1	Interrupts enabled

Note:

Normal reception via the RDRF can be used as a receiver interrupt source in addition to PE, ORE, and FRE errors.

[Bit 0] TIE (Transmitter Interrupt Enable)

This bit is used to control transmitter interrupts.

TIE	Function
0	Interrupts prohibited (initial value)
1	Interrupts enabled

Note:

Send requests using the TDRE can also be used as a transmitter interrupt source.

14.3 UART Operation

The UART has the following three operation modes, each of which can be switched by writing a value to the SMR and SCR registers.

- Asynchronous normal mode
- Asynchronous multiprocessor mode
- CLK synchronous mode

■ UART Operation Modes

Table 14.3-1 "UART Operation Modes" lists the UART operation modes.

For the asynchronous modes, the stop bit length can be specified only for sending. For reception, the stop bit length is always one bit. The UART only operates in the following modes. Do not specify another mode.

Table 14.3-1 UART Operation Modes

Mode	Parity	Data length	Operation mode	Stop bit length
0	Yes/No	7	Asynchronous normal mode	1 bit or 2 bits
	Yes/No	8		
1	No	8+1	Asynchronous multiprocessor mode	No
2	No	8	CLK synchronous mode	

■ UART Clock Selection

○ Internal timer

If CS0 has been set to "0" to select the U-TIMER, the reload value set in the U-TIMER determines the baud rate. Use the following formulas to calculate the baud rate for this case:

- Asynchronous: $\phi / (16 \times \beta)$
 - CLK synchronous: ϕ / β
- ϕ : Peripheral machine clock frequency
- β : Cycle set by the U-TIMER ($2n + 2$ or $2n + 3$, n : Reload value)

In asynchronous mode, data can be transferred within the range of -1% to +1% of the specified baud rate.

○ External clock

If CS0 has been set to "1" to select an external clock, use the following formulas to calculate the baud rate of the external clock frequency "f":

- Asynchronous: $f/16$
- CLK synchronous: f

Note that the maximum value of f is 3.125 MHz.

14.3.1 Asynchronous Modes

The UART can handle data only in nonreturn-to-zero (NRZ) format.

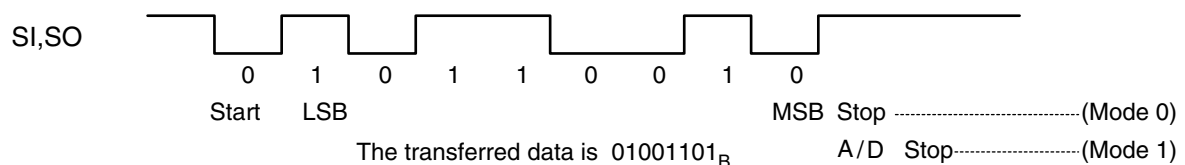
Items of transfer data always start from the start bit ("L" level data). Data items with the length specified by LSB first are transferred. The transfer data ends with the stop bit ("H" level data). If an external clock has been selected, always input the clock signal.

■ Transfer Data Format in Asynchronous Modes

Figure 14.3-1 "Format of Transferred Data Format in Asynchronous Modes (Modes 0 and 1)" shows the format of transferred data in the asynchronous modes.

In asynchronous normal mode (mode 0), the data length can be set to seven or eight bits. In asynchronous multiprocessor mode (mode 1), the data length must be eight bits. A parity bit cannot be added in asynchronous multiprocessor mode, but an A/D bit must be added.

Figure 14.3-1 Format of Transferred Data Format in Asynchronous Modes (Modes 0 and 1)



○ Receiving

If the SCR register RXE bit (bit 1) is "1", reception is performed.

When the start bit is detected on the receiving line, one data frame in the format determined by the SCR register is received. If an error occurs after one data frame has been received, an error flag will be set and the RDRF flag (SSR register bit 4) will be set subsequently. If the RIE bit (bit 1) of the same SSR register has been set to 1 at this time, a receiver interrupt for the CPU will be thrown. Check each flag of the SSR register. If reception is normal, read the SISR register. If an error has occurred, respond accordingly.

The RDRF flag will be cleared when the SISR register is read.

○ Sending

When the SSR register TDRE flag (bit 11) is "1", the send data will be written to the SODR register and the data is sent if the SCR register TXE flag (bit 0) is "1".

When the data set in the SODR register is loaded to the sending shift register and sending starts, the TDRE flag will be set again so that the next item of sending data can be set. If the TIE bit (bit 0) of the same SSR register has been set to 1 at this time, a transmitter interrupt for the CPU will be thrown. A request will then be issued to set the send data in the SODR register.

The TDRE flag will be cleared as soon as the send data is set in the SODR register.

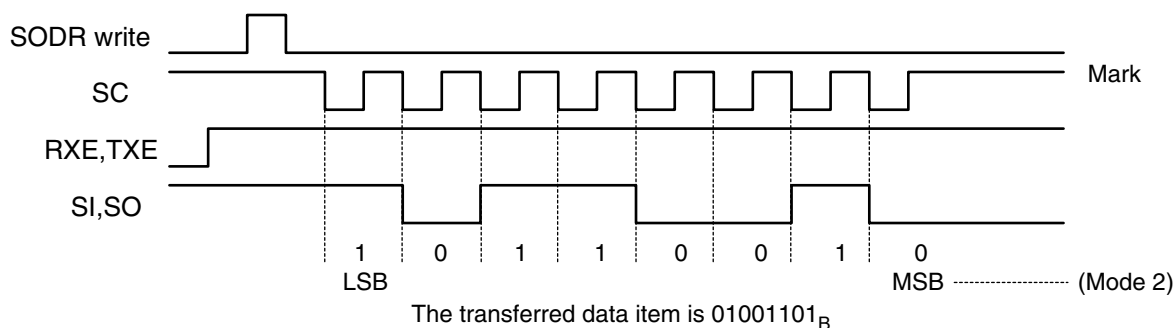
14.3.2 CLK Synchronous Mode

The UART can handle data only in nonreturn-to-zero (NRZ) format.

■ Transfer Data Format in CLK Synchronous Mode

Figure 14.3-2 "Format of Transferred Data in CLK Synchronous Mode (Mode 2)" shows the relationship between send/receive clock pulses and data in CLK Synchronous Mode.

Figure 14.3-2 Format of Transferred Data in CLK Synchronous Mode (Mode 2)



If CS0 has been set to "0" to select output from the U-TIMER, a synchronous clock pulse for receiving data will be generated automatically when data is sent.

If an external clock has been selected, it must first be determined whether data is present in the send data buffer SODR register of the sending UART (for which the TDRE flag is "0"). A clock pulse of precisely 1 byte must be supplied in this case. Always set the mark level before sending starts and when sending ends.

Because the data is handled as eight bit-data, no parity bit can be added. Moreover, because there are no start and stop bits, errors other than overrun errors will not be detected.

○ Initialization

The setting values of the individual control registers when using CLK synchronous mode are as follows:

- SMR register
 - MD1 and MD0: 10
 - CS: Specify clock input.
 - SCKE: Specify 1 for an internal timer or 0 for an external clock.
 - SOE: Specify 1 to enable sending or 0 for receiving only.
- SCR register
 - PEN: 0
 - P, SBL, and A/D: The settings of these bits have no effect.
 - CI: 1
 - REC: 0 (for initialization)
 - RXE and TXE: Specify at least one of these bits as 1.
- SSR register

- RIE: Specify 1 to use interrupts or 0 to not use interrupts.
- TIE: 0

○ **Starting communication**

Writing to the SODR register starts communication. Even when only reception is to be performed, dummy send data must be written to the SODR register.

○ **Ending communication**

Whether the SSR register RDRF flag has changed to 1 can be used to detect whether communication has ended. Use the SSR register ORE bit to confirm that communication has been performed normally.

14.3.3 UART Interrupts and Timings for Setting Flags

The UART has five flags and two types of interrupt sources.
The five flags are PE, ORE, FRE, RDRF, and TDRE.
The two types of interrupt sources are receiver and transmitter interrupts.

■ Interrupts and Flags

PE means a parity error, ORE means an overrun error, and FRE means a framing error. These flags are set when a reception error occurs. Setting the SCR register REC bit to "0" clears these flags.

The RDRF flag is set when reception data is loaded to the SISR register. The RDRF flag is cleared when the SISR register is read. In mode 1, however, there is no parity detection function. In mode 2, there is no parity detection function or framing error detection function.

The TDRE flag is set when the SODR register becomes empty and writing is enabled. Writing to the SODR register clears the TRDE flag.

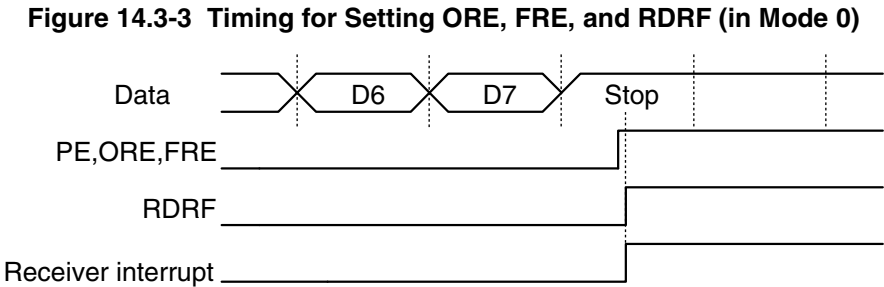
During reception, interrupts are requested using the PE, ORE, FRE, and RDRF flags.

During sending, interrupts are requested using the TDRE flag.

■ Timing for Setting Receiver Interrupt Flags in Mode 0

When reception and transfer ends and the final stop bit is detected, the PE, ORE, FRE, and RDRF flags are set and an interrupt request for the CPU is generated. If the PE, ORE, and FRE flags are set, the SISR data will become invalid.

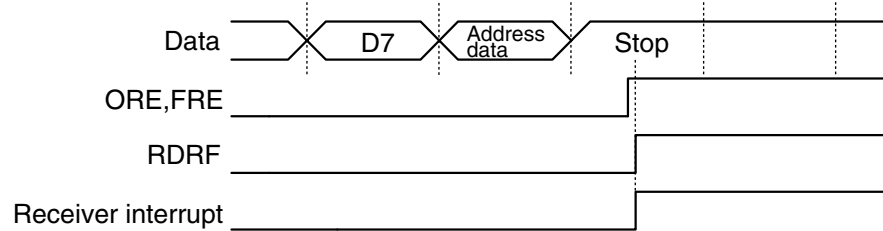
Figure 14.3-3 "Timing for Setting ORE, FRE, and RDRF (in Mode 0)" shows the timing for setting ORE, FRE, and RDRF in mode 0.



■ Timing for Setting Receiver Interrupt Flags in Mode 1

When reception and transfer ends and the final stop bit is detected, the ORE, FRE, and RDRF flags are set and an interrupt request for the CPU is generated. Because the data length for reception is eight bits, the data indicating the address/data of the final bit 9 will become invalid. If the ORE or FRE flags are active, the SISR data will become invalid.

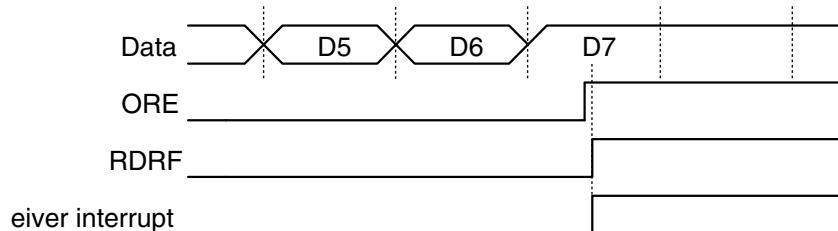
Figure 14.3-4 "Timing for Setting ORE, FRE, and RDRF (in Mode 1)" shows the timing for setting ORE, FRE, and RDRF in mode 1.

Figure 14.3-4 Timing for Setting ORE, FRE, and RDRF (in Mode 1)

■ Timing for Setting the Receiver Interrupt Flags in Mode 2

When receiving transfer ends and the final data item (D7) is detected, the ORE and RDRF flags are set and an interrupt request is generated for the CPU. If the ORE flag is active, the SDR data will become invalid.

Figure 14.3-5 "ORE and RDRF Set Timings (Mode 2)" shows the timing for setting ORE and RDRF in mode 2.

Figure 14.3-5 ORE and RDRF Set Timings (Mode 2)

■ Timing for Setting the Transmitter Interrupt Flags of Modes 0, 1, and 2

The TDRE flag is cleared when data is written to the SODR register and data is transferred to the internal shift register. As soon as writing the next data item is enabled, the TDRE flag is set and an interrupt request for the CPU is generated.

During sending, setting the SCR register TXE bit (including the RXE bit in mode 2) to "0" will set the SSR register TDRE flag to "1" and UART sending will be prohibited when the operation of the sending shifter stops. After setting the SCR register TXE bit (including the RXE bit in mode 2) to "0" during sending, the data written to the SODR register will be sent before stopping the sending operation.

Figure 14.3-6 "Timing for Setting TDRE (in Modes 0 and 1)" shows the timing for setting TDRE in modes 0 and 1. Figure 14.3-7 "Timing for Setting TDRE (in Mode 2)" shows the timing for setting TDRE in mode 2.

Figure 14.3-6 Timing for Setting TDRE)(in Modes 0 and 1)

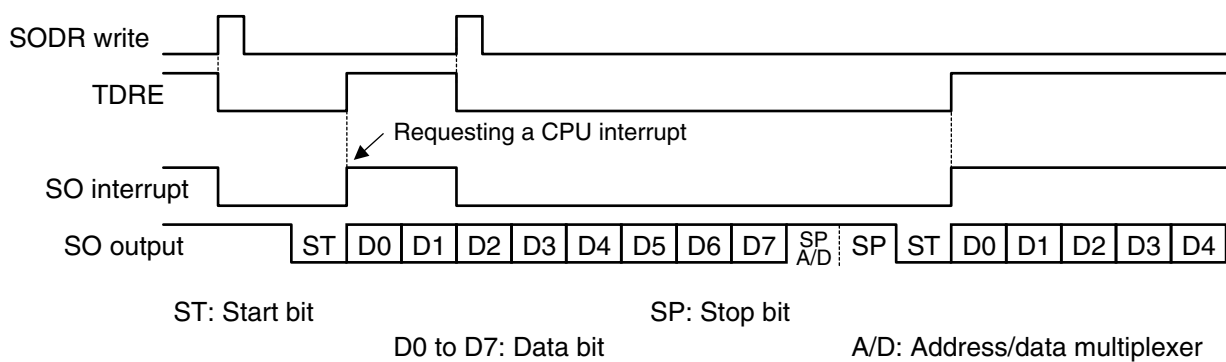
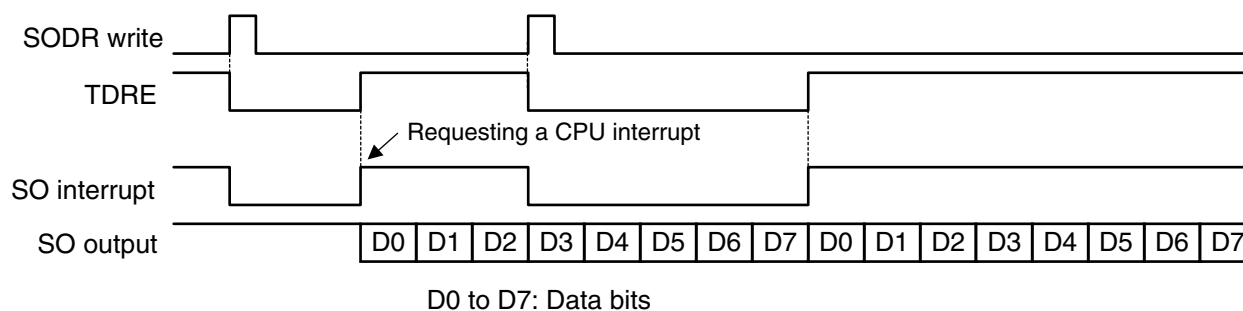


Figure 14.3-7 Timing for Setting TDRE (in Mode 2)



14.4 Example of Using UART

This section provides notes on using UART and shows an example.

■ Note on Using UART

Stop operation when setting the communication mode. The contents of sending and reception data will become unpredictable during setting the mode.

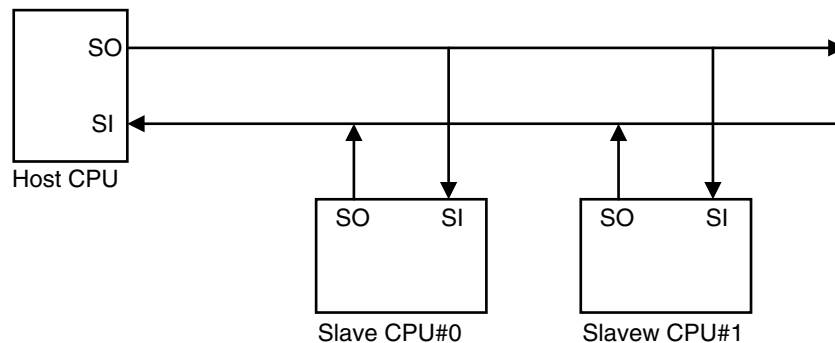
If the timing for writing to the serial output data register (SODR) matches the timing for requesting a receiver interrupt (RDRF=1) during UART operation in synchronous transfer mode (mode 2), the communication control circuit may stop. To prevent this problem, write to the SODR after data transfer or immediately after transmission begins.

■ UART Sample Application

In this example, multiple slave CPUs are connected to a single host CPU in mode 1. Figure 14.4-1 "Sample System Configuration in Mode 1" shows a sample system configuration in mode 1.

Only the communication interface on the host side is supported for this resource.

Figure 14.4-1 Sample System Configuration in Mode 1



Communication starts when the host CPU transfers address data.

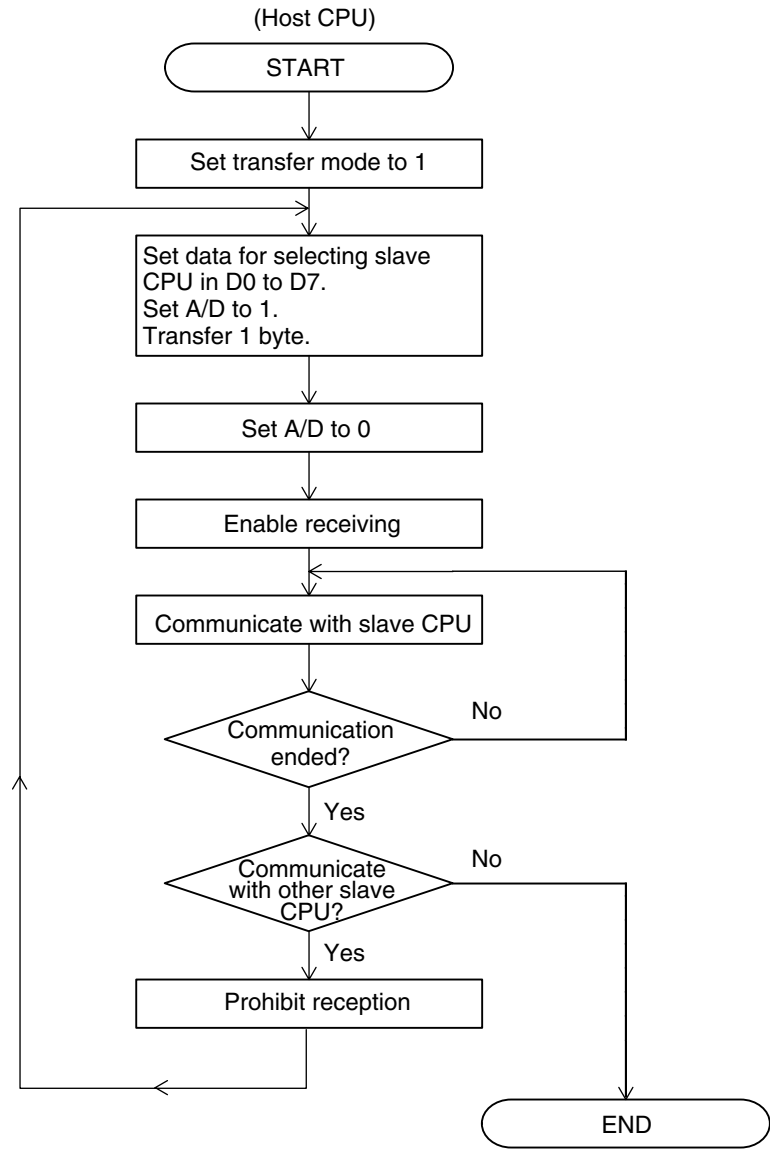
Address data is indicated by the fact that the SCR register A/D bit is "1". This address data is used to select the slave CPU becoming the communication destination and enable communication with the host CPU.

Normal data is indicated by the fact that the SCR register A/D bit is "0".

Figure 14.4-2 "Flowchart for Communication in Mode 1" shows the flowchart.

Because the parity check function cannot be used in mode 1, set the SCR register PEN bit to "0".

Figure 14.4-2 Flowchart for Communication in Mode 1



14.5 Sample Settings for Baud Rate and U-TIMER Reload Value

This section provides an example of setting the baud rate and reload value of the U-TIMER.

■ Sample Settings for Baud Rate and U-TIMER Reload Value

The frequency values in the tables indicate the clock frequencies of peripheral machine clocks. The value UCC1 indicates the value to which the UCC1 bit of the U-TIMER control register (UTIMC) is set.

The dashes in the table indicate that the corresponding baud rate cannot be used because the error exceeds plus or minus 1%.

■ Sample settings for baud rate and U-TIMER reload value

Table 14.5-1 Setting Values in Asynchronous (Start-stop Control) Mode

Baud rate	μs	25MHz	20MHz	12.5MHz	10MHz
1200	833.33	650 _D (UCC1=0)	520 _D (UCC1=0)	324 _D (UCC1=1)	259 _D (UCC1=1)
2400	416.67	324 _D (UCC1=1)	259 _D (UCC1=1)	162 _D (UCC1=0)	129 _D (UCC1=0)
4800	208.33	162 _D (UCC1=0)	129 _D (UCC1=0)	80 _D (UCC1=1)	64 _D (UCC1=0)
9600	104.17	80 _D (UCC1=1)	64 _D (UCC1=0)	39 _D (UCC1=1)	31 _D (UCC1=1)
19200	52.08	39 _D (UCC1=1)	31 _D (UCC1=1)	19 _D (UCC1=1)	-
38400	26.04	19 _D (UCC1=1)	-	12 _D (UCC1=1)	-
57600	17.36	12 _D (UCC1=1)	-	-	-
10400	96.15	74 _D (UCC1=0)	59 _D (UCC1=0)	36 _D (UCC1=1)	29 _D (UCC1=0)
31250	32.00	24 _D (UCC1=0)	19 _D (UCC1=0)	11 _D (UCC1=1)	9 _D (UCC1=0)
62500	16.00	11 _D (UCC1=1)	9 _D (UCC1=0)	-	4 _D (UCC1=0)

Table 14.5-2 Setting Values in CLK Synchronous Mode

Baud rate	μs	25MHz	20MHz	12.5MHz	10MHz
250K	4.00	49 _D (UCC1=0)	39 _D (UCC1=0)	24 _D (UCC1=0)	19 _D (UCC1=0)
500K	2.00	24 _D (UCC1=0)	19 _D (UCC1=0)	11 _D (UCC1=1)	9 _D (UCC1=0)
1M	1.00	11 _D (UCC1=1)	9 _D (UCC1=0)	5 _D (UCC1=0)*	4 _D (UCC1=0)

*: Error exceeding plus or minus 1%

CHAPTER 15 DMA CONTROLLER (DMAC)

This chapter outlines the DMA controller and describes the register configuration/ functions and operations of the DMA controller.

15.1 "Outline of the DMA Controller"

15.2 "DMA Controller Block Diagram"

15.3 "DMA Controller Registers"

15.4 "DMA Controller Transfer Modes"

15.5 "DMA Controller Timing Diagrams"

15.6 "Notes on the DMA Controller"

15.1 Outline of the DMA Controller

The DMA controller is used for direct memory access (DMA) transfer.

■ DMA Controller Features

- Eight channels
- Three types of modes: Single/block transfer, burst transfer, and continuous transfer
- Transfer between total address area and total address area
- Maximum transfer count of 65,536
- Transfer end interrupt function
- Transfer address increase or reduction can be selected by software

■ List of DMA Transfer Request Sources

Table 15.1-1 "DMA Transfer Request Sources" shows a list of DMA transfer request sources.

Table 15.1-1 DMA Transfer Request Sources

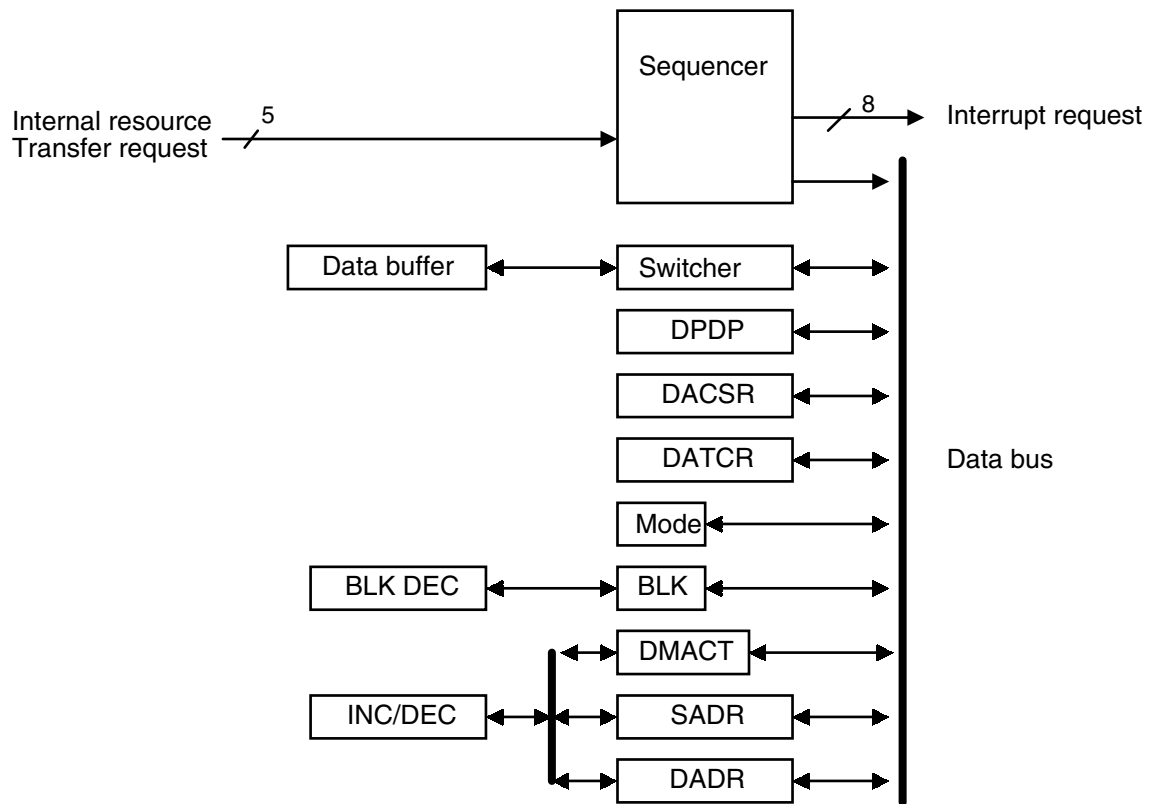
Channel number	Transfer request source
0	None
1	None
2	None
3	PPG ch0
4	UART ch0 reception
5	UART ch0 transmission
6	16-bit reload timer ch0
7	A/D converter

15.2 DMA Controller Block Diagram

Figure 15.2-1 "DMA Controller Block Diagram" shows a block diagram of the DMA Controller.

■ Block Diagram of the DMA Controller

Figure 15.2-1 DMA Controller Block Diagram



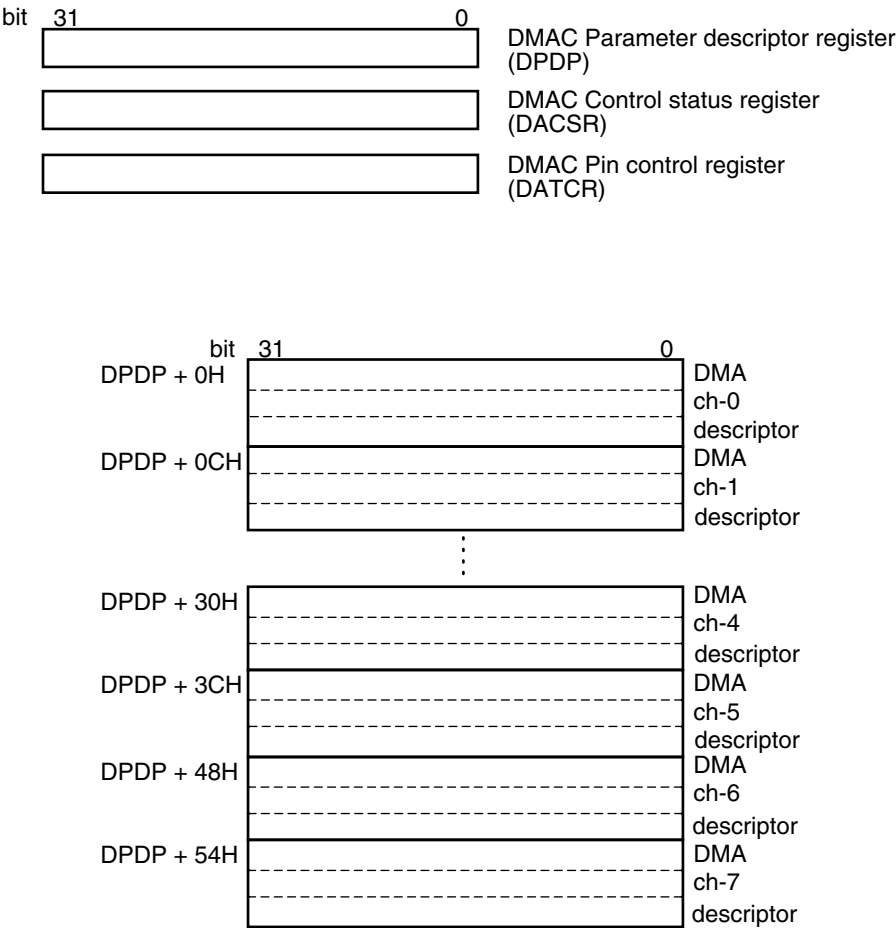
15.3 DMA Controller Registers

Figure 15.3-1 "DMA Controller Registers" shows the configurations of the DMA controller registers.

■ List of DMA Controller Registers

Figure 15.3-1 "DMA Controller Registers" shows a list of DMA controller registers.

Figure 15.3-1 DMA Controller Registers

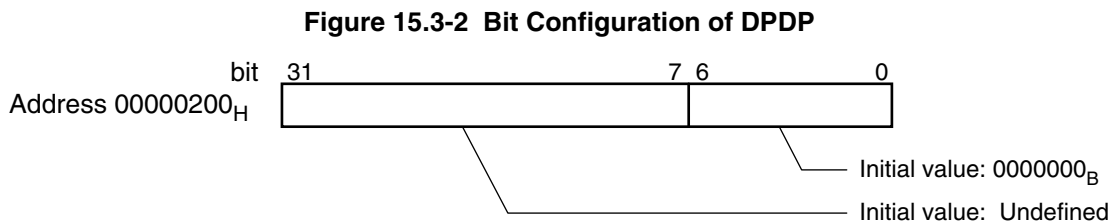


15.3.1 DMAC Parameter Descriptor Pointer (DPDP)

The DMAC parameter descriptor pointer (DPDP) is a DMAC-internal register. The DPDP is used to store the leading address of the DMAC descriptor table in RAM. DPDP bits 6 to 0 are always 0. The leading addresses of descriptors can be set in units of 128 bytes.

■ DPDP Configuration

Figure 15.3-2 "Bit Configuration of DPDP" shows the bit configuration of the DPDP.



After a reset, this register is not initialized.

This register can be read or written to.

Use 32-bit transfer instructions to access this register.

The DPDP specifies the addresses to which the descriptors that specify the operation mode of each channel are assigned. Table 15.3-1 "Descriptor Address of Each Channel" lists these addresses.

Table 15.3-1 Descriptor Address of Each Channel

DMA channel	Descriptor address	DMA channel	Descriptor address
0	DPDP + 0 (00H)	4	DPDP + 48 (30H)
1	DPDP + 12 (0CH)	5	DPDP + 60 (3CH)
2	DPDP + 24 (18H)	6	DPDP + 72 (48H)
3	DPDP + 36 (24H)	7	DPDP + 84 (54H)

15.3.2 DMAC Control Status Register (DACSR)

The DMAC control status register (DACSR) is a DMAC internal register. The DACSR is used to control the entire DMAC and indicate the DMAC status.

■ DACSR Configuration

Figure 15.3-3 "Bit Configuration of DACSR" shows the bit configuration of the DACSR.

Figure 15.3-3 Bit Configuration of DACSR

bit	31	30	29	28	27	26	25	24	Initial value
Address 00000204 _H	DER7	DED7	DIE7	DOE7	DER6	DED6	DIE6	DOE6	00000000 _H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
bit	23	22	21	20	19	18	17	16	
	DER5	DED5	DIE5	DOE5	DER4	DED4	DIE4	DOE4	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
bit	15	14	13	12	11	10	9	8	
	DER3	DED3	DIE3	DOE3	DER2	DED2	DIE2	DOE2	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
bit	7	6	5	4	3	2	1	0	
	DER1	DED1	DIE1	DOE1	DER0	DED0	DIE0	DOE0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

■ Bit Functions of DACSR

The functions of bits of the DACSR are explained below.

[Bits 31, 27, 23, 19, 15, 11, 7, and 3] DER_n (DMA Error)

These bits indicate that an error has occurred at the source of the DMA request in channel *n* and that the corresponding DMA transfer has been canceled.

DER	Function
0	No error has occurred.
1	An error has occurred.

Whether an error occurs depends on the source for the DMA request (resource). Not all DMA request sources cause an error.

During a reset, these bits are initialized to "0".

These bits can be read or overwritten, but only writing operations during which these bits are set to "0" are valid.

Reading operations using read-modify-write instructions always return "1" for these bits.

Note:

Only channel 4 can report that an error has occurred for the DMA transfer request source using the DERn bit in the DMAC control status register (DACSR).

If receiver interrupt of UART channel 0 is being used for DMA transfer requests, the DER4 bit will be set to "1" when the following errors occur:

- Parity error
- Overrun error
- Framing error

[Bits 30, 26, 22, 18, 14, 10, 6, and 2] DEDn (DMA EnD)

These bits indicate that DMA transfer of channel n has ended.

DED	Function
0	DMA transfer has not ended.
1	The counter is 0 or an error has occurred for the transfer request source.

During a reset, these bits are initialized to "0".

These bits can be read or overwritten, but only writing operations during which these bits are set to "0" are valid.

Reading operations using read-modify-write instructions always return "1" for these bits.

[Bits 29, 25, 21, 17, 13, 9, 5, and 1] DIEn (DMA Interrupt Enable)

These bits specify whether to generate an interrupt request when DMA transfer of channel n ends (when DEDn is 1).

DIE	Function
0	Interrupts prohibited
1	Interrupts enabled

During a reset, these bits are initialized to "0".

These bits can be read or overwritten.

[Bits 28, 24, 20, 16, 12, 8, 4, and 0] DOEn (DMA Operation Enable)

These bits enable DMA transfer of channel n.

DOE	Function
0	DMA transfer prohibited
1	DMA transfer enabled

When DMA transfer of the relevant channel is completed (when DEDn is 1), DOEn is cleared by setting it to 0.

If clearing because of transfer completion and setting by write operations from the bus are performed at the same time, the setting operation has priority.

During a reset, these bits are initialized to "0".

These bits can be read or overwritten.

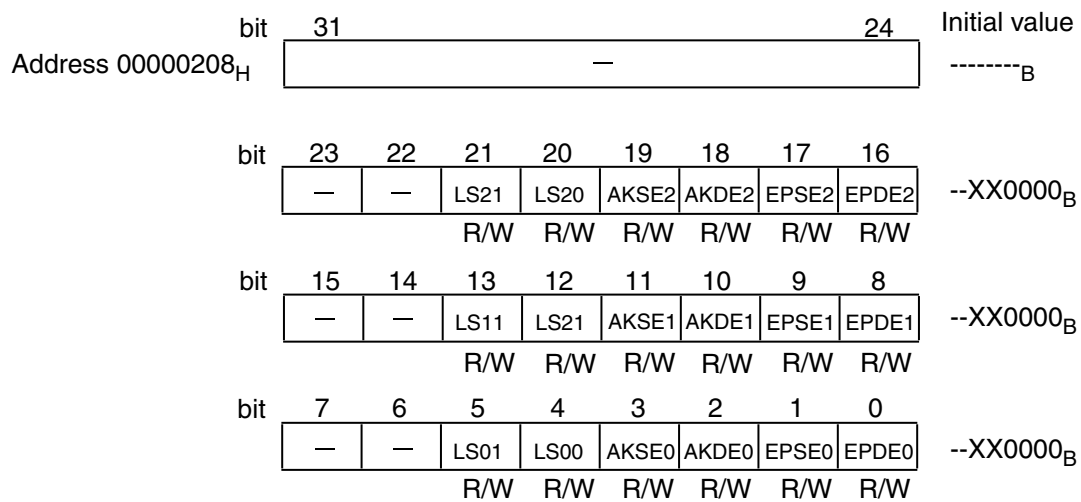
15.3.3 DMAC Pin Control Register (DATCR)

The DMAC pin control register (DATCR) is a DMAC-internal register. The DATCR is used to control the external transfer request input, external transfer request acceptance output, and external transfer end output pins.

■ DATCR Configuration

Figure 15.3-4 "Bit Configuration of DATCR" shows the bit configuration of the DATCR.

Figure 15.3-4 Bit Configuration of DATCR



■ Bit Functions of DATCR

The functions of bits of the DATCR are explained below.

[Bits 21, 20, 13, 12, 5, and 4] LSn1 and LSn0: Select the transfer request input detection level.

These bits are used to select the detection level of the corresponding external transfer request input pin DREQn as listed in Table 15.3-2 "Selecting the Transfer Request Input Detection Level".

Table 15.3-2 Selecting the Transfer Request Input Detection Level

LSn1	LSn0	Operation control function
0	0	Rising edge detection
0	1	Falling edge detection
1	0	"H" level detection
1	1	"L" level detection

After a reset, these bits are undefined.

CHAPTER 15 DMA CONTROLLER (DMAC)

These bits can be read or overwritten.

If continuous transfer mode is used, set "H" level detection or "L" level detection.

[Bits 19, 11, and 3] AKSEn

[Bits 18, 10, and 2] AKDn

These bits specify the timing for issuing the transfer request acceptance output signal. In addition, these bits specify whether to enable or prohibit the output function corresponding to the pin of the transfer request acceptance output signal.

Table 15.3-3 "Specifying Transfer Request Acceptance Output" shows settings for specifying transfer request acceptance output.

Table 15.3-3 Specifying Transfer Request Acceptance Output

AKSEn	AKDn	Operation control function
0	0	Transfer acceptance output is prohibited.
0	1	Transfer acceptance output is enabled. Transfer acceptance is output at transfer destination data access.
1	0	Transfer acceptance output is enabled. Transfer acceptance is output at transfer source data access.
1	1	Transfer acceptance output is enabled. Transfer acceptance is output at transfer destination data access and transfer source data access.

During a reset, these bits are initialized to "00".

These bits can be read or overwritten.

[Bits 17, 9, and 1] EPSEn

[Bits 16, 8, and 0] EPDn

These bits specify the timing for issuing the transfer end output signal. In addition, these bits specify whether to enable or prohibit the output function corresponding to the pin of the transfer end output signal.

Table 15.3-4 "Specifying Transfer End Output" shows settings for specifying transfer end output.

Table 15.3-4 Specifying Transfer End Output

EPSEn	EPDn	Operation control function
0	0	Transfer end output is prohibited.
0	1	Transfer end output is enabled. Transfer end is output at transfer destination data access.

Table 15.3-4 Specifying Transfer End Output

EPSEn	EPDEn	Operation control function
1	0	Transfer end output is enabled. Transfer end is output at transfer source data access.
1	1	Transfer end output is enabled. Transfer end is output at transfer destination data access and transfer source data access.

During a reset, these bits are initialized to "00".

These bits can be read or overwritten.

15.3.4 RAM Descriptor-Internal Register

The RAM descriptor-internal register is used to store the setting information for each channel during DMA transfer.

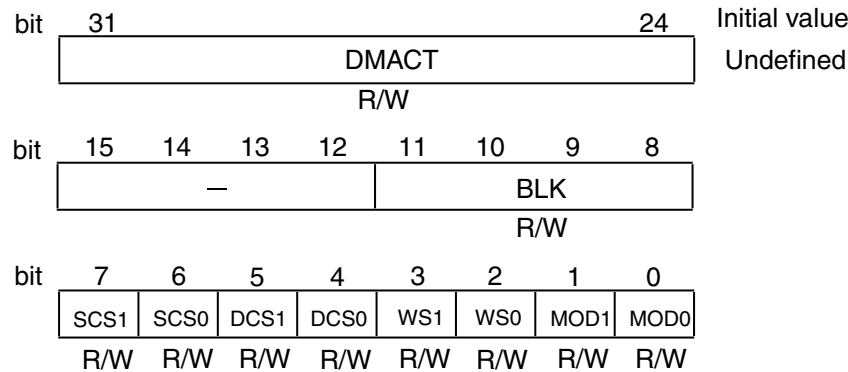
This register has a size of 12 bytes per channel and is allocated at the addresses indicated by the DPDP.

See Table 15.3-1 "Descriptor Address of Each Channel" for the leading addresses of the descriptors for each channel.

■ Configuration of Leading Word of the Descriptor

Figure 15.3-5 "Bit Configuration of Leading Word of Descriptor" shows the bit configuration of the leading word of the descriptor.

Figure 15.3-5 Bit Configuration of Leading Word of Descriptor



The functions of bits of the leading word of the descriptor are explained below.

[Bits 31 to 16] DMACT: Specify the transfer count

These bits are used to specify the transfer count. When 0000_H is specified, 65,536 transfer operations will be performed.

The specified count is decremented by 1 during each transfer operation.

[Bits 15 to 12] Empty

[Bits 11 to 8] BLK: Specify the block size

These bits are used to specify the transfer block size for single/block transfer mode.

When "0" is specified, a default block size of 16 will be assumed. Specify "1" for single transfer.

[Bits 7 and 6] SCS1 and SCS0: Specify the transfer source address update mode

[Bits 5 and 4] DCS1 and DCS0: Specify the transfer destination address update mode

These bits are used to specify the update mode for transfer operations of the transfer source and transfer destination addresses individually.

The combinations listed in Table 15.3-5 "Specifying Transfer Source/Transfer Destination Update Modes" can be specified.

Table 15.3-5 Specifying Transfer Source/Transfer Destination Update Modes

SCS1	SCS0	DCS1	DCS0	Transfer source address	Transfer destination address
0	0	0	0	Address increase	Address increase
0	0	0	1	Address increase	Address reduction
0	0	1	0	Address increase	Address fixed
0	1	0	0	Address reduction	Address increase
0	1	0	1	Address reduction	Address reduction
0	1	1	0	Address reduction	Address fixed
1	0	0	0	Address fixed	Address increase
1	0	0	1	Address fixed	Address reduction
1	0	1	0	Address fixed	Address fixed
Others				Setting prohibited	

Table 15.3-6 "Units of Address Increase and Reduction" lists the units in which individual addresses increases and decreases in address update mode according to the specified transfer data size.

Table 15.3-6 Units of Address Increase and Reduction

Transfer data size	Unit of address increase and reduction
byte (8 bit)	± 1 byte
halfword (16 bit)	± 2 bytes
word (32 bit)	± 4 bytes

[Bits 3 and 2] WS1 and WS0

Table 15.3-7 "Specifying the Transfer Data Size" shows settings for specifying the transfer data size.

Table 15.3-7 Specifying the Transfer Data Size

WS1	WS0	Transfer data size
0	0	byte
0	1	halfword
1	0	word

Table 15.3-7 Specifying the Transfer Data Size

WS1	WS0	Transfer data size
1	1	Setting prohibited

[Bits 1 and 0] MOD1 and MOD2

Table 15.3-8 "Specifying the Transfer Mode" shows settings for specifying the transfer mode.

Table 15.3-8 Specifying the Transfer Mode

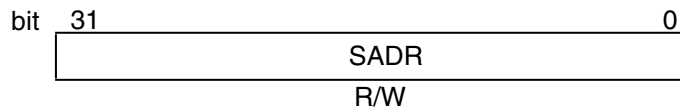
MOD1	MOD0	Operation mode
0	0	Single/block mode
0	1	Burst mode
1	0	Continuous transfer mode
1	1	Setting prohibited

Only channels 0 to 2 can be used in continuous transfer mode.

■ **Configuration of Second Word of the Descriptor**

Figure 15.3-6 "Bit Configuration of Second Word of Descriptor" shows the bit configuration of the second word of the descriptor.

Figure 15.3-6 Bit Configuration of Second Word of Descriptor



The transfer source address is stored in the second word of the descriptor.

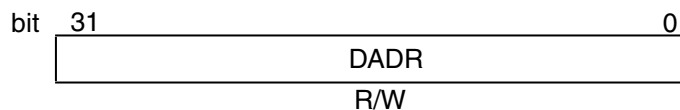
Based on the specified address update mode (SCS1 and SCS0 bits), the value is updated corresponding to the transfer operation.

If the transfer data size is halfword, specify an address that is a multiple of two. If the transfer data size is word, specify an address that is a multiple of four.

■ **Configuration of Third Word of the Descriptor**

Figure 15.3-7 "Bit Configuration of Third Word of Descriptor" shows the bit configuration of the third word of the descriptor.

Figure 15.3-7 Bit Configuration of Third Word of Descriptor



The transfer destination address is stored in the third word of the descriptor.

Based on the specified address update mode (DCS1 and DCS0 bits), the value is updated corresponding to the transfer operation.

If the transfer data size is halfword, specify an address that is a multiple of two. If the transfer data size is word, specify an address that is a multiple of four.

15.4 DMA Controller Transfer Modes

The DMA controller has the following three transfer modes. The operation procedures for these modes are explained below.

- **Single/block transfer mode**
 - **Continuous transfer mode**
 - **Burst transfer mode**
-

■ Single/Block Transfer Mode

1. Use an initialization routine to set the descriptor.
2. Use a program to initialize the DMA transfer request source. If an internal peripheral circuit is selected as the DMA transfer request source, enable interrupt requests. At the same time, set the ICR of the interrupt controller to interrupts prohibited.
3. Use a program to set the respective DACSR DOEn bits to 1.
--- This completes setting of the DMA. ---
4. When the DMAC detects a DMA transfer request input, it requests bus authority from the CPU.
5. When the CPU transfers the bus authority to the DMAC, the DMAC accesses the information in the three descriptor words through the bus.
6. The value in the DMACT is decremented. Transfer operations are performed based on the information in the descriptor for the number of times specified by the BLK or until the DMACT reaches 0. The transfer request acceptance output signal is output during data transfer (if external transfer request input is used). When the decremented DMACT reaches 0, the transfer end output signal is output during data transfer.
7. Transfer request input is cleared.
8. The SADR or DADR is incremented or decremented. This value and the DMACT value are written back to the descriptor.
9. The bus authority is returned to the CPU.
10. If the DMACT value is 0, an interrupt is issued to the CPU if DACSR DEDn has been set to 1 and interrupts enabled.

The minimum number of required cycles per transfer operation is listed below. The descriptor is stored in internal RAM, data is transferred between external buses, and the data length is bytes.

- If the transfer source and transfer destination addresses are fixed: $(6 + 5 \times \text{BLK})$ cycles
- If the transfer source or transfer destination address is fixed: $(7 + 5 \times \text{BLK})$ cycles
- If the transfer source and transfer destination addresses are increased or reduced: $(8 + 5 \times \text{BLK})$ cycles

■ Continuous Transfer Mode

1. Use an initialization routine to set the descriptor.
2. Use a program to initialize the DMA transfer request source. Set the external transfer request input pin to H or L level detection.
3. Use a program to write 1 to the desired DACSR DOEn bits.
--- This completes setting of the DMA. ---
4. When the DMAC detects DMA transfer request input, bus authority is requested from the CPU.
5. When the CPU transfers the bus authority to the DMAC, the DMAC accesses the information of the three words in the descriptor through the bus.
6. The DMACT is decremented. One transfer operation is performed based on the information in the descriptor. The transfer request acceptance output signal is output during data transfer. When the decremented DMACT reaches 0, the transfer end output signal is output during data transfer.
7. If the DMACT value is not 0 and there is a DMA request from a resource, the operation is repeated starting from step 6. (The steps may include step 8, depending on the bus state.)
8. If the DMACT value is 0 or DMA requests from the resources have been released, the SADR or DADR is incremented or decremented. This value and the DMACT value are written back to the descriptor.
9. The bus authority is returned to the CPU.

If the counter value is 0, an interrupt is issued to the CPU if DACSR DEDn has been set to "1" and interrupts enabled.

The minimum number of required cycles per transfer operation is as described below. The descriptor is stored in internal RAM, data is transferred between external buses, and the data length is bytes.

- If the transfer source and transfer destination addresses are fixed: $(6 + 5 \times n)$ cycles
- If the transfer source or transfer destination address is fixed: $(7 + 5 \times n)$ cycles
- If the transfer source and transfer destination addresses are increased or reduced: $(8 + 5 \times n)$ cycles

■ Burst Transfer Mode

1. Use an initialization routine to set the descriptor.
2. Use a program to initialize the DMA transfer request source. If an internal peripheral circuit is selected as the DMA transfer request source, enable interrupt requests. ICR interrupts of the interrupt controller are prohibited at this time.
3. Use a program to set the appropriate DACSR DOEn bits to "1".
--- This completes the DMA setting procedure. ---
4. When the DMAC detects DMA transfer request input, it requests bus authority from the CPU.
5. After the CPU transfers the bus authority to the DMAC, the DMAC accesses the information in the three words of the descriptor through the bus.
6. The value in the DMACT is decremented. Transfer operations are performed based on the information in the descriptor for the number of times specified by the DMACT. The transfer request acceptance output signal is output during data transfer (if external transfer request input is used). When the decremented DMACT reaches "0", the transfer end output signal is output during data transfer.

7. The value in the SADR or DADR is incremented or decremented. This value and the DMACT value are written back to the descriptor.
8. The bus authority is returned to the CPU.
9. An interrupt is issued to the CPU if DACSR DEDn has been set to 1 and interrupts enabled.

The minimum numbers of cycles required per transfer operation is as indicated below. Here, the descriptor is stored in internal RAM, data is transferred between external buses, and the data length is bytes.

- If the transfer source and transfer destination addresses are fixed: $(6 + 5 \times n)$ cycles
- If the transfer source or transfer destination address is fixed: $(7 + 5 \times n)$ cycles
- If the transfer source and transfer destination addresses are increased or reduced: $(8 + 5 \times n)$ cycles

15.5 DMA Controller Timing Diagrams

This section provides the operation timing diagrams of the DMA controller.

- Timing diagrams of the descriptor access block
- Timing diagrams of the data transfer block
- Timing diagrams of transfer stop in continuous transfer mode
- Timing diagrams of transfer end

■ Explanation of Symbols Used in the Timing Diagrams

Table 15.5-1 "Explanation of Symbols Used in the Timing Diagrams" lists explanations of symbols used in the timing diagrams.

Table 15.5-1 Explanation of Symbols Used in the Timing Diagrams

Symbol	Explanation
#0	Descriptor No. 0
#0H	Bits 31 to 16 of descriptor No. 0
#0L	Bits 15 to 0 of descriptor No. 0
#1	Descriptor No. 1
#1H	Bits 31 to 16 of descriptor No. 1
#1L	Bits 15 to 0 of descriptor No. 1
#2	Descriptor No. 2
#2H	Bits 31 to 16 of descriptor No. 2
#2L	Bits 15 to 0 of descriptor No. 2
#1/2	Descriptor No. 1 or No. 2 (Depends on SCS1, SCS0, DSC1, and DSC0.)
#1/2H	Bits 31 to 16 of descriptor No. 1 or No. 2
#1/2L	Bits 15 to 0 of descriptor No. 0 or No. 2
S	Transfer source
SH	Bits 31 to 16 of the transfer source
SL	Bits 15 to 0 of the transfer source
D	Transfer destination
DH	Bits 31 to 16 of the transfer destination
DL	Bits 15 to 0 of the transfer destination

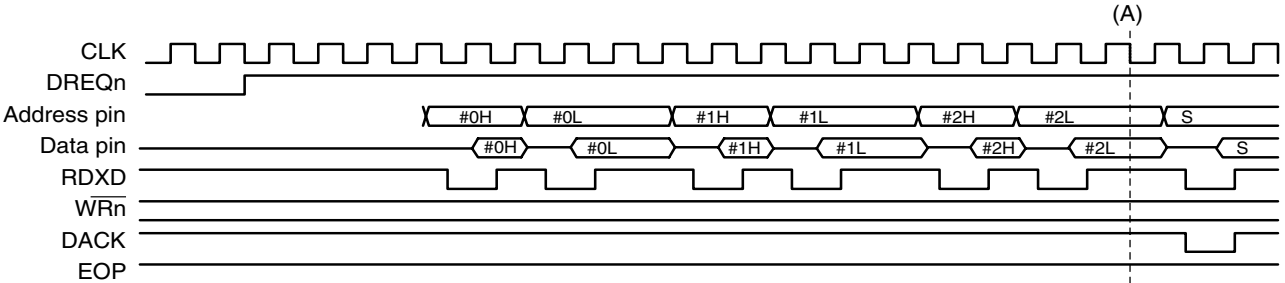
15.5.1 Timing Diagrams of the Descriptor Access Block

This section provides the timing diagrams of the descriptor access block.

■ Descriptor Access Block

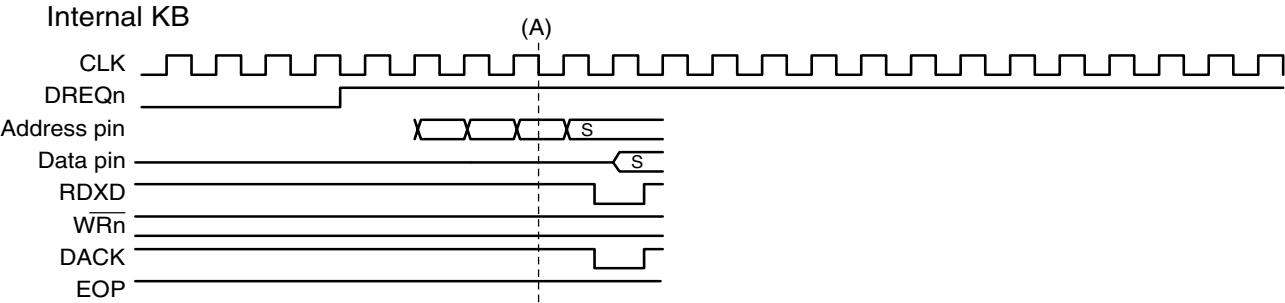
- Request pin input mode: Level, descriptor address: External

Figure 15.5-1 Descriptor Access Block Timing Diagram 1



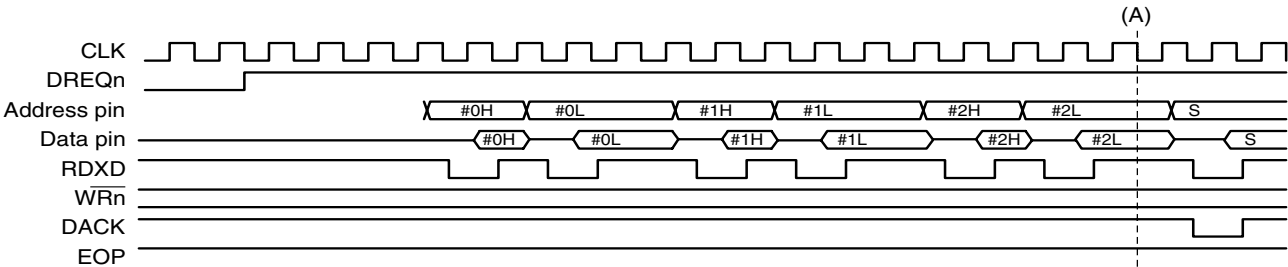
- Request pin input mode: Level, descriptor address: Internal

Figure 15.5-2 Descriptor Access Block Timing Diagram 2



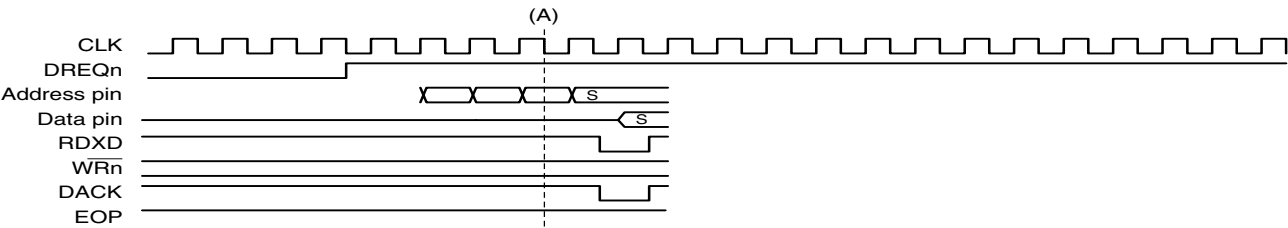
- Request pin input mode: Edge, descriptor address: External

Figure 15.5-3 Descriptor Access Block Timing Diagram 3



- Request pin input mode: Edge, descriptor address: Internal

Figure 15.5-4 Descriptor Access Block Timing Diagram 4



Note:

The above conditions reflect the case in which the time from DREQ occurrence to start of DMAC operation is shortest.

For actual operation, the start of DMAC operation can be delayed due to bus conflicts related to CPU instruction fetch or data access.

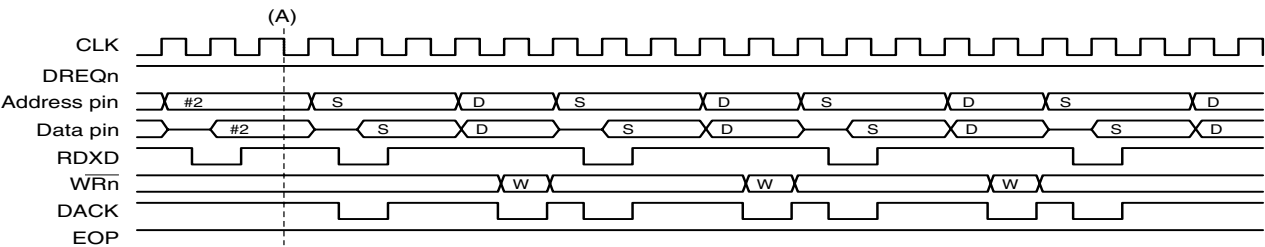
15.5.2 Timing Diagrams of the Data Transfer Block

This section provides the timing diagrams of the data transfer block.

■ Data Transfer Block, 16-bit/8-bit Data

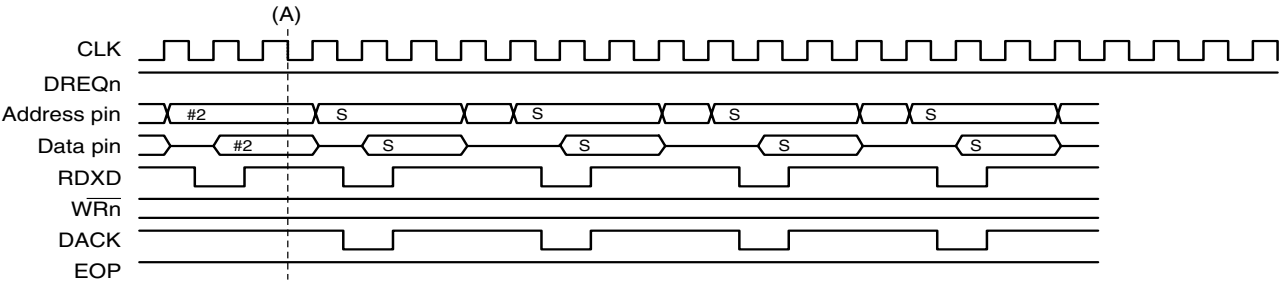
○ Transfer source area: External, transfer destination area: External

Figure 15.5-5 Data Transfer Block Timing Diagram 1



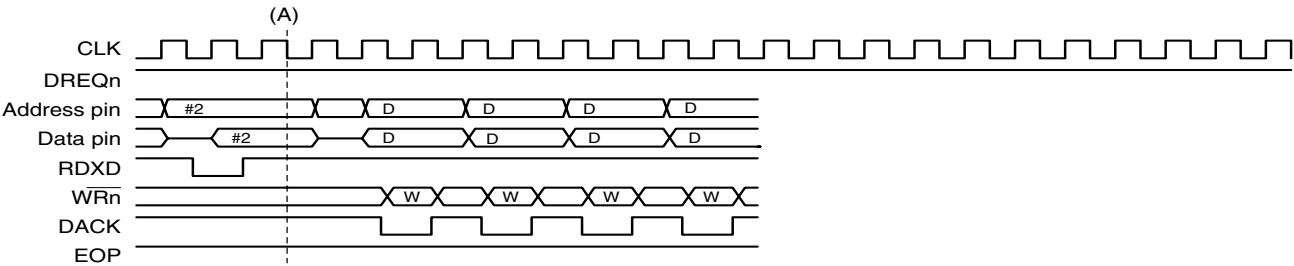
○ Transfer source area: External, transfer destination area: Internal RAM

Figure 15.5-6 Data Transfer Block Timing Diagram 2



- Transfer source area: Internal RAM, transfer destination area: External

Figure 15.5-7 Data Transfer Block Timing Diagram 3



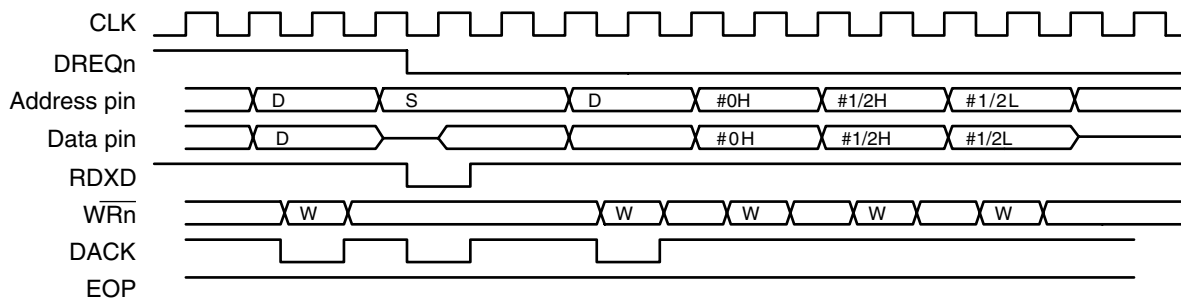
15.5.3 Timing Diagrams of Transfer Stop in Continuous Transfer Mode

This section provides the timing diagrams of transfer stop in continuous transfer mode.

■ Transfer Stop in Continuous Transfer Mode (When Either of the Addresses is Fixed), 16-bit/8-bit Data

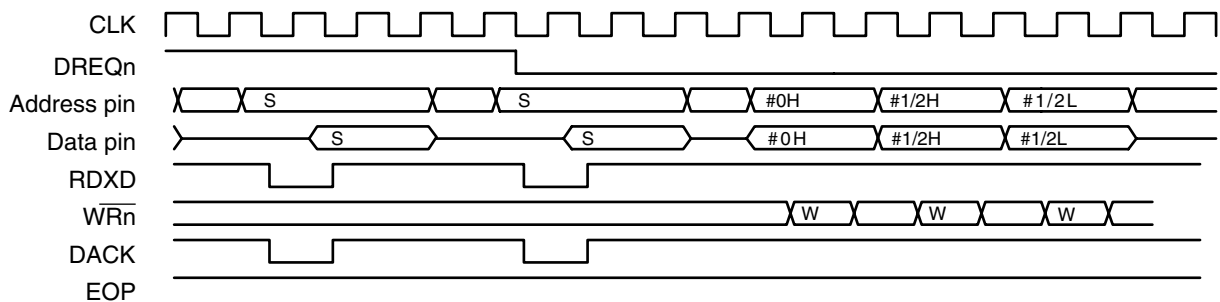
- Transfer source area: External, transfer destination area: External

Figure 15.5-8 Timing Diagram 1 of Transfer Stop in Continuous Transfer Mode



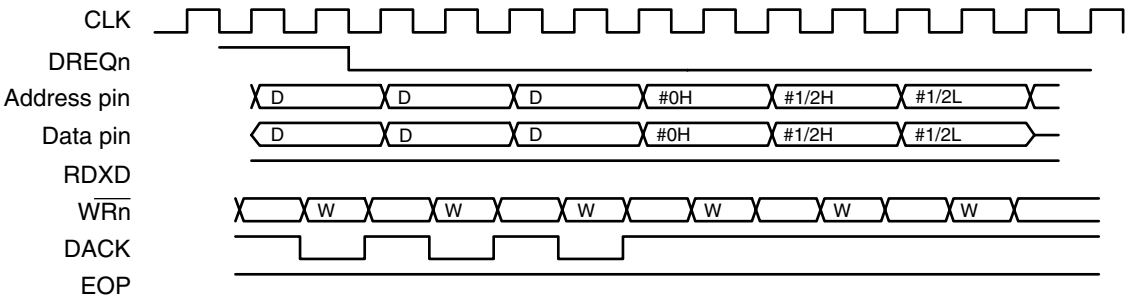
- Transfer source area: External, transfer destination area: Internal RAM

Figure 15.5-9 Timing Diagram 2 of Transfer Stop in Continuous Transfer Mode



○ Transfer source area: Internal RAM, transfer destination area: External

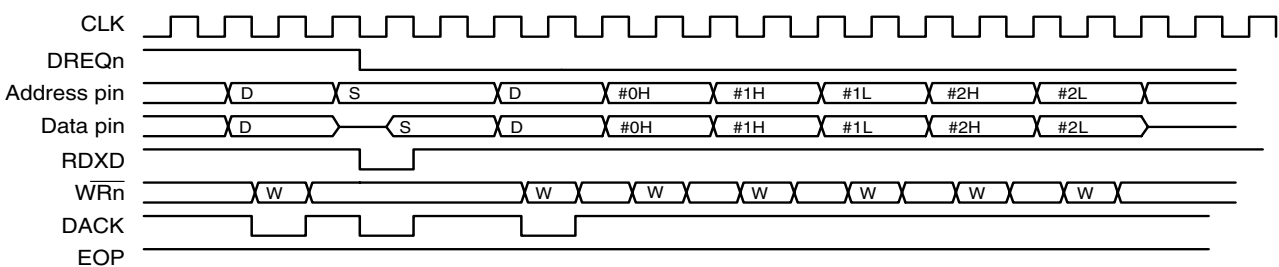
Figure 15.5-10 Timing Diagram 3 of Transfer Stop in Continuous Transfer Mode



■ Transfer Stop in Continuous Transfer Mode (When Both Addresses Change), 16-bit/8-bit Data

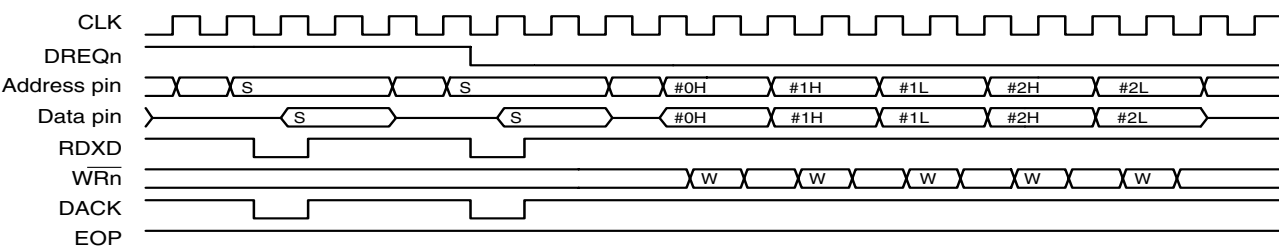
○ Transfer source area: External, transfer destination area: External

Figure 15.5-11 Timing Diagram 4 of Transfer Stop in Continuous Transfer Mode



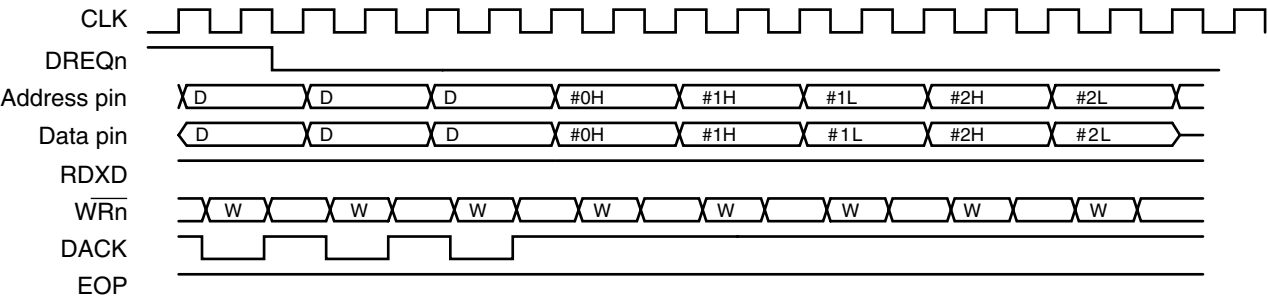
○ Transfer source area: External, transfer destination area: Internal RAM

Figure 15.5-12 Timing Diagram 5 of Transfer Stop in Continuous Transfer Mode



- Transfer source area: Internal RAM, transfer destination area: External

Figure 15.5-13 Timing Diagram 6 of Transfer Stop in Continuous Transfer Mode

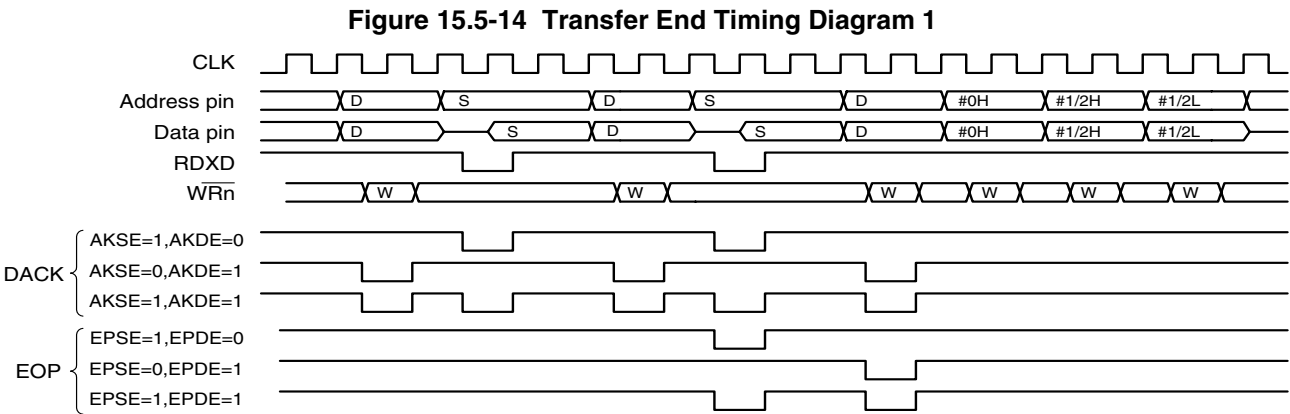


15.5.4 Timing Diagrams of Transfer End

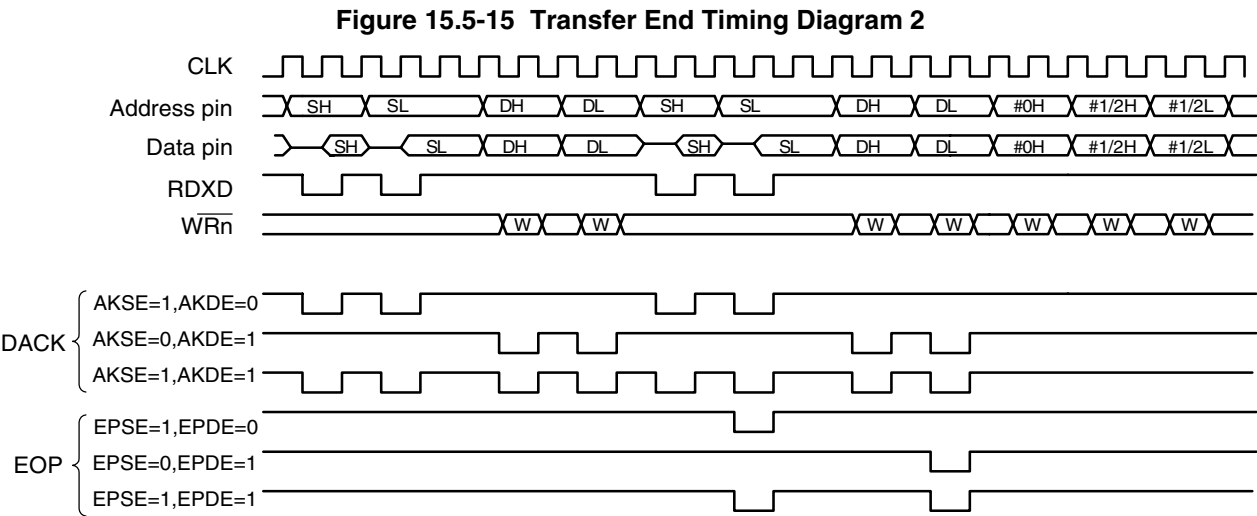
This section provides the timing diagrams of transfer end.

■ Transfer End (When Either of the Addresses is Fixed)

- Bus width: 16 bits, data length: 8 bits/16 bits



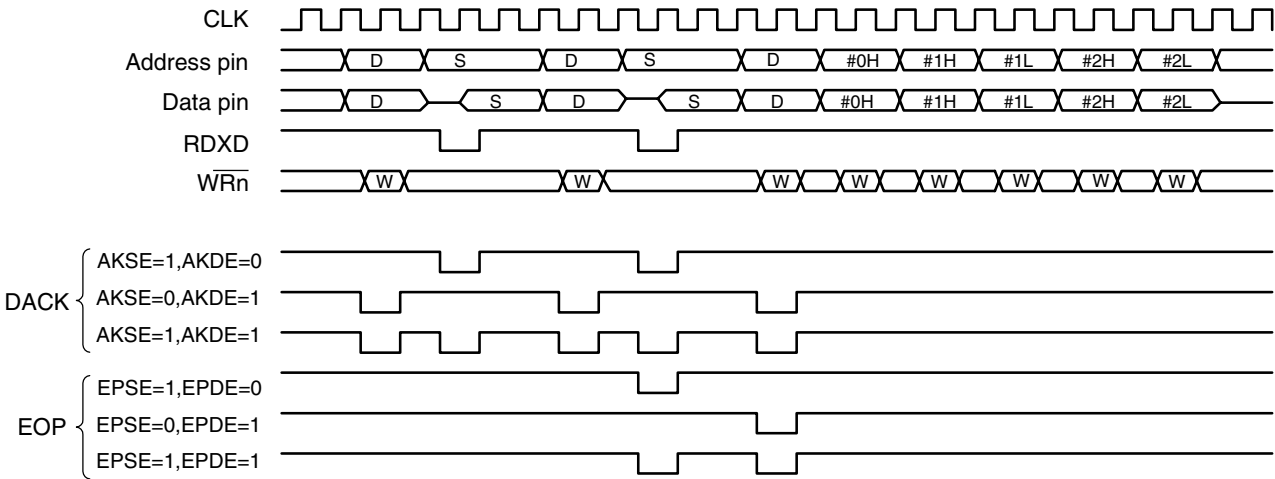
- Bus width: 16 bits, data length: 32 bits



■ Transfer End (when both addresses change)

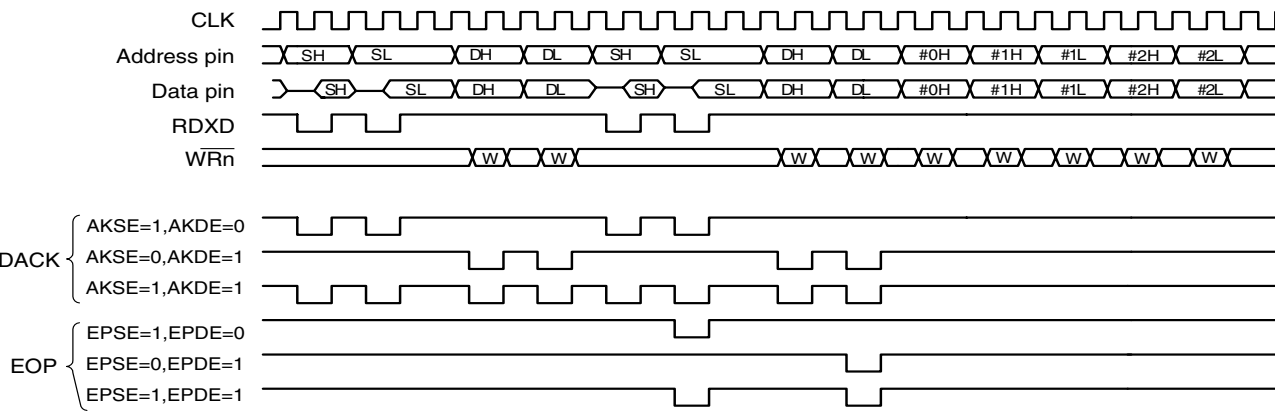
○ Bus width: 16 bits, data length: 8 bits/16 bits

Figure 15.5-16 Transfer End Timing Diagram 3



○ Bus width: 16 bits, data length: 32 bits

Figure 15.5-17 Transfer End Timing Diagram 4



15.6 Notes on the DMA Controller

This section provides notes on using the DMA controller.

■ Order of Priority Between the Channels

When a DMA transfer request has activated a channel, the DMA controller will hold the channel and will not accept transfer requests from other channels until the current transfer operation ends.

If multiple channel requests are active at the same time the DMAC detects a DMA transfer request, the DMAC will determine the order of priority of the channels as follows:

(Highest) channel 0 --> channel 1 --> channel 2 --> channel 3 --> channel 4 (lowest)

When multiple channel requests occur at the same time, DMA transfer of one channel will be performed. Control will then be returned to the CPU before DMA transfer of the next channel is performed.

■ Using Resource Interrupt Requests as DMA Transfer Requests

To enable DMAC transfer, the interrupt level of the interrupt controller of the relevant interrupt must be set to interrupt prohibited.

Conversely, to enable interrupts, the DMAC operation enable bit in the DMAC must be set to prohibited state. In addition, the interrupt level must be set to an appropriate value.

■ Suppressing DMA Transfer When an Interrupt Having Higher Priority Occurs

The MB91F127/128 has a function for stopping DMA transfer when an interrupt with higher priority occurs. This applies to DMA transfer operations that are performed using the occurrence of a DMA transfer request.

○ HRCL register

The hold request cancel level register (HRCL) of the interrupt controller can be used to stop DMA transfer when an interrupt request occurs.

When an interrupt request from a peripheral circuit occurs, the DMAC will suppress DMA transfer if the interrupt level set for the interrupt request is higher than the interrupt level set by the HRCL register. If DMA transfer is being executed, the DMAC will cancel the transfer operation at the break of the transfer operation and open bus authority for the CPU. If the DMAC is waiting for the occurrence of a DMA transfer request, the DMAC will maintain hold status even if a DMA transfer request occurs.

After a reset, the HRCL register will be set to the lowest level (31). As a result, DMA transfer will be suppressed for all interrupt requests. To perform DMA transfer even when an interrupt request occurs, set the HRCL register to the required value.

○ PDRR register

Suppression of DMA transfer operations using the HRCL register is valid only when higher priority interrupt requests are active. Therefore, if interrupt requests are cleared in a program such as an interrupt handler, suppression of DMA transfer using the HRCL register can be released at that time and the CPU can lose bus authority.

To enable the reception of other interrupt requests, a PDRR register is provided in the clock controller for clearing interrupt requests and suppressing DMA transfer operations.

Write a value other than 0 to the PDRR in the interrupt handler to suppress DMA transfer operations. To release suppression of DMA transfer operations, write "0" to the PDRR.

■ DMA Transfer Operations in Sleep Mode

The DMA transfer operation cannot be used in sleep mode. Be sure to disable the DMA transfer operation before switching to sleep mode.

■ Transfer Operation to the DMA Controller Internal Register

Do not specify a DMAC internal register as the transfer destination address.

■ Continuous Transfer

In continuous transfer mode, a descriptor may be written back even during transfers, depending on the status of the internal bus buffer of the device. Even if this occurs, the transfer operation continues and does not end.

■ Operation of External Transfer of Internal Memory

In block transfer mode, a DMA transfer is performed twice for each DREQ.

In continuous transfer mode, a DMA transfer is performed one extra time even if a DREQ is ceased. Select and carry out one of the following measures to resolve this problem:

- Use a DREQ in edge detection mode (effective only in block mode).
- Specify a destination address that is in an external area to cause a DACK to occur during access to the destination.
- For both the source and destination addresses, set a descriptor in external memory unless they are fixed addresses.

CHAPTER 16 BIT SEARCH MODULE

This chapter outlines the bit search module and describes the register configuration/ functions, operations of the bit search module, and save/return processing.

16.1 "Outline of the Bit Search Module"

16.2 "Bit Search Module Registers"

16.3 "Bit Search Module Operation and Saving/Restoring"

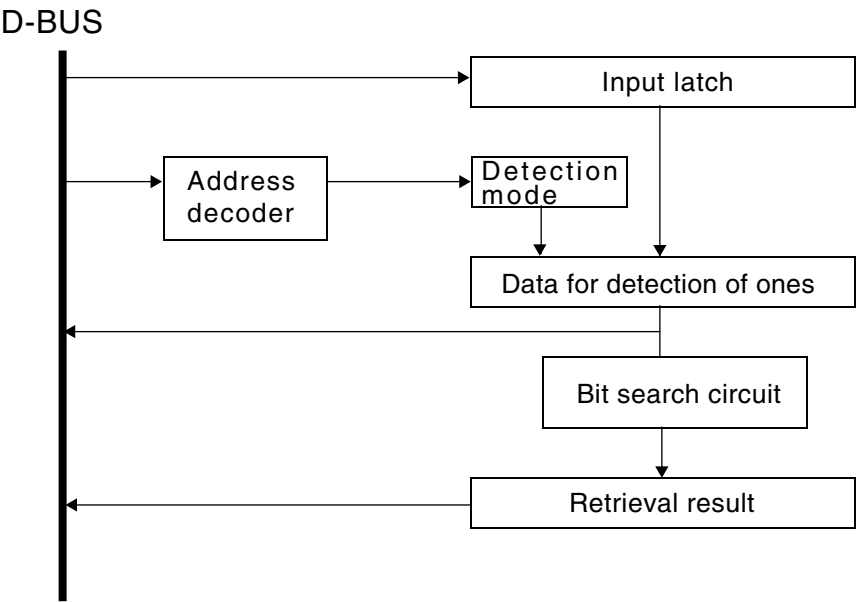
16.1 Outline of the Bit Search Module

The bit search module identifies the positions of 0, 1, or bit changes in data written to input registers and returns the positions of the detected bits.

■ Block Diagram

Figure 16.1-1 "Block Diagram of the Bit Search Module" shows a block diagram of the bit search module.

Figure 16.1-1 Block Diagram of the Bit Search Module



16.2 Bit Search Module Registers

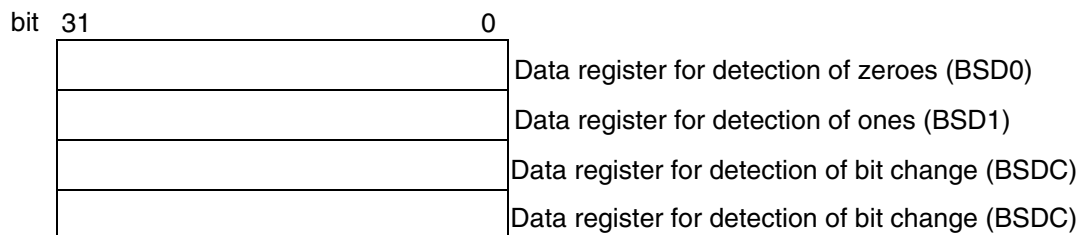
The four bit search module registers are as follows:

- Data register for detection of zeroes (BSD0)
- Data register for detection of ones (BSD1)
- Data register for detection of bit change (BSDC)
- Detection result register (BSRR)

■ Bit Search Module Registers

Figure 16.2-1 "Bit Search Module Registers" Bit Search Module Registers

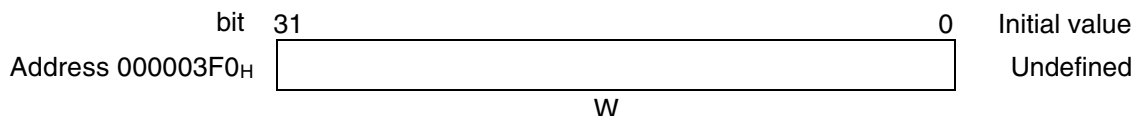
Figure 16.2-1 Bit Search Module Registers



■ Data Register for Detection of Zeroes (BSD0)

Figure 16.2-2 "Register Configuration of BSD0" shows the register configuration of BSD0.

Figure 16.2-2 Register Configuration of BSD0



Zeros in the written value are detected.

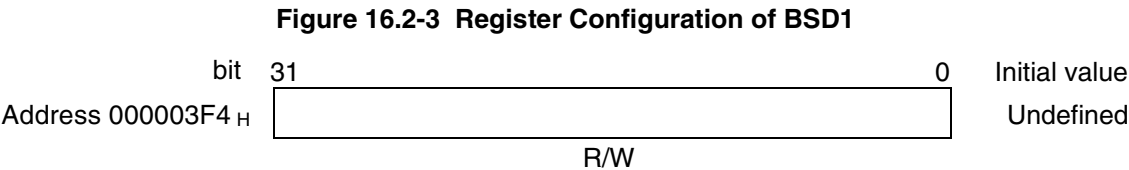
The initial value after a reset is undefined.

The read value is undefined.

For data transfer, use 32-bit data transfer instructions. (Do not use 8-bit or 16-bit data transfer instructions.)

■ Data Register for Detection of Ones (BSD1)

Figure 16.2-3 "Register Configuration of BSD1" shows the register configuration of BSD1.



For data transfer, use 32-bit data transfer instructions. Do not use 8-bit or 16-bit data transfer instructions.

○ During writing

"Ones" in the written values are detected.

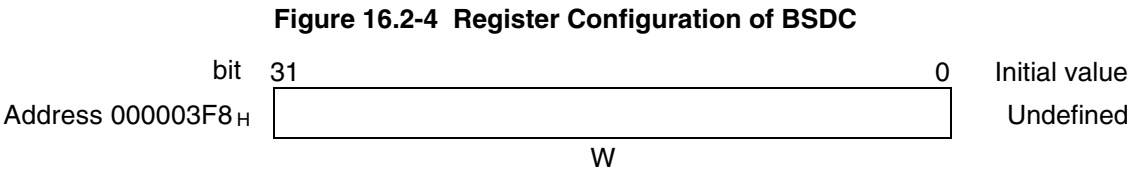
○ During read operations

The data for saving the internal state of the bit search module is read. If a program such as an interrupt handler uses the bit search module, use this data for saving and restoring the original state. Even if data is written to the data register for detection of zeroes or data register for detection of bit change, the original state can be saved and restored by using only the data register for detection of ones.

The initial value after a reset is undefined.

■ Data register for detection of bit change (BSDC)

Figure 16.2-4 "Register Configuration of BSDC" shows the register configuration of the BSDC.



Bit changes in the written value are detected.

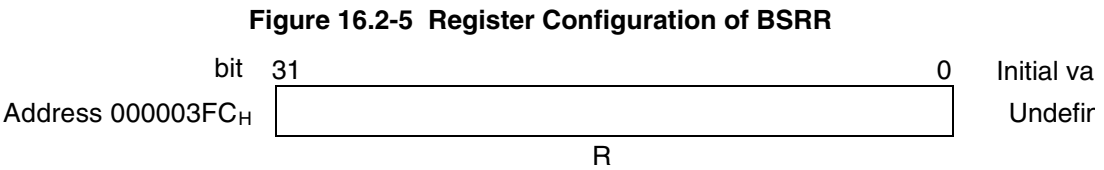
The initial value after a reset is undefined.

The read value is undefined.

For data transfer, use 32-bit data transfer instructions. (Do not use 8-bit or 16-bit data transfer instructions.)

■ Detection result register (BSRR)

Figure 16.2-5 "Register Configuration of BSRR" shows the register configuration of the BSRR.



The result of detection of zeroes, detection of ones, or bit change detection is read.

The detection result that is read is determined by the last data register written to.

16.3 Bit Search Module Operation and Saving/Restoring

This section describes the operation of the bit search module as well as the process of saving/restoring for detection of 0, detection of 1, and bit change detection by the bit search module.

■ Detection of Zeroes

The data written to the data register for detection of zeroes is scanned from the MSB to the LSB. The position where the first zero is detected is returned.

The detection result can be obtained by reading the detection result register.

Table 16.3-1 "Bit Positions and Return Values" shows the relationship between the detected positions and return values.

If no zero can be found (that is, the value is FFFFFFFF_H), 32 will be returned as the search result.

[Execution example]

Written data	Read value (decimal)
11111111 11111111 11110000 00000000 _B	(FFFFF000) _H → 20
11111000 01001001 11100000 10101010 _B	(F849E0AA) _H → 5
10000000 00000010 10101010 10101010 _B	(8002AAAA) _H → 1
11111111 11111111 11111111 11111111 _B	(FFFFFFFF) _H → 32

■ Detection of Ones

The data written to the data register for detection of ones is scanned from the MSB to the LSB. The position at which the first one is detected is returned.

The detection result can be obtained by reading the detection result register.

Table 16.3-1 "Bit Positions and Return Values" shows the relationship between the detected positions and return values.

If no one can be found (that is, the value is 00000000_H), 32 will be returned as the search result.

[Execution example]

Written data	Read value (decimal)
00100000 00000000 00000000 00000000 _B	(20000000) _H → 2
00000001 00100011 01000101 01100111 _B	(01234567) _H → 7
00000000 00000011 11111111 11111111 _B	(0003FFFF) _H → 14
00000000 00000000 00000000 00000001 _B	(00000001) _H → 31
00000000 00000000 00000000 00000000 _B	(00000000) _H → 32

■ Detection of Bit Change

The data written to the data register for detection of bit change is scanned from bit 30 to the LSB and compared with the value of the MSB. The positions of the first bit that is different from the respective bit in the MSB is detected are returned.

The detection result can be obtained by reading the detection result register.

Table 16.3-1 "Bit Positions and Return Values" shows lists the relationship between the detected positions and return values.

If no bit change can be found, 32 will be returned.

Bit change detection will never return 0 as the result of detection.

[Execution example]

Written data		Read value (decimal)	
00100000 00000000 00000000 00000000 _B	(20000000) _H	→	2
00000001 00100011 01000101 01100111 _B	(01234567) _H	→	7
00000000 00000011 11111111 11111111 _B	(0003FFFF) _H	→	14
00000000 00000000 00000000 00000001 _B	(00000001) _H	→	31
00000000 00000000 00000000 00000000 _B	(00000000) _H	→	32
11111111 11111111 11110000 00000000 _B	(FFFFFF00) _H	→	20
11111000 01001001 11100000 10101010 _B	(F849E0AA) _H	→	5
10000000 00000010 10101010 10101010 _B	(8002AAAA) _H	→	1
11111111 11111111 11111111 11111111 _B	(FFFFFFFF) _H	→	32

Table 16.3-1 Bit Positions and Return Values

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 found	32

■ Saving and Returning

Use the following procedure if the internal state of the bit search module must be saved and returned such as when the bit search module is used by an interrupt handler:

1. Read the data register for detection of ones and retain (save) its contents.
2. Use the bit search module.
3. Write the data saved in step 1 to the data register for detection of ones (returning).

By executing the above operations, the value obtained when the detection result register is read next will correspond to the contents written to the bit search module.

Even if the data register written to last is the data register for detection of zeroes or data register for detection of bit change, the above procedure will return to the original state correctly.

CHAPTER 17 FLASH MEMORY

This chapter provides an outline of flash memory and explains its register configuration, register functions, and operations.

17.1 "Outline of Flash Memory"

17.2 "Flash Memory Registers"

17.3 "Flash Memory Access Modes"

17.4 "Starting the Automatic Algorithm"

17.5 "Execution Status of the Automatic Algorithm"

17.6 "Sector Protect Operation"

17.1 Outline of Flash Memory

MB91F127/128 has an internal flash memory of 256 kilobytes (2 megabits) or 510 kilobytes (4 megabits) that enables to perform the following functions with a single +3 V power supply: simultaneous erasure of all sectors, erasure in sector units, and writing in half-word (16 bits) units via the FR-CPU.

■ Outline of Flash Memory

The flash memory that is employed is an internal 256-kilobyte (MB91F127) or 510-kilobyte (MB91F128) flash memory operated at 3 V.

The flash memory employed here is basically the same as the Fujitsu 4-megabit (510 kilobits × 8 or 254 kilobits × 16) flash memory MBM29LV400TC (except for a part of the sector configuration) and enables writing with a device-external ROM writer.

When this memory is used as FR-CPU internal ROM, it becomes possible to read instructions and data in word units (32 bits), in addition to features equivalent to the features of the MBM29LV400TC. This enables high-speed device operation.

Along with this manual, refer to the MBM29LV400TC Data Sheet.

The following features are implemented by combining flash memory macros and FR-CPU interface circuits:

- Features for use as CPU memory, for storing programs and data
 - Accessibility through 32-bit bus when used as ROM
 - Allowing read, write, and erase (automatic program algorithm*1) by the CPU
 - Features of a single flash memory product equivalent to MBM29LV200T
 - Allowing read or write (automatic program algorithm *1) by a ROM writer
- *1: Automatic program algorithm: embedded algorithm™

This section explains use of the flash memory accessed from the FR-CPU.

For information on using the flash memory accessed from a ROM writer, see the instruction manual provided with the ROM writer.

■ Execution Status of the Automatic Algorithm

When the automatic algorithm is started in CPU programming mode, its operation status can be checked with the internal Busy or Ready signal (RDY/BUSYX). The level of this signal can be read from the "RDY" bit of the flash memory status register.

When the "RDY" bit is "0", the automatic algorithm performs a write or read and another Read or Erase command cannot be accepted. Data cannot be read from a flash memory address either.

Data read when the "RDY" bit is "0" determines the setting of a hardware sequence flag indicating flash memory status (see Section 17.5 "Execution Status of the Automatic Algorithm").

■ Interrupt Control

When the automatic algorithm sequence ends, an interrupt request can be issued to the CPU, thereby making it possible to quickly recognize the end of an automatic algorithm sequence that has continued for an extended period.

The "RDYINT" and "INTE" bits of the flash memory status register control the interrupt at the end of the automatic algorithm.

The "RDYINT" bit is an interrupt flag set at the end of the automatic algorithm. When the rising edge of the internal Ready or Busy signal (RDY/BUSYX) from "0" to "1" is detected, the interrupt flag is set to "1". When the "INTE" bit is "1" and the "RDYINT" bit is set, an interrupt request is output to the CPU.

When canceling the interrupt request, set the "RDYINT" or "INTE" bit to "0".

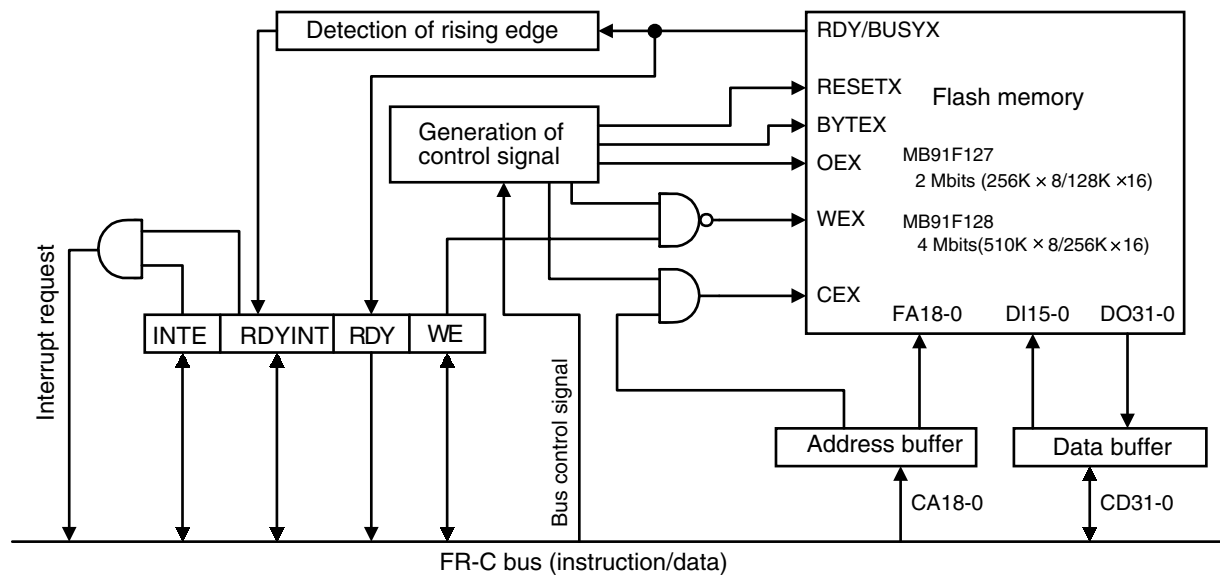
■ Writing by ROM Writer

This flash memory enables writing by a device-external ROM writer.

During writing by a device-external ROM writer, the pin functions equivalent to the functions of the single flash memory MBM29LV400TC are assigned to the external pins of the device and the FR-CPU stops operation. In CPU mode, address line connections are changed and the mapping in the memory area changes. For details, refer to the specification of "the corresponding ROM writer".

■ Block Diagram of Flash Memory

Figure 17.1-1 Block diagram of flash memory



■ Flash Memory Sector Configuration (MB91F127)

The flash memory has different address mapping depending on whether it is accessed from the FR-CPU or from the ROM writer. Figure 17.1-2 "Mapping for Access from FR-CPU (MB91F127)" and Table 17.1-1 "Sector Addresses (MB91F127)" show the mapping for access from the FR-CPU. Figure 17.1-3 "Mapping and Sector Configuration for Access from ROM Writer (MB91F127)" shows the mapping for access from the ROM writer.

○ Mapping for access from the FR-CPU

Figure 17.1-2 Mapping for Access from FR-CPU (MB91F127)

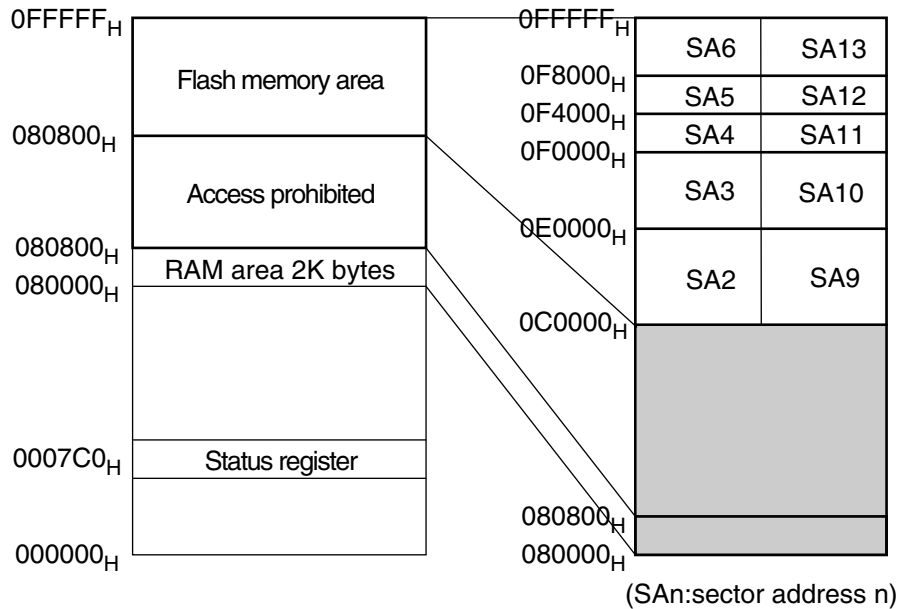
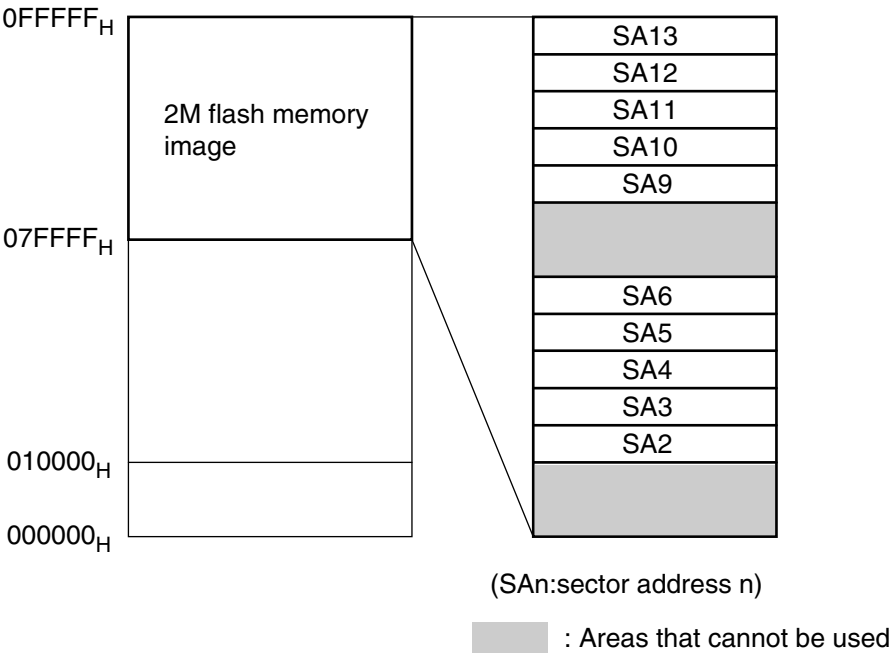


Table 17.1-1 Sector Addresses (MB91F127)

Sector address	Address range	Corresponding bit location	Sector capacity
SA9	0C0002, 3 _H to 0DFFFE, F _H (16 bits on LSB side)	Bits 15 to 0	64K bytes
SA10	0E0002, 3 _H to 0EFFFF, F _H (16 bits on LSB side)	Bits 15 to 0	32K bytes
SA11	0F0002, 3 _H to 0F3FFE, F _H (16 bits on LSB side)	Bits 15 to 0	8K bytes
SA12	0F4002, 3 _H to 0F7FFE, F _H (16 bits on LSB side)	Bits 15 to 0	8K bytes
SA13	0F8002, 3 _H to 0FFFFE, F _H (16 bits on LSB side)	Bits 15 to 0	16K bytes
SA2	0C0000, 1 _H to 0DFFFC, D _H (16 bits on MSB side)	Bits 31 to 16	64K bytes
SA3	0E0000, 1 _H to 0EFFFF, D _H (16 bits on MSB side)	Bits 31 to 16	32K bytes
SA4	0F0000, 1 _H to 0F3FFC, D _H (16 bits on MSB side)	Bits 31 to 16	8K bytes
SA5	0F4000, 1 _H to 0F7FFC, D _H (16 bits on MSB side)	Bits 31 to 16	8K bytes
SA6	0F8000, 1 _H to 0FFFFC, D _H (16 bits on MSB side)	Bits 31 to 16	16K bytes

○ Mapping and sector configuration for access from the ROM writer

Figure 17.1-3 Mapping and Sector Configuration for Access from ROM Writer (MB91F127)



Flash Memory Sector Configuration (MB91F128)

The flash memory has different address mapping depending on whether it is accessed from the FR-CPU or from the ROM writer. Figure 17.1-4 "Mapping for Access from FR-CPU (MB91F128)" and Table 17.1-2 "Sector Addresses (MB91F128)" show the mapping for access from the FR-CPU. Figure 17.1-5 "Mapping and Sector Configuration for Access from ROM Writer (MB91F128)" shows the mapping for access from the ROM writer.

Mapping for access from the FR-CPU

Figure 17.1-4 Mapping for Access from FR-CPU (MB91F128)

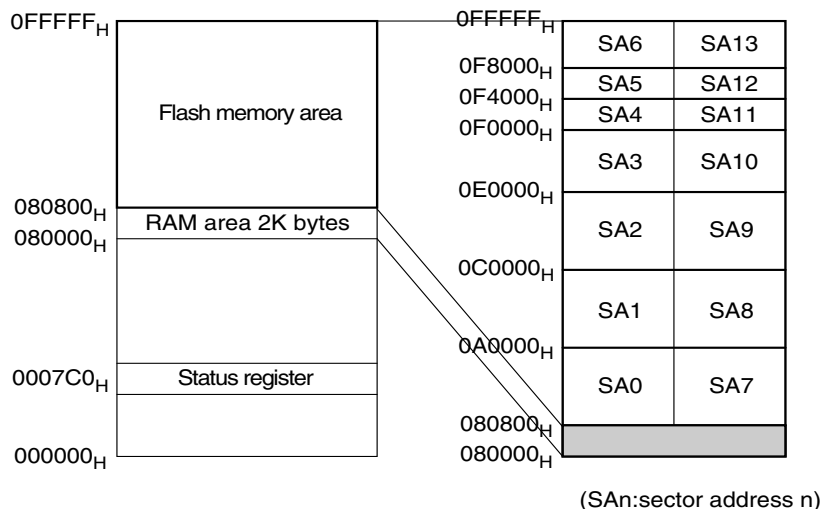


Table 17.1-2 Sector Addresses (MB91F128)

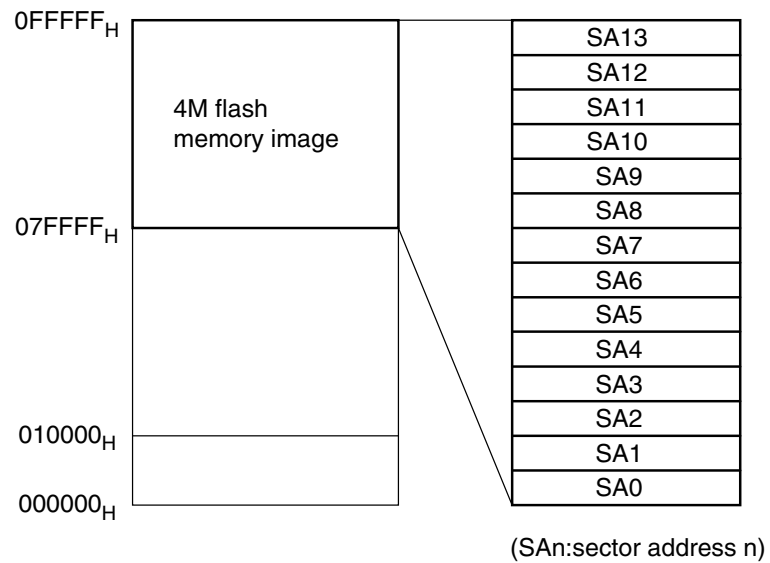
Sector address	Address range	Corresponding bit location	Sector capacity
SA7	080802, 3 _H to 09FFFE, F _H (16 bits on LSB side)	Bits 15 to 0	
SA8	0A0002, 3 _H to 0BFFFE, F _H (16 bits on LSB side)	Bits 15 to 0	64K bytes
SA9	0C0002, 3 _H to 0DFFFE, F _H (16 bits on LSB side)	Bits 15 to 0	64K bytes
SA10	0E0002, 3 _H to 0E7FFE, F _H (16 bits on LSB side)	Bits 15 to 0	64K bytes
SA11	0F0002, 3 _H to 0F3FFE, F _H (16 bits on LSB side)	Bits 15 to 0	32K bytes
SA12	0F4002, 3 _H to 0F7FFE, F _H (16 bits on LSB side)	Bits 15 to 0	8K bytes
SA13	0F8002, 3 _H to 0FFFFE, F _H (16 bits on LSB side)	Bits 15 to 0	8K bytes
SA0	080800, 1 _H to 09FFFC, D _H (16 bits on MSB side)	Bits 31 to 16	16K bytes
SA1	0A0000, 1 _H to 0BFFFC, D _H (16 bits on MSB side)	Bits 31 to 16	64K bytes
SA2	0C0000, 1 _H to 0D3FFC, D _H (16 bits on MSB side)	Bits 31 to 16	64K bytes
SA3	0E0000, 1 _H to 0E7FFC, D _H (16 bits on MSB side)	Bits 31 to 16	64K bytes
SA4	0F0000, 1 _H to 0F3FFC, D _H (16 bits on MSB side)	Bits 31 to 16	32K bytes
SA5	0F4000, 1 _H to 0F7FFC, D _H (16 bits on MSB side)	Bits 31 to 16	8K bytes

Table 17.1-2 Sector Addresses (MB91F128) (Continued)

Sector address	Address range	Corresponding bit location	Sector capacity
SA6	0F8000, 1 _H to 0FFFFC, D _H (16 bits on MSB side)	Bits 31 to 16	8K bytes

- Mapping and sector configuration for access from the ROM writer

Figure 17.1-5 Mapping and Sector Configuration for Access from ROM Writer (MB91F128)



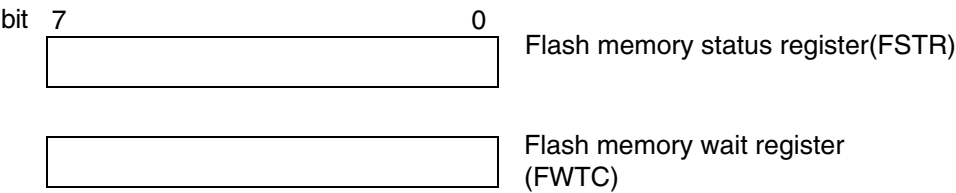
17.2 Flash Memory Registers

The flash memory has types of two registers: flash memory status register (FSTR) and flash memory wait register (FWTC).

■ List of Flash Memory Registers

Figure 17.2-1 "Flash Memory Registers" shows a list of flash memory registers.

Figure 17.2-1 Flash Memory Registers



17.2.1 Flash Memory Status Register (FSTR)

The flash memory status register (FSTR) indicates the operation status of the flash memory.

■ Flash Memory Status Register (FSTR)

This register also controls interrupts to the CPU and writing to the flash memory. Only the CPU can access this register. Even if a writer is provided, it cannot access this register. Do not access this register with Read Modify Write instructions.

Figure 17.2-2 "Bit Configuration of FSTR" shows the bit configuration of the FSTR.

Figure 17.2-2 Bit Configuration of FSTR

bit	7	6	5	4	3	2	1	0	Initial value
Address 000007C0 _H	INTE	RDYINT	WE	RDY	—	—	—	—	000XXXX0 _B
	R/W	R/W	R/W	R	—	—	—	R/W	

The functions of bits of the FSTR are explained below.

[bit 7] INTE (INTerrupt Enable)

The INTE bit controls interrupts generated by the termination of the automatic algorithm in flash memory (for a write/erase operation etc.).

This bit is initialized to "0" during a reset. Read and write operations are enabled.

INTE	Function
0	disables issuing interrupts at termination of the automatic algorithm. (This is the initial value)
1	enables issuing interrupts at termination of the automatic algorithm.

[bit 6] RDYINT (ReaDY INTerrupt)

The PDYINT bit is set to "1" when the automatic algorithm (for a write/erase operation etc.) in flash memory terminates.

When bit 7 (INT = "1") enables interrupt output and this bit (bit 6) is set to "1", an interrupt request for terminating the automatic algorithm is generated.

After a reset, the bit is initialized "0". Read/Write operations for this bit are enabled. However, only write operations with the value "0" are valid: even when a write operation attempts to set "1", the value of this bit remains unchanged.

Cause for clearing: Clear is performed by writing "0" through an instruction.

Cause for setting: The bit is set by termination of the automatic algorithm (when the rising edge of the RDY/BUSYX signal is detected).

[bit 5] WE (Write Enable)

The WE bit controls writing data and commands to the flash memory in CPU mode.

When this bit is "0", writing data and commands to the flash memory becomes invalid. Data from flash memory is read in 32-bit access mode.

When this bit is "1", writing data and commands to the flash memory becomes valid and the automatic algorithm can be started. However, data from flash memory is read in 16-bit access mode, during which flash memory cannot be used as program memory because 32-bit access is inhibited.

When overwriting this bit, ensure that the RDY bit has caused a stop of the automatic algorithm (write/erase). When the RDY bit is "0", the value of this bit cannot be changed.

This bit is initialized to "0" during a reset. Read and write operations are enabled.

WE	Function
0	inhibits writing to the flash memory and enables 32-bit read operations (ROM mode) [this is the initial value].
1	enables writing to flash memory and inhibits 32-bit read operations (programming mode).

[bit 4] RDY (ReaDY)

The RDY bit indicates the operation status of the automatic algorithm (write/erase).

When this bit is "0", the automatic algorithm is executing a write or erase operation and another Write or Erase command cannot be accepted. Data also cannot be read from an address in flash memory. Reading this bit indicates the status of flash memory.

For details, see Section 17.5 "Execution Status of the Automatic Algorithm."

This bit is initialized to "0" during a reset. Read and write operations are enabled.

RDY	Function
0	Writing or erasing is in progress, flash memory is not ready to accept a new Read, Write, or Erase command
1	Flash memory is ready to accept a new Read, Write, or Erase command.

[bit 3 to 1] (reserved bit)

These bits are reserved bits. Their values during read operations are undefined, and they do not affect write operations.

[bit 0] (reserved bit)

This bit is a reserved bit. Read operations for this bit return "0". Always set this bit to "0".

When the bit is set to "1", the results of subsequent operations may become uncertain.

This bit is initialized to "0" during a reset.

17.2.2 Flash Memory Wait Register (FWTC)

The flash memory wait register (FWTC) controls the wait status of flash memory in CPU mode. This register also controls high-speed reading of flash memory.

■ Configuration of flash memory wait register (FWTC)

Figure 17.2-3 "Bit configuration of flash memory wait register (FWTC)" shows the bit configuration of the flash memory wait register (FWTC).

Figure 17.2-3 Bit configuration of flash memory wait register (FWTC)

bit	7	6	5	4	3	2	1	0	Initial value
Address 000007C4 _H	—	—	—	—	—	FACH	WTC1	WTC0	XXXXX000 _B
	—	—	—	—	—	W	R/W	R/W	

The functions of the bits of the flash memory wait register (FWTC) are explained below.

Bits 1 and 0: WTC1 and WTC0

These bits control wait state of flash memory.

WTC1, WTC0	Function
00	+0 wait, 2 cycles (initial value)
01	+1 wait, 3 cycles
10	Do not use
11	Do not use

Bit 2: FACH

This bit controls the speed at which flash memory is read.

FACH	Function
0	Normal reading (initial value)
1	High-speed reading

17.3 Flash Memory Access Modes

The following two types of access mode are available for the FR-CPU:

- **FR-CPU mode:** The CPU is accessed through the internal bus.
One word (32 bits) can be read but not written in a single cycle.
 - **Programming mode:** The CPU is directly accessed from the external pins.
Access to data with a length defined in words (32 bits) is prohibited but writing data with a length defined in half-words (16 bits) is enabled.
-

■ FR-CPU ROM Mode (32 Bits, Read only)

In this mode, the flash memory serves as FR-CPU internal ROM. This mode enables to read one word (32 bits) in one cycle but does not enable to write to flash memory or to start the automatic algorithm.

○ Mode specification

When specifying this mode, set the "WE" bit of the flash memory status register to "0".

This mode is always set after a reset occurs at CPU run time.

This mode can be set only when the CPU is running.

○ Detailed operation

In this mode, one word (32 bits) can be read from the flash memory area in one cycle.

Depending on the read operation, two cycles may be required per word (when 1 wait cycle is included), thereby making it possible to issue instructions to the FR-CPU with no wait.

○ Restrictions

Address assignment and endians in this mode differ from those for writing with the ROM writer.

In this mode, commands and data cannot be written to flash memory together.

■ FR-CPU Programming Mode (16 Bits, Read/Write)

This mode enables data to be written and erased. As one word (32 bits) cannot be accessed in one cycle, program execution in flash memory is disabled in this mode.

○ Mode specification

When specifying this mode, set the "WE" bit of the flash memory status register to "1".

When a reset occurs at CPU run time, the "WE" bit indicates "0". When setting this mode, set the "WE" bit to "1". If the "WE" bit is set again to "0" through a writing operation or because of a reset, the device enters ROM mode.

When the "RDY" bit of the flash memory status register is "0", the "WE" bit cannot be overwritten. When overwriting the "WE" bit, ensure that the "RDY" bit is set to "1".

○ Detailed operation

One half-word (16 bits) can be read from the flash memory area in one cycle.

Depending on the read operation, two cycles can be required for reading a half-word (when 1 wait cycle is included).

The automatic algorithm can be started by writing a command to flash memory.

When the automatic algorithm starts, data can be written to or erased from flash memory.

For details on the automatic algorithm, see Section 17.4 "Starting the Automatic Algorithm."

○ Restrictions

Address assignment and endians in this mode differ from those for writing with the ROM writer.

This mode inhibits reading data in words (32 bits).

■ Flash Memory Mode

Set the MD2, 1, and 0 pins to "1, 1, 1" and perform a reset to stop the functions of the CPU. At this time, a function of the flash memory interface circuit connects signals from ports 2, 3, 4, 5, and 6 directly to control signals of the flash memory, thereby enabling direct control of the flash memory from the external pins. It seems like that this mode allows the flash memory to appear as a single unit on the external pins. Use setting of this mode mainly for writing and erasing data using the flash memory writer.

In this mode, all operations provided by the automatic algorithms of the 4M-bit flash memory are available.

■ Correspondence of Flash Memory Control Signals to MBM29LV400TC

Table 17.3-1 shows the correspondence of the flash memory control signals to the MBM29LV400TC.

Table 17.3-1 Correspondence of the Flash Memory Control Signals to the MBM29LV400TC

MBM29LV400 TC external pin	MB91F127/MB91F128 external control pin		
	FR-CPU mode	Flash memory mode	
		Normal operation	VID application pin
RESET	RSTX	RSTX	MD1
RY/BY	None (Interrupt request to CPU)	RY/BYX	-
BYTE	Fixed to internal H	BYTEX	-
WE	Control by internal control signal and I/F circuit	WEX	-
OE		OEX	MD2
CE		CEX	-

Table 17.3-1 Correspondence of the Flash Memory Control Signals to the MBM29LV400TC (Continued)

MBM29LV400 TC external pin	MB91F127/MB91F128 external control pin		
	FR-CPU mode	Flash memory mode	
		Normal operation	VID application pin
A17 to A10	Internal address bus	AQ18 to AQ11	-
A9		AQ10	AQ0
A8 to A0		AQ9 to AQ1	-
A-1		MD0	-
DQ15 to DQ8	Internal data bus	None	-
DQ7 to DQ0		DQ7 to DQ0	-

17.4 Starting the Automatic Algorithm

For writing data to or erasing data from flash memory, start the automatic algorithm stored in flash memory.

■ Command Operation

At the start of the automatic algorithm, one to six half-words (16 bits) are written. This data is called the command.

If the address and data to be written are invalid or are written in an incorrect sequence, the flash memory is reset to read mode.

Table 17.4-1 "Commands" lists the commands of the automatic algorithm.

Table 17.4-1 Commands

Command sequence	Access count	First write cycle		Second write cycle		Third write cycle		Fourth write or read cycle		Fifth read cycle		Sixth write cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset	1	XXXXX	XXF0										
Read/Reset	4	D5555	XXAA	CAAAB	XX55	D5555	XXF0	(RA)	(RD)				
Program	4	D5555	XXAA	CAAAB	XX55	D5555	XXA0	(PA)	(PD)				
Chip Erase	6	D5555	XXAA	CAAAB	XX55	D5555	XX80	D5555	XXAA	CAAAB	XX55	D5555	XX10
Sector Erase	6	D5555	XXAA	CAAAB	XX55	D5555	XX80	D5555	XXAA	CAAAB	XX55	(SA)	XX30
Sector Erase Temporarily stop			XXXXX	XXB0									
Start Sector Erase			XXXXX	XX30									
Auto Select	3	D5555	XXAA	CAAAB	XX55	D5555	XX90						
	3	D5555	XXAA	CAAAB	XX55	D5555	XX20						
	2	XXXXX	XXA0	(PA)	(PD)								
	2	XXXXX	XX90	XXXXX or XX00									

(Notes)

- All addresses and data are in hexadecimal notation.
- RA: read address
- PA: write address
- SA: sector address (specify one sector address arbitrarily. See Table 17.1-1)
- RD: read data
- PD: write data
- The Temporarily Stop Sector Erase or Temporarily Stop Erase command (BOH) and the Start Sector Erase or Restart Erase command (30H) are valid only during a sector erase operation. Both Reset commands can reset flash memory to the read mode.

○ Read/Reset command

When returning to the read mode after the time limit was exceeded, a Read/Reset command sequence can be issued. Data is read from the flash memory in the next read cycle.

The flash memory remains in reading state until another command is entered.

When the power is turned on, flash memory is automatically set to the read or reset state. In this case, data can be read without a command of the automatic algorithm.

○ Program (Write)

In CPU programming mode, data is basically written in half-word units. The write operation is performed in four cycles of bus operation. The command sequence has two "unlock" cycles, which are followed by a Write Setup command and a write data cycle. Writing to memory starts in the last write cycle.

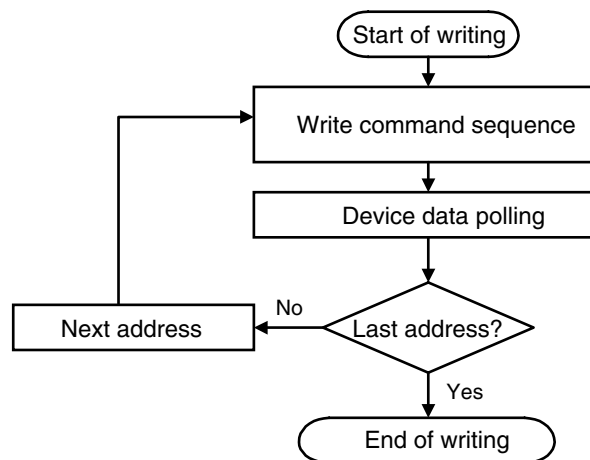
After an automatic write algorithm command sequence was executed, it becomes unnecessary to control the flash memory externally. The flash memory itself internally generates write pulses to check the margin of the cells to which data is written. The data polling function compares bit 7 of the original data with bit 7 of the written data, and if these bits are the same, the automatic write operation ends (see "Hardware sequence flag," in Section 17.5 "Execution Status of the Automatic Algorithm"). The automatic write operation then returns to the read mode and accepts no more write addresses. After that, the flash memory requests the next valid address. In this manner, the data polling function indicates a write operation in memory.

During a write operation, all commands written to the flash memory are ignored. If a hardware reset starts during write operation, the data at the address for writing may become invalid.

Writing operations can be performed in any address sequence and outside of sector boundaries. However, write operations cannot change a data item "0" to "1". If a "0" is overwritten with a "1", the data polling algorithm either determines that the elements are defective, or that "1" has been written. In the latter case, however, the respective data item is read as "0" in reset or read mode. A data item "0" can be changed to "1" only after an erase operation.

Figure 17.4-1 "Write Procedure Using the Write Command" shows the write procedure using the Write command.

Figure 17.4-1 Write Procedure Using the Write Command



○ Erase Chip

The Erase Chip command sequence ("erase all sectors simultaneously") is executed in six access cycles. First, two "unlock" cycles are executed, then a "Setup" command is written. After two more "unlock" cycles, the Erase Chip command is entered.

During the Erase Chip command sequence, the user does not have to write to flash memory before the erase operation. When the automatic erase algorithm is executed, flash memory checks cell states by writing a pattern of zeros before automatically erasing the contents of all cells (preprogram). In this operation, flash memory does not have to be controlled externally.

The automatic erase operation starts with the write operation of the command sequence and ends when bit 7 is set to "1", where flash memory returns to the read mode. The chip erase time can be expressed as follows: time for sector erase x number of all sectors + time for writing to the chip (preprogram).

○ Sector Erase

The Sector Erase command sequence is executed in six access cycles. First, two "unlock" cycles are executed, then a "Setup" command is written. After two more "unlock" cycles, the Sector Erase command is entered in the sixth cycle for starting the sector erase operation. The next Sector Erase command can be accepted within a time-out period of 50 μ s after the last Sector Erase command is written.

As already mentioned, multiple Sector Erase commands can be accepted during the six bus cycles of the writing operation. During the command sequence, Sector Erase commands (30H) for sectors whose contents are to be erased simultaneously are written consecutively to the addresses for these sectors. The sector erase operation itself starts from the end of the time-out period of 50 μ s after the last Sector Erase command is written. When the contents of multiple sectors are erased simultaneously, the subsequent Sector Erase commands must be input within the 50 μ s time-out period to ensure that they are accepted. For checking whether the succeeding Sector Erase command is valid, read bit 3 (see "Hardware sequence flag," in Section 17.5 "Execution Status of the Automatic Algorithm").

During the time-out period, any command other than Sector Erase and Temporarily Stop Erase is reset at read time, and the preceding command sequence is ignored. In the case of the Temporary Stop Erase command, the contents of the sector are erased again and the erase operation is completed.

Any combination and number (from 0 to 6) of sector addresses can be entered in the sector erase buffers.

The user does not have to write to flash memory before the sector erase operation.

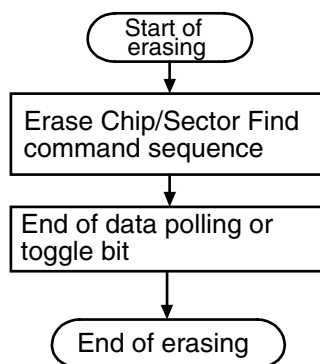
Flash memory automatically writes to all cells in a sector whose data is automatically erased (preprogram). When the contents of a sector are erased, the other cells remain intact. In these operations, flash memory does not have to be controlled externally.

The automatic sector erase operation starts from the end of the 50 μ s time-out period after the last Sector Erase command is written. When bit 7 is set to "1" (see "Hardware sequence flag," in Section 17.5 "Execution Status of the Automatic Algorithm"), the automatic sector erase operation ends and flash memory returns to the read mode. At this time, other commands are ignored.

The data polling function is enabled for any sector address in which data has been erased. The time required for erasing the data of multiple sectors can be expressed as follows: time for sector erase + time for sector write (preprogram) x number of erased sectors.

Figure 17.4-2 "Chip erase procedure using the Erase Chip command" shows the chip erase procedure using the Erase Chip command.

Figure 17.4-2 Chip erase procedure using the Erase Chip command



○ Temporarily Stop Erase

The Temporarily Stop Erase command temporarily stops the automatic algorithm in flash memory when the user is erasing the data of a sector, thereby making it possible to write data to and read data from the other sectors. This command is valid only during the sector erase operation and ignored during chip erase and write operations. The Temporarily Stop Erase command (B0H) is valid only during the sector erase operation including the sector erase time-out period. When this command is entered within the time-out period, waiting for time-out ends and the erase operation is suspended. The erase operation is restarted when a Restart Erase command was written. Temporarily Stop Erase and Restart Erase commands can be entered with any address.

When a Temporarily Stop Erase command is entered during sector erase operation, the flash memory needs a maximum of 20 μ s to stop the erase operation. When flash memory enters temporary erase stop mode, a Ready or Busy signal is output, bit 7 outputs "1", and bit 6 stops to toggle. For checking whether the erase operation has stopped, enter the address of the sector whose data is being erased and read the values of bit 6 and bit 7. At this time, another Temporarily Stop Erase command entry is ignored.

When the erase operation stops, flash memory enters the temporary erase stop and read mode. Data reading is enabled in this mode for sectors that are not subject to temporary erase. Other than that, there is no difference from the standard read operation. In this mode, bit 2 toggles for consecutive reading operations from sectors subject to temporary erase stop (see "Hardware sequence flag," in Section 17.5 "Execution Status of the Automatic Algorithm").

After the temporary erase stop and read mode is entered, the user can write to flash memory by writing a Write command sequence. The write mode in this case is the temporary erase stop and write mode. In this mode, data write operations become valid for sectors that are not subject to temporary erase stop. Other than that, there is no difference from the standard byte writing operation. In this mode, bit 2 toggles for consecutive reading operations from sectors that are subject to temporary erase stop. The temporary erase stop bit (bit 6) can be used to detect this operation.

Note that bit 6 can be read from any address, but bit 7 must be read from write addresses.

To restart the sector erase operation, a Restart Erase command (30H) must be entered. Another Restart Erase command entry is ignored in this case. On the other hand, a Temporarily Stop Erase command can be entered after flash memory restarts the erase operation.

17.5 Execution Status of the Automatic Algorithm

This flash memory has two hardware components for performing a Write or Erase sequence in the automatic algorithm. These components indicate the internal operation status of flash memory and the completion of operations to external components. One is a Ready/Busy signal and the other is a hardware sequence flag.

■ Ready/Busy Signal (RDY/BUSYX)

The flash memory uses the Ready/Busy signal in addition to the hardware sequence flag to indicate whether the internal automatic algorithm is running. The Ready/Busy signal is transmitted to the flash memory interface circuit, where it can be read via the "RDY" bit of the flash memory status register. An interrupt signal can also be generated for the CPU at the rising edge of this Ready/Busy signal (see Section 17.1 "Outline of Flash Memory").

When the value of the "RDY" bit is "0", the flash memory is executing a write or erase operation, where new Write and Erase commands are not accepted.

When the value of the "RDY" bit is "1", the flash memory is in read/write or erase operation wait state.

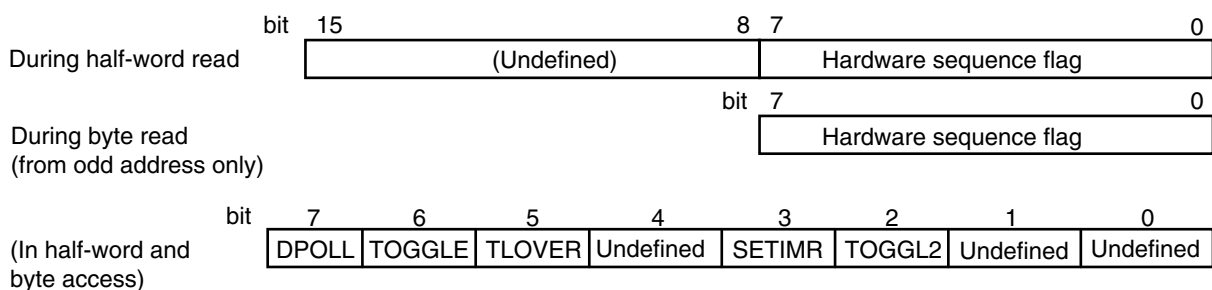
■ Hardware Sequence Flag

For obtaining the hardware sequence flag as data, read an arbitrary address (an odd address in byte access) from flash memory when the automatic algorithm is executed. The data contains five validity bits which indicate the status of the automatic algorithm.

Figure 17.5-1 "Structure of the Hardware Sequence Flag" shows the structure of the hardware sequence flag.

Reading in units of words is inhibited.

Figure 17.5-1 Structure of the Hardware Sequence Flag



The hardware sequence flag becomes invalid in FR-CPU ROM mode. Always use FR-CPU programming mode and write only in half-words or bytes.

Table 17.5-1 "Statuses of the Hardware Sequence Flag" lists the possible statuses of the

hardware sequence flag.

Table 17.5-1 Statuses of the Hardware Sequence Flag

Status		DPOLL	TOGGLE	TLOVER	SETIMR	TOGGL2
Executing	Automatic read operation	Reverse data	Toggle	0	0	1
	Automatic erase operation	0	Toggle	0	1	Toggle
	Temporary erase stop mode	1	1	0	0	Toggle* ¹
		Data	Data	Data	Data	Data
		Reverse data	Toggle* ²	0	0	1* ³
Time limit exceeded	Automatic write operation	Reverse data	Toggle	1	0	1
	Automatic erase operation	0	Toggle	1	1	Undefined
	Write operation during temporary erase stop	0	Toggle	1	1	Undefined

*1: Bit 2 toggles for consecutive read operations from sectors in temporary erase stop.

*2: Bit 6 toggles for consecutive read operations from any address.

*3: During temporary erase stop and write operations, bit 2 indicates "1" while reading the address for the write operation. However, bit 2 toggles for consecutive read operations from sectors in temporary erase stop.

The hardware sequence flags are explained below.

[bit 7] DPOLL (Data polling)

○ Automatic write operation status

When a read operation is performed during execution of the automatic write algorithm, flash memory outputs the inversion of the last written data. When read access is performed at the end of the automatic write algorithm, flash memory outputs the data of bit 7 of the read data in the address indicated by the address signal.

○ Automatic erase operation status

When a read operation is performed during execution of the automatic erase algorithm, flash memory outputs "0" irrespective of the address indicated by the address signal. Similarly, flash memory outputs "1" at the end of the algorithm.

○ Temporary sector erase stop status

When a read operation is performed during temporary sector erase stop, flash memory outputs "1" if the address indicated by the address signal is included in the sector in erase state. If the address is not included in the sector in erase state, flash memory outputs the data of bit 7 of the read value at the address indicated by the address signal.

For checking whether a sector is in temporary sector erase stop state and when determining which sector is in erase state, read toggle bit 6, which is described later.

Note

When the automatic algorithm approaches the end of its operation, bit 7 (data polling) asynchronously varies during a read operation, which means that flash memory outputs operation status information to bit 7 and then outputs the determined data. When flash memory terminates the automatic algorithm, or bit 7 is outputting the determined data, the data of the other bits is undefined. The data of the other bits is read during the execution of consecutive read operations.

[bit 6]: TOGGLE (Toggle bit)

○ Automatic write/erase operation status

When consecutive read operations are performed during the execution of the automatic write or erase algorithm, flash memory outputs the "1" and "0" toggle results to bit 6. When the automatic write or erase algorithm ends, bit 6 stops to toggle for a consecutive read and outputs valid data. The toggle bit becomes valid after the last write cycle of each command sequence.

If a write target sector is protected from overwriting during a write operation, the toggle bit toggles for about 2 μ s and stops to toggle without overwriting. If all selected sectors are write-protected, the toggle bit toggles for about 100 μ s and the system returns to the read mode without changing data.

○ Temporary sector erase stop status

When a read operation is performed during a temporary sector erase stop operation, flash memory outputs "1" if the address indicated by the address signal is included in the sector in erase state. If the address is not included in the sector in erase state, flash memory outputs the data of bit 6 of the read value at the address indicated by the address signal.

[bit 5] TLOVER (Time limit over)

○ Automatic write/erase operation status

Bit 5 indicates by becoming "1" that execution of the automatic algorithm has exceeded the time limit (internal pulse count) specified in flash memory. In other words, when this flag outputs "1" while the automatic algorithm is running, this indicates that a write or erase operation failed.

If an attempt is made to write to a nonblank area without erasing the data of that area, bit 5 also indicates that the attempt failed. In this case, the data of bit 7 (data polling) is undefined, and bit 6 (toggle bit) continues to toggle. If the time limit is exceeded in this status, bit 5 is set to "1".

Note that in this case, flash memory is not defective but is used incorrectly. If this state is entered, perform a Reset.

[bit 3] SETIMR (Sector erase timer)

○ Sector erase operation status

After execution of the Sector Erase command sequence, a sector erase wait period is entered. Bit 3 is "0" in this state and becomes "1" if the limit of the sector erase wait period is exceeded. The data polling and toggle bits become valid after the execution of the first Ease Sector

command sequence.

Suppose that the data polling and toggle bit functions indicate that the erase algorithm is running. If this flag is "1" in this case, an internally controlled erase operation has started and succeeding command entries are ignored until the data polling or toggle bit indicates the end of the erase operation. (Only the input of a temporary erase stop code is accepted.)

When this flag is "1", flash memory accepts another sector erase code entry. In this case, it is recommended to check the status of this flag by software before writing the succeeding sector erase code. If this flag is "1" at the second time of status check, the additional sector erase code may not be accepted.

When a read operation is performed during a temporary sector erase stop operation, flash memory outputs "1" if the address indicated by the address signal is included in the sector that is subject to the erase operation. If the address is not included in the sector that is subject to the erase operation, flash memory outputs the data of bit 3 of the read value at the address indicated by the address signal.

[bit 2] TOGGL2 (Toggle bit)

○ Sector erase operation status

Together with toggle bit 6, this toggle bit is used to indicate whether flash memory is subject to automatic erase operation or temporary erase stop operation. If data is read consecutively from a sector that is subject to erasing during an automatic erase operation, bit 2 toggles. If data is consecutively read from a sector that is subject to a temporary erase stop operation when flash memory is in temporary erase stop and read mode, bit 2 toggles also.

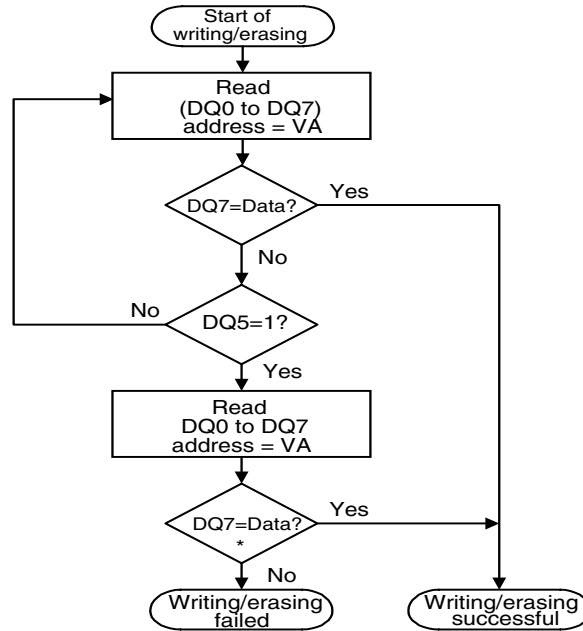
If addresses are read consecutively from a sector not subject to a temporary erase stop operation when flash memory is in temporary erase stop and write mode, bit 2 becomes "1". Unlike bit 2, bit 6 toggles only in usual write and erase or temporary erase stop and write operations.

For example, bit 2 and bit 6 are used together to detect a temporary erase stop and read mode (bit 2 toggles but bit 6 does not). Bit 2 is also used to detect sectors that are subject to erase operations. If data is read from a sector that is subject to an erase operation for the flash memory, bit 2 toggles.

■ Examples of Using Hardware Sequence Flags

The status of an automatic algorithm in the flash memory can be determined with one of the hardware sequence flags described above. As examples, Figure 17.5-2 "Flow Chart of Write/Erase Determination Using Data Polling Function" and Figure 17.5-3 "Flow Chart of Write/Erase Determination Using Toggle Bit Function" show flow charts of write/erase determination using the data polling function and the toggle bit function, respectively.

Figure 17.5-2 Flow Chart of Write/Erase Determination Using Data Polling Function



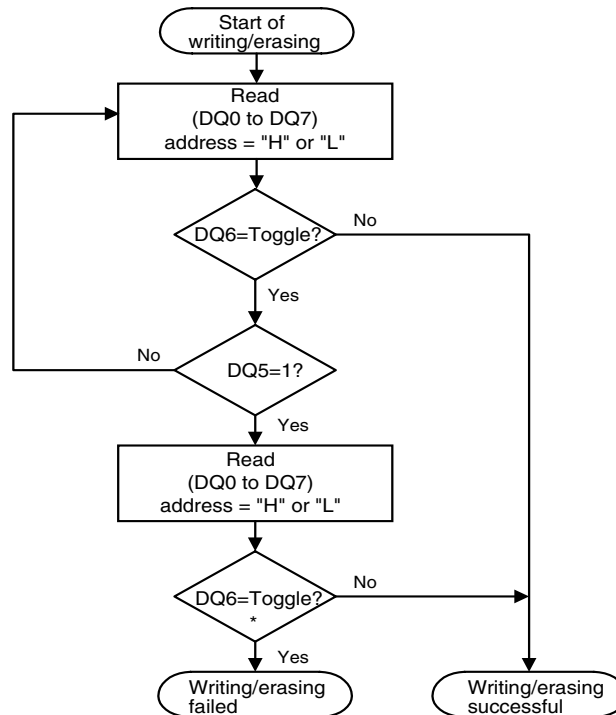
VA = Write address

= Address of sector erased during sector erase operation

= Sector address not protected during chip erase operation

*: DQ7 must be checked again even if DQ5 = "1" because DQ7 changes at the same time as DQ5.

Figure 17.5-3 Flow Chart of Write/Erase Determination Using Toggle Bit Function



*: DQ6 must be checked again even if DQ5 = "1" because DQ6 stops toggling at the same time as DQ5 changes to "1".

17.6 Sector Protect Operation

This flash memory has the sector protect function that disables any unauthorized writing or erasing for individual sectors. Once protection of a sector is set, the sector retains the setting unless the device becomes damaged. However, a sector may be unprotected temporarily in order to enable the write or erase operation in it. This can be done with the sector protect operations.

The sector protect operations do not have an automatic algorithm such as write or erase. The operations can be executed only in flash memory mode, not in normal mode. Thus, the operation must be done mainly with external pin control using a flash memory writer.

■ List of Sector Protect Operations

There are three sector protect operations:

- Enable sector protect
- Verify sector protect
- Temporary sector unprotect

Table 17.6-1 "Pin Settings" lists the pin settings for these operations.

Table 17.6-1 Pin Settings

Operation	CEX	OEX	WEX	AQ1	AQ2	AQ7	AQ13 to AQ18	DQ0 to DQ15	RSTX	MD2	MD1	MD0
Enable sector protect	L	H	L	L	H	L	Sector address	-	H	V _{ID}	H	V _{ID}
Verify sector protect	L	L	H	L	H	L	Sector address	Code output	H	H	H	V _{ID}
Temporary sector unprotect	-	-	-	-	-	-	-	-	H	H	V _{ID}	H

■ Enable Sector Protect

Enable sector protect writes to the protection circuit in the flash memory.

This operation can disable write and erase to any combination of the seven sectors. In the factory default setting of the MB91F127 and MB91F128, all sectors are unprotected.

As required by this operation, the sector address (AQ18, AQ17, AQ16, AQ15, AQ14, or AQ13) of a sector to be protected must be set in an address signal, and AQ8 must be set to "0". For the correspondence between sectors and sector addresses, see 17.1-1 "Sector Addresses (MB91F127) (Continued)".

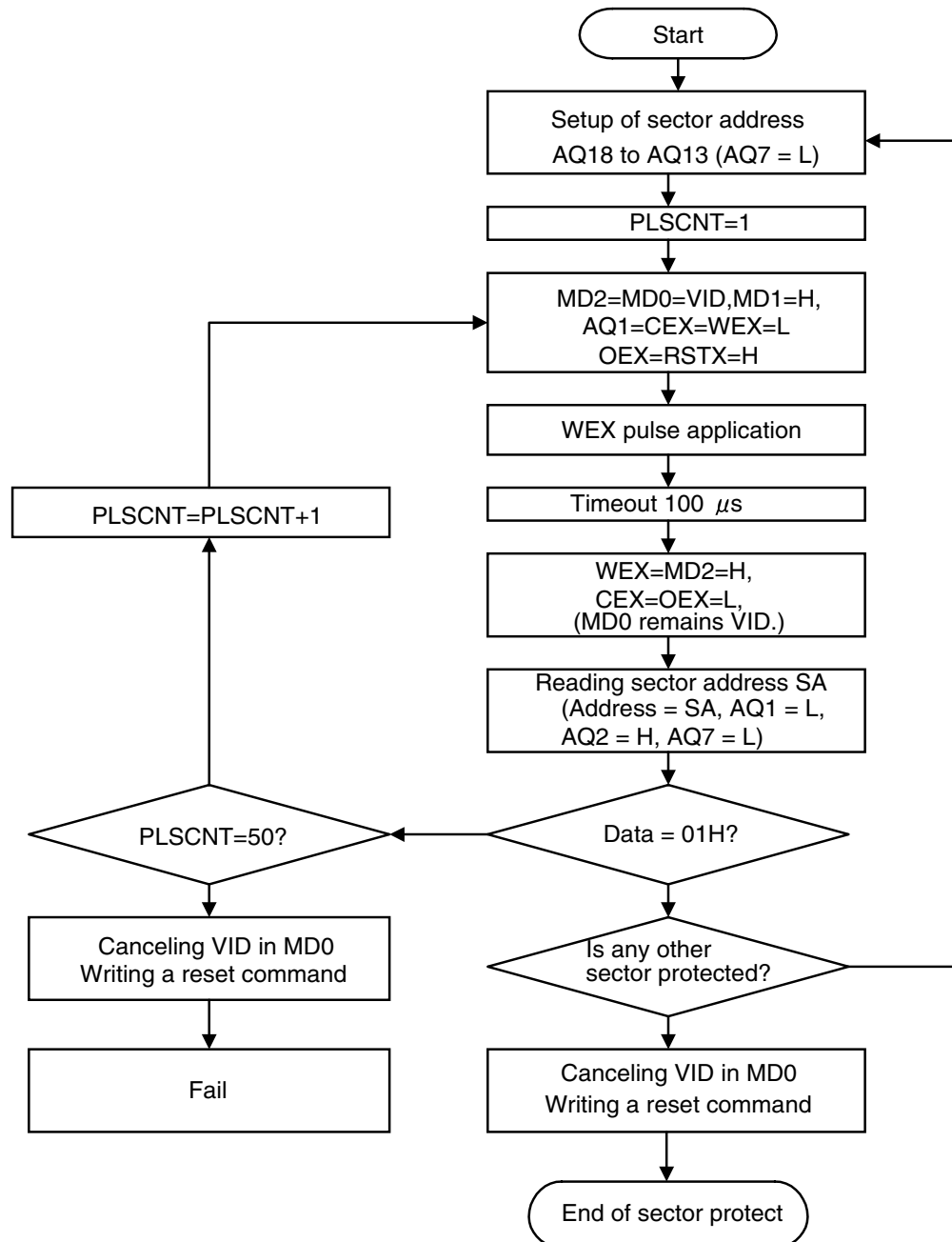
Writing to a protected circuit starts at a falling edge and ends at a rising edge of the WEX pulse after V_{ID} (= 12V) is applied to MD2 and MD0 to set CEX to "0". A sector address must remain constant throughout the WEX pulse. Once sector protect is set, it cannot be canceled. The write or erase operation to a protected sector is disabled after it is set.

■ Verify Sector Protect

Verify sector protect verifies writing to a protected circuit in the flash memory. In this operation, CEX and OEX are first set to "0", and with WEX set to "1", VID is applied to MD0 (margin mode). Under the condition of (AQ7, AQ2, AQ1) = (0, 1, 0), when an address signal set to a sector address is read, "1" is output to the output DQ0 in a protected sector. 00H is read in an unprotected sector.

Figure 17.6-1 "Sector Protect Algorithm Using Enable Sector Protect And Verify Sector Protect" shows the sector protect algorithm using enable sector protect and the verify sector protect.

Figure 17.6-1 Sector Protect Algorithm Using Enable Sector Protect And Verify Sector Protect



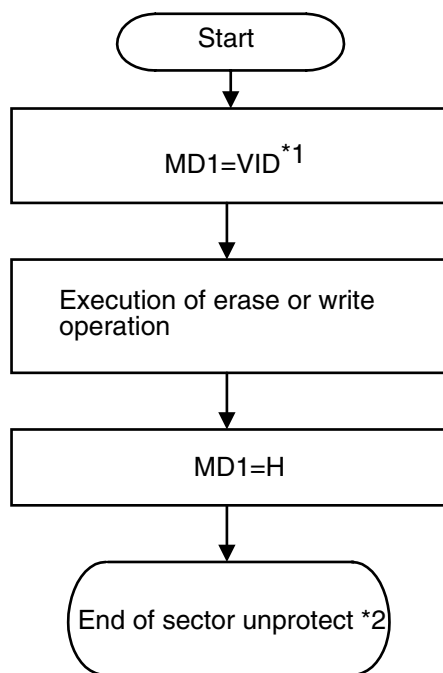
■ Temporary Sector Unprotect

Once a sector becomes protected using enable sector protect, it cannot be written to or erased unless the device becomes damaged. The temporary sector unprotect operation can temporarily cancel the current setting of sector protect. Execute this operation to set temporary sector unprotect by continuing to apply VID to MD1. During this time, the current setting that keeps the sector protect is ignored, and write or erase to all sectors is enabled.

Setting MD1 back to "1" (= 5V) cancels this operation, and all of the previously protected sectors are protected again.

Figure 17.6-2 "Algorithm of Temporary Sector Unprotect" shows the algorithm of temporary sector unprotect.

Figure 17.6-2 Algorithm of Temporary Sector Unprotect



*1: All of the protected sectors are unprotected.

*2: All of the previously protected sectors are protected again.

APPENDIX

The appendix shows the I/O map, interrupt vector, and pin status for each CPU state.

A "I/O MAP"

B "INTERRUPT VECTORS"

C "PIN STATES FOR EACH CPU STATE"

APPENDIX A I/O MAP

The addresses listed in Table A-1 "I/O Map" are allocated to each register of the peripheral devices contained in MB91F127/128.

I/O Map

[How to view the table]

Address	Register				Internal resource
	+0	+1	+2	+3	
000000H	PDR0[R/W] XXXXXXXX	PDR1[R/W] XXXXXXXX	PDR2[R/W] XXXXXXXX	PDR3[R/W] XXXXXXXX	Port data register

Read/write attribute

Register initial value after a reset

Register name (The register of column 1 has address 4n, the register of column 2 has address 4n + 1, and so on.)

Leftmost register address (At word access, the register of column 1 becomes the MSB of the data.)

Notes:

The initial bit values of the register are indicated as follows:

- "1": Initial value "1"
- "0": Initial value "0"
- "X": Initial value "X"
- "-": No physical register at the location

Table A-1 I/O Map

Address	Register				Internal resource
	+0	+1	+2	+3	
000000 _H	PDR3 [R/W] XXXXXXXX	PDR2 [R/W] XXXXXXXX	-	-	Port data register
000004 _H	PDR7 [R/W] -----X	PDR6 [R/W] XXXXXXXX	PDR5 [R/W] XXXXXXXX	PDR4 [R/W] XXXXXXXX	
000008 _H	-	PDRA [R/W] -XXXXXXXX	-	PDR8 [R/W] --XXXXXX	
00000C _H	-				
000010 _H	-	-	PDRE [R/W] XXXXXXXX	PDRF [R/W] ----XXXX	
000014 _H	PDRG [R/W] XXXXXXXX	-	-	PDRJ [R/W] XXXXXXXX	
000018 _H	-	-		-	Unused
00001C _H	SSR [R/W] 00001-00	SIDR [R] SODR [W] XXXXXXXX	SCR [R/W] 00000100	SMR [R/W] 00--0-00	UART 0
000020 _H	SSR [R/W] 00001-00	SIDR [R] SODR [W] XXXXXXXX	SCR [R/W] 00000100	SMR [R/W] 00--0-00	UART 1
000024 _H	SSR [R/W] 00001-00	SIDR [R] SODR [W] XXXXXXXX	SCR [R/W] 00000100	SMR [R/W] 00--0-00	UART 2
000028 _H	TMRLR [W] XXXXXXXX XXXXXXXX		TMR [R] XXXXXXXX XXXXXXXX		Reload Timer 0
00002C _H	-		TMCSR [R/W] ----0000 00000000		
000030 _H	TMRLR [W] XXXXXXXX XXXXXXXX		TMR [R] XXXXXXXX XXXXXXXX		Reload Timer 1
000034 _H	-		TMCSR [R/W] ----0000 00000000		
000038 _H	-		-		Unused
00003C _H	TMRLR [W] XXXXXXXX XXXXXXXX		TMR [R] XXXXXXXX XXXXXXXX		Reload Timer 2
000040 _H	-		TMCSR [R/W] ----0000 00000000		

APPENDIX

Table A-1 I/O Map (Continued)

Address	Register				Internal resource
	+0	+1	+2	+3	
000044 _H	IPCP1 [R] XXXXXXXX XXXXXXXX		IPCP0 [R] XXXXXXXX XXXXXXXX		Multifunctional timer
000048 _H	IPCP3 [R] XXXXXXXX XXXXXXXX		IPCP2 [R] XXXXXXXX XXXXXXXX		
00004C _H	-	ICS23 [R/W] 00000000	-	ICS01 [R/W] 00000000	
000050 _H	ADCR [R] 000000XX XXXXXXXX		ADCS [R/W] 00000000 00000000		A/D converter (sequential comparison)
000054 _H	OCCP1 [R/W] XXXXXXXX XXXXXXXX		OCCP0 [R/W] XXXXXXXX XXXXXXXX		Multifunctional timer
000058 _H	OCCP3 [R/W] XXXXXXXX XXXXXXXX		OCCP2 [R/W] XXXXXXXX XXXXXXXX		
00005C _H	-		-		Unused
000060 _H	-		-		
000064 _H	OCS2, 3 [R/W] XXX00000 0000XX00		OCS0, 1 [R/W] XXX00000 0000XX00		Multifunctional timer
000068 _H	-		-		Unused
00006C _H	TCDT [R/W] 00000000 00000000		TCCS [R/W] 0----- 00000000		Multifunctional timer
000070 _H	-		-		Unused
000074 _H	-		-		Unused
000078 _H	UTIM [R] UTIMR [W] 00000000 00000000		-	UTIMC [R/W] 0--00001	U-TIMER 0
00007C _H	UTIM [R] UTIMR [W] 00000000 00000000		-	UTIMC [R/W] 0--00001	U-TIMER 1
000080 _H	UTIM [R] UTIMR [W] 00000000 00000000		-	UTIMC [R/W] 0--00001	U-TIMER 2
000084 _H	-		-		Unused
000088 _H	-		-		
00008C _H	-		-		Unused
000090 _H	-		-		

Table A-1 I/O Map (Continued)

Address	Register				Internal resource
	+0	+1	+2	+3	
000094 _H	EIRR [R/W] 00000000	ENIR [R/W] 00000000	-		External interrupt
000098 _H	EHVR [R/W] 00000000	ELVR [R/W] 00000000	-		
00009C _H	-				Unused
0000A0 _H	-				
0000A4 _H	-				
000048 _H	-				
0000AC _H	-				
0000B0 _H	-				
0000B4 _H	-				
0000B8 _H	-				
0000BC _H	-				
0000C0 _H	-				
0000C4 _H	-				
0000C8 _H	-				
0000CC _H	-				
0000D0 _H	-	-	DDRE [W] 00000000	DDRF [W] 00000000	Port direction register
0000D4 _H	-	AIC3 [R/W] 11111111	-	-	A/D converter
0000D8 _H	DDRG [W] 00000000	-	-	DDRJ [W] 00000000	Port direction register
0000DC _H	GCN1 [R/W] 00110010 00010000		-	GCN2 [R/W] 00000000	PPG
0000E0 _H	PTMR0 [R] 11111111 11111111		PCSR0 [W] XXXXXXXX XXXXXXXX		PPG0
0000E4 _H	PDUT0 [W] XXXXXXXX XXXXXXXX		PCNH0 [R/W] 0000000-	PCNL0 [R/W] 00000000	
0000E8 _H	PTMR1 [R] 11111111 11111111		PCSR1 [W] XXXXXXXX XXXXXXXX		PPG1
0000EC _H	PDUT1 [W] XXXXXXXX XXXXXXXX		PCNH1 [R/W] 0000000-	PCNL1 [R/W] 00000000	

APPENDIX

Table A-1 I/O Map (Continued)

Address	Register				Internal resource
	+0	+1	+2	+3	
0000F0 _H	PTMR2 [R] 11111111 11111111		PCSR2 [W] XXXXXXXXXX XXXXXXXXXX		PPG2
0000F4 _H	PDUT2 [W] XXXXXXXXXX XXXXXXXXXX		PCNH2 [R/W] 0000000-	PCNL2 [R/W] 00000000	
0000F8 _H	PTMR3 [R] 11111111 11111111		PCSR3 [W] XXXXXXXXXX XXXXXXXXXX		PPG3
0000FC _H	PDUT3 [W] XXXXXXXXXX XXXXXXXXXX		PCNH3 [R/W] 0000000-	PCNL3 [R/W] 00000000	
000100 _H to 0001FC _H	-				Unused
000200 _H	DPDP [R/W] ----- ----- 00000000				DMA controller
000204 _H	DACSR [R/W] 00000000 00000000 00000000 00000000				
000208 _H	DATCR [R/W] ----- --XX0000 --XX0000 --XX0000				
00020C _H	-				
000210 _H to 0002FC _H	-				Unused
000300 _H to 0003EC _H	-				Unused
0003F0 _H	BSD0 [W] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				Bit search module
0003F4 _H	BSD1 [R/W] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
0003F8 _H	BSDC [W] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				
0003FC _H	BSRR [R] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				

Table A-1 I/O Map (Continued)

Address	Register				Internal resource
	+0	+1	+2	+3	
000400 _H	ICR00 [R/W] ---11111	ICR01 [R/W] ---11111	ICR02 [R/W] ---11111	ICR03 [R/W] ---11111	Interrupt controller
000404 _H	ICR04 [R/W] ---11111	ICR05 [R/W] ---11111	ICR06 [R/W] ---11111	ICR07 [R/W] ---11111	
000408 _H	ICR08 [R/W] ---11111	ICR09 [R/W] ---11111	ICR10 [R/W] ---11111	ICR11 [R/W] ---11111	
00040C _H	ICR12 [R/W] ---11111	ICR13 [R/W] ---11111	ICR14 [R/W] ---11111	ICR15 [R/W] ---11111	
000410 _H	ICR16 [R/W] ---11111	ICR17 [R/W] ---11111	ICR18 [R/W] ---11111	ICR19 [R/W] ---11111	
000414 _H	ICR20 [R/W] ---11111	ICR21 [R/W] ---11111	ICR22 [R/W] ---11111	ICR23 [R/W] ---11111	
000418 _H	ICR24 [R/W] ---11111	ICR25 [R/W] ---11111	ICR26 [R/W] ---11111	ICR27 [R/W] ---11111	
00041C _H	ICR28 [R/W] ---11111	ICR29 [R/W] ---11111	ICR30 [R/W] ---11111	ICR31 [R/W] ---11111	
000420 _H	ICR32 [R/W] ---11111	ICR33 [R/W] ---11111	ICR34 [R/W] ---11111	ICR35 [R/W] ---11111	
000424 _H	ICR36 [R/W] ---11111	ICR37 [R/W] ---11111	ICR38 [R/W] ---11111	ICR39 [R/W] ---11111	
000428 _H	ICR40 [R/W] ---11111	ICR41 [R/W] ---11111	ICR42 [R/W] ---11111	ICR43 [R/W] ---11111	
00042C _H	ICR44 [R/W] ---11111	ICR45 [R/W] ---11111	ICR46 [R/W] ---11111	ICR47 [R/W] ---11111	Interrupt controller
000430 _H	DICR [R/W] -----0	HRCL [R/W] ---11111	-	-	Delayed interrupt
000434 _H to 00047C _H	-				Unused
000480 _H	RSRR [R] WTCR [W] 1XXXX-00	STCR [R/W] 000111--	PDRR [R/W] ----0000	CTBR [W] XXXXXXXXX	Clock control section
000484 _H	GCR [R/W] 110011-1	WPR [W] XXXXXXXXX	-	-	
000488 _H	PTCR [R/W] 00--0---	-			PLL control
00048C _H to 0005FC _H	-				Unused

APPENDIX

Table A-1 I/O Map (Continued)

Address	Register				Internal resource
	+0	+1	+2	+3	
000600 _H	DDR3 [W] 00000000	DDR2 [W] 00000000	-	-	Data orientation register
000604 _H	DDR7 [W] -----0	DDR6 [W] 00000000	DDR5 [W] 00000000	DDR46 [W] 00000000	
000608 _H	-	DDRA [W] -000000-	-	DDR8 [W] --0--000	
00060C _H	ASR1 [W] 00000000 00000001		AMR1 [W] 00000000 00000001		External bus interface
000610 _H	ASR2 [W] 00000000 00000010		AMR2 [W] 00000000 00000001		
000614 _H	ASR3 [W] 00000000 00000011		AMR3 [W] 00000000 00000001		
000618 _H	ASR4 [W] 00000000 00000100		AMR4 [W] 00000000 00000001		
00061C _H	ASR5 [W] 00000000 00000101		AMR5 [W] 00000000 00000001		
000620 _H	AMD0 [R/W] ---00111	AMD1 [R/W] 0--00000	AMD32 [R/W] 00000000	AMD4 [R/W] 0--00000	
000624 _H	AMD5 [R/W] 0--00000	DSCR [W] 00000000	RFCR [R/W] --XXXXXX 00---000		
000628 _H	EPCR0 [W] ----1100 -1111111		EPCR1 [W] -----1 11111111		
00062C _H	DMCR4 [R/W] 00000000 0000000-		DMCR5 [R/W] 00000000 0000000-		
000630 _H to 0007BC _H	-				Unused
0007C0 _H	FSTR [R/W] 000XXXX0	-	-	-	Flash memory
0007C4 _H	FWTC [R/W] XXXXX000	-	-	-	
0007C8 _H to 0007F8 _H	-				Unused
0007FC _H	-		LER [W] -----000	MODR [W] XXXXXXXXX	Little endian register mode register

Notes:

Do not execute any RMW instruction on a register with a write-only bit.

RMW instruction (RMW: Read-modify-write)

AND Rj, @Ri	OR Rj, @Ri	E OR Rj, @Ri
ANDH Rj, @Ri	OR H Rj, @Ri	E OR H Rj, @Ri
ANDB Rj, @Ri	OR B Rj, @Ri	E OR B Rj, @Ri
BANDL #u4, @Ri	B OR L #u4, @Ri	BE OR L #u4, @Ri
BANDH #u4, @Ri	B OR H #u4, @Ri	BE OR H #u4, @Ri

Data in areas with "Unused" or "-" is undefined.

APPENDIX B INTERRUPT VECTORS

Table B-1 "Interrupt Vector Table" shows an interrupt vector table. The interrupt vector table lists the interrupt sources and interrupt vector/interrupt control register allocations of MB91F127/128.

■ Interrupt vectors

Table B-1 Interrupt Vector Table

Interrupt source	Interrupt number		Interrupt level* ¹	Offset	TBR default address * ²
	Decimal	Hexa-decimal			
Reset	0	00	-	3FC _H	000FFFFC _H
System reserved	1	01	-	3F8 _H	000FFFF8 _H
System reserved	2	02	-	3F4 _H	000FFFF4 _H
System reserved	3	03	-	3F0 _H	000FFFF0 _H
System reserved	4	04	-	3EC _H	000FFFE _C
System reserved	5	05	-	3E8 _H	000FFFE8 _H
System reserved	6	06	-	3E4 _H	000FFFE4 _H
System reserved	7	07	-	3E0 _H	000FFFE0 _H
System reserved	8	08	-	3DC _H	000FFFD _C
System reserved	9	09	-	3D8 _H	000FFFD8 _H
System reserved	10	0A	-	3D4 _H	000FFFD4 _H
System reserved	11	0B	-	3D0 _H	000FFFD0 _H
System reserved	12	0C	-	3CC _H	000FFF _{CC}
System reserved	13	0D	-	3C8 _H	000FFF _{C8}
Undefined instruction exception	14	0E	-	3C4 _H	000FFF _{C4}
NMI request	15	0F	F _H only	3C0 _H	000FFF _{C0}
External interrupt 0	16	10	ICR00	3BC _H	000FFF _{BC}
External interrupt 1	17	11	ICR01	3B8 _H	000FFF _{B8}
External interrupt 2	18	12	ICR02	3B4 _H	000FFF _{B4}
External interrupt 3	19	13	ICR03	3B0 _H	000FFF _{B0}
UART 0 receive completed	20	14	ICR04	3AC _H	000FFF _{AC}
UART 1 receive completed	21	15	ICR05	3A8 _H	000FFF _{A8}

Table B-1 Interrupt Vector Table

Interrupt source	Interrupt number		Interrupt level* ¹	Offset	TBR default address * ²
	Decimal	Hexa-decimal			
UART 2 receive completed	22	16	ICR06	3A4 _H	000FFFA4 _H
UART 0 send completed	23	17	ICR07	3A0 _H	000FFFA0 _H
UART 1 send completed	24	18	ICR08	39C _H	000FFF9C _H
UART 2 send completed	25	19	ICR09	398 _H	000FFF98 _H
DMAC 0 (end, error)	26	1A	ICR10	394 _H	000FFF94 _H
DMAC 1 (end, error)	27	1B	ICR11	390 _H	000FFF90 _H
DMAC 2 (end, error)	28	1C	ICR12	38C _H	000FFF8C _H
DMAC 3 (end, error)	29	1D	ICR13	388 _H	000FFF88 _H
DMAC 4 (end, error)	30	1E	ICR14	384 _H	000FFF84 _H
DMAC 5 (end, error)	31	1F	ICR15	380 _H	000FFF80 _H
DMAC 6 (end, error)	32	20	ICR16	37C _H	000FFF7C _H
DMAC 7 (end, error)	33	21	ICR17	378 _H	000FFF78 _H
A/D (successive type)	34	22	ICR18	374 _H	000FFF74 _H
Reload timer 0	35	23	ICR19	370 _H	000FFF70 _H
Reload timer 1	36	24	ICR20	36C _H	000FFF6C _H
Reload timer 2	37	25	ICR21	368 _H	000FFF68 _H
External interrupt 4	38	26	ICR22	364 _H	000FFF64 _H
External interrupt 5	39	27	ICR23	360 _H	000FFF60 _H
System reserved	40	28	ICR24	35C _H	000FFF5C _H
System reserved	41	29	ICR25	358 _H	000FFF58 _H
U-TIMER 0	42	2A	ICR26	354 _H	000FFF54 _H
U-TIMER 1	43	2B	ICR27	350 _H	000FFF50 _H
U-TIMER 2	44	2C	ICR28	34C _H	000FFF4C _H
Flash memory	45	2D	ICR29	348 _H	000FFF48 _H
System reserved	46	2E	ICR30	344 _H	000FFF44 _H
System reserved	47	2F	ICR31	340 _H	000FFF40 _H
PPG0	48	30	ICR32	33C _H	000FFF3C _H
PPG1	49	31	ICR33	338 _H	000FFF38 _H
PPG2	50	32	ICR34	334 _H	000FFF34 _H
PPG3	51	33	ICR35	330 _H	000FFF30 _H

APPENDIX

Table B-1 Interrupt Vector Table

Interrupt source	Interrupt number		Interrupt level* ¹	Offset	TBR default address * ²
	Decimal	Hexa-decimal			
ICU0 (Capture)	52	34	ICR36	32C _H	000FFF2C _H
ICU1 (Capture)	53	35	ICR37	328 _H	000FFF28 _H
ICU2 (Capture)	54	36	ICR38	324 _H	000FFF24 _H
ICU3 (Capture)	55	37	ICR39	320 _H	000FFF20 _H
OCU0 (Matching)	56	38	ICR40	31C _H	000FFF1C _H
OCU1 (Matching)	57	39	ICR41	318 _H	000FFF18 _H
OCU2 (Matching)	58	3A	ICR42	314 _H	000FFF14 _H
OCU3 (Matching)	59	3B	ICR43	310 _H	000FFF10 _H
System reserved	60	3C	ICR44	30C _H	000FFF0C _H
16-bit free-running timer	61	3D	ICR45	308 _H	000FFF08 _H
System reserved	62	3E	ICR46	304 _H	000FFF04 _H
Delayed interrupt source bit	63	3F	ICR47	300 _H	000FFF00 _H
System reserved (used by REALOS)* ³	64	40	-	2FC _H	000FFEFC _H
System reserved (used by REALOS)* ³	65	41	-	2F8 _H	000FFE8 _H
Used by INT instruction	66 to 255	42 to FF	- to -	2F4 _H to 000 _H	000FEF4 _H to 000FFC00 _H

*1: ICR is a register set up within the interrupt controller to set the interrupt level for each interrupt request. ICR is provided for each interrupt request.

*2: TBR is a register that points to the start address of the EIT vector table. The vector address is obtained by adding the offset value determined for each EIT source to TBR.

*3: If REALOS/FR is used, use interrupts of 0x40 and 0x41 for the system code.

Reference

The 1-KB area from the address indicated by TBR is the vector area for EIT.

Each vector is 4 bytes in size. The relationship between the vector number and vector address can be represented as follows:

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

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

APPENDIX C PIN STATES FOR EACH CPU STATE

Table C-1 "Pin Status in External Bus 16-bit Mode " to Table C-3 "Pin Status in External Bus 16-bit Mode " list the pin status for each CPU state.

Note that the pin status for reset in external bus mode is different from that in single chip mode.

■ Explanation of terms in the pin status list

Terms used to indicate pin status in Table C-1 "Pin Status in External Bus 16-bit Mode" to Table C-3 "Pin Status in External Bus 16-bit Mode" have the following meanings:

- Input ready

The input function can be used.

- Input 0 fixed

State in which, at an input gate just beside the pin, "0" is transmitted inside by cutting off external input.

- Output Hi-Z

The pin is put into high impedance by disabling the transistor driving the pin.

- Hold output

The status output just before this mode is entered is output. That is, if a built-in peripheral device with output is operating, the output is conducted according to the peripheral device. If it is operating as an output port, the output is held.

- Hold status just before

The status output just before this mode is entered is output or is ready for input.

■ Pin status list

Table C-1 Pin Status in External Bus 16-bit Mode

Pin name	Function	Sleeping	Stopped		Reset
			HIZX=0	HIZX=1	
P20 to P27	Port	Hold status just before	Hold status just before	Output Hi-Z Input 0 fixed	Output Hi-Z All pins input ready
P30 to P37					
P40 to P47					
P50 to P57					
P60 to P65					
P66	IN2				
P67	IN3				
P70	FRCK				
	TCI2				
P80	Port				
P81	IN0				
P82	IN1				
P83 to P86	Port				
PA0 to PA2					
PA3	SO1				
PA4	SI1				
PA5	SC1				
PA6	Port				
PG0	OCPA0				
PG1	OCPA1				
PG2	OCPA2				
PG3	OCPA3				
PG4	OC0				
PG5	OC1				
PG6	OC2				
PG7	OC3				

Table C-1 Pin Status in External Bus 16-bit Mode (Continued)

Pin name	Function	Sleeping	Stopped		Reset
			HIZX=0	HIZX=1	
PE0	INT0	Hold status just before	Input ready	Input ready	Output Hi-Z All pins input ready
PE1	INT1				
PE2	INT2				
PE3	INT3				
PE4	INT4		Hold status just before	Output Hi-Z Input 0 fixed	
	TCI1				
PE5	INT5				
	SC0				
PE6	SI0				
PE7	SO0				
PF0	TCI0				
PF1	SI2				
PF2	SO2				
PF3	SC2				
	$\overline{\text{ATG}}$				
PJ0 to PJ7	AN0 to AN7				

Table C-2 Pin Status in External Bus 16-bit Mode

Pin name	Function	Sleeping	Stopped		Bus open (BGRNT)	Reset	
			HIZX=0	HIZX=1			
P20 to P27	D16 to D23	Hold output or Hi-Z	Same as at left	Output Hi-Z Input 0 fixed	Output Hi-Z	Output Hi-Z All pins input ready	
---	D24 to D31						
---	A00 to A15						Hold output (address output)
P60 to P67	A16 to A23	P: Hold status just before F: Address output	Same as at left				
P70	A24	Hold status just before	Same as at left				
P80	RDY	P: Hold status just before F: RDY Input	P, F: Hold status just before		P: Hold status just before F: Input ready	Output Hi-Z All pins input ready	
P81	$\overline{\text{BGRNT}}$	P: Hold status just before F: H output	P, F: Hold status just before		L output		
P82	BRQ	P: Hold status just before F: BRQ output	P, F: Hold status just before		BGQ input		
---	$\overline{\text{RD}}$	Hold output	Same as at left		Output Hi-Z		H output
---	$\overline{\text{WR0}}$						
P85	$\overline{\text{WR1}}$	P: Hold status just before F: H output	P, F: Hold status just before	H output			
P86	ALE	P: Hold status just before F: L output	P, F: Hold status just before	L output			
PA0	$\overline{\text{CS0}}$	Hold status just before	H output	L output			
PA1	$\overline{\text{CS1}}$	P: Hold status just before F: CS output	P: Same as at left F: H output	H output			
PA2	$\overline{\text{CS2}}$						
PA3	$\overline{\text{CS3}}$						
PA4	$\overline{\text{CS4}}$						

Table C-2 Pin Status in External Bus 16-bit Mode (Continued)

Pin name	Function	Sleeping	Stopped		Bus open (BGRNT)	Reset	
			HIZX=0	HIZX=1			
PA5	CS5	P: Hold status just before F: CS output	P:Same as at left F:H outout	Output Hi-Z Input 0 fixed	Output Hi-Z	H output	
PA6	CLK	P: Hold status just before F: Clock output	P, F: Hold status just before		Clock output	Clock output	
PG0	OCPA0	Hold status just before	Hold status just before		Hold status just before	Output Hi-Z All pins input ready	
PG1	OCPA1						
PG2	OCPA2						
PG3	OCPA3						
PG4	OC0						
PG5	OC1						
PG6	OC2						
PG7	OC3						
PE0	INT0			Input ready			Input ready
PE1	INT1						
PE2	INT2						
PE3	INT3						
PE4	INT4		Hold status just before	Output Hi-Z Input 0 fixed			
	TCI1						
PE5	INT5						
	SC0						
PE6	SI0						
PE7	SO0						
PF0	TCI0						
PF1	SI2						
PF2	SO2						
PF3	SC2						
	ATG						
PJ0 to PJ7	AN0 to AN7						

P: If general-purpose port is specified

F: If specified function is selected

Table C-3 Pin Status in External Bus 8-bit Mode

Pin name	Function	Sleeping	Stopped		Bus open (BGRNT)	Reset	
			HIZX=0	HIZX=1			
P20 to P27	Port	Hold status just before	Same as at left	Output Hi-Z Input 0 fixed	Hold status just before	Output Hi-Z All pins input ready	
---	D24 to D31	Hold output or Hi-Z	Same as at left		Output Hi-Z		
---	A00 to A15	Hold output (address output)	Same as at left				FF _H output
P60 to P67	A16 to A23	P: Hold status just before F: Address output	Same as at left				
P70	A24	Hold status just before	Same as at left				
P80	RDY	P: Hold status just before F: Input ready	P, F: Hold status just before		P: Hold status just before F: RDY input	Output Hi-Z All pins input ready	
P81	$\overline{\text{BGRNT}}$	P: Hold status just before H output	P, F: Hold status just before		L output		
P82	BRQ	P: Hold status just before BRQ output	P, F: Hold status just before		BRQ input		
---	$\overline{\text{RD}}$	Hold status just before	Same as at left		Output Hi-Z	H output	
---	$\overline{\text{WR0}}$						
P85	Port	Hold status just before	Same as at left		Hold status just before	Output Hi-Z All pins input ready	
P86	ALE	P: Hold status just before F: L output	P, F: Hold status just before		Output Hi-Z	L output	
PA0	$\overline{\text{CS0}}$	Hold status just before	H output			L output	
PA1	$\overline{\text{CS1}}$	P: Hold status just before F: CS output	P: Same as at left F: H output				H output
PA2	$\overline{\text{CS2}}$						
PA3	$\overline{\text{CS3}}$						
PA4	$\overline{\text{CS4}}$						
PA5	$\overline{\text{CS5}}$						

Table C-3 Pin Status in External Bus 8-bit Mode (Continued)

Pin name	Function	Sleeping	Stopped		Bus open (BGRNT)	Reset
			HIZX=0	HIZX=1		
PA6	CLK	P: Hold status just before F: Clock output	P, F: Hold status just before	Output Hi-Z Input 0 fixed	Clock output	Clock output
PG0	OCPA0	Hold status just before	Hold status just before		Hold status just before	Output Hi-Z All pins input ready
PG1	OCPA1					
PG2	OCPA2					
PG3	OCPA3					
PG4	OC0					
PG5	OC1					
PG6	OC2					
PG7	OC3					
PE0	INT0		Input ready	Input ready		
PE1	INT1					
PE2	INT2					
PE3	INT3					
PE4	INT4	Hold status just before	Output Hi-Z Input 0 fixed			
	TCI1					
PE5	INT5					
	SC0					
PE6	SI0					
PE7	SO0					
PF0	TCI0					
PF1	SI2					
PF2	SO2					
PF3	SC2					
	ATG					
PJ0 to PJ7	AN0 to AN7					

P: If general-purpose port is specified

F: If specified function is selected

INDEX

**The index follows on the next page.
This is listed in alphabetic order.**

Index

Numerics

16-bit free-running timer, clear timing of.....	219
16-bit free-running timer, count timing of.....	219
16-bit input capture unit, operation of.....	223
16-bit input capture unit, timing of.....	224
16-bit output compare unit, operation explanation of	220
16-bit output compare unit, timing of.....	221
16-bit reload timer register	169
16-bit reload timer, activation using.....	201
16-bit reload timer, outline of.....	168

A

A/D converter (successive approximation conversion type)	3
A/D converter feature	262
A/D converter register configuration.....	264
A/D converter, note on using.....	278
A/D converter, other note on using	278
ADCR conversion	271
ADCS configuration.....	265
ADCS, bit details of	265
address at which program that can enter stop or sleep mode must be placed	88
AMD0 configuration.....	109
AMD0, bit function of	109
AMD1 configuration.....	111
AMD1, bit functions of	111
AMD32 configuration.....	113
AMD32, bit functions of	113
AMD4 configuration.....	114
AMD4, bit functions of	114
AMD5 configuration.....	115
AMD5, bit function of	115
analog input control register (AIC), configuration of	273
area mask register (AME) and area selection register (ASR)	105
area selection register (ASR) and area mask register (AME).....	105
automatic algorithm, execution status of	342
automatic erase operation status	360
automatic wait cycle timing.....	150
automatic write operation status	360
automatic write/erase operation status.....	361

B

basic read cycle timing.....	141
basic write cycle timing	143
baud rate and U-TIMER reload value, sample setting for	301
baud rate calculation.....	230
big endian bus access	122
bit ordering.....	35
bit search module.....	4
bit search module register.....	335
block diagram..... 168, 226, 247, 263, 281, 334	
branch instruction with delay slot.....	41
branch instruction with delay slot, CPU operation for	41
branch instruction with delay slot, restriction on	43
branch instruction without delay slot.....	44
branch instruction without delay slot, CPU operation for	44
burst transfer mode.....	318
bus authority acquisition	154
bus authority release.....	154
bus interface	2, 101
bus interface block diagram.....	103
bus interface register	104
bus size specification	102
byte access.....	137
byte ordering.....	35

C

cascade mode.....	231
Causes of Reset Delays other than Programs.....	75
chip select area.....	100
clock control.....	4
clock doubler function on or off, note on turning	83
clock doubler function starting	82
clock doubler function stopping.....	82
clock frequency selection method.....	155
clock generator and controller, register for	60
clock generator, block diagram of.....	61
clock system diagram for reference	85
combination of operation frequency when clock doubler function is on or off	83
command operation	355
compare register	207

compare register (OCCP0 to OCCP3), configuration of	211	DMA controller (DMAC)	3
comparison of external access operation of big endian and little endian type	122	DMA controller feature	304
configuration of flash memory wait register (FWTC)	351	DMA controller internal register, transfer operation to	331
continuous conversion mode	274	DMA controller register, list of	306
continuous transfer	331	DMA controller, block diagram of	305
continuous transfer mode	318	DMA suppression circuit, block diagram of	80
control register	258	DMA suppression setting	80
control status register (PCNH and PCNL), register configuration of	184	DMA transfer operation in sleep mode	331
conversion data protection function	276	DMA transfer request sources, list of	304
coprocessor absent trap	57	DMA transfer request, using resource interrupt request as	330
coprocessor error trap	58	DPDP configuration	307
counter operation state	178		
counter operation states	178	E	
CPU operation for branch instruction with delay slot	41	EIRR configuration	237
CPU operation for branch instruction without delay slot	44	EIT (exception, interrupt, trap)	45
CTBR configuration	67	EIT cause	45
		EIT cause acceptance priority	53
D		EIT operation	55
DACSR configuration	308	EIT vector table	51
data access	36	EIT, note on	46
data bus width	125, 131	EIT, return from	45
data bus width and control signal, relationship between	122, 123	enable sector protect	364
data direction register (DDR)	162	ENIR configuration	236
data format	124, 130	EPCR0 configuration	116
data register (TCDT), configuration of	207	EPCR0, bit function of	116
data register for detection of bit change (BSDC)	336	EPCR1 configuration	120
data register for detection of one (BSD1)	336	EPCR1, bit function of	120
data register for detection of zero (BSD0)	335	erase chip	357
data transfer block, 16-bit/8-bit data	323	execution status of automatic algorithm	342
DATCR configuration	311	external access operation of big endian and little endian type, comparison of	122
DATCR, bit function of	311	external bus access	126
dedicated register	26	external bus operation, program example of	156
delayed interrupt module register	243	external bus operation, program specification for	156
delayed interrupt module, block diagram of	242	external bus request	140
descriptor access block	321	external clock, precaution on using	17
detection of bit change	338	external device connection, example for	132
detection of one	337	external event count	177
detection of zero	337	external interrupt and NMI controller, block diagram of	234
detection result register (BSRR)	336	external interrupt controller register	235
DICR configuration	243	external interrupt request level	240
DICR DLYI bit	244	external interrupt request level setting register (ELVR)	238
direct addressing area	20	external interrupt, operating procedure for	239
		external interrupt, operation of	239

INDEX

external pin, selecting function (I/O pins or control pin) of	163
external reset input.....	17
external wait cycle timing	151

F

feature	23, 45, 100
flash memory.....	4
flash memory control signal to MBM29LV400TC, correspondence of	353
flash memory mode.....	353
flash memory sector configuration (MB91F127) ..	344
flash memory sector configuration (MB91F128) ..	346
flash memory status register (FSTR)	349
flash memory, block diagram of	343
flash memory, outline of	342
FPT-100P-M05, package dimension.....	6
FPT-120P-M21, package dimensions of.....	6
FR series, common memory map for.....	38
FR-CPU	2
FR-CPU programming mode (16 bits, read/write)	352
FR-CPU ROM mode (32 bits, read only).....	352

G

GCR configuration.....	68
gear control unit, block diagram of	76
gear function setting.....	77
general control register 1 (GCN1), bit configuration of	191
general control register 1 (GCN1), bit detail of.....	191
general-purpose register	34

H

halfword access	135
hardware configuration.....	258
hardware sequence flag.....	359
hold request cancellation request (HRLC)	256
hold request cancellation request Level setting register (HRCL)	252
hold request cancellation request sequence.....	258

I

I flag	48
I/O circuit type	13
I/O map	368
I/O port register	160
I/O port, basic block diagram of	160
ICR configuration	49, 250

ICR mapping.....	49
initializing CPU by reset.....	59
input capture control register (ICS01 and ICS23), configuration of	215
input capture data register (IPCP0 to IPCP3), configuration of	215
input pin function.....	176
instruction, overview of	39
INT instruction operation.....	56
INTE instruction operation	56
internal architecture	23
internal clock operation.....	175
internal memory, operation of external transfer of	331
internal RAM: 14 KB	3
interrupt.....	199
interrupt and flag	296
interrupt control	343
interrupt controller	3
interrupt controller register	248
interrupt controller, hardware configuration of	246
interrupt level	47
interrupt level mask register (ILM)	48
interrupt number.....	244
interrupt source, clearing	256
interrupt stack	50
interrupt vector.....	376
interrupt, level mask for.....	48

L

leading word of descriptor, configuration of	314
LER configuration	121
LER, bit function of.....	121
level mask for interrupt or NMI.....	48
list of flash memory register.....	348
little endian bus access.....	122
little endian order, outline of.....	130
low-power consumption mode	4

M

major function	246
MB91107/MB91108 block diagram.....	5
MB91F127/128 (FPT-100P-M05), pin assignment of	7
MB91F127/128 block diagram.....	5
MB91F127/128 memory map	37
memory map.....	20
mode data	97
mode pin	96

mode pin (MD0 to MD2).....	16
mode register (MODR).....	97
MODR register, note on writing data to.....	97
multifunctional timer unit, block diagram of.....	205
multifunctional timer unit, register of.....	206
multifunctional timer, configuration of.....	204
multiplication and division result register (MDH and MDL).....	28

N

NC pin, handling of.....	16
normal bus access.....	140
Notes on during operation of PLL clock mode.....	17

O

one conversion mode.....	274
one-shot mode.....	197
operation frequency when clock doubler function is on or off, combination of.....	83
operation mode.....	96
operation of 16-bit free-running timer, explanation of	218
operation of multifunctional timer unit, explanation of	217
order of priority between channels.....	330
oscillation input after power-on.....	18
other.....	4
other interval timer.....	3
output control register (Ocs0 to Ocs3), configuration of	211

P

package dimension of FPT-100P-M05.....	6
PCNH and PCNL, bit function of.....	184
PCTR configuration.....	72
PDRR configuration.....	66
pin assignment of MB91F127/128 (FPT-100P-M05)	7
Pin Assignment of the MB91107/MB91108 (FPT-120P-M21).....	7
pin description.....	8
pin status after power-on.....	18
pin status list.....	380
PLL clock setting, example of.....	84
port data register (PDR).....	161
power supply pin (VCC, VSS), connection of.....	16
power-on.....	18
power-on reset, initialization at.....	18

PPG cycle setting register (PCSR), bit configuration of.....	188
PPG duty setting register, bit configuration of (PDUT)	189
PPG mode.....	195
PPG output all-L and all-H.....	200
PPG timer by software, activation of multiple channel of.....	201
PPG timer register (PTMR), bit configuration of...190	
PPG timer, block diagram for one channel of.....	182
PPG timer, characteristic of.....	180
PPG timer, overall block diagram of.....	181
PPG timer, register of.....	183
prevent latchup, effective way to.....	16
priority between channels, order of.....	330
priority decision.....	253
program (read).....	356
program access.....	36
program counter (PC).....	26
program status (PS).....	27
program status register (PS).....	30

Q

quartz oscillation circuit.....	16
---------------------------------	----

R

read cycle timing.....	145
read/reset command.....	356
read/write mixed cycle timing.....	149
ready/busy signal (RDY/BUSYX).....	359
receiver interrupt flag in mode 0, timing for setting	296
receiver interrupt flag in mode 1, timing for setting	296
receiver interrupt flag in mode 2, timing for setting	297
relationship between data bus width and control signal.....	122, 123
reload register (UTIMR).....	227
reload register (UTIMR), bit configuration of.....	227
reload timer.....	3
reset cause.....	4, 59
reset cause holding circuit, block diagram of.....	78
reset cause holding setting.....	79
reset sequence.....	59
reset suspension.....	74
resource interrupt request as DMA transfer request, using.....	330
restart operation in sleep status, how to.....	93
restart operation in stop status, how to.....	90

INDEX

RET instruction operation	58
return pointer (RP)	27
ROM writer, writing by	343
RSRR and WTCR configuration	62

S

sample assembler source code	85
sample external device connection	129
sample setting for baud rate and U-TIMER reload value	301
sample time-division I/O interface timing	152
saving and returning	339
SCR configuration	285
SCR, bit function of	285
second word of descriptor, configuration of	316
sector erase	357
sector erase operation status	361, 362
sector protect operation, list of	364
serial input data register (SIDR) and serial output data register (SODR)	288
serial output data register (SODR)	288
series configuration	4
single/block transfer mode	317
sleep control circuit, block diagram of	92
sleep status, outline of	87
sleep/stop state, return from	17
SMR configuration	283
SMR, bit function of	283
SSR configuration	289
SSR, bit function of	289
standby mode (sleep/stop), return from	256
standby mode, operation in	88
standby mode, status switch in	95
STCR configuration	64
step trace trap operation	56
stop control circuit, block diagram of	89
stop conversion mode	275
stop erase, temporary	358
stop status, outline of	87
stopped state, return from	239
suppressing DMA transfer when interrupt having higher priority occurs	330
switch to sleep status, how to	92
switch to stop status, how to	89
symbol used in timing diagram, explanation of	320
system stack pointer (SSP)	27, 50

T

table base register (TBR)	27, 51
---------------------------------	--------

temporary sector erase stop status	361
temporary sector unprotect	366
temporary stop erase	358
term in pin status list, explanation of	379
third word of descriptor, configuration of	316
time-based timer	75
time-division I/O specification	102
timer control status register (TCCS), configuration of	207
timing for setting receiver interrupt flag in mode 0	296
timing for setting receiver interrupt flag in mode 1	296
timing for setting receiver interrupt flag in mode 2	297
timing for setting transmitter interrupt flag of mode 0, 1 and 2	297
TMCSR configuration	170
TMCSR, bit function of	170
TMR configuration	173
TMRLR configuration	174
transfer data format in asynchronous mode	293
transfer data format in CLK synchronous mode ..	294
transfer end (when both addresses change)	329
transfer end (when either of addresses is fixed) ..	328
transfer mode, single/block	317
transfer stop in continuous transfer mode (when both of addresses change), 16-bit/8-bit data	326
transfer stop in continuous transfer mode (when either of addresses is fixed), 16-bit/8-bit data	325
transmitter interrupt flag of mode 0, 1 and 2, timing for setting	297

U

UART	3
UART clock selection	292
UART feature	280
UART operation mode	292
UART register configuration	282
UART sample application	299
UART, note on using	299
undefined instruction exception, operation for	57
underflow operation	175
unused input pin, handling	16
user interrupt operation	55
user stack pointer (USP)	28
using hardware sequence flag, examples of	362
U-TIMER control register (UTIMC)	228
U-TIMER control register (UTIMC), bit detail of ...	228

U-TIMER control register (UTIMC), note on using	229	watchdog control unit, block diagram of	74
U-TIMER value register (UTIM)	227	Watchdog timer function	17
U-TIMER, list of register of	227	watchdog timer starting	74
U-TIMER, outline of	226	word access	134
V		WPR configuration	71
verify sector protect	365	write cycle timing	147
W		writing by ROM writer	343
wait cycle	140	writing data to MODR register, note on	97
		WTCR configuration	62

CM71-10115-1E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL
FR30
32-BIT MICROCONTROLLER
MB91F127/128
HARDWARE MANUAL

June 2003 the first edition

Published **FUJITSU LIMITED** Electronic Devices
Edited Business Promotion Dept.
