FUJITSU SEMICONDUCTOR

CM71-10113-1E

CONTROLLER MANUAL

FR20

32-Bit Micro Controller

MB91191/MB91192 Series Hardware Manual

FUJITSU

FR20

32-Bit Micro Controller

MB91191/MB91192 Series Hardware Manual

FUJITSU LIMITED

PREFACE

■ Purpose of This Document and Intended Reader

The MB91191/MB91192 are developed as one of the "32-bit single-chip microcontroller FR20 series" around the new RISC architecture CPU as its cores, and the specifications for these products are optimized for structures on which high-performance CPU processing power is required.

The functions and operations of the MB91191/MB91192 are described in this document specifically for engineers who actually develop products using the MB91191 and MB91192. Please read through this manual. For more information on various instructions, refer to "Instruction Manual".

Trademarks

■ Trademarks

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

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

Other system and product names used in this manual are trademarks of the related companies and organizations.

■ Organization of This Document

This manual contains the following 21 chapters and an appendix.

CHAPTER 1 Overview of MB91191/MB91192 Series

This chapter includes basic explanations including features of the MB91191/MB91192 series, block diagrams, and a function outline.

CHAPTER 2 Handling Devices

This chapter describes points to note when using the MB91191/MB91192 series.

CHAPTER 3 CPU

This chapter provides basic explanations for such elements as the architecture, specifications, and commands, etc., required to understand the CPU core functions of the FR series.

CHAPTER 4 External Bus Interface

This chapter describes an outline of the external bus interface, the register configuration/functions, the bus operation, and the bus timing, and program examples for the bus operation are explained.

CHAPTER 5 I/O Port

This chapter describes an outline of the I/O port and the register configuration/functions.

CHAPTER 6 FG Input

This chapter describes an outline of the FG input section, the register configuration/functions, and their operation.

CHAPTER 7 FRC Capture

This chapter describes an outline of the FRC capture section, the register configuration/functions, and each input section operation.

CHAPTER 8 Programmable Pulse Generator (PPG0, 1)

This chapter describes an outline of the programmable pulse generator (PPG0, 1), the register configuration/functions, and their operation.

CHAPTER 9 Real Timing Generator (RTG)

This chapter describes an outline of the real timing generator (RTG), the register configuration/functions, and their operation.

CHAPTER 10 Timer

This chapter describes an outline of the timer section, the register configuration/functions, and timer section operation.

CHAPTER 11 12-bit PWM

This chapter describes an outline of the PWM, the register configuration/functions, and the PWM operations.

CHAPTER 12 8-bit Pulse Width Counter

This chapter describes an outline of the 8-bit pulse width counter, the register configuration/functions, and 8-bit pulse width counter operations.

CHAPTER 13 External Interrupt

External interrupt comprises of the key input interrupt and external interrupt sections. This chapter describes an outline of the external interrupt 1 (key input circuit) and external interrupt (INT0 to 2), and the register configuration/functions, and their operation.

CHAPTER 14 Delayed Interrupt Module

This chapter describes an outline of the delayed interrupt module, the register configuration/functions, and delayed interrupt module operations.

CHAPTER 15 Interrupt Controller

This chapter describes an outline of the interrupt controller, the register configuration/functions, and the interrupt controller operations.

CHAPTER 16 10-bit A/D Converter

This chapter describes an outline of the 10-bit A/D converter, the register configuration/functions, and the 10-bit A/D converter operations.

CHAPTER 17 Serial I/O

This chapter describes an outline of the serial I/O, the register configuration/functions, and the serial data RAM and serial I/O operations.

CHAPTER 18 10-bit General-purpose Prescaler

This chapter describes an outline of the 10-bit general-purpose prescaler, the register configuration/functions, and the 10-bit general-purpose prescaler operations.

CHAPTER 19 Bit Search Module

This chapter describes an outline of the bit search module, the register configuration/functions, the bit search module operations, and save/return processes.

CHAPTER 20 Wait Controller

This chapter describes an outline of the wait control section, and the register configuration/functions.

CHAPTER 21 Flash Memory

This chapter describes an outline of the flash memory, the register configuration/functions and the flash memory operations.

Appendix

Details that could not be described within the body text, such as I/O map, interrupt vector, peripheral circuit measurement speed, restrictions and commands list for use of the MB91191/MB91192 series are described in the appendix.

- The contents of this document are subject to change without notice. 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.

How to Read This Document.

■ Format of This Book

The explanation concerning the main term used in this book is shown as follows.

Term	Meaning
I-bus	It is a bus of the width of 16 bits for an internal instruction. Because the FR20 series has adopted internal Harvard Architecture, the instruction and data are the independent buses. The bus converter is connected with I-BUS.
D-bus	It is a data bus of the width of the internal 32 bits. An internal resource is connected with D-bus.
C-bus	It is an internal multiplex bus. The C-bus connects to the I-bus and D-bus via the switch. The external interface module is connected with C-BUS. The external data bus multiplexes data and instructions.
R-bus	It is a data bus of the width of the internal 16 bits. R-bus is connected with D-bus through the adaptor. Various I/O, the clock generation block, and the interruption controller are connected with R-bus. The R-bus is 16-bit width, so the address and data are multiplexed. If the CPU accesses these resources, it takes a number of cycles.
E-unit	It is an operation execution unit.
ϕ	It is a system clock. It is a clock output from the clock generation block to the each internal resource connected with R-bus. The system clock at the highest speed shows the same cycle as source oscillation but is divided into 1, 1/2, 1/4, and 1/8 (or 1/2, 1/4, 1/8, and 1/16) by PCK1 and PCK0 of the clock generator GCR register.
θ	It is a system clock. It is an operation clock of resource and CPU connected with buses other than R-bus. The system clock at the highest speed shows the same cycle as source oscillation but is divided into 1, 1/2, 1/4, and 1/8 (or 1/2, 1/4, 1/8, and 1/16) by CCK1 and CCK0 of the clock generator GCR register.

CONTENTS

CHAPTER 1

Overview of MB91191/ MB91192 Series

This chapter includes basic explanations including features of the MB91191/MB91192 series, block diagrams, and a function outline.

1.1 Feature of MB91191/MB91192 Series

The MB91191/MB91192 series is a single-chip microcontroller with a built-in peripheral I/O resource suited to software servo control of VTRs that require high-speed CPU processing, featuring a 32-bit RISC-CPU (FR20 series) at its core.

■ Feature of MB91191/MB91192 Series

● CPU

- 32-bit RISC (FR20), load/store architecture, 5 stages pipeline
- 32-bit general-purpose register x 16
- One instruction/one cycle, 16-bit fixed length instructions (basic instruction)
- Commands for memory to memory transfer, bit processing, and barrel shift, etc.: Commands suitable for embedded applications
- Commands for function entry/exit, command for multi loading/storing of register contents: Commands supporting high-level languages
- Register interlock function: Simplification of assembler description
- Branch instruction with a delay slot: Decrease of overhead for branch processing
- Support at internal into/instruction level of multipliers
  - 32-bit multiplication with sign: 5 cycles
  - 16-bit multiplication with sign: 3 cycles
- Interruption (save of PC and PS): 6 cycles and 16 priority levels

● Bus interface

- 16-bit address output, 8-/16-bit data I/O
- Basic bus cycle: 2 clock cycle
- Support interface to various memories
- Multiplexed data/address input/output
- Auto-wait cycle: 0 to 7 cycles can be set randomly per area.
- Unused data and address pins can be used as I/O ports.
- Support of little endian mode

● Bit search module

- 1-cycle search for the change bit position of the first 1/0 from the MSB within a word

● Serial I/O

- Internal buffer RAM x 3ch (up to 128 bytes can be transferred automatically)
- Independent mode of the transmission/reception buffer (up to 64 bytes can be transferred automatically)

● A/D converter (Successive Approximation Type)

- 10-bit x 16ch
- Successive approximation conversion method (conversion time: 8.4 µs @20MHz)
- Channel scan function
- Hardware and software conversion start functions
- Internal FIFO (Software conversion: 6 stages, Hardware conversion: 6 stages)

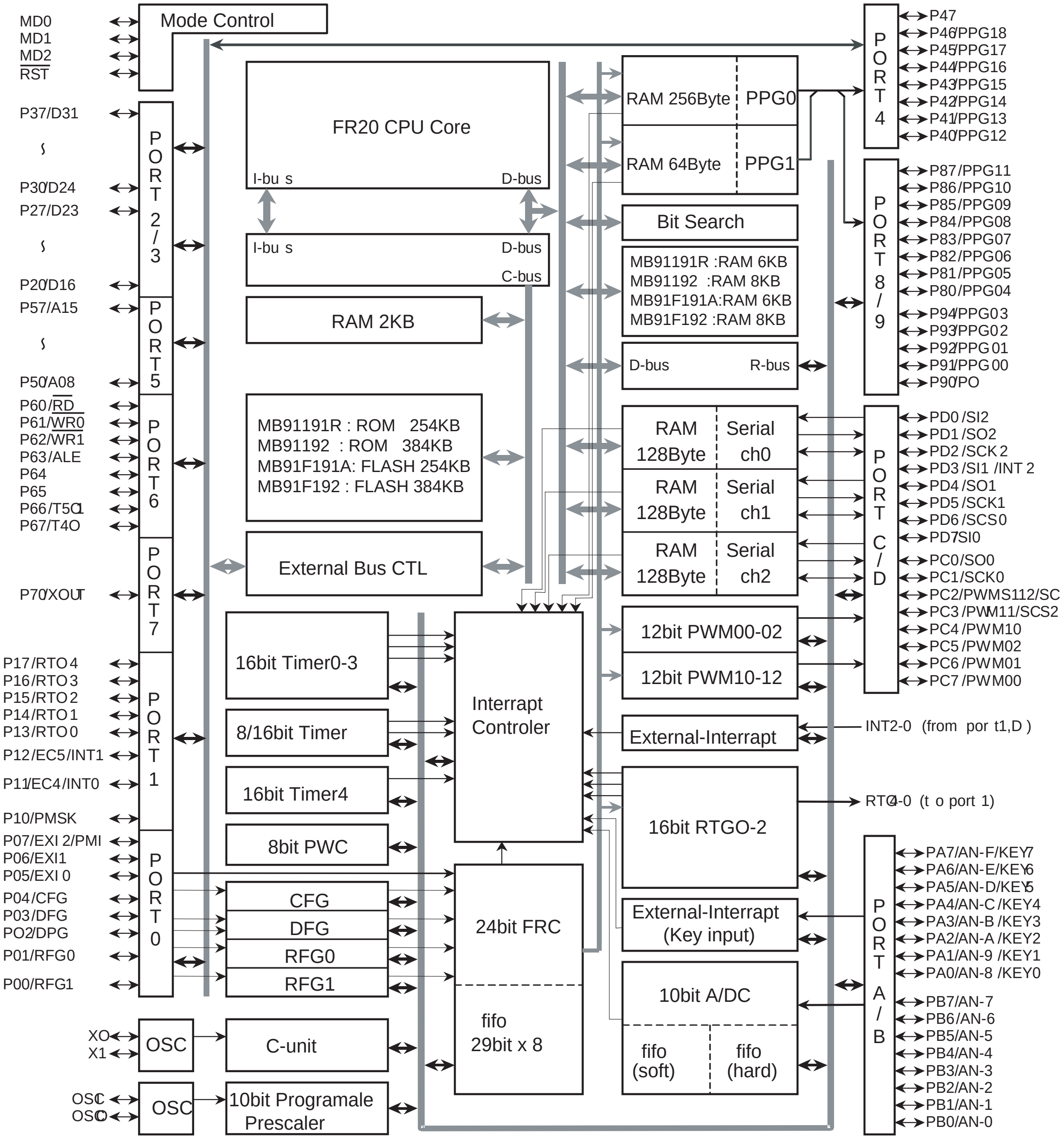
- Timer
    - 16-bit x 4ch
    - 16-bit timer/Counter x 1ch (with square wave output)
    - 8-/16-bit timer/Counter x 1ch (with square wave output)
- FG Input
    - Incorporates capstan, drum, and reel input circuit
- Capture
    - Internal 24-bit free-run counter (Minimum resolution = 50ns@20 MHz)
    - Internal FIFO (Data: 21-bit x 8, Detection: 8-bit x 8)
- Programmable pules generator
    - Internal buffer RAM (PPG0: 256 bytes, PPG1: 64 bytes)
    - Output timing precision: 800 ns (@20 MHz)
    - Includes an A/DC hardware start function
- Real time timing generator
    - RTG: incorporates 3 circuits
    - Output timing precision: 400 ns/800 ns selectable
    - Timing output ports: 5 ports
- PWM
    - 12-bit PWM x 6ch (rate, multi-type)
    - Base frequency = 78.1 KHz/39.0 KHz (@20MHz) selectable
- PWC
    - 8-bit PWC x 1ch (with mask input)
    - Measurement precision: 400 ns (@20 MHz)
- General-purpose Prescaler
    - 10-bit prescaler x 1ch (with square wave and pulse outputs)
    - Dedicated internal oscillator circuit
    - Includes load function driven by PPG output
- Interrupt control
    - External interrupt input: 3 inputs
    - Key input interrupt: 8 inputs

1.2 Block Diagram of All MB91191/MB91192 Series

Figure 1.2-1 shows the block diagram of all MB91191/MB91192 series.

■ Block Diagram of All MB91191/MB91192 Series

Figure 1.2-1 Block diagram of all MB91191/MB91192 series



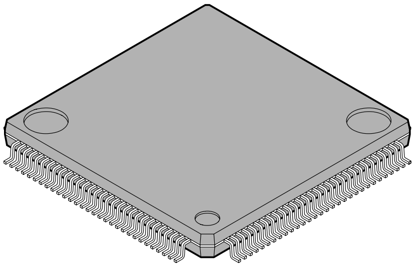
1.3 Package Dimension

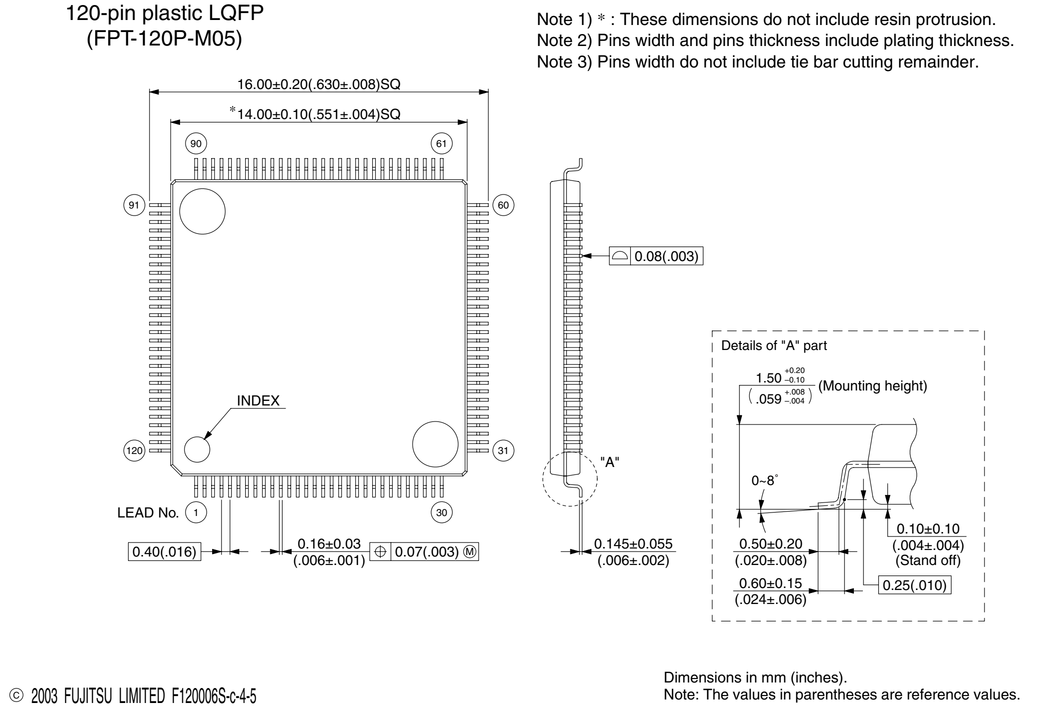
MB91191/MB91192 series is available in one type of packages.

■ Package Dimension (LQFP-120)

Figure 1.3-1 Package Dimension of FTP-120-M05

FPT-120P-M05

120-pin plastic LQFP  (FPT-120P-M05)	Lead pitch	0.40 mm
	Package width × package length	14.0 × 14.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.62 g
	Code (Reference)	P-LFQFP120-14×14-0.40


120-pin plastic LQFP
(FPT-120P-M05)

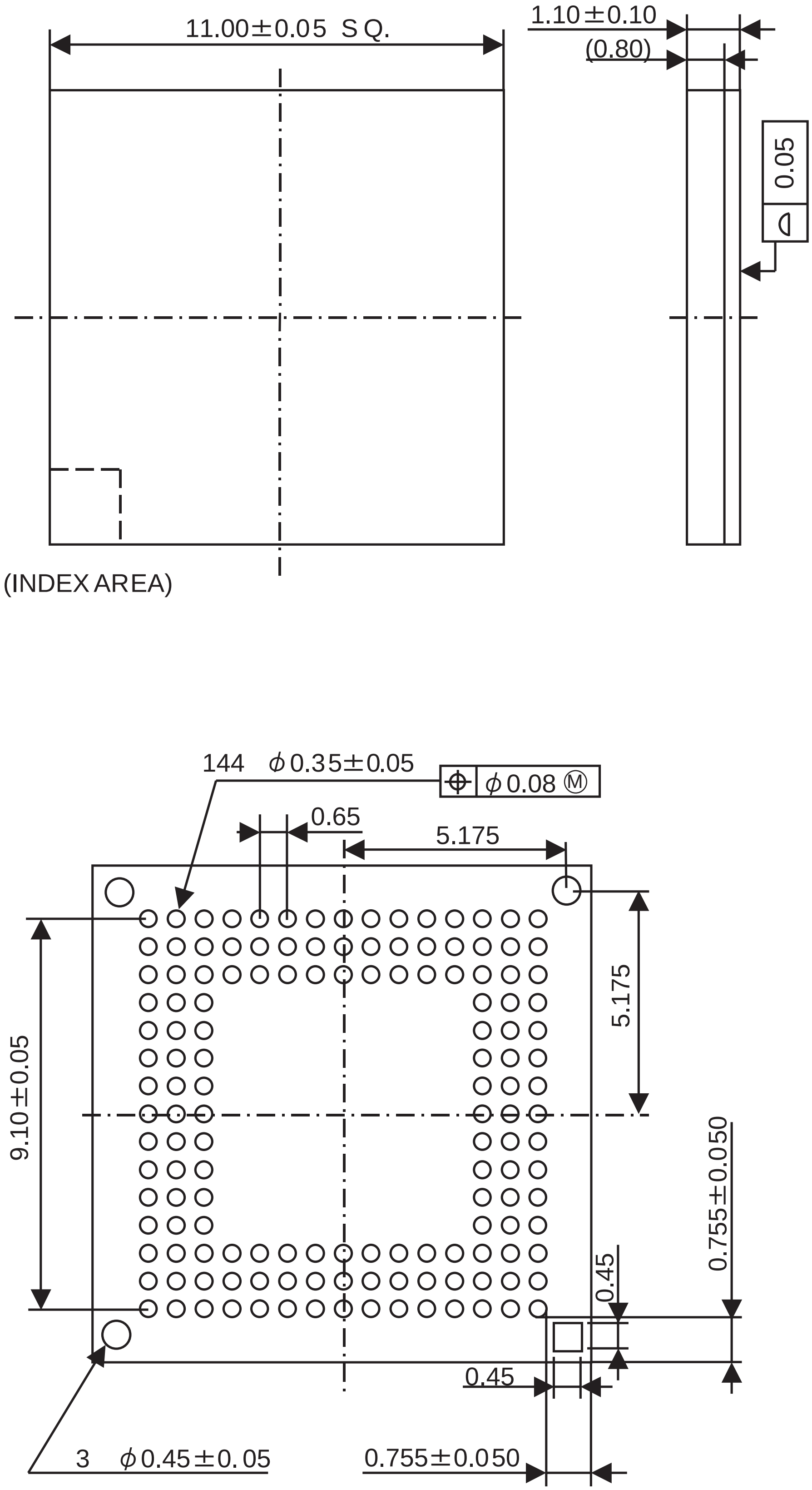
Note 1) ∗ : These dimensions do not include resin protrusion.
Note 2) Pins width and pins thickness include plating thickness.
Note 3) Pins width do not include tie bar cutting remainder.

© 2003 FUJITSU LIMITED F120006S-c-4-5

Dimensions in mm (inches).
Note: The values in parentheses are reference values.

■ Package Dimension (FLGA-144)

Figure 1.3-2 Package dimension of FLGA-144

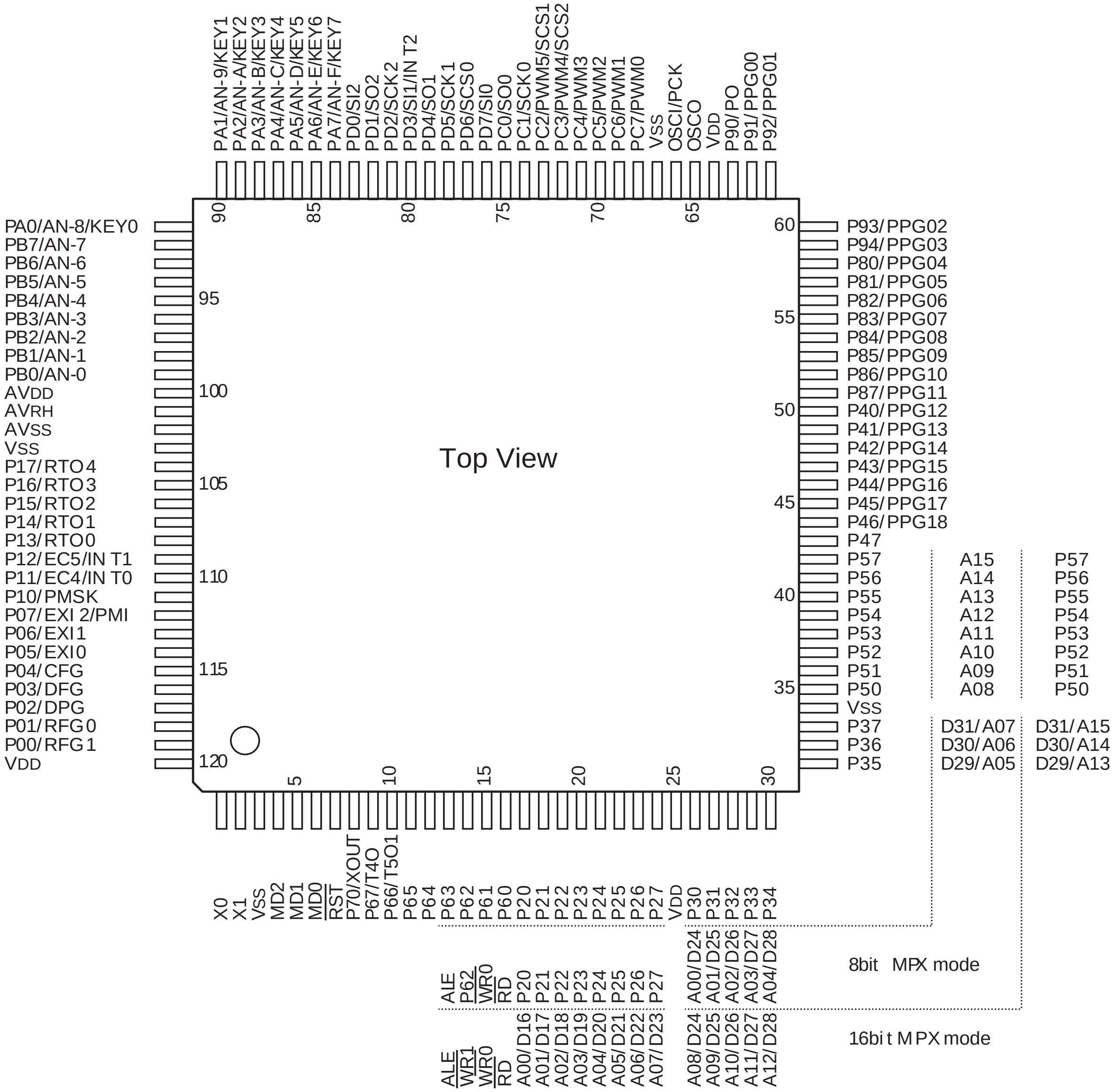


1.4 Pin Assignment

Figure 1.4-1 and Figure 1.4-2 show the pin assignment of the MB91191/MB91192 series.

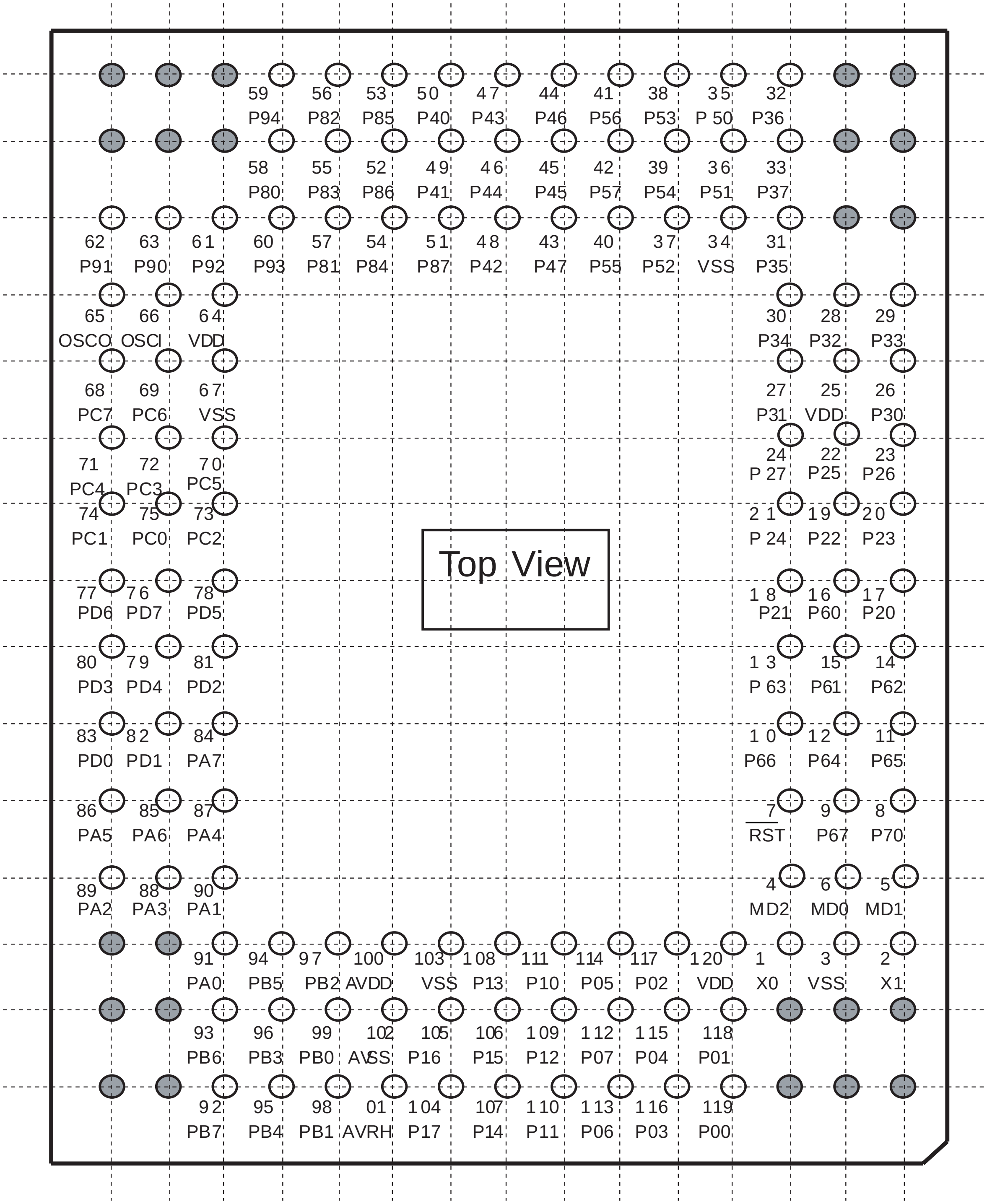
■ Pin Assignment (LQFP-120)

Figure 1.4-1 Pin assignment of LQFP-120



■ Pin Assignment (FLGA-144)

Figure 1.4-2 Pin assignment of FLGA-144



Note:

For the MB91191 series, the FLGA-144 package does not supply. It only supplies for the MB91192 series.

1.5 Pin Function Description

Table 1.5-1 lists the pin function of MB91191/MB91192 series.
The numbers shown in the tables has nothing to do with package pin numbers. For pin numbers, see "1.4 Pin Assignment".

■ Pin Function List

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
1	X0(I)	A	It is crystal oscillation pin.
2	X1(O)		
3	Vss	-	It is Vss pin.
4	MD2	B	It is operating mode specification pin. It is CMOS schmitt input.
5	MD1		
6	MD0		
7	$\overline{RST}$	B	It is reset input pin. It is CMOS schmitt input.
8	P70/XOUT	C	This pin is shared with clock output (X0/2, PCK/2). It is CMOS input.
9	P67/T4O	C	This pin is shared with timer 4 square wave output. It is CMOS input.
10	P66/T5O1		This pin is shared with timer 5 square wave output. It is CMOS input.
11	P65		It is general-purpose I/O port. It is CMOS input.
12	P64		It is general-purpose I/O port. It is CMOS input.
13	P63/ALE/ALE		This pin is shared with address strobe output. It is CMOS input.
14	P62/P62/$\overline{WR1}$		This pin is shared with write strobe output 1. It is CMOS input.
15	P61/$\overline{WR0}$/$\overline{WR0}$		This pin is shared with write strobe output 0. It is CMOS input.
16	P60/$\overline{RD}$ /$\overline{RD}$		This pin is shared with read strobe output. It is CMOS input.

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
17	P20/P20/D16:A00	C	It is general-purpose I/O port. It is CMOS input.
18	P21/P21/D17:A01		
19	P22/P22/D18:A02		
20	P23/P23/D19:A03		
21	P24/P24/D20:A04		
22	P25/P25/D21:A05		
23	P26/P26/D22:A06		
24	P27/P27/D23:A07		
25	V_{DD}	-	It is power supply pin.
26	P30/D24:A00/D24:A08	C	It is high-current I/O port and shared with external bus pins. It is CMOS input.
27	P31/D25:A01/D25:A09		
28	P32/D26:A02/D26:A10		
29	P33/D27:A03/D27:A11		
30	P34/D28:A04/D28:A12		
31	P35/D29:A05/D29:A13		
32	P36/D30:A06/D30:A14		
33	P37/D31:A07/D31:A15		
34	Vss	-	It is Vss pin.
35	P50/A08/P50	C	It is high-current I/O port and shared with external bus pins. It is CMOS input.
36	P51/A09/P51		
37	P52/A10/P52		
38	P53/A11/P53		
39	P54/A12/P54		
40	P55/A13/P55		
41	P56/A14/P56		
42	P57/A15/P57		

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
43	P47	C	It is general-purpose I/O port. It is CMOS input.
44	P46/PPG18		This pin is shared with PPG output. It is CMOS input.
45	P45/PPG17		
46	P44/PPG16		
47	P43/PPG15		
48	P42/PPG14		
49	P41/PPG13		
50	P40/PPG12		
51	P87/PPG11	C	This pin is shared with PPG output. It is CMOS input.
52	P86/PPG10		
53	P85/PPG09		
54	P84/PPG08		
55	P83/PPG07		
56	P82/PPG06		
57	P81/PPG05		
58	P80/PPG04		
59	P94/PPG03	C	This pin is shared with PPG output. It is CMOS input.
60	P93/PPG02		
61	P92/PPG01	C	This pin is shared with PPG output. It is CMOS input.
62	P91/PPG00		
63	P90/PO		This pin is shared with general-purpose prescaler output. It is CMOS input.
64	V_{DD}	-	It is power supply pin.
65	OSCO (O)	A	It is crystal oscillation pin for dedicated general-purpose prescaler.
66	OSCI/PCK(I)		
67	Vss	-	It is Vss terminal.
68	PC7/PWM0	C	This pin is shared with PWM output. It is CMOS input.
69	PC6/PWM1		
70	PC5/PWM2		
71	PC4/PWM3		

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
72	PC3/PWM4/SCS2	F	This pin is shared with PWM output and serial 2 chip select. It is CMOS schmitt input.
73	PC2/PWM5/SCS1		This pin is shared with PWM output and serial 1 chip select. It is CMOS schmitt input.
74	PC1/SCK0		This pin is shared with serial 0 shift clock. It is CMOS schmitt input.
75	PC0/SO0	C	This pin is shared with serial 0 serial output. It is CMOS input.
76	PD7/SI0	F	This pin is shared with serial 0 serial input. It is CMOS schmitt input.
77	PD6/SCS0		This pin is shared with serial 0 chip select input. It is CMOS schmitt input.
78	PD5/SCK1		This pin is shared with serial 1 shift clock. It is CMOS schmitt input.
79	PD4/SO1	C	This pin is shared with serial 1 serial input. It is CMOS input.
80	PD3/SI1/INT2	F	This pin is shared with serial 1 serial input and external interrupt 2. It is CMOS schmitt input.
81	PD2/SCK2		This pin is shared with serial 2 shift clock. It is CMOS schmitt input.
82	PD1/SO2	C	This pin is shared with serial 2 serial output. It is CMOS input.
83	PD0/SI2	F	This pin is shared with serial 2 serial input. It is CMOS schmitt input.
84	PA7/AN-F/KEY7	E	This pin is shared with analog input and key input. It is CMOS schmitt input.
85	PA6/AN-E/KEY6		
86	PA5/AN-D/KEY5		
87	PA4/AN-C/KEY4		
88	PA3/AN-B/KEY3		
89	PA2/AN-A/KEY2		
90	PA1/AN-9/KEY1		
91	PA0/AN-8/KEY0		

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
92	PB7/AN-7	D	This pin is shared with analog input. It is CMOS schmitt input.
93	PB6/AN-6		
94	PB5/AN-5		
95	PB4/AN-4		
96	PB3/AN-3		
97	PB2/AN-2		
98	PB1/AN-1		
99	PB0/AN-0		
100	AV_{DD}	-	It is A/D converter power supply pin.
101	AVRH	-	It is A/D converter reference power supply pin.
102	AVss	-	It is Vss pin of A/D converter.
103	Vss	-	It is Vss pin.
104	P17/RTO4	C	This pin is shared with RTG output. It is CMOS input.
105	P16/RTO3		
106	P15/RTO2		
107	P14/RTO1		
108	P13/RTO0		
109	P12/EC5/INT1	F	This pin is shared with timer 5 clock input and external interrupt input. It is CMOS schmitt input.
110	P11/EC4/INT0		This pin is shared with timer 4 clock input and external interrupt input. It is CMOS schmitt input.
111	P10/PMSK		This pin is shared with PWC mask input. It is CMOS schmitt input.

Table 1.5-1 Pin Function List

Pin No. (LQFP)	Pin name	Form	Function
112	P07/EXI2/PMI	F	This pin is shared with external capture input and PWC input. It is CMOS schmitt input.
113	P06/EXI1		This pin is shared with external capture input. It is CMOS schmitt input.
114	P05/EXI0		
115	P04/CFG		This pin is shared with capstan FG input. It is CMOS schmitt input.
116	P03/DFG		This pin is shared with drum FG input. It is CMOS schmitt input.
117	P02/DPG		This pin is shared with drum pulse input. It is CMOS schmitt input.
118	P01/RFG0		This pin is shared with reel FG input. It is CMOS schmitt input.
119	P00/RFG1		
120	V_{DD}	-	It is power supply pin.

1.6 I/O Circuit Type

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

■ I/O Circuit Type

Table 1.6-1 I/O circuit types

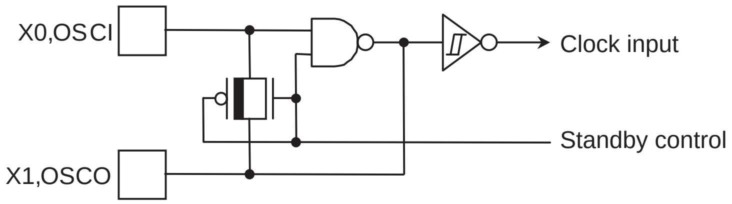
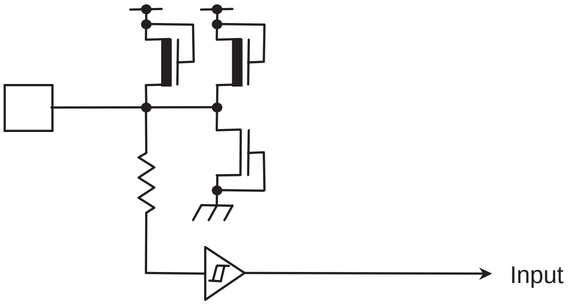
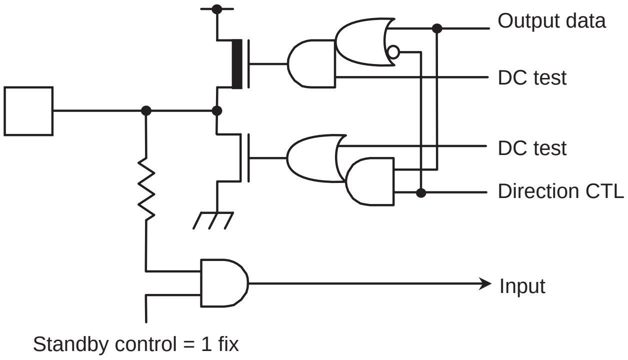
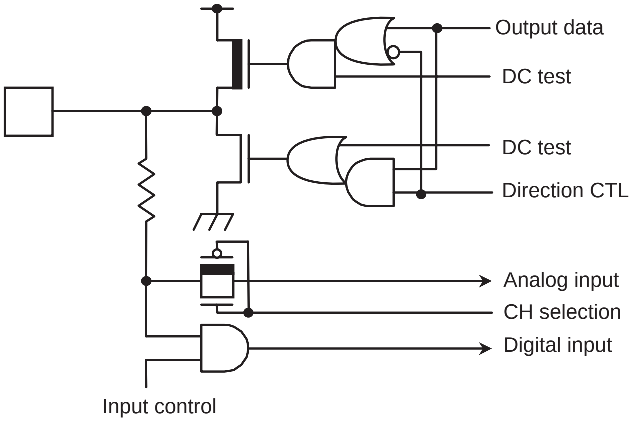
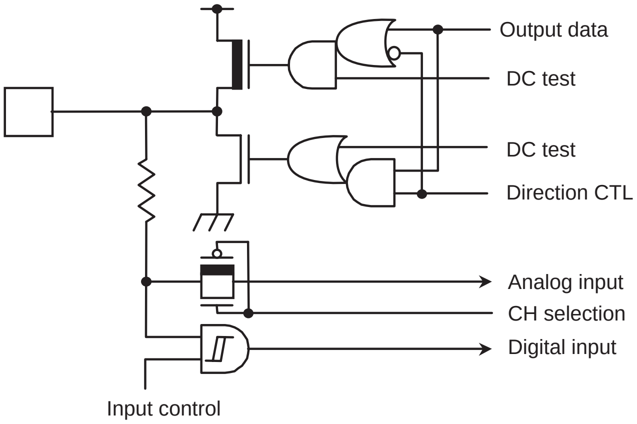
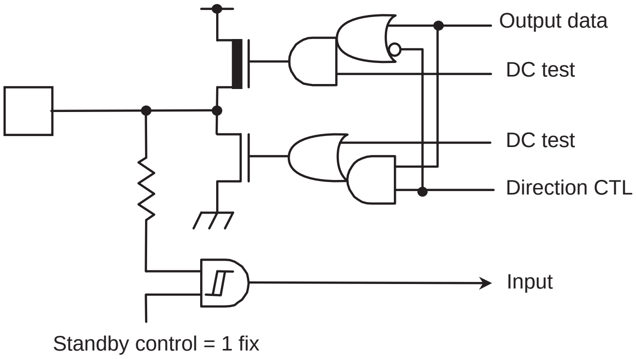
Classifi-cation	Circuit Type	Remark
A		• Oscillation return resistance about 1MΩ
B		• CMOS schmitt input
C	 Output data DC test DC test Direction CTL Input Standby control = 1 fix	• CMOS level output • CMOS input without standby control

Table 1.6-1 I/O circuit types

Classification	Circuit Type	Remark
D	 Output data DC test DC test Direction CTL Analog input CH selection Digital input Input control	• CMOS level output • CMOS input with input control • Analog input
E	 Output data DC test DC test Direction CTL Analog input CH selection Digital input Input control	• CMOS level output • CMOS schmitt input with input control • Analog input
F	 Output data DC test DC test Direction CTL Input Standby control = 1 fix	• CMOS level output • CMOS schmitt input without standby control

CHAPTER 2

Handling Devices

This chapter describes points to note when using the MB91191/MB91192 series.

2.1 Precautions When Handling Devices

The semiconductor device breaks down at a certain probability. Moreover, the failure of the semiconductor device is greatly controlled by the condition (circuit condition and environmental condition, etc.) used.
To have the high use, the semiconductor device is explained about reliability about the matter which should be noted and considered as follows.

■ Precautions when Designing

Here, the matter which should be noted when an electronic equipment is designed with a semiconductor device is described.

● Observance of absolute maximum rating

When an excessive stress (voltage, current, and temperature, etc.) adds, the semiconductor device has the possibility to destroy. The value that this threshold was provided is an absolute maximum rating. Therefore, care must be taken not to exceed the rating even for one item.

● Observance of recommended operation condition

The recommended operation condition is a condition to guarantee normal movement of the semiconductor device. All standard values for electric features are assured within this condition range. Always use under the recommended operation condition. When this condition is exceeded and used, the adverse effect is occasionally caused for reliability.

Use by the item, the condition, and the logical combination not described to this material is not guaranteed. Please consult with the section in charge of sales of our company about use by conditions other than being described for the idea beforehand.

● Processing and protection of terminal

In the semiconductor device, there are a power supply and various input/output terminals. The following attention is necessary for these.

- Prevention of over-voltage/over-current

Deterioration is caused in the device when the voltage/current which exceeds the maximum rating is applied to each terminal, and when it is remarkable, it becomes destruction. Please prevent such an over-voltage/over-current occurring when you design the equipment.

- Protection of output terminal

If the output terminal is short circuited with the power terminal or other output terminal, or when large capacity load is connected, a large electrical current may result. If this condition is prolonged, the device will be damaged, so this kind of connection should not be made.

- Processing of unused input terminal

If the input terminal with very high impedance is used while opened, the operation might become unstable. Ensure connections to the power terminal and ground terminal have the appropriate resistance.

● Latch up

The semiconductor device is composed by the formation of the region of the P-type and the N-type on the substrate. Internal parasitism PN junction (thyristor structure) might keep doing on-line when the voltage of an external abnormal voltage is added, and the heavy-current which exceeds hundreds of mA flow to the power supply terminal. This is called a latch up. The reliability of the device is not only damaged when this phenomenon occurs but also there is dreading the arrival heat generation, smoking, and the ignition to destruction. Please note the following points to prevent this.

- There must not be what the voltage more than the maximum rating adds the terminal. Please note abnormal noise and surge etc.
- An abnormal current must not flow in consideration of the power supply turning on sequence.

● Restriction of safety etc. and observance of standard

All over the world, various restrictions and standards of safety and the EMI, etc. have been installed. Please suit these restriction and standard when the customer designs the equipment.

● Fail safe design

The semiconductor device breaks down at a certain probability. The customer safely designs such as the device redundancy, fire spreading prevention, exceeding current prevention, and prevention of malfunction not to occur the injury accident, fire accident, and social damage consequently when the semiconductor device is broken.

● Attention concerning usage

The our company semiconductor device is intended to be used for a standard usage (associated equipment for office appliances such as computers/OA and industries/communications/the measurements and personal/home equipment etc.). The customer concerns the usage of threatening the life by the breakdown and malfunction, dreading the damage to the human body, or the special application which the extreme high reliability is requested (fro aviation/space, atomic control, device for relayed the bottom of the sea, running control, the medical device to keep the life, etc.), be sure to consult with the sales division of our company. When you use without the consultation and acknowledge that the responsibility cannot be assumed about the occurring damage etc.

■ Precautions when Mounting Package

In the package, there are a lead insertion type and a surface mounting type. In both cases, quality assurance for heat resistance at the soldering stage only applies to the mounting under conditions recommended by us. Please inquire the section in charge of sales of our company about details of the mounting condition.

● Lead insertion type

There are two ways to mount the lead insertion type package onto the printing board-the first is to directly solder it onto the printing board, and the second is to mount it on the printing board using the socket.

When soldering it directly to the printing board, the flow soldering method (wave soldering method) whereby the solder is melted after inserting the lead through the hole in the printing board is generally used. In this case, heat stress in excess of the usual maximum rating preservation temperature is added to the lead part at the soldering stage. Please mount under the mounting recommendation condition of our company.

When the mounting method using the socket is used, if surface processing of the socket contact point and surface processing of the IC lead are different, contact failure may be generated after a prolonged period. Therefore, checking the surface processing of the socket contact and surface processing of the IC lead is recommended before mounting.

● Surface mounting type

The lead used in the surface mounting type package is thinner than that for the insertion type, so its shape is easily changed. In line with the increased number of pins in the package, the lead pitch is also narrow, and opening defects due to the lead change and short circuits due to the solder bridge can easily be caused, so an appropriate mounting technique is required. We recommend the solder re-flow method, and implement rank classification of the mounting conditions per product. Please mount according to the rank classification of the our company recommendation.

● About keeping the semiconductor device

The plastic package is made of resin, so moisture is absorbed if left in a natural environment. When the heat when mounting on the moisture absorption package joins, the decrease and the package crack of the wet-proofing by the interfacial flaking off generation might be generated. Please note the following points.

- The be dewy of moisture happens to the product in the place with a rapid temperature change. Store it in a place with minimum temperature fluctuations that avoid such an environment.
- The depository of the product recommends the use of a dry box. Please keep under the relative humidity to 70% RH and the temperature to 5 to 30 °C.
- Silica-gel is used as a dry medicine in our company with a damp-proof and high as packing material of the semiconductor device if necessary an aluminum laminate bag. Put the semiconductor device in the laminated aluminum bag and close it tightly for storage.
- Please avoid the place where a lot of places and dust where the corrosively gas is generated.

● About the baking

The moisture absorption package can be dehumidified by executing the baking (heating dryness). Please execute the baking by the condition which our company recommends.

● Static electricity

Please note the following points so that the semiconductor device may cause destruction by static electricity easily.

- Please adjust the relative humidity of the working environment to 40% to 70% RH. Please examine the use of TEL device (apparatus for generating ion) etc. if necessary.
- Earth the conveyer, soldering tank and iron, and peripheral incidental equipment to be used.
- In order to prevent electrocution of human body, try to earth through a ring or armlet with high resistance (about 1MΩ), wear conductive clothing and shoes, and place a conductive mat on the floor to minimize static build-up.
- Please execute the earth or the electrification prevention to the treatment device and meters.
- Avoid using materials that are easily charged, such as polystyrene foam, to store the assembled board.

■ Precautions

The reliability of the semiconductor is affected by the peripheral temperature as mentioned before, and other environmental conditions. Please note the following points.

● Humidity

As for environment long-term use under the high humidity environment, something wrong with the leak character might occur in not only the device but also printed wiring board, etc. Please consider the damp-proof processing such as giving when the high humidity is assumed.

● Static electricity discharge

The semiconductor device might cause the malfunction by being generate the electrical discharge in Kon when electrified thing exists in high voltage near the device. For this case, please treat the prevention of electrification or the prevention of the electrical discharge.

● Corrosively gas, dust, and oil

If the device is used within a corrosive gas environment or where dust and oil, etc. may become attached, the effect may be detrimental due to chemical reactions. Protective measures should be considered when used under such an environment.

● Radiation and cosmic ray

A general device does not assume the environment exposed to the radiation and the cosmic ray in the design. Therefore, use this to shield.

● Smoking and ignition

The device of the resin molding type is not nonflammability. Do not use the device near any ignitable substance. In the event of smoke or fire may be generated toxic gases.

Additionally, please consult the section in charge of sales of our company about use under a special environment for the idea.

2.2 Others

The others are explained

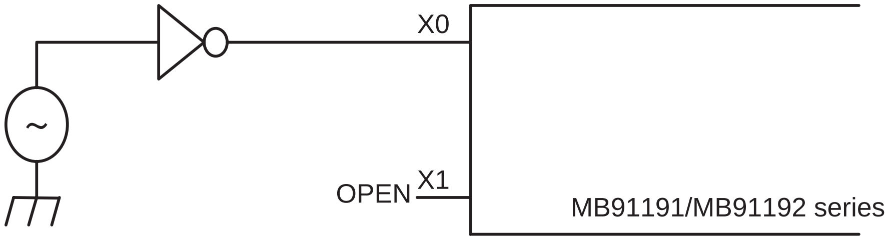
■ External Reset Input

When "L" level is input to the $\overline{\text{RST}}$ pin, to ensure the inside achieves reset status, "L" level input to the $\overline{\text{RST}}$ pin is required for at least five machine cycles.

■ Note on Using External Clock

When using the external clock, drive the X0. Figure 2.2-1 shows the example of using an external clock.

Figure 2.2-1 Example of using an external clock



*:Be sure to make X1 pin open.

■ Power Supply Pin

If there are multiple V_{DD} and V_{SS} pins, from the point of view of device design, pins to be of the same potential are connected the inside of the device to prevent such malfunctioning as latch up. To reduce unnecessary radiation, prevent malfunctioning of the strobe signal due to the rise of ground level, and observe the standard for total output current, be sure to connect the V_{DD} and V_{SS} pins to the power supply and ground externally.

Also try to ensure that connection to the V_{DD}, Vss on this device is at the lowest impedance possible from the power supply source. In addition, We will recommend the ceramic capacitor of about 0.1 µF to be connected as bypass capacitor between V_{DD} and V_{SS} near this device.

■ Crystal Oscillation Circuit

The noise near X0 and the X1 terminal becomes original of the malfunction of this device. The printing board should be designed so that the X0, X1, crystal oscillator (or ceramic oscillator), and bypass capacitor to the ground are arranged as close as possible.

Printing board artwork around the X0 and X1 terminals to the ground is strongly recommended, as steady operation can be expected.

■ Handling NC Pin

Use the Non Connect (N.C.) terminal while open.

■ Mode (MD0 to MD2) Pin

Please tie directly to V_{DD} or V_{SS} and use these terminals.

In order to prevent erroneous entry to test mode due to noise, the pattern length between each mode terminal and V_{DD} or V_{SS} on the printing board should be as short as possible, and they should be connected at low impedance.

■ At Power On

When the power is turned on, the $\overline{\text{RST}}$ pin must be started from "L" level status, and changed to "H" level after at least five cycles of the internal operation clock have passed, after the power source reaches the V_{DD} level.

CHAPTER 3

CPU

This chapter provides basic explanations for such elements as the architecture, specifications, and commands, etc., required to understand the CPU core functions of the FR series.

3.1 Memory Space

The logical address space of the FR20 series is 4 Gbytes (2^{32} addresses), and the CPU performs linear access.

■ Direct Addressing Area

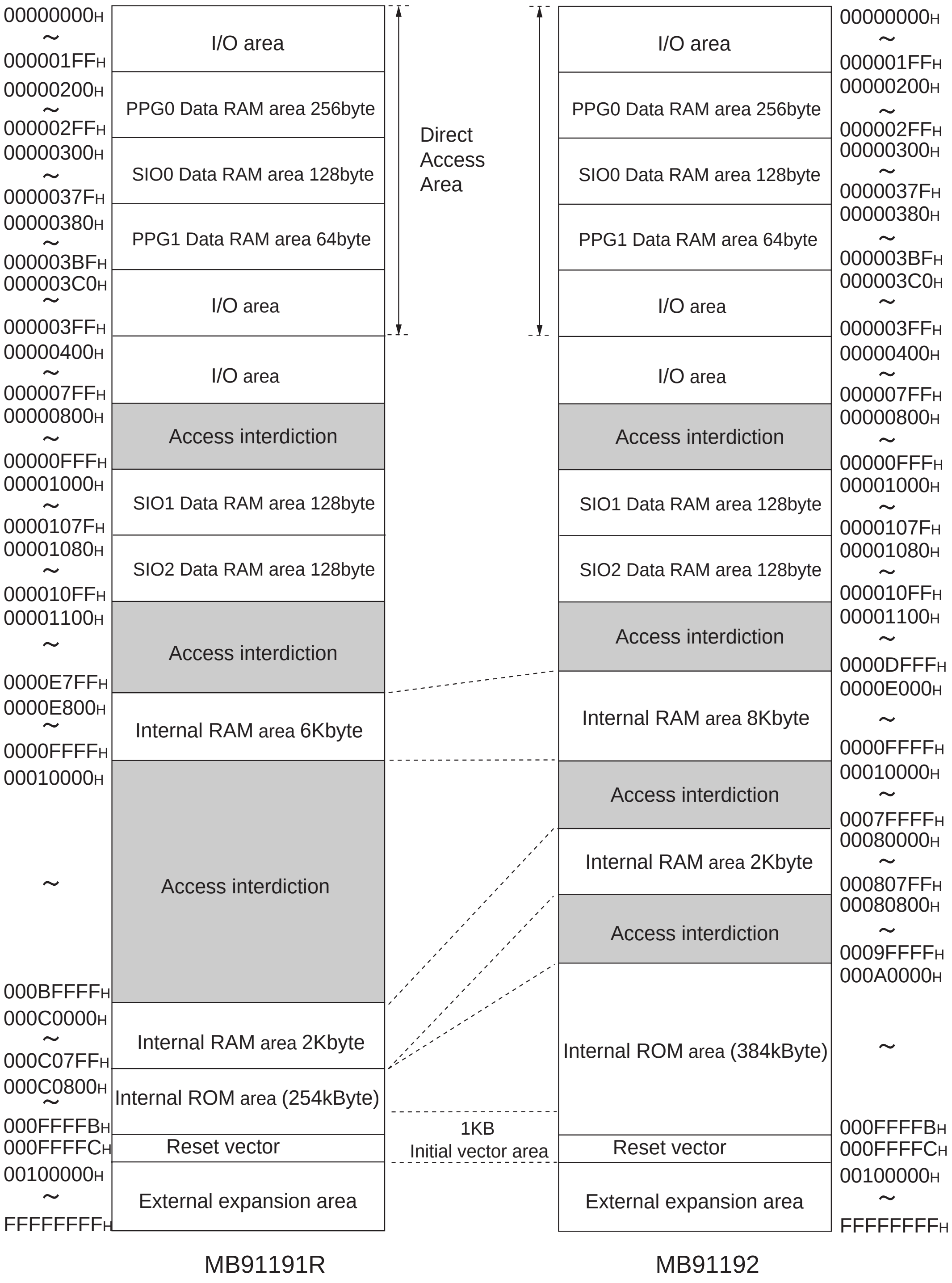
The under-mentioned area of the address space is used for I/O. This area is called the "direct addressing area" and operand addresses can be specified directly within the command. A direct area is different as follows depending on the size of the accessed data.

- Byte data access:0-0FF$_H$
- Half word data access:0-1FF$_H$
- Word data access:0-3FF$_H$

■ Memory Map

Figure 3.1-1 shows the memory map of the MB91191/MB91192.

Figure 3.1-1 MB91191/MB91192 Memory Map



Note:

Under single-chip mode, access to the external extension area is impossible.

Select internal ROM external bus mode using the mode register to access the external extension area.

3.2 CPU Architecture

The FR20 CPU is a high performance core that adopts highly functional commands for the embedded application as well as RISC architecture.

■ Feature of CPU Architecture

● Adoption of RISC architecture

- Basic instruction: one instruction one cycle

● 32 bit architecture

- General-purpose register 32 bits × 16

● Linear memory space of 4 GB

● Installing of multipliers

- 32 bits x multiplication 32 bits: 5 cycles
- 16 bits x multiplication 16 bits: 3 cycles

● Reinforcement of interruption processing function

- High-speed response speed (6 cycles)
- Support for multiple interrupts
- Level mask function (16 levels)

● Reinforcement of instruction for I/O operation

- Memory memory transfer operation
- Bit processing instruction

● High code efficiency

- 16 bits in basic instruction word length

● Low power consumption

- Sleep mode, stop mode

■ Construction of Internal Architecture

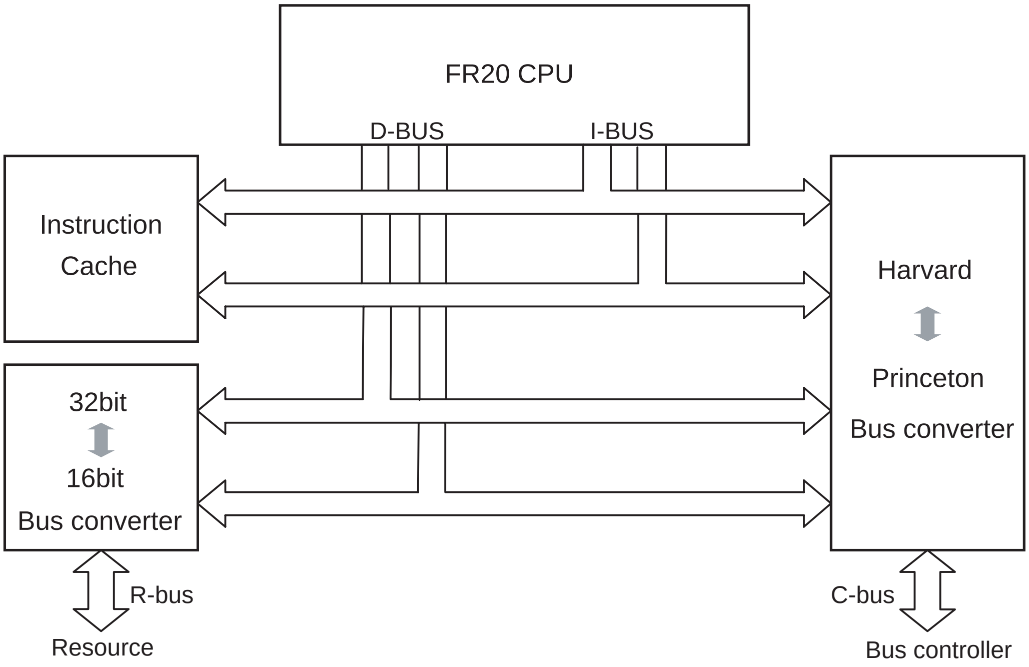
The FR20 CPU adopts the Harvard architecture structure whereby the command bus and data bus are independent.

The on chip command cache is connected to the command bus (T-bus). A 32-bit <--> 16-bit bus converter is connected to the data bus (D-bus), and performed interfaces between the CPU and peripheral resources.

A Harvard <--> Princeton bus converter is connected to both the I-bus and D-bus, performed and interfaces between the CPU and bus controller.

Figure 3.2-1 shows the construction of internal architecture.

Figure 3.2-1 Construction of Internal architecture



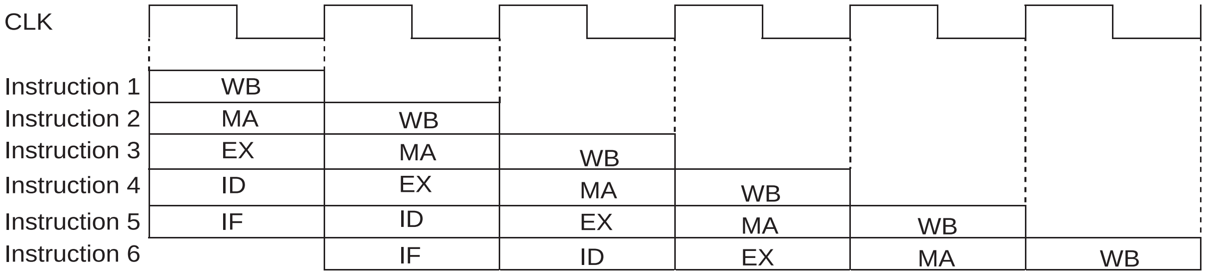
● CPU

The FR20's 32-bit RISC architecture is compactly implemented on the CPU. A five-level command pipeline method is adopted to execute one command per cycle. The pipeline is composed of the following stages.

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

Figure 3.2-2 shows the instruction pipeline.

Figure 3.2-2 Instruction pipeline



Instruction 1	WB					
Instruction 2	MA	WB				
Instruction 3	EX	MA	WB			
Instruction 4	ID	EX	MA	WB		
Instruction 5	IF	ID	EX	MA	WB	
Instruction 6		IF	ID	EX	MA	WB

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

As a rule, the instruction is executed at the speed of one instruction per cycle. However, a number of cycles are required to execute commands for the load/store command to which memory wait is attached, branch commands that do not have delay slots, and multi-cycle commands. Also, when the supplied instruction is slow, the execution speed of the instruction decrease.

Refer to "3.8 Overview of Instructions" for details.

● 32-bit ↔ 16-bit bus converter

Interfaces between the D-BUS that quickly accesses at 32-bit width and the R-BUS that accesses at 16-bit width, and realizes data access from the CPU to built-in peripheral circuit.

When 32-bit width access is performed from the CPU, this bus converter accesses the R-BUS by converting it to 16-bit width access twice. Some of built-in peripheral circuits have access width-related restrictions.

● Harvard ↔ Princeton bus converter

Coordinates between the CPU command access and data access, and realizes smooth interface with the external bus.

In CPU, the instruction bus and the data bus are the independent Harvard architecture structures. On the other hand, the bus controller that controls the external bus has a Princeton architectural structure with a single bus. This bus converter ranks the priority order for command access and data access of the CPU, and controls access to the bus controller. This operation always optimizes the external bus access ranking.

It also has a two-word write buffer to eliminate CPU bus waiting time and a one-word pre-fetch buffer to fetch commands.

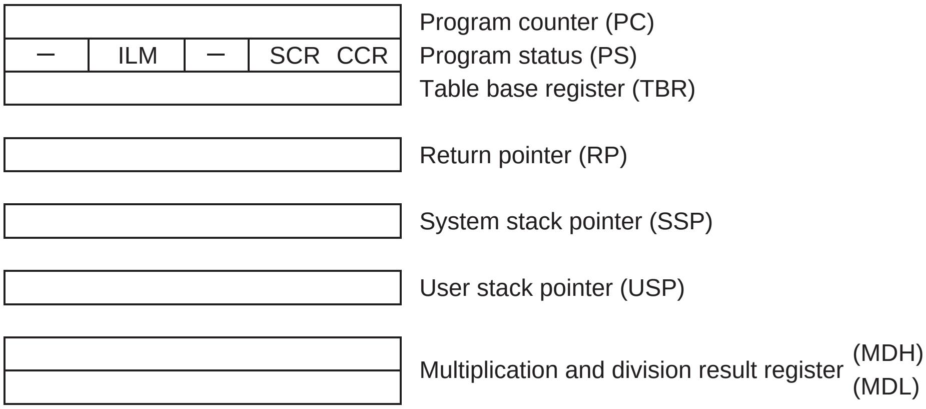
3.3 Dedicated Registers

Use the dedicated registers for specified purposes. Program counter (PC), program status (PS), table base register (TBR), return pointer (RP), system stack pointer (SSP), user stack pointer (USP), and multiplication/division results registers (MDH/MDL) are prepared.

■ Dedicated Registers List

Figure 3.3-1 shows the dedicated register list.

Figure 3.3-1 Dedicated registers list



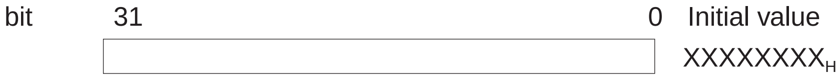
■ Program Counter (PC)

Function of program counter (PC: Program Counter) is described.

The program counter (PC) consists of 32-bit.

Figure 3.3-2 shows the bit configuration of the program counter (PC).

Figure 3.3-2 Program counter (PC)



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

If the PC is updated when an instruction is executed, Bit 0 is set to "0". Bit 0 may be "1" only when an odd address is specified as the branch destination address.

Even in that case, bit 0 is invalid, and the command must be placed in the address of the multiple of two.

The initial value by reset is irregular.

■ Program Status Register (PS)

This register retains the program status, and is separated into three parts, namely, ILM, SCR, and CCR. Refer to "3.3.1 Program Status Register (PS)" for details.

The undefined bits are all reserved bits. When the register is read, "0" is always read. Writing is invalid.

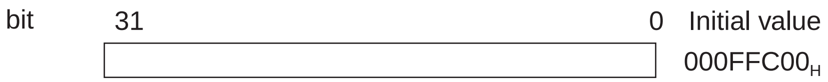
■ Table Base Register (TBR)

Function of table base register (TBR: Table Base Register) is described.

The table base register (TBR) consists of 32-bit.

Figure 3.3-3 shows the bit configuration of the table base register (TBR).

Figure 3.3-3 Table base register (TBR)



The table base register retains the starting address of the vector table used for EIT processing.

The initial value by reset is 000FFC00$_H$.

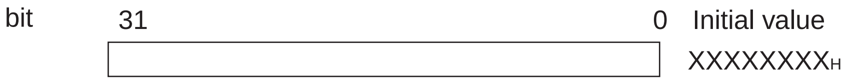
■ Return Pointer (RP)

Function of Return Pointer (RP: Return Pointer) is described.

The return pointer (RP) consists of 32-bit.

Figure 3.3-4 shows the bit configuration of the return pointer (RP).

Figure 3.3-4 Return pointer (RP)



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

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

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

The initial value by reset is irregular.

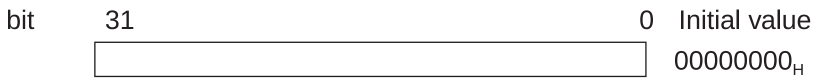
■ System Stack Pointer (SSP)

Function of system stack pointer (SSP: System Stack Pointer) is described.

The system stack pointer (SSP) consists of 32-bit.

Figure 3.3-5 shows the bit configuration of the system stack pointer (SSP).

Figure 3.3-5 System stack pointer (SSP)



SSP is the system stack pointer.

When the S flag is "0", the SSP functions as R15. The SSP can be specified explicitly.

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

The initial value by reset is 00000000$_H$.

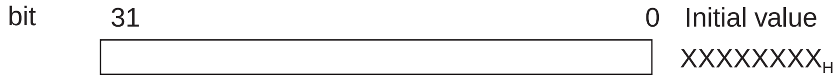
■ User Stack Pointer (USP)

Function of user stack pointer (USP: User Stack Pointer) is described.

The user stack pointer (USP) consists of 32-bit.

Figure 3.3-6 shows the bit configuration of the user stack pointer (USP).

Figure 3.3-6 User stack pointer (USP)



USP is the user stack pointer.

When the S flag is "1", the USP functions as R15. The USP can be specified explicitly. The initial value by reset is irregular.

The USP cannot be used for the RETI instruction.

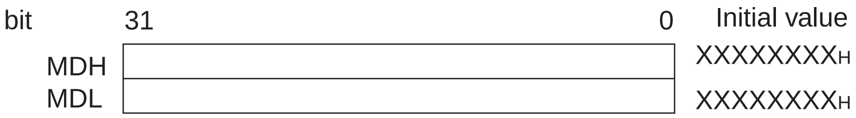
■ Multiplication and Division Result Register (MDH/MDL)

Function of multiplication and division result register (MDH/MDL: Multiply & Divide register) is described.

The multiplication and division result register (MDH/MDL) consists of 32-bit.

Figure 3.3-7 shows the bit configuration of the multiplication and division result register (MDH/MDL).

Figure 3.3-7 Multiplication and division result storage register (MDH/MDL)



MDH and MDL are the multiplication and division register. Each register is 32-bit long.

The initial value by reset is irregular.

● Functions when multiplication is executed

In the case of 32-bit x 32-bit multiplication, the 64-bit long operation results are stored in the multiplication/division results register in the following format.

- MDH: Higher 32-bit
- MDL: Lower 32-bit

In the case of 16-bit x 16-bit multiplication, the results are stored in the multiplication/division results register as follows.

- MDH: Indeterminate
- MDL: 32-bit

● Functions when division is executed

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

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

- MDH: surplus
- MDL: commerce

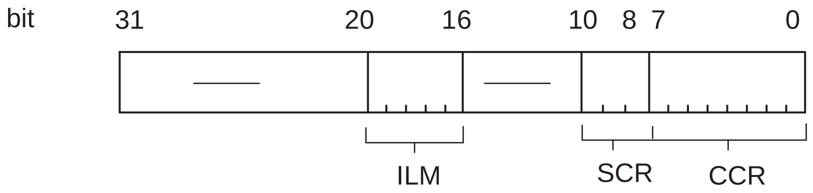
3.3.1 Program Status Register (PS)

The register retains the program status, and is separated into three parts, ILM, SCR, and CCR.
A bit undefined in figure is reservation all bit. When the register is read, "0" is always read. Writing is invalid.

■ Program Status Register (PS)

Figure 3.3-8 shows the configuration of the program status register (PS).

Figure 3.3-8 Program status register (PS)



● Condition code register (CCR)

Figure 3.3-9 shows the configuration of the condition code register (CCR).

Figure 3.3-9 Condition code register (CCR)

bit	7	6	5	4	3	2	1	0	Initial value
			S	I	N	Z	V	C	--00XXXX$_H$

The function of each bit is explained as follow.

[bit5] S: Stack flag

This bit specifies the stack pointer to be used as R15.

Value	Function
0	The system stack pointer (SSP) is used as R15. When the EIT occurs, this bit is automatically set to "0". (Note that a value saved on the stack is the value before it is cleared)
1	The user stack pointer (USP) is used as R15.

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

Select the SSP when the RETI instruction is executed.

[bit4] I: Interrupt enable flag

Permission and the interdiction of the user interruption demand are controlled.

Value	Function
0	User interruption interdiction. When the INT instruction is executed, this bit is cleared to "0". (Note that a value saved on the stack is the value before it is cleared)
1	User interruption permission. Mask processing of user interrupt requests is controlled by the value retained by ILM.

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

[bit3] N: Negative flag

This bit indicates the code when the operation results are defined as integers expressed as complements of 2.

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

Initial state by reset is irregular.

[bit2] Z: Zero flag

This bit indicates whether or not the operation result is 0.

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

Initial state by reset is irregular.

[bit1] V: Overflow flag

This bit is considered to be the integer expressing the operand used for operations as complements of 2, and indicates whether or not an overflow was generated as the result of such operation.

Value	Function
0	It is indicated that no overflow has occurred as a result of the operation.
1	It is indicated that an overflow has occurred as a result of the operation.

Initial state by reset is irregular.

[bit0] C: carrying flag

This bit indicates whether or not carry or borrow was generated from the highest bit through the operation.

Value	Function
0	It is indicated that no carry or borrow has occurred.
1	It is indicated that a carry or borrow has occurred.

Initial state by reset is irregular.

- System Condition code Register (SCR)

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

Figure 3.3-10 System condition code register (SCR)

bit	10	9	8	Initial value
	D1	D0	T	XX0$_H$

Each bit function of the system condition code register (SCR) is explained as follows.

[bit10, 9] D1, D0: Flag for step division

The middle data of step division execution time is maintained. Do not change while executing the division processing.

Restart of the step division is guaranteed by saving/returning the PS register value when other processes are carried out during the step division. Initial state by reset is irregular.

When the DIV0S instruction is executed, the divided and the divisor are referenced and set.

When the DIV0U instruction is executed, the bits clear forcibly.

[bit8] T: step trace trap flag

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

Value	Function
0	Step trace trap invalidity
1	Step trace trap effective In this case, all NMIs for user and user interrupts will be interrupt disabled.

This bit is initialized to "0" by a reset.

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

● Interrupt level mask register (ILM)

Figure 3.3-11 shows the configuration of the interrupt level mask register (ILM).

Figure 3.3-11 Interrupt level register (ILM)

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

The interrupt level mask register (ILM) retains the interrupt level mask value, and values retained by this ILM are used as the level mask.

Interrupt requests are received only when the supported interrupt level is higher than the level indicated by this ILM out of the interrupt requests to be input to the CPU.

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

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

If the original value is between 16 to 31, the new value must be between 16 to 31. When the command setting 0 to 15 is executed, the value (specified value + 16) is transferred.

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

This register is initialized to 15 (01111_B) by a reset.

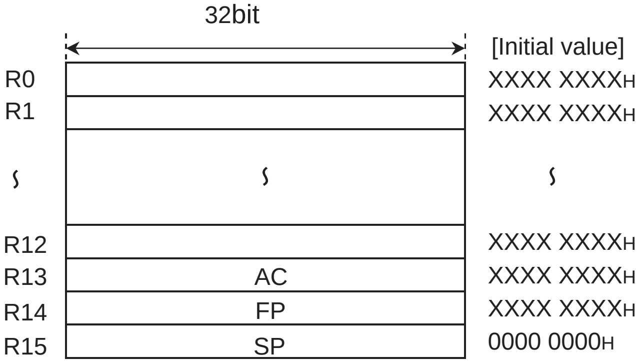
3.4 General-purpose Register

Register R0 to R15 is a general-purpose register. These registers are used as the pointer for memory access and accumulator for various operations.

■ General-purpose Register

Figure 3.4-1 shows the configuration of the general-purpose register.

Figure 3.4-1 Configuration of general-purpose register



Register R0-R15 is a general register. These registers are used as the pointer for memory access and accumulator for various operations. The following of the 16 registers are expected to have special uses, so some commands are emphasized.

- R13: Virtual accumulator (AC)
- R14: Frame pointer (FP)
- R15: Stack pointers (SP)

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

3.5 Data Construction

The data allocation of the FR20 series uses as follow.

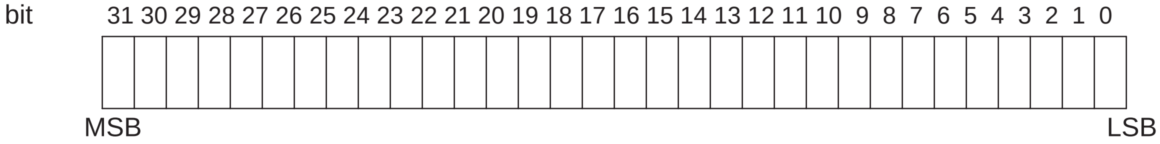
- **Bit ordering: Little endian**
- **Byte ordering: Big endian**

■ Bit Ordering

In the FR20 series, the little endian has been adopted as a bit ordering.

Figure 3.5-1 shows the data allocation of the bit ordering.

Figure 3.5-1 Data allocation of bit ordering

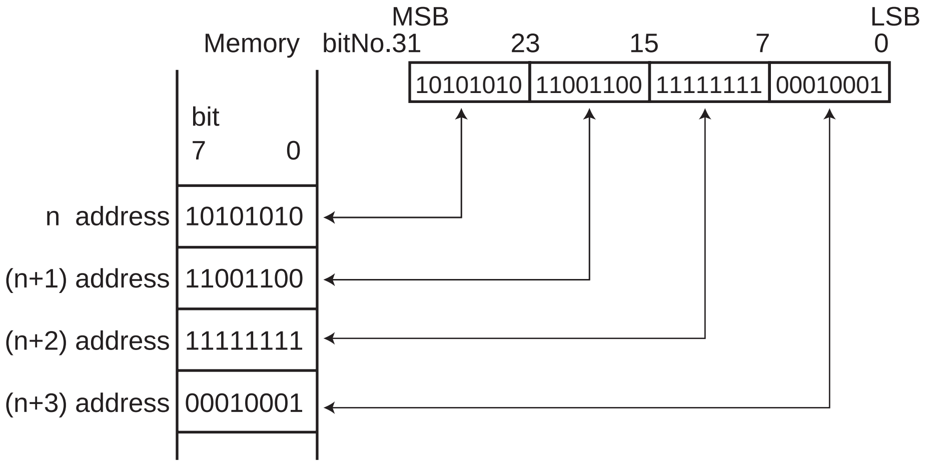


■ Byte Ordering

Big endian is adopted as the byte ordering for the FR family.

Figure 3.5-2 shows the data allocation of the byte ordering.

Figure 3.5-2 Data allocation of byte ordering



3.6 Word Alignment

As commands and data are accessed per byte, addresses to be allocated differ depending on the command length and data width.

■ Program Access

It is necessary to arrange the program of the FR20 series in the address of the multiple of two. Bit 0 of the program counter (PC) is set to "0" when updating the PC in line with execution of the command. It may be "1" only when an odd address is specified as the branch destination address. Even in that case, bit 0 is invalid, and the command must be placed in the address of the multiple of two.

There is no odd number address exception.

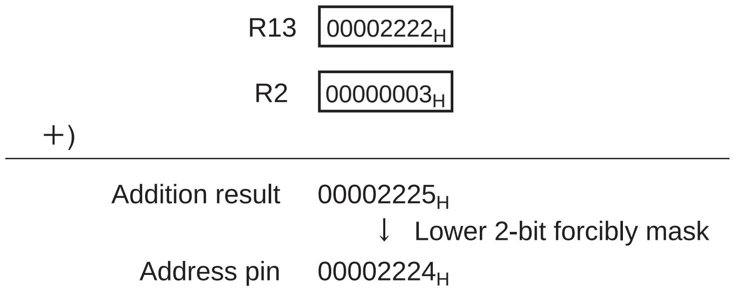
■ Data Access

In the FR20 series, when data is accessed, forced alignment is applied to the address depending on its width as follows.

- Word Access: An address must be a multiple of 4. (The lowest 2-bit is forcibly "00".)
- Half-word access: An address must be a multiple of 2. (The lowest bit is forcibly "0".)
- Byte Access: -

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

(example) LD @(R13,R2),R0



3.7 Memory Map

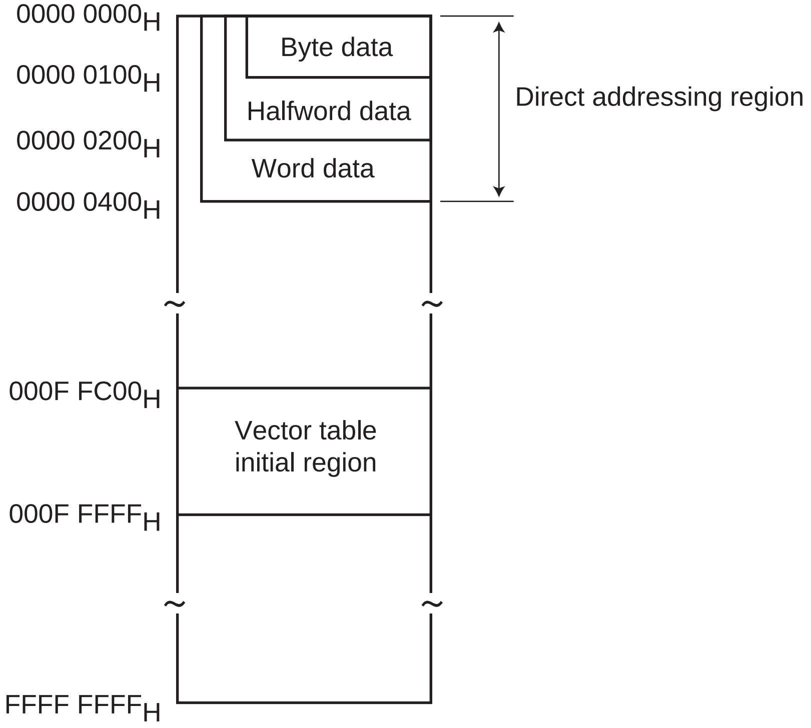
The memory map for the FR20 series is shown.

■ Memory Map

The address space of the memory is 32 bit linear.

Figure 3.7-1 shows the memory map.

Figure 3.7-1 Memory map



● Direct addressing

The under-mentioned region of the address space is a region for I/O. In this area, the operand address can be specified directly within the command through direct addressing.

The size of the address area which an address can be directly specified is different in each data length.

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

● Vector table initial region

The region of $000FFC00_H$ - $000FFFFF_H$ is EITT vector table initial area.

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

3.8 Overview of Instructions

The FR20 series supports logical operation and bit operations that are optimized for embedded application, and direct addressing commands as well as a general RISC command system. The set list shows the appendix. As each command is 16 bits length (some commands are 32 or 48 bits), memory usage is more efficient.

■ Overview of Instructions

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

- Arithmetic operation
- Load and store
- Divergence
- Logical operation and bit operation
- Direct addressing
- Others

● Arithmetic operation

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

Furthermore, it has 32-bit x 32-bit and 16-bit x 16-bit multiplication commands, and the 32-bit/32-bit step division commands.

Also equipped with an immediate transfer command that sets immediate values to the register, and an inter-register transfer command.

The arithmetic operation command executes all operations using a general-purpose register and multiplication/division register within the CPU.

● Load and store

Load and store are the commands that read and write to external memories. They are also used to read and write the peripheral circuits (I/O) within the chip.

Load and store have three types of access length, namely byte, half-word, and word. In addition to normal indirect register memory addressing, memory addressing is also possible for certain commands such as indirect displacement registers and indirect increment/decrement registers.

● Branch

In the FR20 series, whether the operations are with or without delay slots can be specified for the branch command.

It is an instruction of the branch, the call, the interruption, and the return. There are two types of branch command. One type has a delay slot and the other does not. They can be optimized to suit the purpose.

Refer to "3.8.1 Branch Command with Delay Slot" and "3.8.2 Branch Command without Delay Slot" for details of the branch commands.

- Logical operation and bit operation

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

- Direct addressing

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

- Others

  This command executes flag set up within the PS register, stack operation, code/zero expansion, etc. Also equipped with high-level language supported function entry/exit, and register multi-load/store commands.

3.8.1 Branch Command with Delay Slot

The operation with the delay slot branches prior to execute the command at the branch destination after executing the command located immediately after the branch command.

■ Branch Command with Delay Slot

The following commands execute the branch command with delay slot.

JMP:D	@Ri	CALL:Dlabel12		CALL:D	@Ri	RET:D	
BRA:D	label9	BNO:D	label9	BEQ:D	label9	BNE:D	label9
BC:D	label9	BNC:D	label9	BN:D	label9	BP:D	label9
BV:D	label9	BNV:D	label9	BLT:D	label9	BGE:D	label9
BLE:D	label9	BGT:D	label9	BLS:D	label9	BHI:D	label9

■ Operation of Branch Command with Delay Slot

Operations with delay slots branch out after executing the command placed just after the branch command (called a "delay slot") prior to execute the branch destination command.

As the delay slot command is executed before the branch operation, the apparent execution speed is 1 cycle. The NOP command must be placed as an alternative if an effective command cannot be inserted in the delay slot.

[example]

```
                          ; Row of instruction
   ADD      R1, R2        ;
   BRA:D    LABEL         ; Branch instructions
   MOV      R2, R3        ; Delay slot ......Executed before branch
      ...
   LABEL:STR 3 and @R4    ; The divergence ahead
```

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

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

Concrete examples are shown below.

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

[example]

```
   LDI:32   #Label, R0
   JMP:D    @R0           ; Branches out to Label
   LDI:8    #0,    R0     ; No effects on the branch destination address
   ...
```

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

[example]

```
RET:D                ; Branch to address defined beforehand in RP
MOV     R8,     RP   ; No effect on the return operation
...
```

- The flag referred by the Bcc: Drel command is not effected by the delay slot command either.

[example]

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

- When RP is referred to for the command within the delay slot under the CALL:D command, the updated contents will be read by the CALL:D command.

[example]

```
CALL:D  Label        ; Update RP and branch
MOV RP and R0        ; RP of an execution above-mentioned CALL:D result is forwarded.
..
```

■ Limitations for Branch Command with Delay

- Instruction that can be placed in the delay slot

Only commands that satisfy the following conditions can be executed within the delay slot.

- 1 cycle instruction.
- No branch instruction.
- Instruction which does not influence operation even when order changes

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

- Step trace trap

Step trace trap will not be generated between execution of the branch command with delay slot and the delay slot.

- Interrupt/NMI

No interrupt/NUM is received between execution of the branch command with delay slot and the delay slot.

- Undefined instruction exception

If an undefined command exists with the delay slot, undefined command exception will not be generated. At this time, undefined instruction operates as NOP instruction.

3.8.2 Branch Command without Delay Slot

Branch Command without Delay Slot is described.

■ Branch Command without Delay Slot

The following commands execute the branch command without delay slot.

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

■ Operation of Branch Command without Delay Slot

Operation of ones without delay slots are absolutely executed in command rank order. The instruction provided immediately before the branch instruction is not executed after branching.

[example]

```
                           ; Row of instruction
    ADD      R1, R2        ;
    BRA       LABEL        ; Branch instruction (delay slot none)
    MOV      R2, R3        ; Not executed
    ...
    LABEL:STR 3 and @R4        ; The branch ahead
```

Execution cycle number for branch commands without delay slots will be 2 cycles branched, or 1 cycle non-branched. As the appropriate command cannot be inserted into the delay slot, the command code efficiency is better than the branch command with the delay slot described the NOP instruction. A balance between execution speed and code efficiency can be struck by selecting either the operation with the delay slot when effective commands can be set in the delay slot, or the operation without the delay slot when effective commands cannot be set.

3.9 EIT (Exception, Interruption, and Trap)

EIT indicates suspension of program execution due to generation of an event while executing the current and other programs. It is a general term for Exception, Interrupt, and Trap.

■ EIT (Exception, Interruption, and Trap)

The exception is an incident which occurs in relation to the context under execution. Execution restarts from the instruction that caused the exception.

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

The trap is an incident which occurs in relation to the context under execution. There is something directed by the program like the system call. Execution restarts from the instruction following the one that caused the trap.

■ EIT Factor

The EIT factors are as follow.

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

■ Return from EIT

Use the RETI instruction to return from EIT.

3.9.1 Interrupt Level of EIT

Interrupt levels is controlled by 0 to 31 by five bits.

■ Interrupt level of EIT

The allocation of each level is as follow.

Table 3.9-1 Interrupt level

Interrupt level		Factor	Remark
2 decimal number	10 decimal number		
00000	0	(System reservation)	When the original value of the ILM is 16 to 31, values within this range cannot be set as the ILM by the program.
00001	1		
00010	2		
00011	3		
00100	4	INTE instruction, step trace trap	
00101 to 01110	5 to 14	(System reservation)	
01111	15	NMI (for user)	
10000 to 11110	16 to 30	Interrupt	When ILM is set, it is a user interruption interdiction.
11111	31	-	When ICR is set, it is an interruption interdiction.

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

Undefined command exception, coprocessor absence trap, coprocessor error trap and INT instruction are unaffected by the interrupt level. Moreover, ILM is occasionally changed.

■ Level Mask to Interruption/NMI

When NMI and interrupt requests are generated, the interrupt level held by the interrupt factor is compared with the level mask value held by the ILM. And, when the following condition consists, the mask is done, and the demand is not accepted.

Interrupt level held by factor ≥ Level mask value

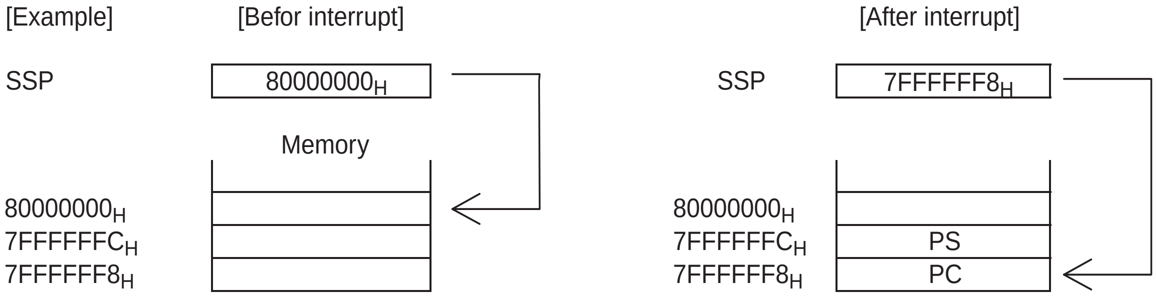
3.9.2 Interrupt Stack Operation

The value of PC and PS is saved and revived in the area shown by SSP. After an interrupt, PC is stored in the address indicated by the SSP, and PS is stored in the address (SSP+4).

■ Interrupt Stack

Figure 3.9-1 shows the example of the interrupt stack.

Figure 3.9-1 Interrupt stack operation diagram



3.9.3 EIT Vector Table

From address which TBR shows to vector region for EIT region of 1KB

■ EIT Vector Table

Each vector is 4 bytes, and the relationship between the vector number and vector address is expressed below.

vctadr = TBR + vctofs = TBR +(03FCH - 4 × vct)

vctadr: Vector Address

vctofs: Vector offset

vct: Vector number

The lowest 2-bit of the addition result are always handled as "00".

The area of $000FFC00_H$ to $000FFFFF_H$ is an initial area of the vector table by reset. A special function is allocated partially of the vector. The vector table on the architecture is shown in Table 3.9-2 .

Table 3.9-2 Vector table

Vector number		Vector Address	Description
0	00_H	$000FFFFC_H$	Reset
1	01_H	TBR + $03F8_H$	System reservation
2	02_H	TBR + $03F4_H$	System reservation
3	03_H	TBR + $03F0_H$	System reservation
4	04_H	TBR + $03EC_H$	System reservation
5	05_H	TBR + $03E8_H$	System reservation
6	06_H	TBR + $03E4_H$	System reservation
7	07_H	TBR + $03E0_H$	Coprocessor absent trap
8	08_H	TBR + $03DC_H$	Coprocessor error trap
9	09_H	TBR + $03D8_H$	INTE Instruction
10	$0A_H$	TBR + $03D4_H$	Instruction break exception
11	$0B_H$	TBR + $03D0_H$	Operand break trap
12	$0C_H$	TBR + $03CC_H$	Step trace trap
13	$0D_H$	TBR + $03C8_H$	System reservation
14	$0E_H$	TBR + $03C4_H$	Undefined instruction exception
15	$0F_H$	TBR + $03C0_H$	NMI (for user)
16	10_H	TBR + $03BC_H$	Mask enable interrupt factor #0 (IRQ0)
17 to 63	11_H to $3F_H$	TBR + $03B8_H$ to TBR + 0300_H	Mask enable interrupt factor #1 (IRQ2) to Mask enable interrupt factor #47 (IRQ47)
64	40_H	TBR + $02FC_H$	System reservation (used for REALOS)
65	41_H	TBR + $02F8_H$	System reservation (used for REALOS)
66 to 255	42_H to FF_H	TBR + $02F4_H$ to TBR + 0000_H	INT Instruction

3.9.4 Multiple EIT Processing

When a number of EIT factors are generated simultaneously, one of the EIT factors is selected and accepted in the CPU, and after the EIT sequence is executed, such EIT factors are detected again. This operation is repeated as necessary.
When EIT factors are detected, if there are no more EIT factors that can be accepted, the handler command for the last EIT factor accepted will be executed.
Therefore, the handler order for each factor when a number of EIT factors are generated simultaneously is determined by the following two elements.

- **Priority level of EIT factor acceptance**
- **How to mask the other factor at receiving**

■ Priority Level of EIT Factor Acceptance

Priority for acceptance of the EIT factors indicates the order when selecting the factors executing the EIT sequence that saves PS and PC, updates the PC (on demand) and performs mask processing of other factors. The handler of the factor previously accepted is not always executed first.

Table 3.9-3 shows the priority level of the EIT factor acceptance.

Table 3.9-3 Priority level of EIT factor acceptance and masking other factor

Priority order of acceptance	Factor	Masking of other factor
1	Reset	Other factors are abandoned.
2	Undefined instruction exception	Cancellation
3	INT Instruction	I Flag=0
	Coprocessor absent trap	None
	Coprocessor error trap	
4	User Interrupt	ILM = Level of accepted factor
5	NMI (for user)	ILM=15
7	INTE Instruction	ILM=4
8	Step trace trap	ILM=4

A tinge of mask processing onto other factors after accepting the EIT factors is added, and each handling procedure for the generated EIT factors are also mentioned in Table 3.9-4 .

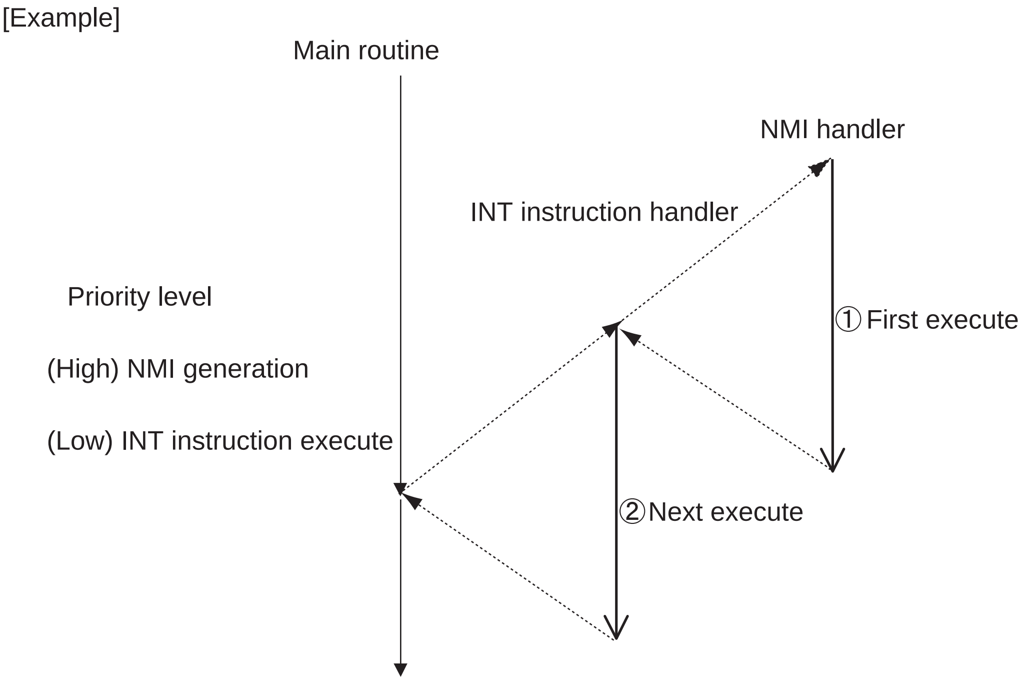
Table 3.9-4 Execution sequence of EIT handler

Execution sequence of handler	Factor
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 factors are abandoned.
*2: If the INTE instruction is executed in steps, only EIT of the step trace trap generates INTE instruction.
The factor by INTE is disregarded.

Figure 3.9-2 shows the example of multiple EIT processing.

Figure 3.9-2 Example of multiple EIT processing



3.9.5 Operation of EIT

This section explains operation of EIT

■ Operation of EIT

As per the following explanation, the "PC" at the transfer origin indicates the address of the command that detected each EIT factor.

The "following command address" means that the command that detected EIT is as follows.

- If LDI is 32: PC+6
- If LDI: 20 and COPOP, COPLD, COPST and used: PC+4
- Other instructions: PC+2

● Operation of user interruption/NMI

When a user interrupt or interrupt request of NMI for user is generated, the request is accepted or not is determined in the following order.

● Right or wrong judgment of interruption demand acceptance

1. The interruption levels of requests that are generated simultaneously are compared, and the one with the highest level (the smallest numeric value) will be selected. In terms of the levels used for comparisons, the value retained by the supported ICR is used for mask enable interrupts, and a predetermined constant is used for NMI.
2. When a number of interrupt requests of the same level are generated, the one with the youngest interrupt number is selected.
3. Compares the interrupt level with the selected interrupt request with the level mask value determined by ILM.

   When the interrupt level is greater than or equal to the level mask value, the interrupt request is masked and not accepted.

   To (4) at interrupt levels < level mask value.
4. When the selected interruption request is an interruption that can be masked, the interruption request will be masked and will not be accepted when the I flag is 0. If the I flag is 1, go to 5). When the selected interrupt request is NMI, go to 5) regardless of the I-flag value.
5. When the above condition occurs, the interrupt request is accepted at the command-processing gap.

When an EIT request is detected, if the user interrupt/NMI request is accepted, the CPU operates as follows using the interrupt number supporting the accepted interrupt request. ( ) in the [operation] shows the address which the register indicates.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. Address of the following instruction→(SSP)
5. Interrupt level of accepted request→ ILM
6. "0" → S Flag
7. (TBR + Vector offset of accepted interrupt request)→ PC

Prior to execute the front command of the handler after the interrupt sequence ends, a new EIT is detected. At this stage, if an acceptable EIT is generated, the CPU transits to the EIT processing sequence.

■ Operation of INT Instruction

The INT #u8 instruction operates as follow.

Branch to the interrupt handler for the vector indicated by u8.

[Operation]

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

■ Operation of INT Instruction

The INT instruction operates as follow.

Branch to the interrupt handler for the vector indicated by vector number #9.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. PC+2 → (SSP)
5. "00100" → ILM
6. "0" → S Flag
7. (TBR+3D8H)→ PC

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

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

■ Operation of Step Trace Trap

When the T flag of the SCR within the PS is set, and the step trace function is set to Enabled, a trap is generated per command execution and creates a break.

● Condition of step trace trap detection

- T Flag = 1
- There is no delayed branch instruction.
- While executing other than the INTE command and step trace trap processing routine.

When the above condition occurs, breaks at the command operation gap.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. Address of the following instruction→(SSP)
5. "00100" → ILM
6. "0" → S Flag
7. (TBR+3CCH)→ PC

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

Moreover, EIT by the INTE instruction is not generated.

■ Operation of Undefined Instruction Exception

If an undefined instruction is detected when decoding the command, an undefined instruction exception is generated.

● Detection condition of undefined instruction exception

- 1) It is detected that it is undefined instruction at the decoding of the instruction.
- 2) Placed outside the delay slot (not immediately after the delay branch command).

When the above condition occurs, an undefined instruction exception is generated, and a break is created.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. PC → (SSP)
5. "0" → S Flag
6. (TBR+3C4H)→ PC

The address of the actual command that detected the undefined instruction exception is saved as the PC.

■ Coprocessor Absent Trap

When a coprocessor command using an unmounted coprocessor is executed, a coprocessor absence trap is generated.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. Address of the following instruction→(SSP)
5. "0" → S Flag
6. (TBR+3E0H)→ PC

■ Coprocessor Error Trap

If an error occurs while using a coprocessor, when the coprocessor command that operates the coprocessor is executed next, a coprocessor error trap is generated.

[Operation]

1. SSP-4 → SSP
2. PS → (SSP)
3. SSP-4 → SSP
4. Address of the following instruction→(SSP)
5. "0" → S Flag
6. (TBR+3DCH)→ PC

Note:

This product does not contain the coprocessor.

■ Operation of RETI Instruction

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

[Operation]

1. (R15) → PC
2. R15+4 → R15
3. (R15) → PS
4. R15+4 → R15

Care must be taken that the stack pointer to be referred to returning of the PS and PC is selected in accordance with the S flag contents. When the command that operates R15 (stack pointer) within the interrupt handler is executed, set the S flag to "1" and use the USP as R15. The S flag must be returned to "0" before the RETI command.

■ Delay Slot

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

Refer to "3.8.1 Branch Command with Delay Slot" for details.

3.10 Reset Sequence

This section explains the reset when the CPU is the operation state.

■ Reset Factor

The reset factor is as follow.

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

■ Initialization by Reset

The reset factor is generated, the CPU is initialized.

● Releasing from the external reset pin or software reset

- The pin is set to the predetermined state.
- Each resource in the device is put in the reset state. The control register is initialized to the predetermined value.
- The lowest gear is selected for the clock.

■ Reset Sequence

When a reset factor is released, the CPU executes the following reset sequence.

- $(000FFFFC_H) \rightarrow PC$

Note:

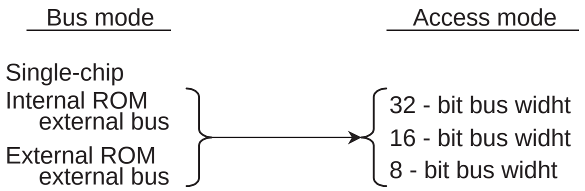
After reset, operation mode needs to be set by the mode register setting.

3.11 Memory Access Mode

In the FR20 series, operation mode is controlled by the mode pins (MD2, 1, 0) and the mode register (MODR).

■ Operation Mode

In the operation mode, there are a bus mode and an access mode.



● Bus Mode

The bus mode controls the internal ROM operation and external access function operation, and is specified by the mode set up pins (MD2, 1, 0) and M1, M0 bit of the mode register (MODR).

● Access mode

The access mode controls the external data bus width, and is specified by the mode set up pins (MD2, 1, 0) and BW1, 0 bits within the AMD0/AMD1/AMD32/AMD4 and AMD5 (address mode registers).

■ Mode Pin

Operation is specified by three pins (MD2, 1, 0) as per Table 3.11-1 .

Table 3.11-1 Mode Pin and setting mode

Mode pin			Mode name	Reset vector access area	External data bus width	Remark
MD2	MD1	MD0				
0	0	0	External vector mode 0	External	8 bits	The use of the terminal is prohibited.
0	0	1	External vector mode 1	External	16 bit	
0	1	0	External vector mode 2	External	32 bit	
0	1	1	Internal vector mode	Internal	(Mode register)	Single-chip mode
1	-	-	-	-	-	The use of the terminal is prohibited.

■ Mode Data

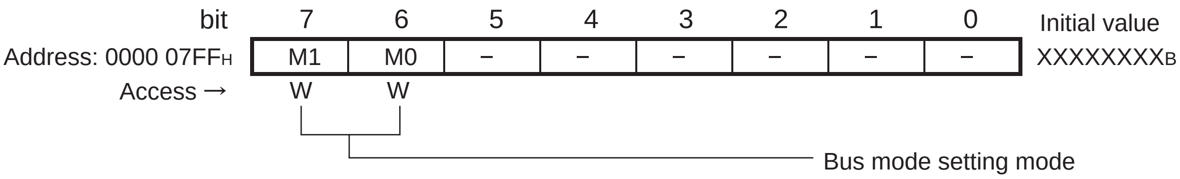
Data that the CPU writes at "0000 07FF$_H$" after reset is called mode data.

The mode register (MODR) exists in "0000 07FF$_H$" and after setting to this register, operation is carried out under the set up mode of this register. The mode register can be written only once after resetting.

The set value of this register is valid immediately after writing.

■ Mode Register (MODR)

Figure 3.11-1 Mode register (MODR)



[bit7, 6]: M1, M0

There are bus mode setting bits. These bits specify the bus mode after writing the mode register.

Table 3.11-2 Bus mode setting bits and function

M1	M0	Function	Remark
0	0	Single-chip mode	
0	1	Internal ROM external bus mode	
1	0	External ROM external bus mode	Setting disabled
1	1	-	Setting disabled

Note:

Only set the "00" and "01" for this product above.

[bit5-0]:-

System Reserved bit

Always write "0" to these bits.

■ Notes on Writing to Mode Register (MODR)

Before writing to the MODR, AMD0 to AMD5 must be set, and the bus width in each chip select (CS) area must be decided.

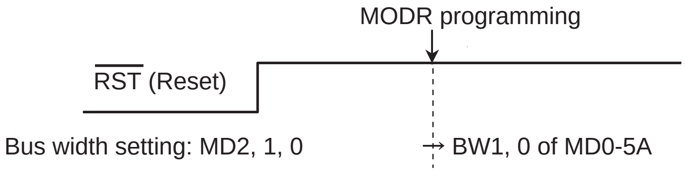
The MODR has no bits used to set the bus width.

For the bus width, before writing to the MODR, the mode pins (MD2 to 0) are valid. After MODR writing, set values (BW1, BW0) of the AMD0 to AMD5 will be valid.

For example, the external reset vector performs in the normal area 0 (the area in which the CSOX is active), but in this case, the bus width is decided by the MD2 to MD0 pins. The bus width at that time is set to 32 or 16 bits by MD2 to MD0, and if MODR is written while nothing is set to AMD0, the initial value of the AMD0 bus width has been set to 8 bits. So bus operation will be performed by transiting the area 0 to 8 bit bus mode after writing the MODR, and erroneous operation will result.

AMD0 to AMD5 must be set before writing the MODR to prevent this kind of problem.

Figure 3.11-2 Notes on writing to mode register (MODR)



3.12 Clock Generation Section (Low Power Consumption Mechanism)

The clock generation section is the modules that have the following functions:

- **CPU clock generation (including the gear function)**
- **Peripheral clock generation (including the gear function)**
- **Reset generation and cause retention**
- **Standby function**
- **Built-in PLL (duty correction circuit)**

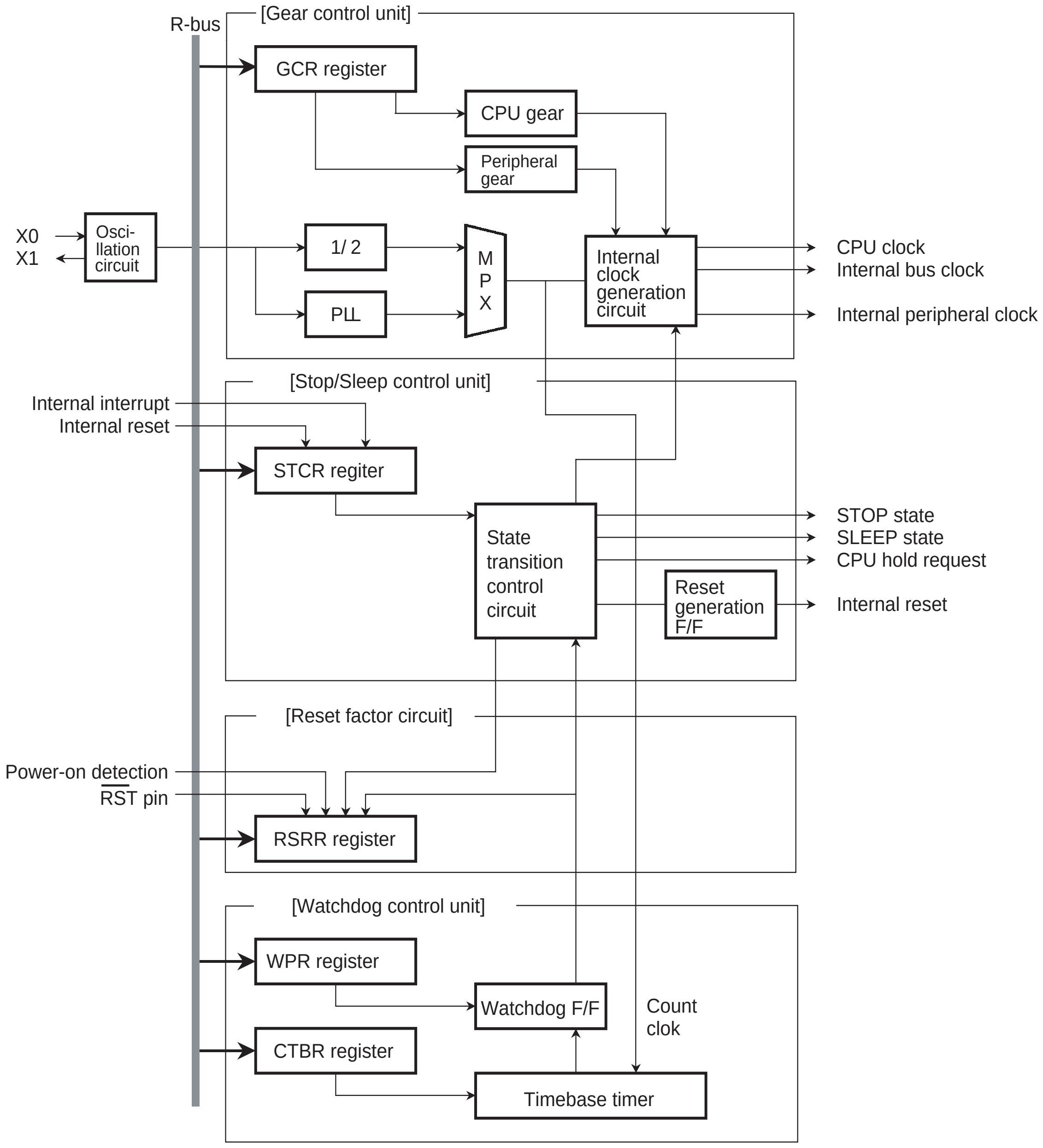
■ Register in Clock Generation Section

Figure 3.12-1 Register in Clock Generation Section

bit	7 - - - - - - - - - - - - - - - - - - 1	
Address: 000480_H	RSRR/WTCR	Reset Factor/Watchdog Cycle Control Register
000481_H	STCR	Standby Control Register
000482_H	Reserve	(Access interdiction)
000483_H	CTBR	Timebase Timer Clear Register
000484_H	GCR	Gear Control Register
000485_H	WPR	Watchdog Reset Generation Dalayed Register

■ Block Diagram of Clock Generation Section

Figure 3.12-2 Block Diagram of Clock Generation Section



3.12.1 Reset Factor Register (RSRR) and Watchdog Timer Cycle Control Register (WTCR)

The reset factor register (RSRR) retains reset types generated, and the watchdog timer cycle control register (WTCR) specifies the cycle for the watchdog timer.

■ Reset Factor Register (RSRR) and Watchdog Timer Cycle Control Register (WTCR)

Figure 3.12-3 Reset factor register (RSRR) and watchdog timer cycle control register (WTCR)

bit	7	6	5	4	3	2	1	0	Initial value after Power ON
Address: 000480H	PONR	−	WDOG	ERST	SRST	−	WT1	WT0	1XXX XXXXB
Access →	R	−	R	R	R	−	W	W	

[bit7]: PONR

When this bit is "1", it indicates that the previously generated reset was a power-on reset. When this bit is "1", contents other than this bit of this register are invalid.

[bit6]: (Reserved)

It is reserved bit. The reading value is undefined.

[bit5]: WDOG

When this bit is "1", it indicates that the previously generated reset was a watchdog reset.

[bit4]: ERST

When this bit is "1", it indicates that the previously generated reset was caused by the external reset pin.

[bit3]: SRST

When this bit is "1", it indicates that the previously generated reset was caused by the software reset request.

[bit2]: (Reserved)

It is reserved bit. The reading value is undefined.

[bit1, 0]: WT1, 0

These bits specify the cycle of the watchdog timer. The bits and the selected cycle have the following relationship. These bits are initialized when the entire reset is generated.

Table 3.12-1 Watchdog timer cycle specified by WT1 and WT0

WT1	WT0	Writing spacing to at least necessary for control generation of watchdog reset WPR	Timer from last $5A_H$ write to WPR to occurrence of watchdog resetting
0	0	$\phi \times 2^{15}$ (Initial value)	$\phi \times 2^{15}$ 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 $\phi \times 2^{22}$

Note: ϕ is twice as large as X0 when GCR CHC is 1, and is one time as large as X0 when GCR CHC is 0.

3.12.2 Standby Control Register (STCR)

This register controls standby operations and specifies the oscillation stabilization wait time.

■ Standby Control Register (STCR)

Figure 3.12-4 Standby control register (STCR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000481_H	STOP	SLEEP	–	SRST	OSC1	OSC0	–	–	000111--_B
Access →	R/W	R/W	–	R/W	R/W	R/W	–	–	

[bit7]:STOP

When "1" is written to this bit, the status will be stop that stops the internal peripheral clock, the internal CPU clock, and oscillation.

[bit6]:SLEP

When "1" is written to this bit, the status will be standby that stops the internal CPU clock. If "1" is written to both the STOP bit and this bit, the STOP bit is handled as the priority, so the status will be stop.

[bit5]:(Reserved)

[bit4]:SRST

When "0" is written to this bit, a software reset request is generated.

[bit3, 2]:OSC1, 0

These bits specify the oscillation stabilization wait time. These bits and selected cycle have the following relationship. This bit is initialized by a power-on reset, and is unaffected by any other reset factors.

Table 3.12-2 Oscillation stabilization wait time specified by 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)

Note: ϕ is twice as large as X0 when GCR CHC is 1, and is one time as large as X0 when GCR CHC is 0.

[bit1, 0]:(Reserved)

There are reserved bits. The reading value is undefined.

3.12.3 Timebase Timer Clear Register (CTBR)

This register initializes the timebase timer contents to 0.

■ Timebase Timer Clear Register (CTBR)

Figure 3.12-5 Timebase timer clear register (CTBR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000483_H	D7	D6	D5	D4	D3	D2	D1	D0	$XXXX\ XXXX_B$
Access→	W	W	W	W	W	W	W	W	

[bit7 to 0]

Writing $A5_H$, $5A_H$ continuously to this register clears the timebase timer to 0 immediately after $5A_H$. The reading value of this register is irregular. There is no restriction on the time interval between $A5_H$ and $5A_H$ writing.

Note:

Clearing the timebase timer using this register temporarily fluctuates oscillation stability wait interval, watchdog cycle, and peripheral cycles that use the timebase.

3.12.4 Gear Control Register (GCR)

The gear control register controls the gear functions of the CPU and peripheral clocks.

■ Gear Control Register (GCR)

Figure 3.12-6 Gear Control Register (GCR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000484_H	CCK1	CCK0	−	−	PCK1	PCK0	−	CHC	$11\text{--}11\text{-}1_B$
Access→	R/W	R/W	−	−	R/W	R/W	−	R/W	

[bit7, 6]:CCK1, 0

These bits specify the CPU gear cycle. The bits and selected cycles have the following relationship. These bits are initialized by a reset.

Table 3.12-3 CPU machine clock

CCK1	CCK0	CHC	CPU machine clock (source oscillation: input frequency from X0)
0	0	0	Source oscillation × 1
0	1	0	Source oscillation × 1/2
1	0	0	Source oscillation × 1/4
1	1	0	Source oscillation × 1/8
0	0	1	Source oscillation × 1/2
0	1	1	Source oscillation × 1/2 × 1/2
1	0	1	Source oscillation × 1/2 × 1/4
1	1	1	Source oscillation × 1/2 × 1/8 (Initial value)

[bit3, 2]:PCK1, 0

These bits specify the peripheral gear cycle. The bits and the selected cycles have the following relationship. These bits are initialized by a reset.

Table 3.12-4 Peripheral machine clock

PCK1	PCK0	CHC	Peripheral machine clock (source oscillation: input frequency from X0)
0	0	0	Source oscillation × 1
0	1	0	Source oscillation × 1/2
1	0	0	Source oscillation × 1/4
1	1	0	Source oscillation × 1/8
0	0	1	Source oscillation × 1/2
0	1	1	Source oscillation × 1/2 × 1/2
1	0	1	Source oscillation × 1/2 × 1/4
1	1	1	Source oscillation × 1/2 × 1/8 (Initial value)

[bit0]:CHC

This bit selects whether 1/2 division cycle or PLL/DCC cycle of the oscillation circuit is used as the basic clock. The setting "1" is the 1/2 division cycle and the setting "0" is the PLL/DCC cycle.

3.12.5 Watchdog Reset Generation Delay Register (WPR)

This register clears the flip-flop for the watchdog timer. Using this register postpones generation of the watchdog reset.

■ Watchdog Reset Generation Delay Register (WPR)

Figure 3.12-7 Watchdog reset generation delay register (WPR)

bit	7	6	4	5	4	3	2	1	0	Initial value
Address: 000485_H	D7	D6	D5	D4	D3	D2	D1	D0		XXXX XXXX$_B$
Access→	W	W	W	W	W	W	W	W		

[bit7 to 0]

Writing $A5_H$, $5A_H$ continuously to this register clears the flip-flop for the watchdog timer to "0" immediately after $5A_H$, and postpones generation of the watchdog reset.

The reading value of this register is irregular. There is no time limit between $A5_H$ and $5A_H$, but if writing both data is not finished within the period as per the following table, a watchdog reset is generated. However, clearing is automatically carried out under stop/sleep mode, so when these conditions are generated, the watchdog reset is automatically postponed.

Table 3.12-5 Watchdog timer cycle specified by WT1 and WT2

WT1	WT0	Writing spacing to at least necessary for control generation of watchdog reset WPR	Time from $5A_H$ final writing in WPR to generation of 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 $\phi \times 2^{22}$

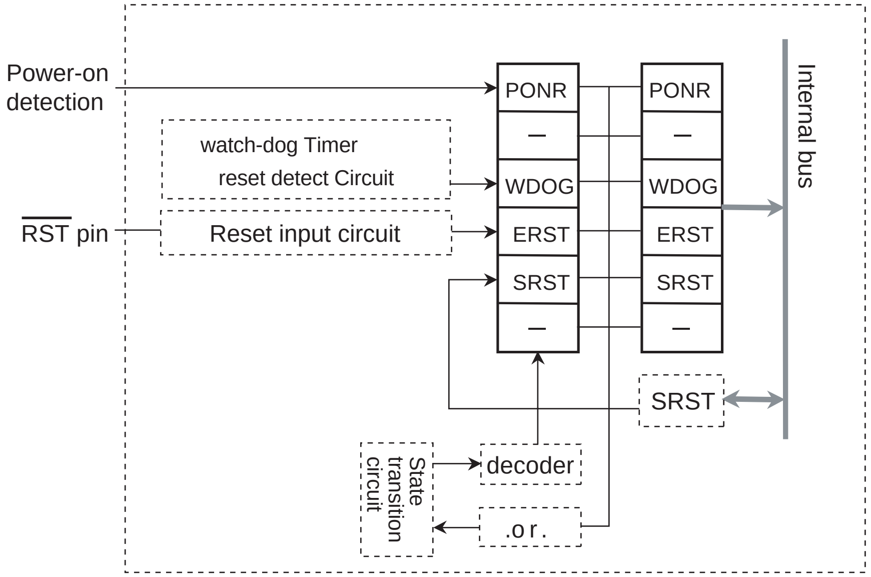
Note: ϕ is twice as large as X0 when GCR CHC is 1, and is one time as large as X0 when GCR CHC is 0.

3.12.6 Reset Factor Retention

The reset factor retention holds the factor of previous generation. All flag is cleared to "0" by reading.
Once a factor flag is set, it is not cleared unless the factor is read.

■ Block Diagram of Reset Factor Retention Circuit

Figure 3.12-8 Reset factor circuit Block Diagram



■ Setting of Reset Factor Retention

No special settings are required to use the rest factor function. Commands for reading the reset factor register and branching to the appropriate program shall be placed near the front of the program to be set in the reset entry address.

[example]

```
RESET-ENTRY
LDI:32 #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 WATCH DOG RESET?
BNE    WDOG-RESET
...
```

Notes:

- When the PONR bit is 1, contents of bits other than that should be handled as indeterminate. Thus, if a reset factor needs to be checked, the command for checking the power-on reset must be placed at the front.
- Checking reset factors other than power-on reset can be done in any position. Priorities are determined in the order of checking.

3.12.7 Stop Status

Stop status indicates the status that stops all internal clocks and oscillation circuit operation. It can be minimized the power consumption.

■ Overview of Stop Status

Stop status indicates the status that stops all internal clocks and oscillation circuit operation. It can be minimized the power consumption.

Transition to the stop status is performed as follow.

- Writing in standby control register (STCR) using the instruction

Returning from the stop status is performed one of the following.

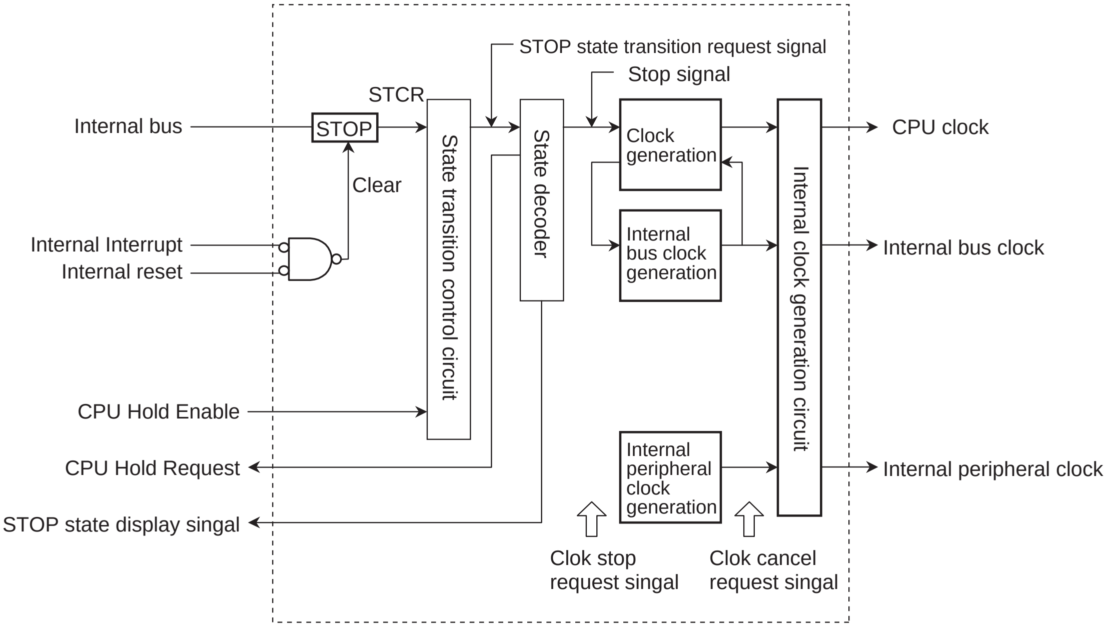
- Interrupt request (only applies to peripherals where interrupt request can be generated even under stop status)
- Applying the L level to the $\overline{\text{RST}}$ pin

Under stop status, all internal clocks are stopped, so built-in peripherals other than those that can generate interrupts for returning will be stopped.

■ Block Diagram of Stop Control Section

Figure 3.12-9 shows the block diagram of stop control section.

Figure 3.12-9 Block diagram of stop control section


STOP state transition request signal
Stop signal
STCR
Internal bus
STOP
Clear
Internal Interrupt
Internal reset
CPU Hold Enable
CPU Hold Request
STOP state display singal
State transition control circuit
State decoder
Clock generation
Internal bus clock generation
Internal peripheral clock generation
Internal clock generation circuit
CPU clock
Internal bus clock
Internal peripheral clock
Clok stop request singal
Clok cancel request singal

■ Transition to Stop Status

● Transition to the stop state using an instruction

Write "1" to bit 7 of the STCR register to enter stop status.

After a stop request is issued, the status is changed such that the CPU does not use the internal bus, and then the clocks are stopped in the following order.

CPU clock →internal bus clock → internal peripheral clock

The oscillation circuit stops when the internal peripheral clock stops.

Notes:

The following routine must be used to change the status to stop using a command.

- Before writing to the STCR, set the same value to the CCK1, 0 and PCK1, 0 bit lots of the GCR, and the gear ratios between the CPU system clock and peripheral system clock should be the same.
- Do not provide the stop state when the CHC bit of GCR is "0". To enter stop status, "1" must be set to the CHC bit of the GCR, and 1/2 division system clock must be selected.
- At least five consecutive NOP instructions must be provided immediately after writing to the STCR.

[Setting]

```
DI:8  #00000001b,R1  ; CPU=Peripheral gear ratio, CHC=1
LDI:32 #GCR,R2
STB    R1,@R2
LDI:8  #10010000b,R1 ; STOP=1
LDI:32 #STCR,R2
STB    R1,@R2
NOP             ;
NOP             ;
NOP             ;
NOP             ;
NOP             ;
```

■ Return by Stop Status

Returning from stop status can be performed by generating an interrupt or reset.

● Return by interrupt

If the interrupt-enabled bit attached to the peripheral function is valid, returns from the stop status by generating a peripheral interrupt.

Returning from the stop status to the normal operation status is carried out in the following procedure.

Generates interrupt -> Restarts oscillation circuit operation -> Waits for oscillation stabilization -> After stabilization, restarts supply of internal peripheral clock -> Restarts supply of internal bus clock -> Restarts supply of internal CPU clock

The program execution after oscillation stabilization waiting time is as follows.

- When the level of the interrupt is enabled by the I flag of CPU ILM

  - After saving the register, fetches in the interrupt vector and executes from the processing routine.
- When the level of the interrupt is disabled by the I flag of CPU ILM
  - Executes the instruction the following instruction that changed to the stop status.

● Return by $\overline{\text{RST}}$ pin

Returning from stop status to normal operating status is as per the following procedure.

L level application to the $\overline{\text{RST}}$ pin -> Generates an internal reset -> Restarts oscillation circuit operation -> Waits for oscillation stabilization -> After stabilization, restarts supply of internal peripheral clock -> Restarts supply of internal bus clock -> Restarts supply of internal CPU clock -> Fetches in the reset vector -> Restarts the command execution from the reset entry address

Notes:

- If an interrupt request has already been generated from a peripheral, status is not changed to stop, and writing is ignored.
- No internal clocks are supplied while waiting for oscillation stabilization except for power-on reset. For power-on reset, the internal status needs to be initialized, so all internal clocks are supplied.

3.12.8 Sleep Status

Sleep status indicates that the CPU clock and internal bus clock are stopped. Power consumption under the status where CPU operation is not required can be reduced in some extent.

■ Overview of Sleep Status

Sleep status indicates that the CPU clock and internal bus clock are stopped.

Power consumption under the status where CPU operation is not required can be reduced in some extent.

The transition to the sleep status is performed as follow.

- Writing in standby control register (STCR) using the instruction

Returning from the sleep status is performed one of the following.

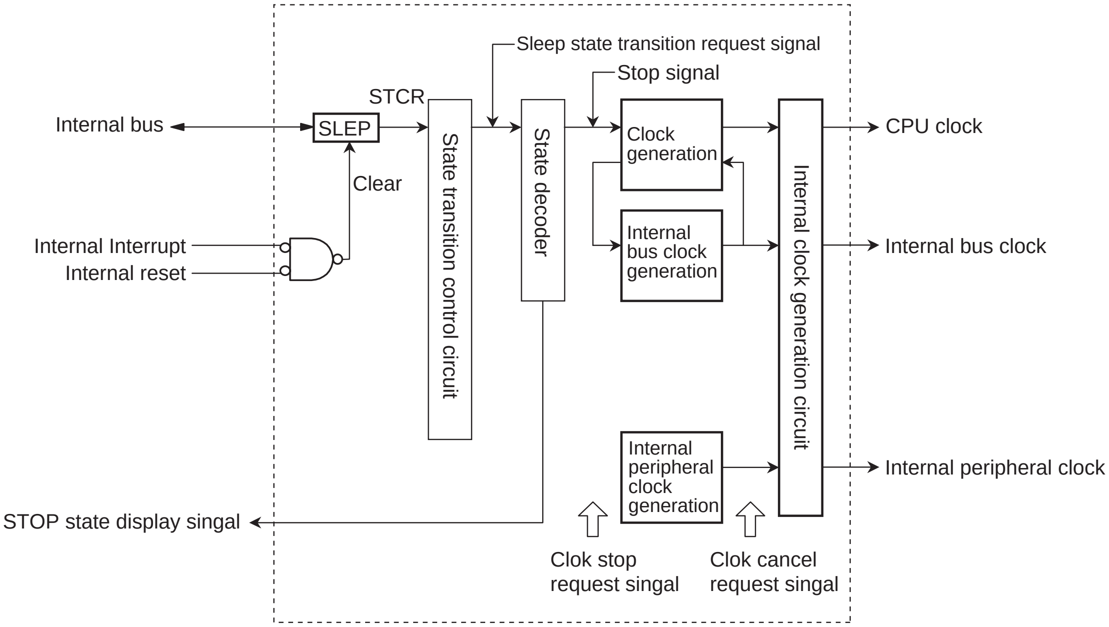
- Interrupt request
- Reset factor generation

Under sleep status, the peripheral clock operates, so reset is possible by interrupt of built-in peripheral.

■ Block Diagram of Sleep Control Section

Figure 3.12-10 shows the block diagram of the sleep control section.

Figure 3.12-10 Block diagram of sleep control section


Sleep state transition request signal
Stop signal
STCR
Internal bus
SLEP
Clear
Internal Interrupt
Internal reset
State transition control circuit
State decoder
Clock generation
Internal bus clock generation
Internal peripheral clock generation
Internal clock generation circuit
CPU clock
Internal bus clock
Internal peripheral clock
STOP state display singal
Clok stop request singal
Clok cancel request singal

■ Transition to Sleep Status

To enter sleep status, write "0" to bit 7 of the STCR register and "1" to bit 6.

Issues the sleep request, and then stops clocks in the following order once the status of the CPU is changed so that the internal bus is not used.

CPU clock → Internal bus clock

Notes:

The following routine must be used to change to sleep status.

- Before writing the STCR, set the same value to the CCK1, 0 and PCK1, 0 bit lots of the GCR, and the gear ratio between the CPU system clock and peripheral system clock should be the same.
- The CHC bit of GCR is can be any value.
- At least five consecutive NOP instructions must be provided immediately after writing to the STCR.

[Setting]

```
LDI:8  #11001100b,R1 ; CPU=Peripheral gear ratio (left example: oscillation x 1/8), CHC is option.
LDI:32 #GCR,R2
STB    R1,@R2
LDI:8  #01010000b,R1 ; SLEP=1
LDI:32 #STCR,R2
STB    R1,@R2
NOP             ;
NOP             ;
NOP             ;
NOP             ;
NOP             ;
```

■ Return by Sleep Status

Returning from sleep status can be done by generating an interrupt or reset.

● Return by interrupt

If the interrupt-enabled bit attached to the peripheral function is valid, returns from sleep status by generating a peripheral interrupt.

Returning from sleep status to normal operating status is as per the following procedure.

Generates interrupt -> Restarts supply of internal bus clock -> Restarts supply of internal CPU clock

The program execution after clock supply is as follow.

- When the level of the interrupt is enabled by the I flag of CPU ILM
  - After saving the register, fetches in the interrupt vector and executes from the processing routine.
- When the level of the interrupt is disabled by the I flag of CPU ILM
  - Executes the instruction of the following instruction that changed to the stop status.

- Return by Reset request

Returning from sleep status to normal operating status is as per the following procedure.

Generates an internal reset -> Restarts supply of internal bus clock -> Restarts supply of internal CPU clock -> Fetches in the reset vector -> Restarts the command execution from the reset entry address

Notes:

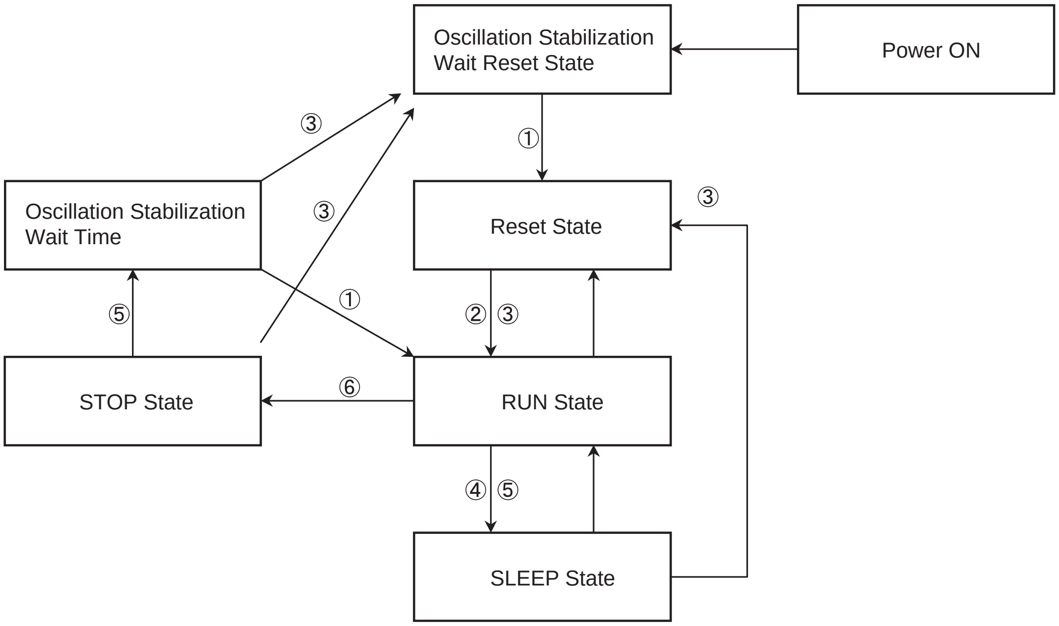
- Execution of the numeric command following the command that writes the STCR may be complete. Therefore, if a cancel or branch command for the interrupt request is placed immediately after, it may seem as though a different operation from that expected is carried out.
- If an interrupt request has already been generated from a peripheral, the status will not be sleep.

3.12.9 State Transition in Standby Mode

Figure 3.12-11 shows the state transition in standby mode.

■ State Transition in Standby Mode

Figure 3.12-11 State transition in standby mode


Oscillation Stabilization Wait Reset State
Power ON
③
①
Oscillation Stabilization Wait Time
③
Reset State
③
①
⑤
② ③
STOP State
⑥
RUN State
④ ⑤
SLEEP State

(1) Oscillation Stabilization Wait Time end

(2) Reset cancellation

(3) Reset input

(4) STCR register SLEP=1

(5) Interrupt input or NMI input

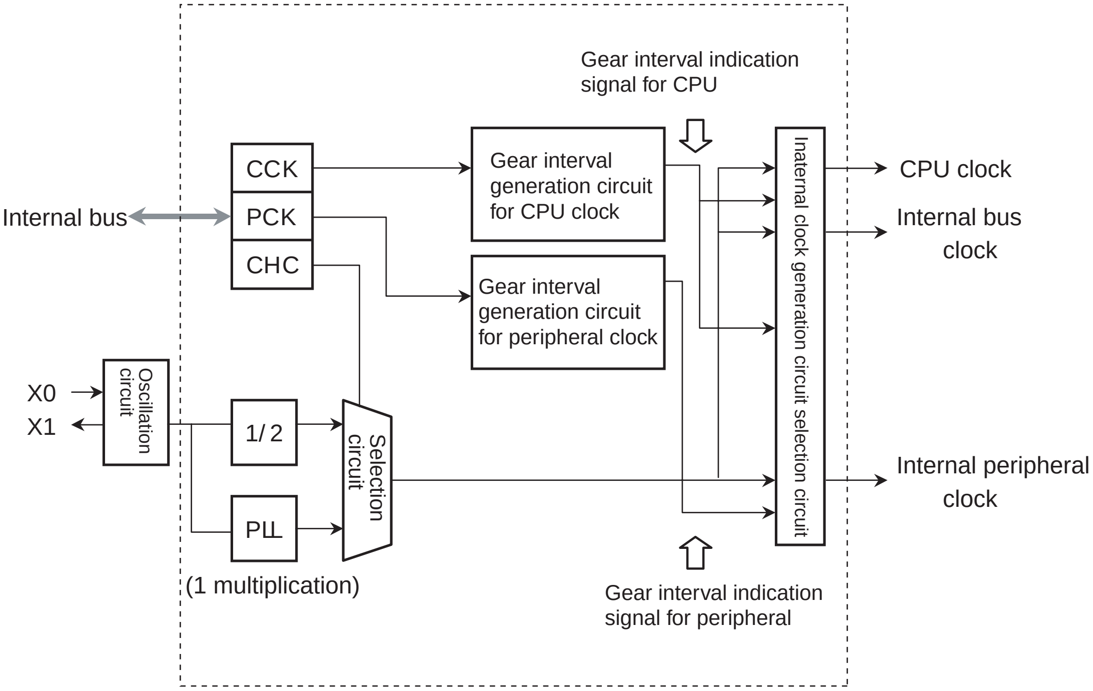
(6) STCR register STOP=1

3.12.10 Gear Function

The gear function supplies to thin out the clock. There are two types of independent circuits (for the CPU and for peripherals), and data can be transmitted and received between the CPU and peripherals even with different gear ratios. Furthermore, whether to use a clock with the same cycle as the clock from the oscillation circuit or a clock via the 1/2 division circuit can be specified as the original clock selection.

■ Block Diagram of Gear Control Section

Figure 3.12-12 Block diagram of gear control section


Gear interval indication signal for CPU
CCK
PCK
CHC
Internal bus
Gear interval generation circuit for CPU clock
Gear interval generation circuit for peripheral clock
Inaternal clock generation circuit selection circuit
CPU clock
Internal bus clock
Internal peripheral clock
X0
X1
Oscillation circuit
1/2
PLL
Selection circuit
(1 multiplication)
Gear interval indication signal for peripheral

■ Setting of Gear Function

The requested gear ratio can be set by setting the CCK1, 0 bit of the gear control register (GCR) to the requested value under the CPU clock control, or by setting the PCK1, 0 bit of the same register to the requested value under the peripheral clock control.

[example]

```
LDI:32 #GCR,R2
LDI:8  #11111100b,R1      ; CCK=11, PCK=11, CHC=0
STB    R1,@R2             ; CPU clock=1/8f, Peripheral clock=1/8f, f=direct
LDI:8  #01111000b,R1      ; CCK=01, PCK=10, CHC=0
STB    R1,@R2             ; CPU clock=1/2f, Peripheral clock=1/4f, f=direct
LDI:8  #00111000b,R1      ; CCK=00, PCK=10, CHC=0
STB    R1,@R2             ; CPU clock=f, Peripheral clock=1/4f, f=direct
```

```
LDI:8  #00110000b,R1        ; CCK=00, PCK=00, CHC=0
STB    R1,@R2               ; CPU clock=f, Peripheral clock=f, f=direct
LDI:8  #10110000b,R1        ; CCK=10, PCK=00, CHC=0
STB    R1,@R2               ; CPU clock=1/4f, Peripheral clock=f, f=direct
```

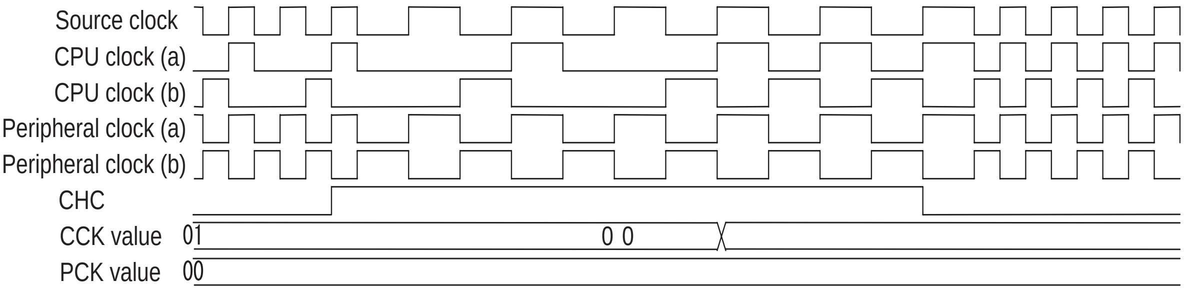
Setting "1" to the CHC bit of the gear control register selects the 1/2 division circuit output as the source clock, and uses "0" and the clock whose cycle is the same as the clock from the oscillation circuit as they are. The CPU system and peripheral system are simultaneously changed to switch the source clock.

[example]

```
LDI:8  #01110001b,R1        ; CCK=01, PCK=00, CHC=1
LDI:32 #GCR,R2
STB    R1,@R2               ; CPU clock=1/2f, Peripheral clock=f, f=1/2xtal
LDI:8  #00110001b,R1        ; CCK=00, PCK=00, CHC=1
STB    R1,@R2               ; CPU clock=f, Peripheral clock=f, f=1/2xtal
LDI:8  #00110000b,R1        ; CCK=00, PCK=00, CHC=0
STB    R1,@R2               ; CPU clock=f, Peripheral clock=f, f=direct
```

Figure 3.12-13 shows the timing.

Figure 3.12-13 Diagram of gear switching timing



■ Restrictions of Gear Function

Table 3.12-6 shows the combination of the gear that can be used in MB91191/MB91192 series.

Table 3.12-6 Restrictions of gear function

For CHC=0		CCK			
		0, 0	0, 1	1, 0	1, 1
PCK	0, 0	△	○	○	○
	0, 1	×	▲	▲	▲
	1, 0	×	×	▲	▲
	1, 1	×	×	×	▲

×: Selection disabled

△: Specify the 1 Wait in the Wait control register before setting. (Refer to "CHAPTER 20 Wait Controller")

▲: The resource of the SIO and PPG is disabled the operation.

Note:

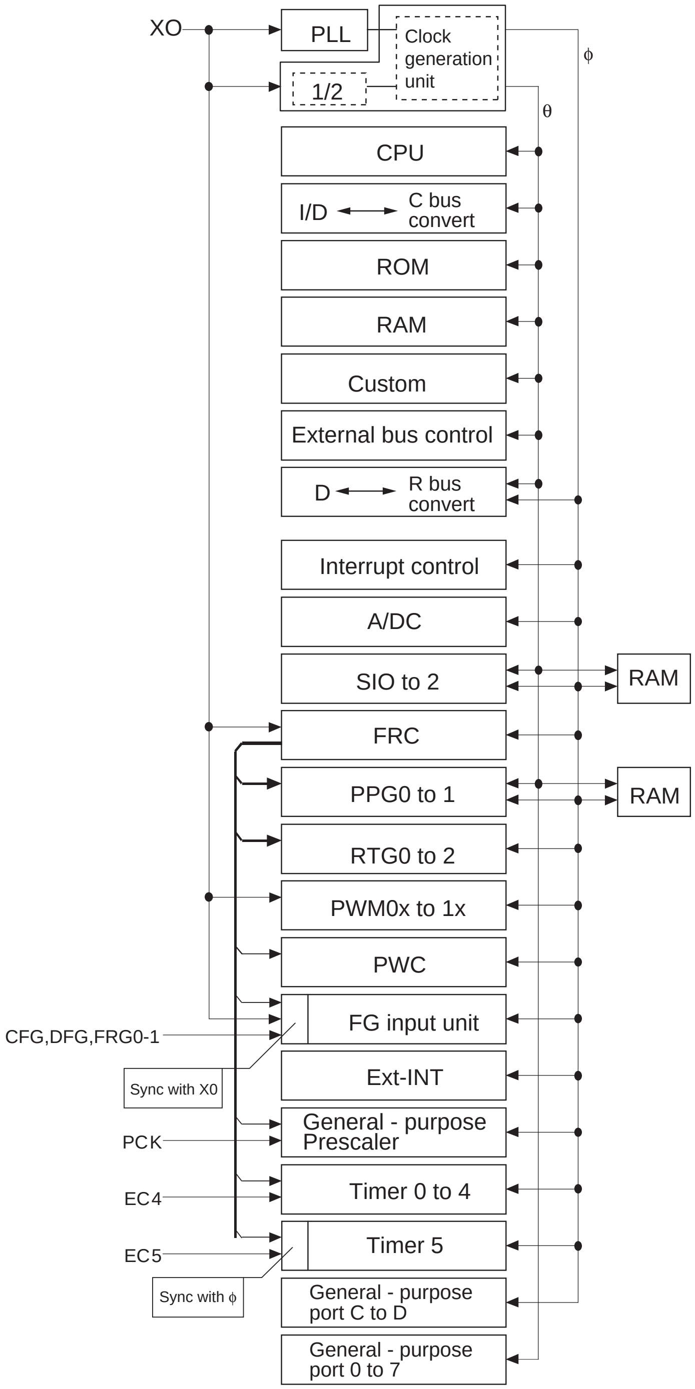
When the CHC is 1, the peripheral resource of SIO and PPG does not operate correctly.

3.12.11 Clock Series Diagram

Figure 3.12-14 shows the clock series diagram.

■ Clock Series Diagram

Figure 3.12-14 Clock series diagram



3.12.12 Clock Series of Peripheral Resource

Table 3.12-7 shows the table for the clock series list of peripheral resource.

■ Table for Clock Series List of Peripheral Resource

Table 3.12-7 Table for clock series list of peripheral resource

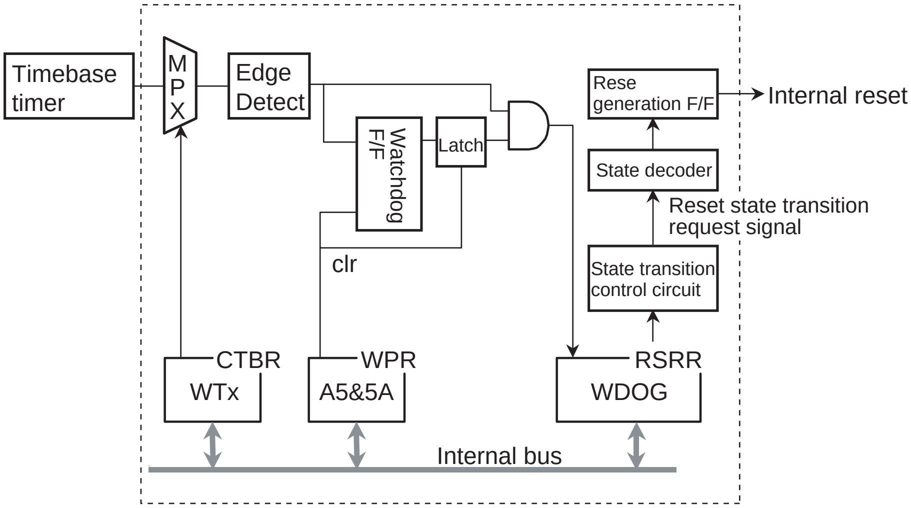
Peripheral resource		Clock	Division clock supply destination	Division clock generation destination	Remark
Servo	Capstan	X0(fch)	-	FRC	Each input is synchronized with X0
	Drum				
	Reel				
FRC, Capture		X0(fch)	PPG RTG Servo General-purpose prescaler Timer 0-5 PWC	-	
PPG	PPG0	FRC ϕ	-	FRC Clock generation block	
	PPG1				
RTG	RTG0	FRC	-	FRC	
	RTG1				
	RTG2				
A/D converter		ϕ	-	Clock generation block	
General-purpose prescaler		FRC OSC	-	FRC OSCI(PCK)	
SIO	SIO0	ϕ	-	Clock generation block	Each input is synchronized with ϕ
	SIO1				
	SIO2				
Timer	TIM0	FRC	-	FRC	Operation by FRC Clock
	TIM1				
	TIM2				
	TIM3				
	TIM4	FRC, EC		FRC	Operation by FRC, external clock
	TIM5	ϕ(FRC, EC)	-	FRC Clock generation block	FRC and external clock is synchronized with ϕ
PWM	PWM0x	X0(fch)	-	-	
	PWM1x				
PWC		FRC	-	FRC	Operation by FRC Clock

3.12.13 Watchdog Function

The watchdog function detects any uncontrolled programs. If writing $A5_H$ and $5A_H$ to the watchdog reset postpone register is not performed within the predetermined period due to an uncontrolled program or suchlike, a watchdog reset request is generated by the watchdog timer.

■ Block Diagram of Watchdog Control Section

Figure 3.12-15 Block Diagram of Watchdog control section



■ Activating Watchdog Timer

The watchdog timer starts operation by writing to the watchdog control register (WTCR). In this case, the interval time for the watchdog timer is set by the WT1 and WT0 bits. In terms of interval time setup, only the time set by the first writing is valid, and subsequent settings will be ignored.

[example]

```
LDI:8  #10000000b,R1     ; WT1, 0=10
LDI:32 #WTCR,R2
STB    R1,@R2            ; Watchdog timer activation
```

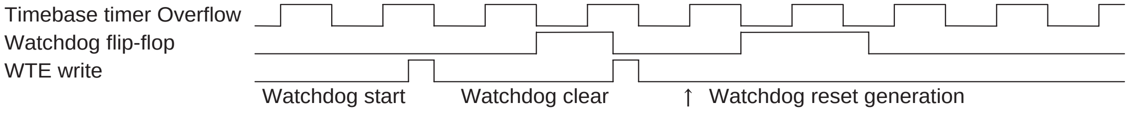
■ Reset Generation Delay

Once the watchdog timer is initiated, $A5_H$ and $5A_H$ must be regularly written to the watchdog reset postpone register (WPR) by the program.

The flip-flop for watchdog reset memorizes the falling edge of the tap selected by the timebase timer if this flip-flop is not cleared by the 2nd falling edge, a reset is generated.

Figure 3.12-16 shows the operations of the watchdog timer.

Figure 3.12-16 Operations of watchdog timer

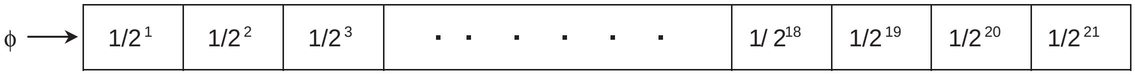


Notes:

- The time interval between the first $5A_H$ and the next $5A_H$ is not specified. Postponement of the watchdog can be performed only when the interval for $5A_H$ writing is within the period specified by the WT bit twice, and $A5_H$ writing is performed once between them.
- After the first $A5_H$ if writing other than $5A_H$ is performed, the first $A5_H$ writing will be invalid. Therefore, $A5_H$ must be written again.

■ Timebase Timer

Figure 3.12-17 Configuration of Timebase Timer



The timerbase timer is used as the timer for the clock supplying to the watchdog timer and timer for the oscillation stabilization wait time. The operation clock ϕ is twice as large as X0 when GCR CHC is 1, and is one time as large as X0 when GCR CHC is 0.

CHAPTER 4

External Bus Interface

This chapter describes an outline of the external bus interface, the register configuration/functions, the bus operation, and the bus timing, and program examples for the bus operation are explained.

4.1 Overview of External Bus Interface

The external bus interface controls the external memory and interface with the external I/O.

■ Feature of External Bus Interface

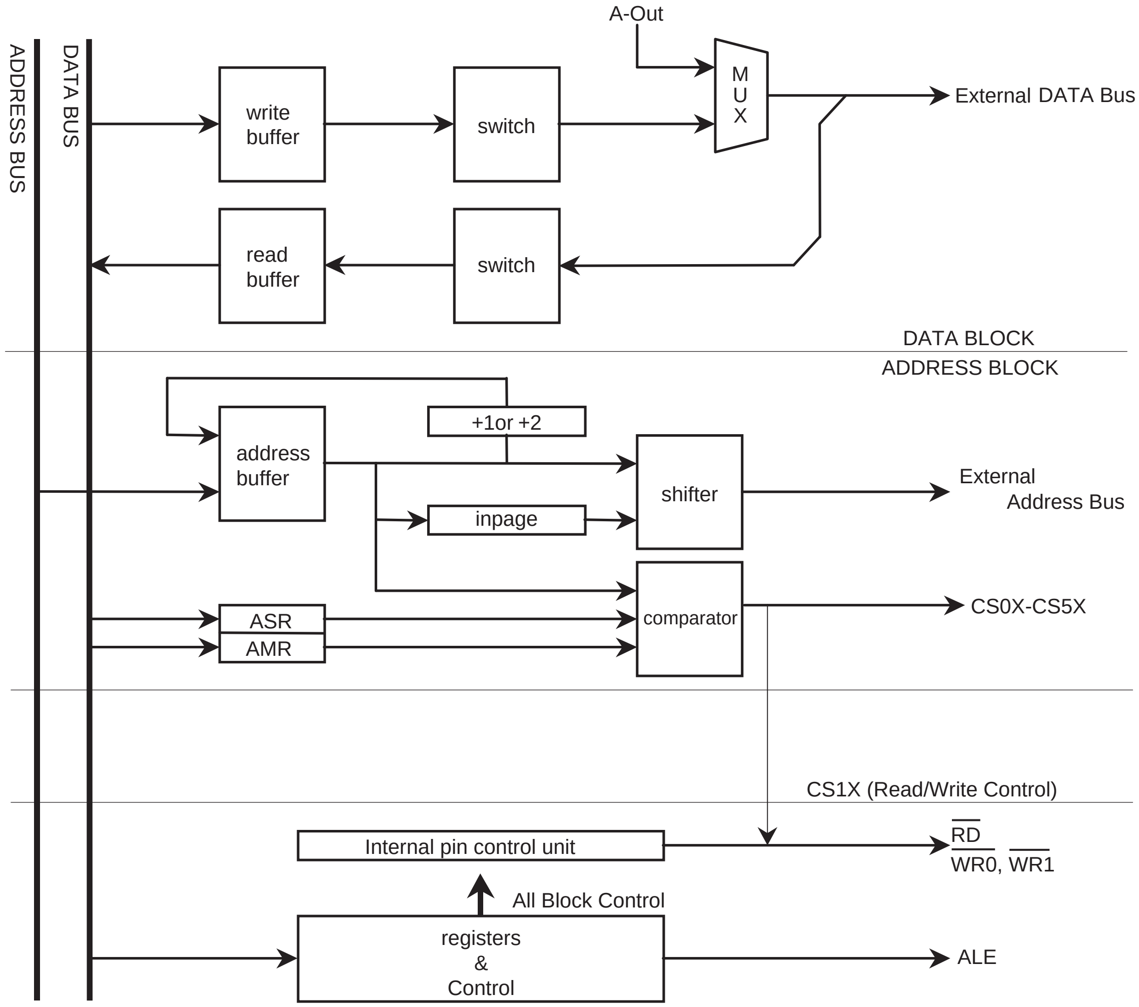
- 16-bit (64KB) address output
- Only 1 bank can be set by the chip select function
  - Capable of setting by the 64KB in the arbitrary position on the logical address space
- Capable of setting a 16-/8-bit bus width
- Insertion of the programmable auto memory wait (maximum for 7 cycles)
- Support for the time division I/O interface of address and data
- Support of little endian mode

4.2 Block Diagram

Figure 4.2-1 shows the block diagram of the external bus interface.

■ Block Diagram of External Bus Interface

Figure 4.2-1 Block diagram of external bus interface



4.3 Area of Bus Interface

A total of six types of chip select areas are prepared as bus interfaces.

■ Area of Bus Interface

Each area position can be arbitrarily allocated in units of at least 64 KB in the 4 GB space by the area select registers (ASR1 to 5) and area mask registers (AMR1 to 5).

Within the area specified by these registers, when the external bus access is performed to area 1 (CS1X), the supported read/write signal ($\overline{RD}$, $\overline{WR0}$, and $\overline{WR1}$) will be active "L".

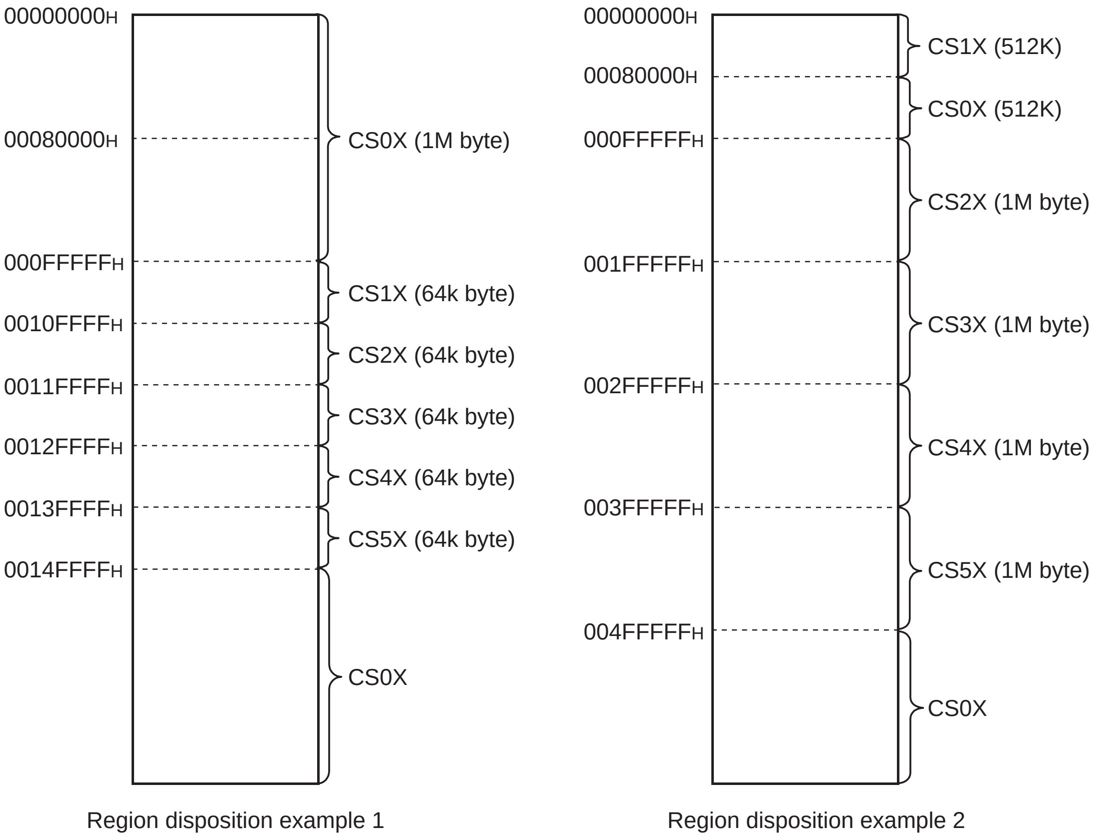
Note:

The area 0 is allocated to space other than the area specified by ASR1 to ASR5.

At a reset, the external area other than 00010000_H to $0005FFFF_H$ becomes area 0.

Figure 4.3-1 shows an example of area 1 to area 5 located in units of 64KB from 00100000_H to $0014FFFF_H$. In the same way, examples in which area 1 is allocated per 512 KB from 00000000_H to $0007FFFF_H$, and areas 2 to 5 are allocated per MB from 00100000_H to $004FFFFF_H$ are shown in Figure 4.3-1 .

Figure 4.3-1 Setting example of chip select area



4.4 Bus Interface

The bus interface has an follow:

- **Normal bus interface**
- **Time division I/O interface of address and data**
- **DRAM control interface**

These interfaces can only be used in predetermined areas.

■ Bus Interface

Table 4.4-1 shows the correspondence between each chip select area and the usable interface function. The area mode register (AMD) is specifies which interface is used.

Table 4.4-1 Each area and usable interface mode

Area	Selectable bus interface			Remark
	Normal Bus	Time division	DRAM	
0	-	-	-	The use of the terminal is prohibited.
1	-	○	-	
2 to 5	-	-	-	The use of the terminal is prohibited.

● Specifying time division I/O

In area 1, the address and data are time division input/output on the bus with the width set by AMD1.

The latch pulse of address is output to the ALE pin.

● Specifying bus size

A bus width can be specified the area 1 by the register setting.

4.5 Register of External bus Interface

This section lists the register of external bus interface.

■ Register List of External Bus Interface

Figure 4.5-1 Register list of external bus interface

<table>
<tr><th></th><th>bit</th><th>15 ←→ 8</th><th>7 ←→ 0</th><th></th></tr>
<tr><td>Address:</td><td>00060CH</td><td colspan="2">ASR 1</td><td>Area Select Register 1</td></tr>
<tr><td></td><td>00060EH</td><td colspan="2">AMR 1</td><td>Area Mask Register 1</td></tr>
<tr><td></td><td>000610H</td><td colspan="2">ASR 2</td><td>Area Select Register 2</td></tr>
<tr><td></td><td>000612H</td><td colspan="2">AMR 2</td><td>Area Mask Register 2</td></tr>
<tr><td></td><td>000614H</td><td colspan="2">ASR 3</td><td>Area Select Register 3</td></tr>
<tr><td></td><td>000616H</td><td colspan="2">AMR 3</td><td>Area Mask Register 3</td></tr>
<tr><td></td><td>000618H</td><td colspan="2">ASR 4</td><td>Area Select Register 4</td></tr>
<tr><td></td><td>00061AH</td><td colspan="2">AMR 4</td><td>Area Mask Register 4</td></tr>
<tr><td></td><td>00061CH</td><td colspan="2">ASR 5</td><td>Area Select Register 5</td></tr>
<tr><td></td><td>00061EH</td><td colspan="2">AMR 5</td><td>Area Mask Register 5</td></tr>
<tr><td></td><td>000620H</td><td>AMD 0</td><td>AMD 1</td><td>Area Mode Register 0 / Area Mode register 1</td></tr>
<tr><td></td><td>000622H</td><td>AMD 32</td><td>AMD 4</td><td>Area Mode Register 32 / Area Mode register 4</td></tr>
<tr><td></td><td>000624H</td><td>AMD 5</td><td>—</td><td>Area Mode Register 5</td></tr>
<tr><td></td><td>000626H</td><td colspan="2">RFC R</td><td>ReFresh Control Register</td></tr>
<tr><td></td><td>00062CH</td><td colspan="2">DMCR4</td><td>DRAM Control Register 4</td></tr>
<tr><td></td><td>00062EH</td><td colspan="2">DMCR5</td><td>DRAM Control Register 4</td></tr>
<tr><td></td><td>0007FEH</td><td>LER</td><td>MODR*</td><td>Little Endian Register / MODe Register</td></tr>
</table>

*: For detail of MODR register, see "3.11 Memory Access Mode".

Note:

Function pins for parts are not prepared under this product, so do not access these registers.

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

Area selection registers (ASR1 to ASR5) and area mask registers (AMR1 to ACR5) specify the address space of the chip select areas 1 to 5.

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

Figure 4.5-2 Area selection register (ASR1 to 5)

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	

Figure 4.5-3 Area Mask register (AMR1 to 5)

AMR1 bit	15	14	13	12	···	2	1	0	Initial valaue
Address 0000 060E$_H$	A31	A30	A29	···	···	A18	A17	A16	0000$_H$
Access →	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$_H$
Access →	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$_H$
Access →	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$_H$
Access →	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$_H$
Access →	W	W	W			W	W	W	

Area selection registers (ASR1 to ASR5) and area mask registers (AMR1 to ANR5) specify the address space of the chip select areas 1 to 5.

Area selection registers (ASR1 to ASR5) specify the upper 16 bits (A31 to A16) of the address, and mask the address bit supported by the area mask registers (AMR1 to AMR 5). Each bit of area mask register

(AMR1 to AMR5) assumes "care" by "0" and "don't care" by "1".

"care" indicates the address space while setting "0" when the ASR set value is "0" or setting "1" when it is "1". "don't care" indicates the address space for both "0" and "1" cases regardless of the ASR set value.

Examples for each chip select area specification by combining the area selection register and area mask register are shown below.

[example 1]

ASR1 = 00000000 000000 $\underline{11}_B$

AMR1 = 00000000 000000 $\underline{00}_B$

When the above is set, the AMR1 bit supporting the bit where "1" is set to ASR1 is "0", so the address space of area 1 will be 64 KB as per the following.

00000000 00000011 00000000 00000000_B (00030000_H)

to

00000000 00000011 11111111 11111111_B ($0003FFFF_H$)

[example 2]

ASR2 = 00001111 11111111_B

AMR2 = 00000000 000000 $\underline{11}_B$

When the above is set, "1" and "0" are left as "care" for the ASR2 set value supporting the bit where "0" is set as AMR2, and the ASR2 bit supporting the bit where "1" is set as AMR2 will be "don't care" for "0" or "1", so the address space of area 2 will be 256 KB as per the following.

00001111 11111100 00000000 00000000_B ($0FFC0000_H$)

to

00001111 11111111 11111111 11111111_B ($0FFFFFFF_H$)

The address space of each area 1 to 5 can be arbitrarily allocated in units of at least 64 KB in the 4 GB space by ASR1 to ASR5 and AMR1 to AMR5. Within the area specified by these registers, when bus access is performed on area 1, the supported read/write pin ($\overline{RD}$, $\overline{WR0}$, and $\overline{WR1}$) will be "L" output.

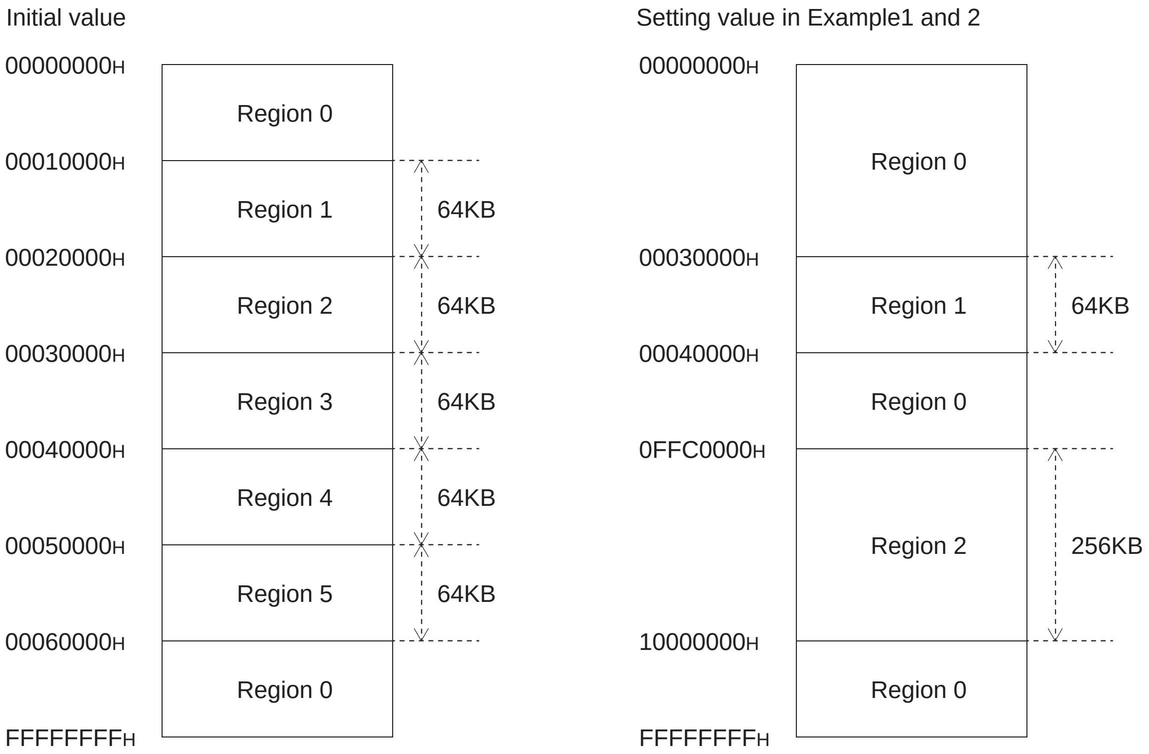
As area 0, space other than the area set by ASR1 to ASR5 and AMR 1 to AMR5 is allocated, the area other than 0001000_H to $0005FFFF_H$ is allocated by the initial values of ASR1 to ASR5 and AMR1 to AMR5 at a reset.

Note:

Set the chip select area so that mutual overlapping is avoided.

Figure 4.5-4 shows a map set in the units of 64KB by the initial value at a reset and a map of area set in the example 1 and 2 above.

Figure 4.5-4 Sample maps specified by chip select area



4.5.2 Area Mode Register 1 (AMD1)

The area mode register 1 (AMD1) specifies the memory access operation mode for each chip select area.

■ Area Mode Register 1 (AMD1)

Figure 4.5-5 Area mode register 1 (AMD1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000 0621H	MPX	—	—	BW1	BW0	WTC2	WTC1	WTC0	0--00000B
Access →	R/W			R/W	R/W	R/W	R/W	R/W	

The area mode register 1 (AMD1) specifies the operation mode for the chip select area 1 (the area specified by ASR1 and AMR1). In area 1, the time division input/output interface can be specified for the address/ data input/output.

The time division input/output interface outputs addresses to the data bus and inputs/outputs data, and supports 8-bit and 16-bit bus widths only as follows.

- 8-bit bus width: A7 to A0 multiplexes to D31 to D24.
- 16-bit bus width: A15 to A0 multiplexes to D31 to D16.

[bit7] :MPX (MultiPleX bit)

The MPX controls the time division input/output interface for the address and data input/output.

0	Specification prohibited
1	Time division input/output interface

When the external bus mode is used for this product, the time division input/output interface must be specified.

[bit4, 3] :BW1, 0 (Bus Width bit)

The BW1 and 0 specifies the bus width of area 1.

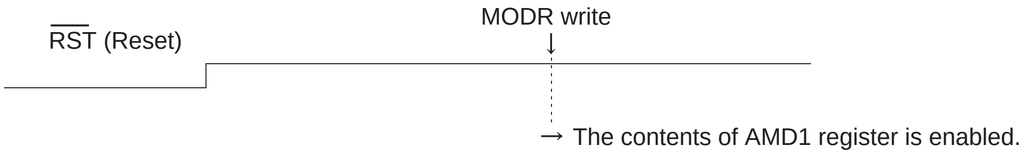
BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	reserved
1	1	reserved

[bit2 to 0] :WTC2 to 0(Wait Cycle bit)

The WTC specifies the auto insert wait cycle number for normal bus interface and time division input/ output interface operation. Operation is the same as WTC2 to WTC0 of the AMD0, but is initialized to "000" by reset, and the insert wait cycle number will be "0".

Note:

Be sure to set the BW1 and BW0 of AMD1 before writing to the MODR.
After setting the mode register (MODR), the bus width set by the AMD is valid for external area 1.
Do not change the BW1 and BW0 after writing the MODR. It causes the malfunction.



→ The contents of AMD1 register is enabled.

4.5.3 Little Endian Register (LER)

Bus access of the MB91191/MB91192 series is usually big endian for all areas, but setting the LER enables one of the areas from 1 to 5 to be handled as a little endian area.
However, area 0 is not for little endian.

■ Little Endian Register (LER)

Figure 4.5-6 Little endian register (LER)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000 07FEH	—	—	—	—	—	LE2	LE1	LE0	-----000B
Access →						W	W	W	

Table 4.5-1 Mode setting using combination of bits (LE2, LE1 and LE0)

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

4.6 Bus Operation

The basic item of the bus operation is explained as follow.

- **The relationship between the data bus width and control signal**
- **Bus access of big endian**
- **Bus access of little endian**
- **Comparison of external access between big endian and little endian**

■ Relationship between Data Bus Width and Control Signal

The relationship between the data bus width and control signal for the next bus interface is described.

- The time division input/output bus interface

■ Bus Access of Big Endian

The external access is described as follow.

- Data format
- Data bus width
- External bus access
- Connection example with external device

■ Bus Access of Little Endian

The external access is described as follow.

- Difference between little endian and big endian
- Data format
- Data bus width
- Connection example with external device

■ Comparison of External Access between Big Endian and Little Endian

As a comparison of the external access between big endian and little endian, word access, half-word access, and byte access to the bus width are described.

4.6.1 Relationship between Data Bus Width and Control Signal

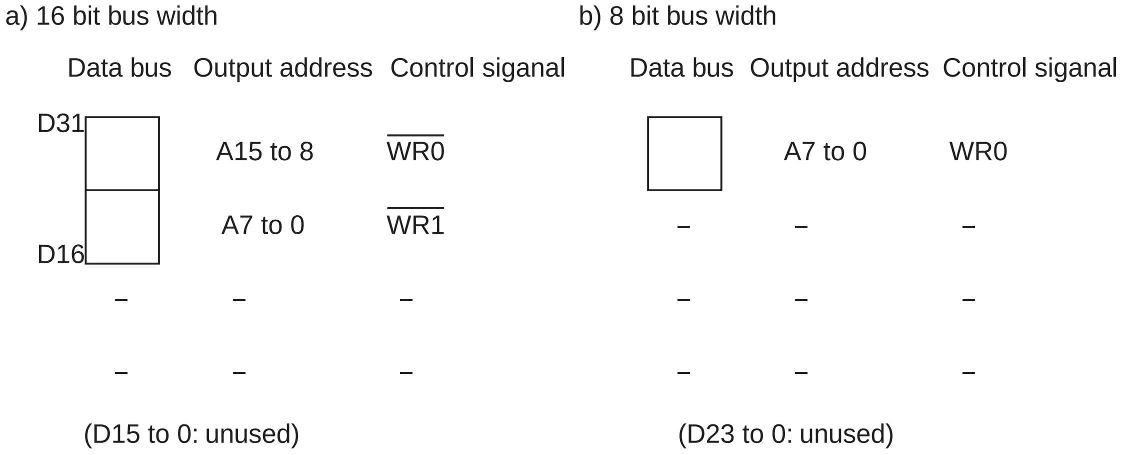
The control signal of $\overline{\text{WR0}}$ to $\overline{\text{WR1}}$ always supports the byte position of the data bus 1:1 regardless of big endian/little endian is used, or the data bus width.

■ The Relationship between the Data Bus Width and Control Signal

In this section, the byte position of the data bus for this product used by the data bus width that has been set, and control signals that support it are summarized.

● The time division input/output bus interface

Figure 4.6-1 Data bus width and control signals under the time division input/output bus interface



a) 16 bit bus width

Data bus	Output address	Control siganal
D31 □	A15 to 8	$\overline{\text{WR0}}$
□ D16	A7 to 0	$\overline{\text{WR1}}$
–	–	–
–	–	–

(D15 to 0: unused)

b) 8 bit bus width

Data bus	Output address	Control siganal
□	A7 to 0	WR0
–	–	–
–	–	–
–	–	–

(D23 to 0: unused)

4.6.2 Bus Access of Big Endian

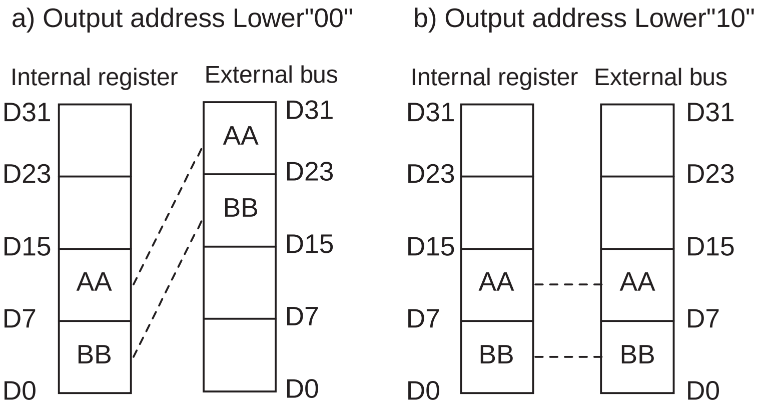
The data format in the FR20 series is normally big endian. Thus, external bus access is big endian for areas in which little endian (LER) is not set.

■ Data Format

The relationship between the internal register and external data bus is indicated per data format.

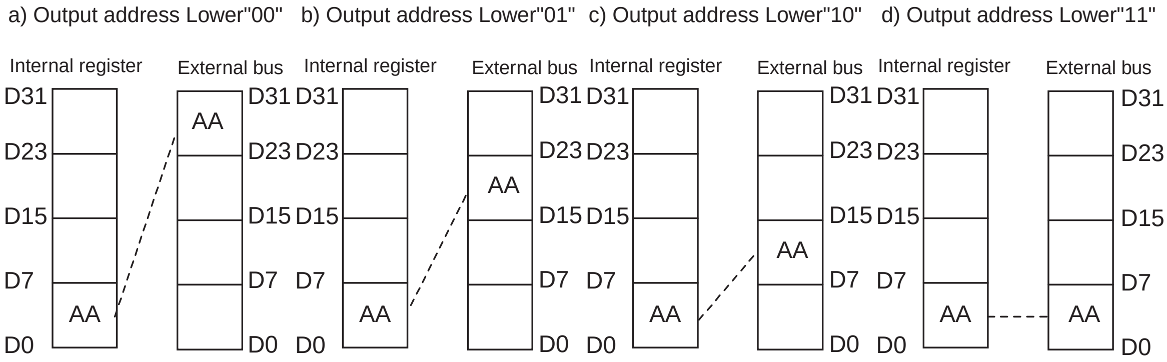
● Half-word access (LDUH, STH instruction execution)

Figure 4.6-2 Relationship between the internal register and external data bus of half-word access



● Byte access (LDUH, STH instruction execution)

Figure 4.6-3 Relationship between internal register and external data bus of byte access

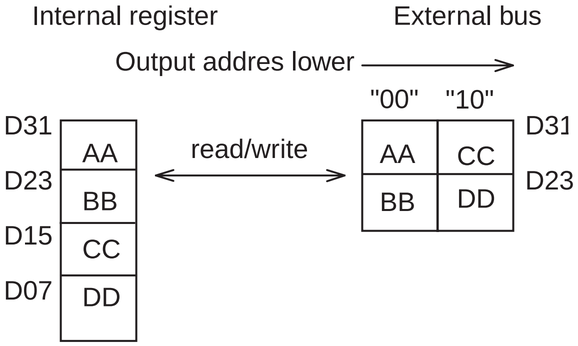

a) Output address Lower"00"
b) Output address Lower"01"
c) Output address Lower"10"
d) Output address Lower"11"
Internal register
External bus
D31
D23
D15
D7
D0
AA
AA

■ Data Bus Width

The relationship between the internal register and external data bus is indicated per data bus width.

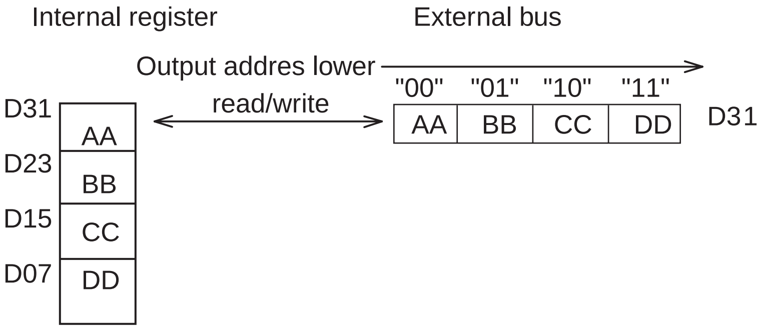
- 16-bit bus width

Figure 4.6-4 Relationship between internal register and external data bus of 16-bit bus width



- 8-bit bus width

Figure 4.6-5 Relationship between internal register and external data bus of 8-bit bus width



■ External Bus Access

External bus access (16-bit/8-bit bus width) is shown per word, half-word, and byte access in Figure 4.6-6 and Figure 4.6-7 . The following items are also shown in Figure 4.6-6 and Figure 4.6-7 .

- Access byte position
- Program address and output address
- Bus access count

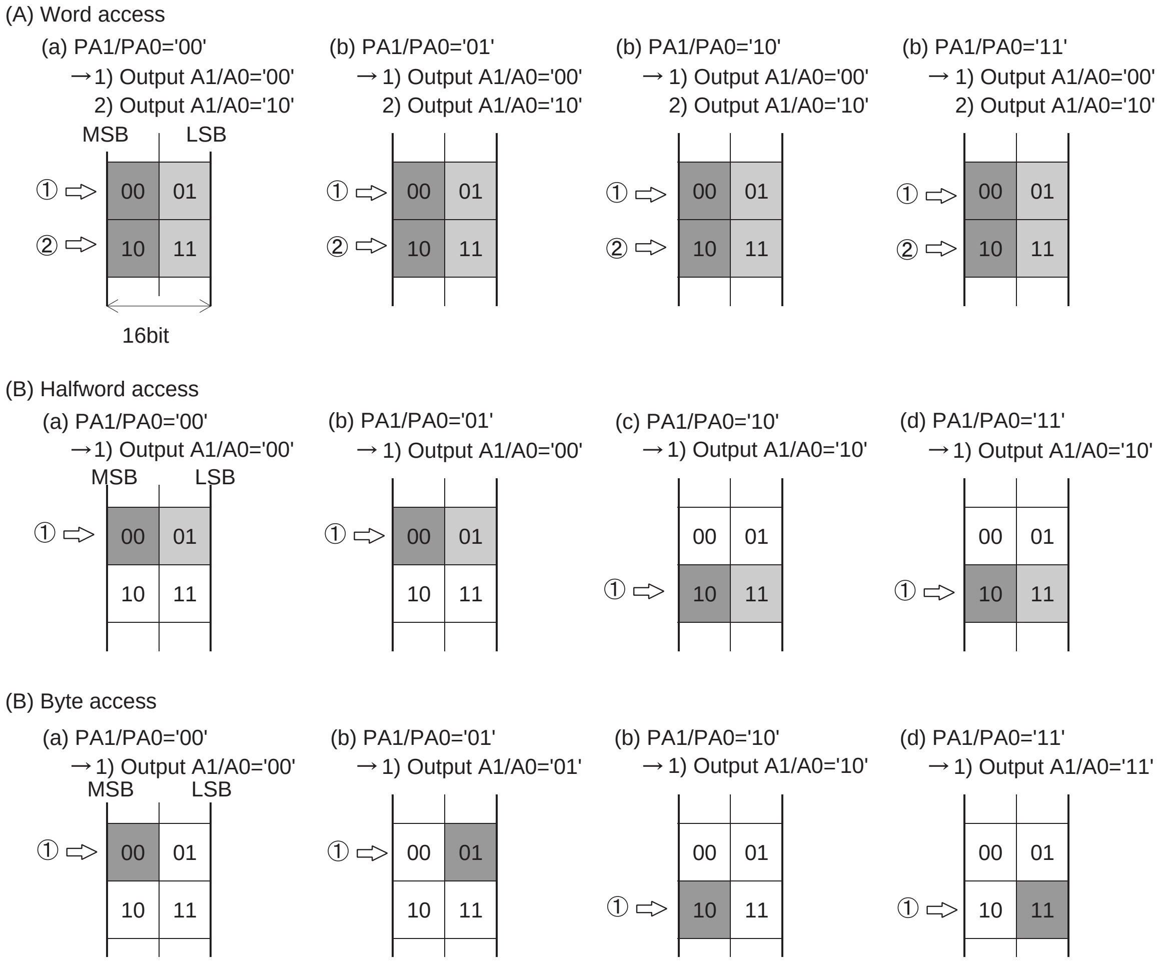
PA1/PA0	: Specified address lower 2-bit by program
Output A1/A0	: Output address lower 2-bit
▨	: Output address start byte position
▨ + ░	: Access data byte position
①~④	: Bus access count

The MB91191/MB91192 series do not detect the misalignment.

According to the word access, even if the address lower 2-bit specified by the program is "00", "01", "10", or "11", the address lower 2-bit to be output will be all "00". In case of half-word access, when the lower address is "00" or "01", "00" will be set, and when it is "10" or "11", "10" will be set.

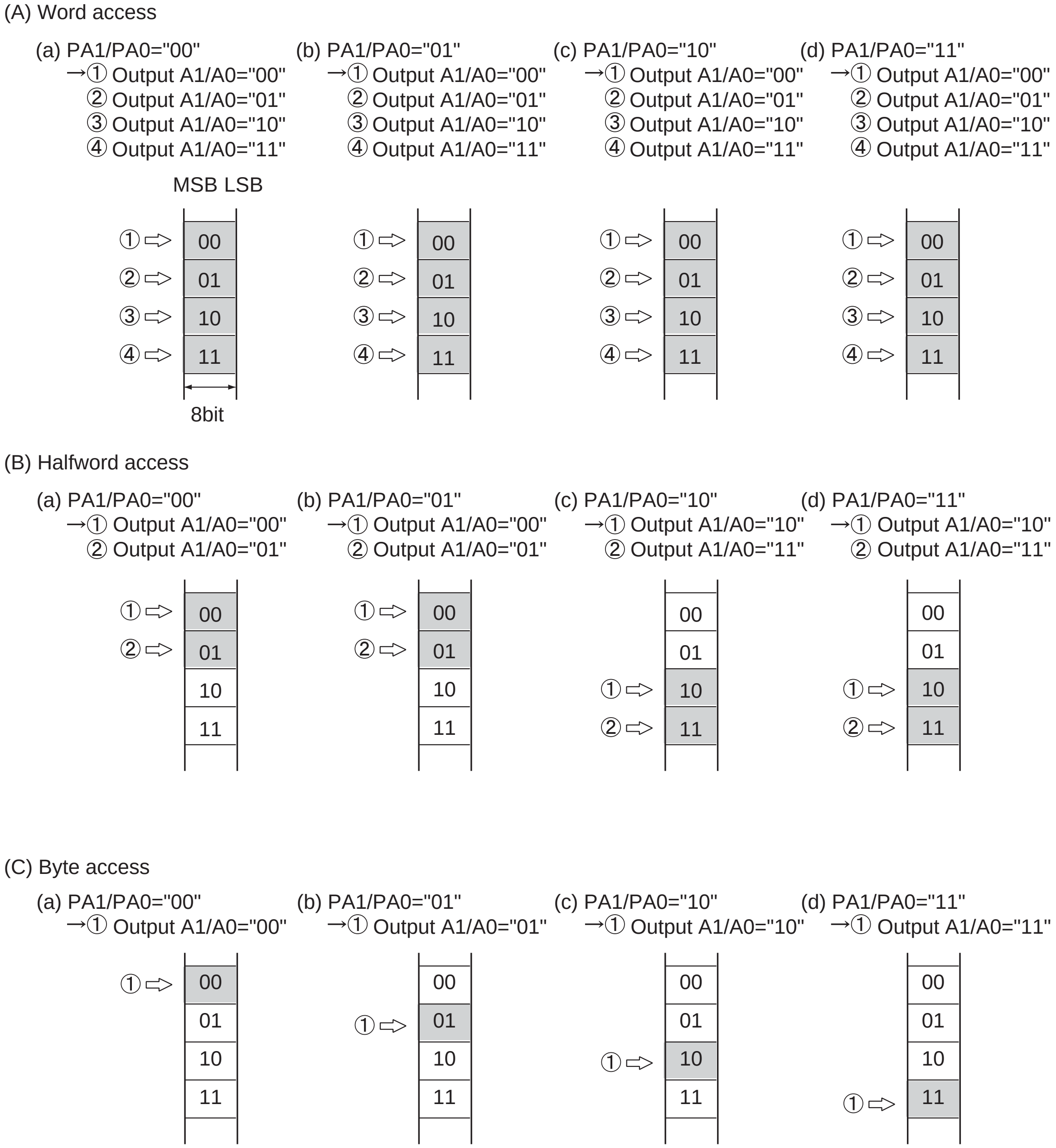
- Width of bus of 16 bits

Figure 4.6-6 External bus access on 16-bit bus width



- Width of bus of 8 bits

Figure 4.6-7 External bus access on 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"

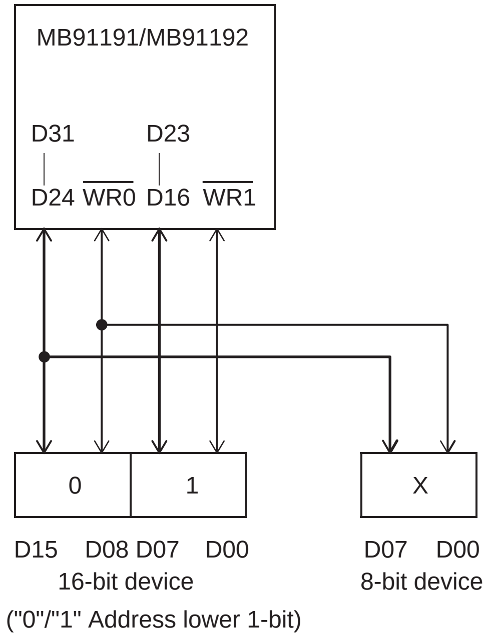
(b) PA1/PA0="01"
→① Output A1/A0="01"

(c) PA1/PA0="10"
→① Output A1/A0="10"

(d) PA1/PA0="11"
→① Output A1/A0="11"

■ Connection Example with External Device

Figure 4.6-8 Connection example with external device



4.6.3 Bus Access of Little Endian

For areas in which little endian (LER) is set, external bus access is little endian.
Bus access of the MB91191/MB91192 series is realized by swapping the byte position of the data bus in accordance with the bus width while the big endian output address order and control signal output are basically the same, using the bus access operation for big endian.
When connected, the big endian and little endian areas must be physically separated, so special care must be taken.

■ Differences between Little Endian and Big Endian

Differences between little endian and big endian are mentioned as follows.

The order of addresses to be output is the same for big endian and little endian.

Data bus control signal to be used for the 32-, 16-, and 8-bit bus width is the same for big endian and little endian.

● Word access

The byte data at the MSB side supporting the big endian address "00" is the little endian byte data at the LSB side. In the case of word access, all of the four-byte positions within the word will be reversed.

"00" → "11", "01" → "10", "10" → "01", "11" → "00"

● Half-word access

The byte data at the MSB side supporting the big endian address "0" is the little endian byte data at the LSB side. In the case of half-word access, the two-byte positions within the half-word will be reversed.

"0" → "1", "1" → "0"

● Byte access

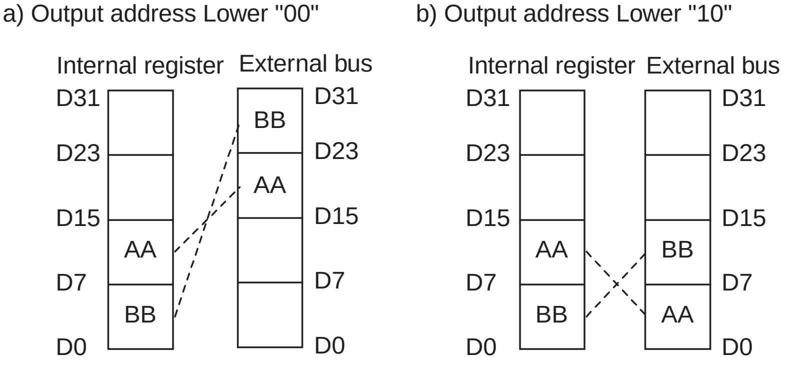
Both big endian and little endian are the same.

■ Data Format

The relationships between the internal register and external data bus is shown per data format.

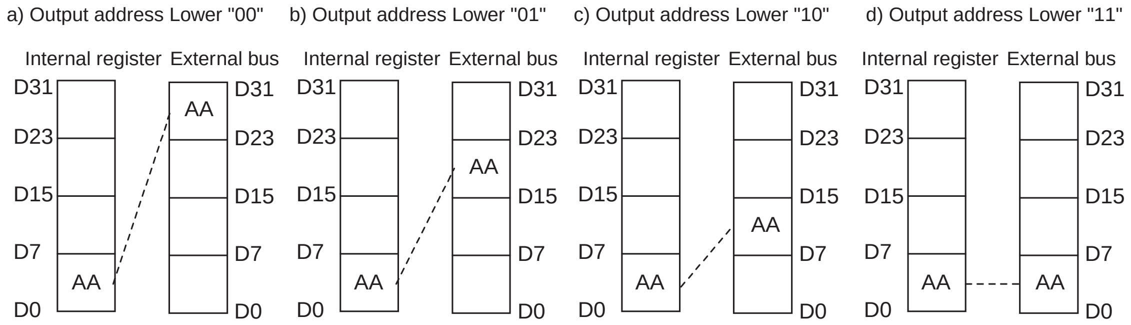
● Half word access (LDUB, STB instruction execution)

Figure 4.6-9 Relationship between internal register and external data bus of half-word access



- Byte access (LDUB, STB instruction execution)

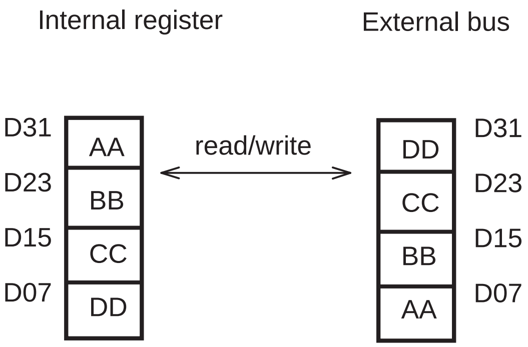
Figure 4.6-10 Relationship between internal register and external data bus of byte access


a) Output address Lower "00" b) Output address Lower "01" c) Output address Lower "10" d) Output address Lower "11"

■ Data Bus Width

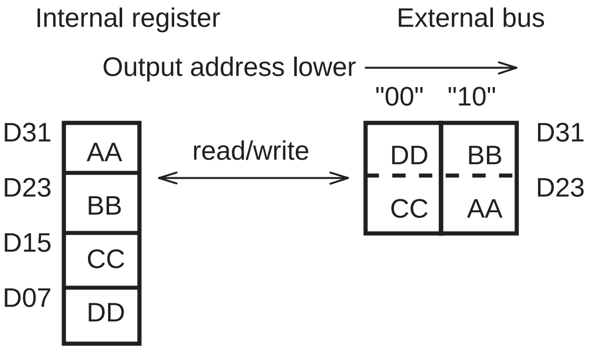
- 32-bit bus width

Figure 4.6-11 Relationship between internal register and external data bus of 32-bit bus width



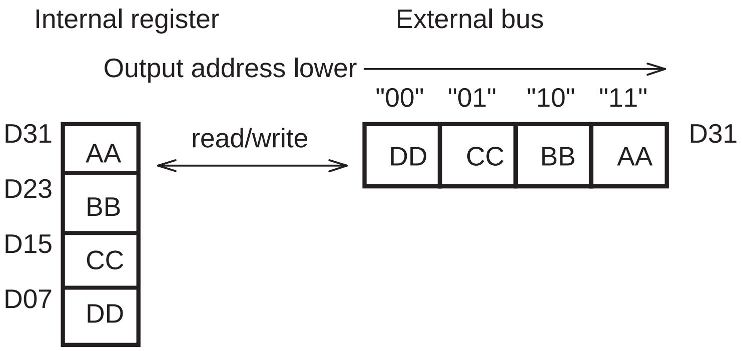
- 16-bit bus width

Figure 4.6-12 Relationship between internal register and external data bus of 16-bit bus width



- 8-bit bus width

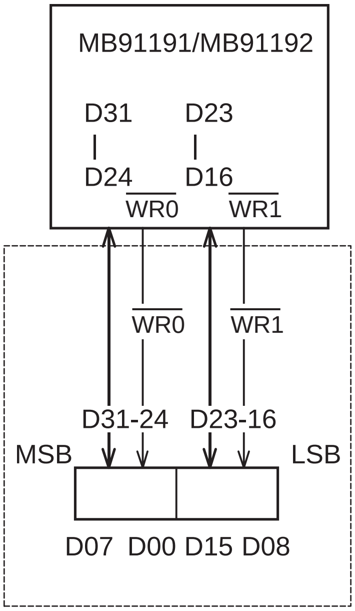
Figure 4.6-13 Relationship between internal register and external data bus of 8-bit bus width



■ Connection Example with External Device

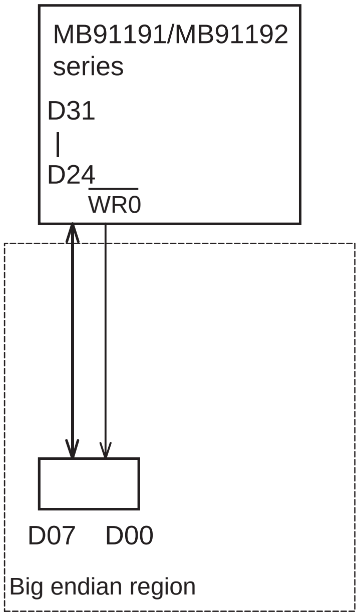
- 16-bit bus width

Figure 4.6-14 Connection example with MB91191/MB91192 series and external device (16-bit bus width)



- 8-bit bus width

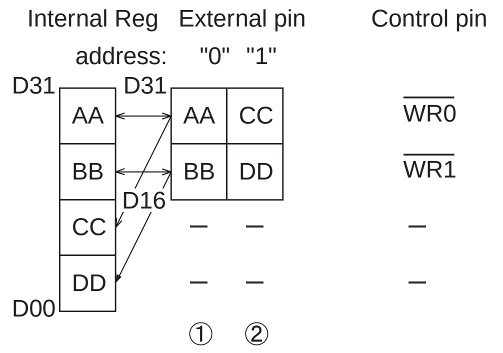
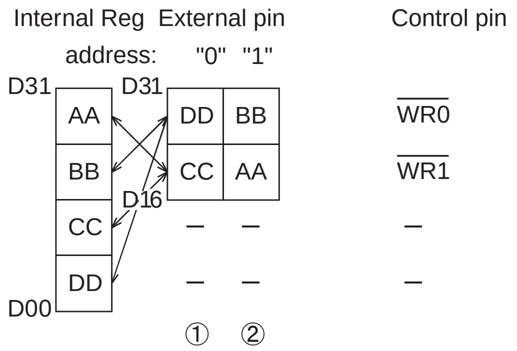
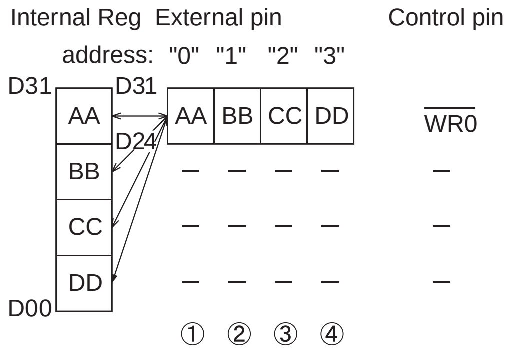
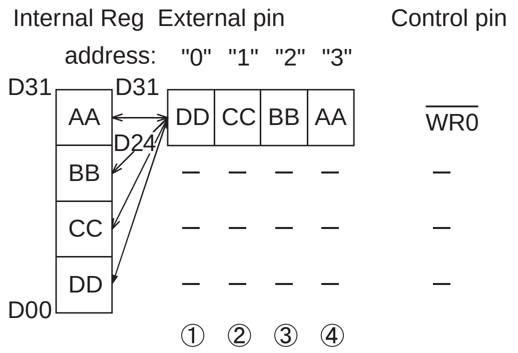
Figure 4.6-15 Connection example with MB91191/MB91192 series and external device (8-bit bus width)



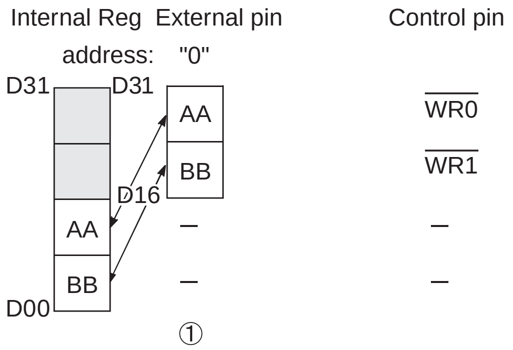
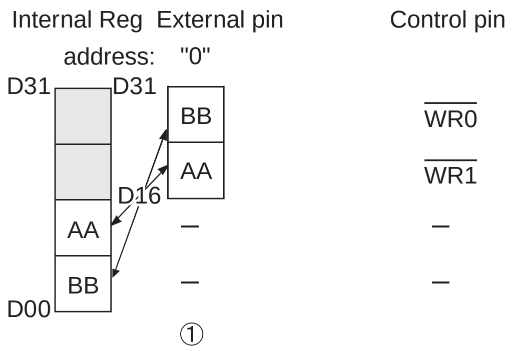
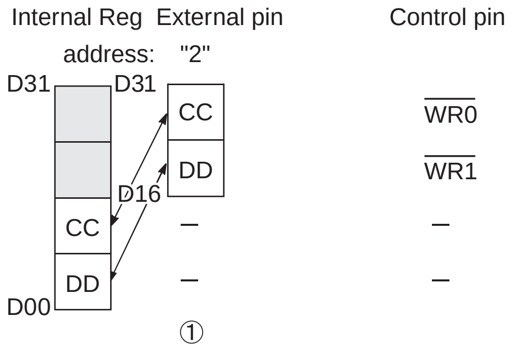
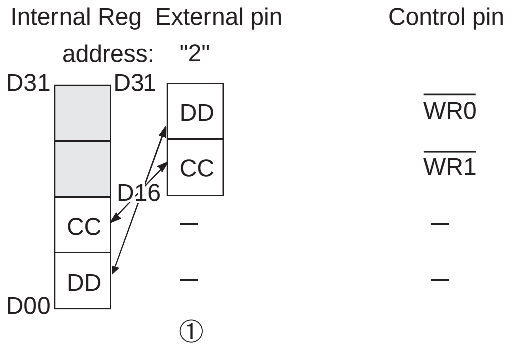
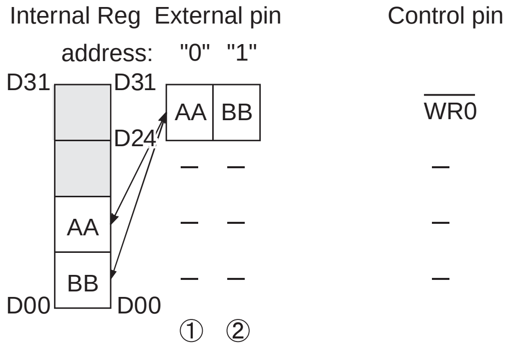
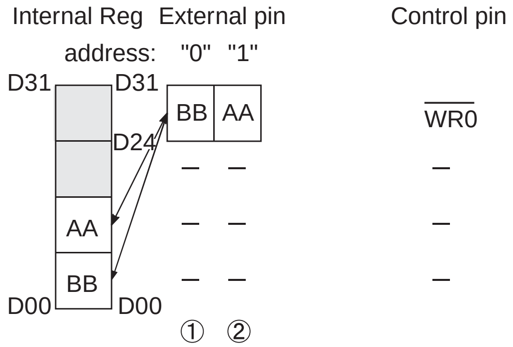
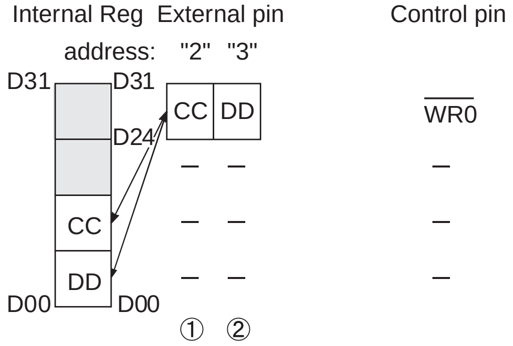
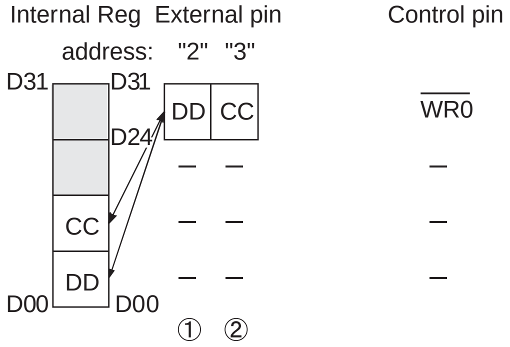
4.6.4 Comparison between Big Endian and Little Endian for External Access

Comparison between big endian and little endian of external access for word access, half-word access, and byte access to the bus width is described.

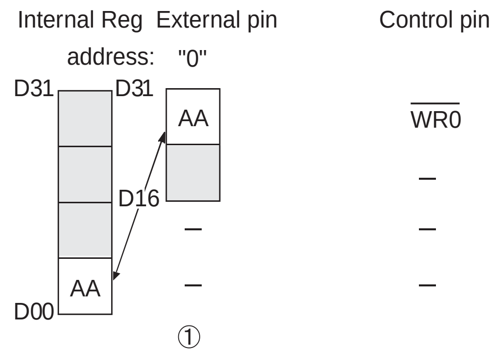
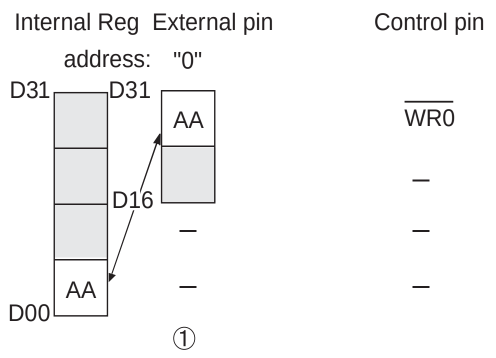
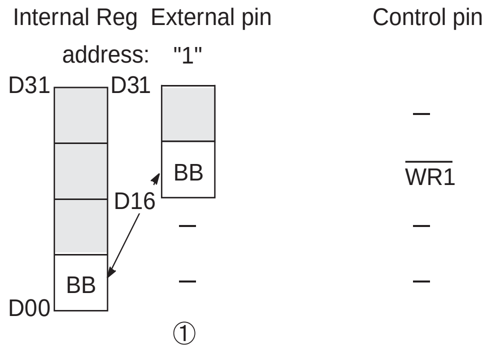
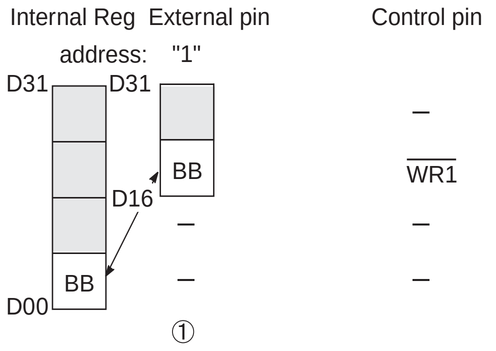
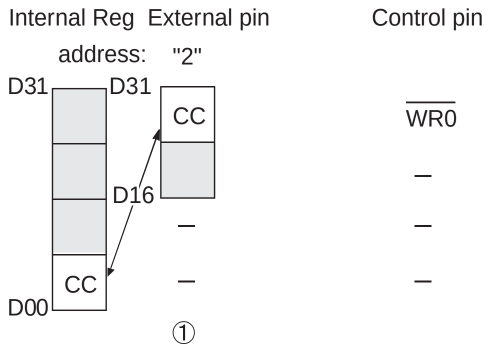
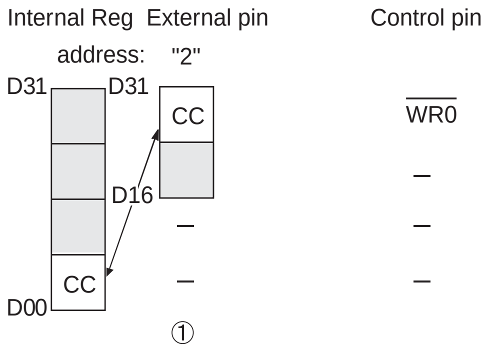
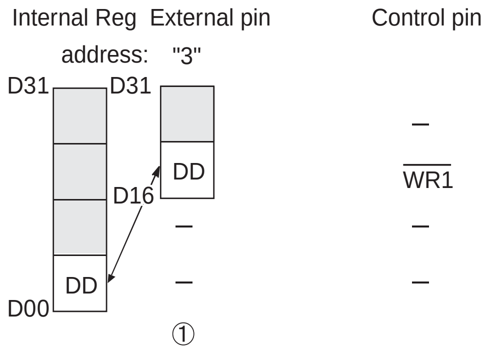
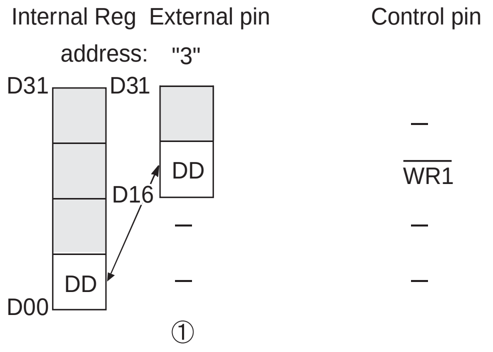
■ Word Access

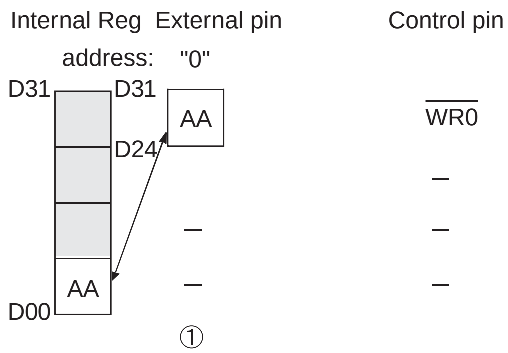
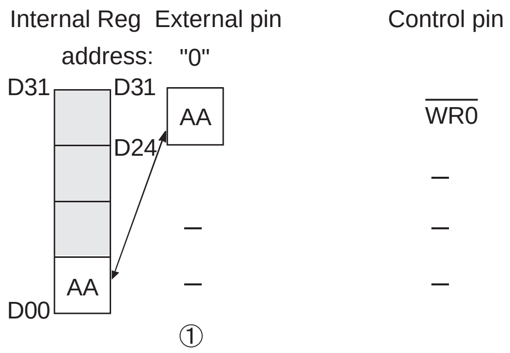
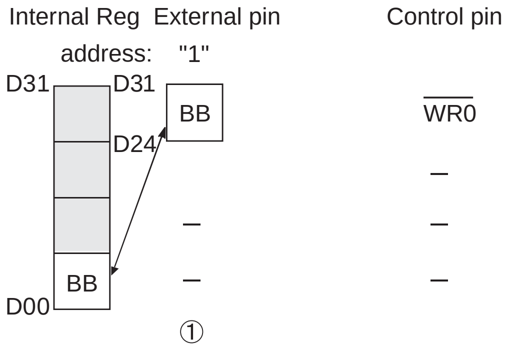
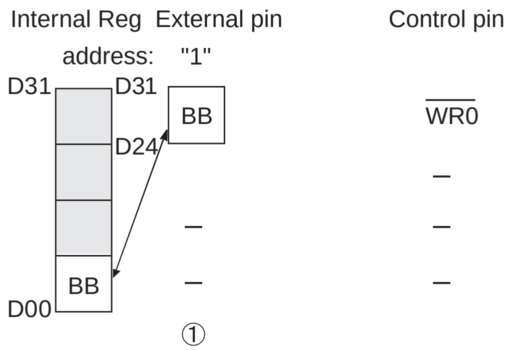
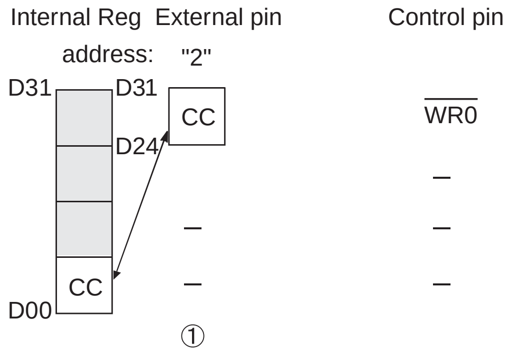
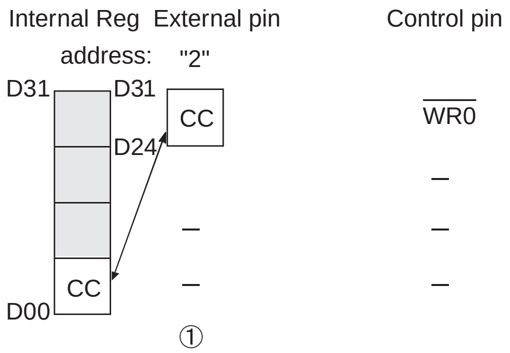
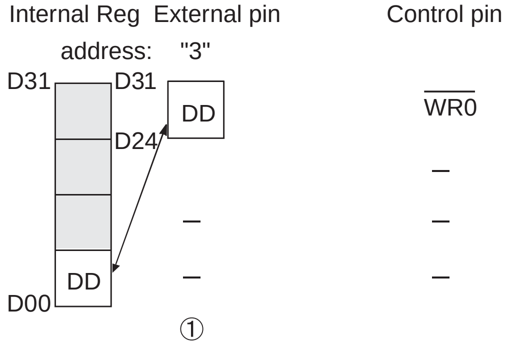
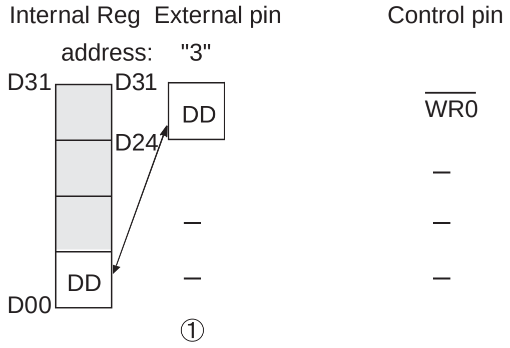
	Big endian mode	Little endian mode
16-bit bus width	 Internal Reg External pin Control pin address: "0" "1" D31 D31 AA AA CC WR0 BB BB DD WR1 D16 CC DD D00 ① ②	 Internal Reg External pin Control pin address: "0" "1" D31 D31 AA DD BB WR0 BB CC AA WR1 D16 CC DD D00 ① ②
8-bit bus width	 Internal Reg External pin Control pin address: "0" "1" "2" "3" D31 D31 AA AA BB CC DD WR0 D24 BB CC DD D00 ① ② ③ ④	 Internal Reg External pin Control pin address: "0" "1" "2" "3" D31 D31 AA DD CC BB AA WR0 D24 BB CC DD D00 ① ② ③ ④

■ Half-word Access

	Big endian mode	Little endian mode
16-bit bus width	 Internal Reg External pin Control pin address: "0" D31 D31 AA BB D16 AA BB D00 WR0 WR1 – – – – ①	 Internal Reg External pin Control pin address: "0" D31 D31 BB AA D16 AA BB D00 WR0 WR1 – – – – ①
	 Internal Reg External pin Control pin address: "2" D31 D31 CC DD D16 CC DD D00 WR0 WR1 – – – – ①	 Internal Reg External pin Control pin address: "2" D31 D31 DD CC D16 CC DD D00 WR0 WR1 – – – – ①
8-bit bus width	 Internal Reg External pin Control pin address: "0" "1" D31 D31 AA BB D24 AA BB D00 D00 WR0 – – – – – – – – – ① ②	 Internal Reg External pin Control pin address: "0" "1" D31 D31 BB AA D24 AA BB D00 D00 WR0 – – – – – – – – – ① ②
	 Internal Reg External pin Control pin address: "2" "3" D31 D31 CC DD D24 CC DD D00 D00 WR0 – – – – – – – – – ① ②	 Internal Reg External pin Control pin address: "2" "3" D31 D31 DD CC D24 CC DD D00 D00 WR0 – – – – – – – – – ① ②

■ Byte Access

	Big endian mode	Little endian mode
16-bit bus width	 Internal Reg External pin Control pin address: "0" D31 D31 AA D16 AA D00 $\overline{\text{WR0}}$ – ①	 Internal Reg External pin Control pin address: "0" D31 D31 AA D16 AA D00 $\overline{\text{WR0}}$ – ①
	 Internal Reg External pin Control pin address: "1" D31 D31 BB D16 BB D00 $\overline{\text{WR1}}$ – ①	 Internal Reg External pin Control pin address: "1" D31 D31 BB D16 BB D00 $\overline{\text{WR1}}$ – ①
	 Internal Reg External pin Control pin address: "2" D31 D31 CC D16 CC D00 $\overline{\text{WR0}}$ – ①	 Internal Reg External pin Control pin address: "2" D31 D31 CC D16 CC D00 $\overline{\text{WR0}}$ – ①
	 Internal Reg External pin Control pin address: "3" D31 D31 DD D16 DD D00 $\overline{\text{WR1}}$ – ①	 Internal Reg External pin Control pin address: "3" D31 D31 DD D16 DD D00 $\overline{\text{WR1}}$ – ①

	Big endian mode	Little endian mode
8-bit Bus width	 Internal Reg External pin Control pin address: "0" D31 D31 AA D24 $\overline{\text{WR0}}$ – – – – – AA D00 ①	 Internal Reg External pin Control pin address: "0" D31 D31 AA D24 $\overline{\text{WR0}}$ – – – – – AA D00 ①
	 Internal Reg External pin Control pin address: "1" D31 D31 BB D24 $\overline{\text{WR0}}$ – – – – – BB D00 ①	 Internal Reg External pin Control pin address: "1" D31 D31 BB D24 $\overline{\text{WR0}}$ – – – – – BB D00 ①
	 Internal Reg External pin Control pin address: "2" D31 D31 CC D24 $\overline{\text{WR0}}$ – – – – – CC D00 ①	 Internal Reg External pin Control pin address: "2" D31 D31 CC D24 $\overline{\text{WR0}}$ – – – – – CC D00 ①
	 Internal Reg External pin Control pin address: "3" D31 D31 DD D24 $\overline{\text{WR0}}$ – – – – – DD D00 ①	 Internal Reg External pin Control pin address: "3" D31 D31 DD D24 $\overline{\text{WR0}}$ – – – – – DD D00 ①

4.7 Bus Timing

The detailed bus access operation in each mode is shown.

■ Time Division I/O Interface

In area 1, time division input/output interface for addresses/data is supported. The time division I/O is performed the bus width specified by the BW1 and BW0.

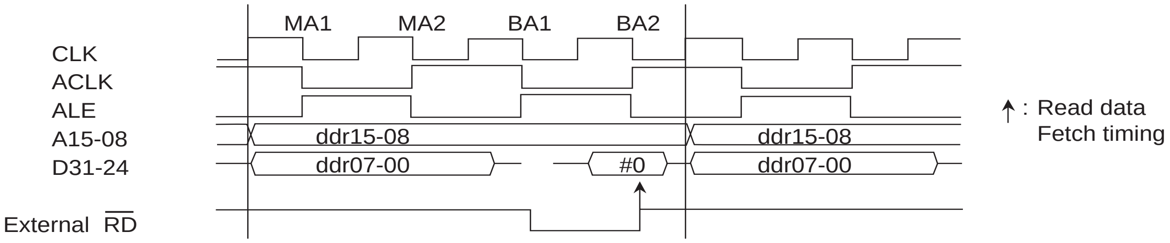
For the time division input/output interface, a total of four clocks cycle, namely address output cycle (2 clocks) + data access cycle (2 clocks) comprises the basic bus cycle, and for the address output cycle, the ALE pin is asserted as the output address latch signal.

Furthermore, the addresses (A15-A08) indicate the start byte for access to the address pins (A15-A08) in the same way as normal during the time-shared bus cycle.

- 8 bits bus width read

Figure 4.7-1 8 bits bus width read

CASE AAMD1: MPX=1, BW=00B, WTC=000B, EPCR0:ALEE=1
access --- 8 bit data read

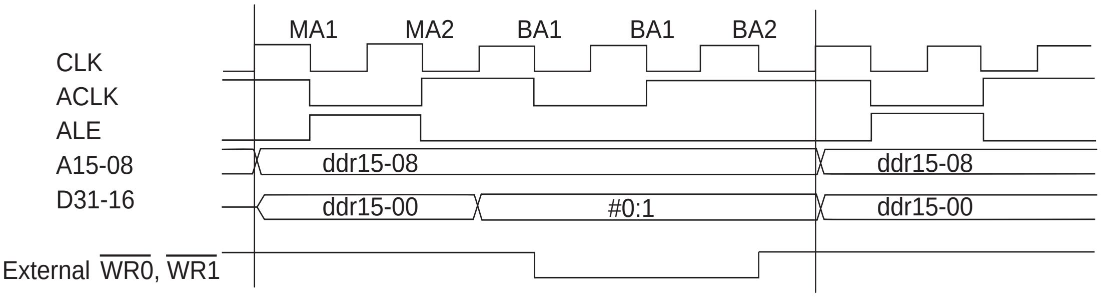


⇒At 8-bit bus width, A07 to A00 addresses output to D31 to D24.

- Automatic wait operation at the time division mode (16-bit bus width write)

Figure 4.7-2 Automatic wait operation at the time division mode (16-bit bus width write)

CASE AMD1: MPX=1, BW=01B, WTC=001B, EPCR0:ALEE=1
aaccess --- 16 bit data write



Note:

To access the external extend area, execute dummy read on built-in ROM area prior to access.

4.8 Program Example of External Bus Operation

A simple program example for operating the external bus is described.

■ Program Specification Example of External Bus Operation

The setting of register is shown as follow.

● Area

- Area 0 (AMD0) : 32-bit, usual bus, automatic wait 0
- Area 1 (AMD1) : 16-bit, time division I/O, automatic wait 2
- Area 2 (AMD32) : 32-bit, usual bus, automatic wait 1
- Area 3 (AMD32) : 32-bit, usual bus, automatic wait 1
- Area 4 (AMD4) :16-bit, DRAM, page size 256, 1CAS/2WE, with wait, CBR refresh
- Area 5 (AMD5) :16-bit, DRAM, page size 512, 2CAS/1WE, without wait, CBR refresh

● The others

- Refresh (RFCR): without wait, 1/8 setting
- External pin (EPCR0): external RDY reception, arbitration of BRQ, BGRNTX
- External pin (DSCR) :DRAM pin setting
- Little endian (LER) : area 2

Note the following about the others.

- MD2, 1, and 0 are "010", and external vector is 32-bit mode.
- Before setting the mode register (MODR), set area 0 to the same bus width.
- Set the area 1 to 5 not to overlap.

■ Program Example of External Bus Operation

Under this program, bytes are used to write the byte register, and half-words are used for the half-word register for explanatory purposes.

```
*****     Example program     *****
//Each register setting
init_gcr    ldi:8     #0x00,r0       //  gcr register  X1 clock mode
ldi:20    #0x484,r1      //  gcr register address setting
stb       r0,@r1         //  gcr register write
init_epcr   ldi:20    #0xffff,r0     //  External pin setting
// External RDY wait, bus arbitration by BRQ, BGRNTX
ldi:20    #0x628,r1      //  epcr0 register address setting
sth       r0,@r1         //  epcr0 register write
init_dscr   ldi:8     #0xff,r0       //  DRAM pin setting
//  RAS, CAS, WE
ldi:20    #0x625,r1      //  dscr register address setting
```

```
stb         r0,@r1            //  dscr register write
init_amd0    ldi:8       #0x10,r0          //  32-bit bus, 0-wait
ldi:20      #0x620,r1         //  amd0 register address setting
stb         r0,@r1            //  amd0 register write
init_amd1    ldi:8       #0x8a,r0          //  time division, 16-bit bus, 2-wait
ldi:20      #0x621,r1         //  amd1 register address setting
stb         r0,@r1            //  amd1 register write
init_amd32   ldi:8       #0x89,r0          //  Normal, 32-bit bus, 1-wait
ldi:20      #0x622,r1         //  amd32 register address setting
stb         r0,@r1            //  amd32 register write
init_amd4    ldi:8       #0x88,r0          //  DRAM, 16-bit bus
ldi:20      #0x623,r1         //  amd4 register address setting
stb         r0,@r1            //  amd4 register write
init_amd5    ldi:8       #0x88,r0          //  DRAM, 16-bit bus
ldi:20      #0x624,r1         //  amd5 register address setting
stb         r0,@r1            //  amd5 register write
init_dmcr4   ldi:20      #0x0c90,r0        //  page size=256, Q1/Q4-wait, Page
//  1CAS-2WE, CBR, No parity
ldi:20      #0x62c,r1         //  dmcr4 register address setting
sth         r0,@r1            //  dmcr4 register write
init_dmcr5   ldi:20      #0x10c0,r0        //  page size=512, Q1/Q4-wait none, Page
//  2CAS-1WE, CBR, No parity
ldi:20      #0x62e,r1         //  dmcr5 register address setting
sth         r0,@r1            //  dmcr5 register write
init_rfcr    ldi:20      #0x0205,r0        //  REL=2, R1W/R3W-wait none, 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          //  CH2 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   //  ch1 address
ldi:32      #0x00151300, r1   //  ch2 address
ldi:32      #0x00196434, r2   //  ch4 address (within the page)
ldi:32      #0x0019657c, r3   //  ch4 address (within the page)
ldi:32      #0x00196600, r4   //  ch4 address (outside of the page)
ldi:32      #0x001a6818, r5   //  ch5 address (within the page)
ldi:32      #0x001a6b8c, r6   //  ch5 address (within the page)
ldi:32      #0x001a6c00, r7   //  ch5 address (outside of the page)
bus_acc     ld          @r0,r8            //  ch1 data word load
lduh        @r1,r9            //  ch2 data half word load
ld          @r2,r10           //  ch4 data word load
ldub        @r3,r11           //  ch4 data byte load
st          r8,@r4            //  ch4 data word store
sth         r9,@r5            //  ch5 data half word store
st          r10,@r6           //  ch5 data word store
stb         r11,@r7           //  ch5 data byte store
```

CHAPTER 5

I/O Port

This chapter describes an outline of the I/O port and the register configuration/functions.

5.1 Overview of I/O Port

The MB91191/MB91192 series have the 102 output ports. In terms of ports, there are ports 2, 3, 5, and 6 that are also used for external bus functions, and ports 0, 1, 4, and 7 to D that are also used for peripheral functions. Ports other than ports 0, 2, and 3 have a function selection bit per bit, and a port function, peripheral function, or external bus function can be selected.

■ Register List of I/O Port

Figure 5.1-1 Register list of I/O Port

bit 7 ← → 0

Address: 000000H	PDR3	Port 3 data register
000001H	PDR2	Port 2 data register
000002H	PDR1	Port 1 data register
000003H	PDR0	Port 0 data register
000004H	PDR7	Port 7 data register
000005H	PDR6	Port 6 data register
000006H	PDR5	Port 5 data register
000007H	PDR4	Port 4 data register
000010H	PDRB	Port B data register
000011H	PDRA	Port A data register
000012H	PDR9	Port 9 data register
000013H	PDR8	Port 8 data register
000016H	PDRD	Port D data register
000017H	PDRC	Port C data register

000008H	DDR3	Port 3 Direction register
000009H	DDR2	Port 2 Direction register
00000AH	DDR1	Port 1 Direction register
00000BH	DDR0	Port 0 Direction register
00000CH	DDR7	Port 7 Direction register
00000DH	DDR6	Port 6 Direction register
00000EH	DDR5	Port 5 Direction register
00000FH	DDR4	Port 4 Direction register
000018H	DDRB	Port B Direction register
000019H	DDRA	Port A Direction register
00001AH	DDR9	Port 9 Direction register
00001BH	DDR8	Port 8 Direction register
00001EH	DDRD	Port D Direction register
00001FH	DDRC	Port C Direction register

000602H	PFS1	Port 1 function selection register

000604H	PFS7	Port 7 function selection register
000605H	PFS6	Port 6 function selection register
000606H	PFS5	Port 5 function selection register
000607H	PFS4	Port 4 function selection register
000020H	PIEB	Port B input enable register
000021H	PIEA	Port A input enable register
000022H	PFS9	Port 9 function selection register
000023H	PFS8	Port 8 function selection register
000026H	PFSD	Port D function selection register
000027H	PFSC	Port C function selection register

5.2 Port 0

Port 0 is the input/output port, and is also used for servo input, capture input, and PWC input.

■ Functions of I/O Port 0

Each port has two registers per bit, namely the data direction register (DDR) and port data register (PDR), and input/output can be set independently per bit. Pins whose DDR is "1" are set to Output, whereas Pins whose DDR is "0" are set to Input. The PDR is undefined by a reset, and the DDR is cleared to "0" and set to Input. The value written to the PDR is output to the pin at the time of output setup by specifying the DDR per bit.

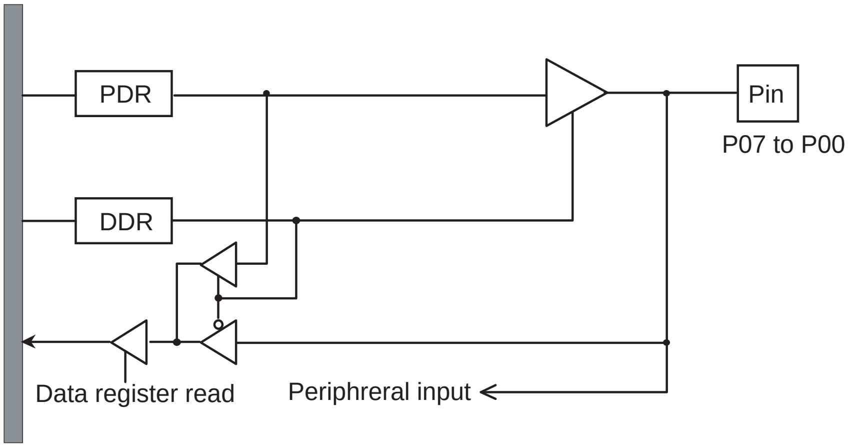
At this time, the contents of the PDR are read from the read value of the PDR.

At the time of the input setup, the pin will have high impedance status.

At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 0

Figure 5.2-1 Block Diagram of Port 0



■ Registers for Port 0

- ● Port 0 data register (PDR0)

Figure 5.2-2 Port 0 data register (PDR0)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000003$_H$	PD07	PD06	PD05	PD04	PD03	PD02	PD01	PD00	XXXX XXXX$_B$
Access→	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- ● Port 0 direction register (DDR0)

Figure 5.2-3 Port 0 direction register (DDR0)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000B$_H$	DR07	DR06	DR05	DR04	DR03	DR02	DR01	DR00	0000 0000$_B$
Access→	W	W	W	W	W	W	W	W	

5.3 Port 1

Port 1 is an input/output port, and is also used for RTG output, timer clock/external interrupt input, and PWC input.

■ Functions of I/O Port 1

The port has three registers per bit, namely DDR, PDR, and PFS (port function select register), and the port input/output setup and function selection can be executed independently per bit. The peripheral function is selected for pins whose PFS is "1", whereas the port function is selected for those whose PFS is "0". Pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input.

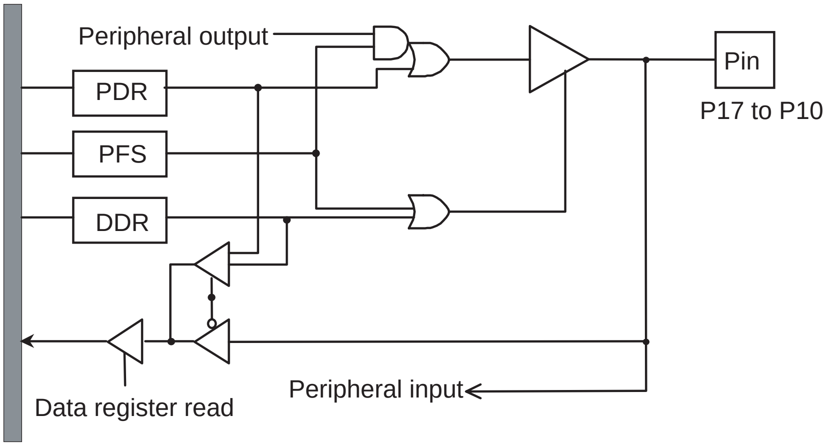
The PFS is initialized by a reset and the port function is selected. Also, the PDR is undefined, and the DDR is cleared to "0" and the port is specified to Input.

The value written to the PDR is output to the pin at the time of output setup by specifying the DDR per bit. At this time, the contents of the PDR are read from the read value of the PDR.

At the time of the input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 1

Figure 5.3-1 Block Diagram of Port 1



■ Registers for Port 1

- ● Port 1 data register (PDR1)

Figure 5.3-2 Port 1 data register (PDR1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000002H	PD17	PD16	PD15	PD14	PD13	PD12	PD11	PD10	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- ● Port 1 direction register (DDR1)

Figure 5.3-3 Port 1 direction register (DDR1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000AH	DR17	DR16	DR15	DR14	DR13	DR12	DR11	DR10	0000 0000B
Access→	W	W	W	W	W	W	W	W	

- ● Port 1 function selection register (PFS1)

Figure 5.3-4 Port 1 function selection register (PFS1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000602H	RT4E	RT3E	RT2E	RT1E	RT0E	−	−	−	0000 0000B
Access→	W	W	W	W	W	W	W	W	

Table 5.3-1 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]: RT4E	General-purpose port	RTG output 4
[bit6]: RT3E	General-purpose port	RTG output 3
[bit5]: RT2E	General-purpose port	RTG output 2
[bit4]: RT1E	General-purpose port	RTG output1
[bit3]: RT0E	General-purpose port	RTG output0
[bit2]:	General-purpose port/EC5, INT1 input	Specification prohibited
[bit1]:	General-purpose port/EC5, INT1 input	Specification prohibited
[bit0]:	General-purpose port/PMSK input	Specification prohibited

5.4 Port 2, 3

Ports 2 and 3 function as input/output ports under single-chip mode, and function as the address/data bus under external bus mode.

■ Functions of I/O Port 2, 3

Each port has two registers per bit, namely DDR and PDR, and input/output can be set independently per bit. Pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input.

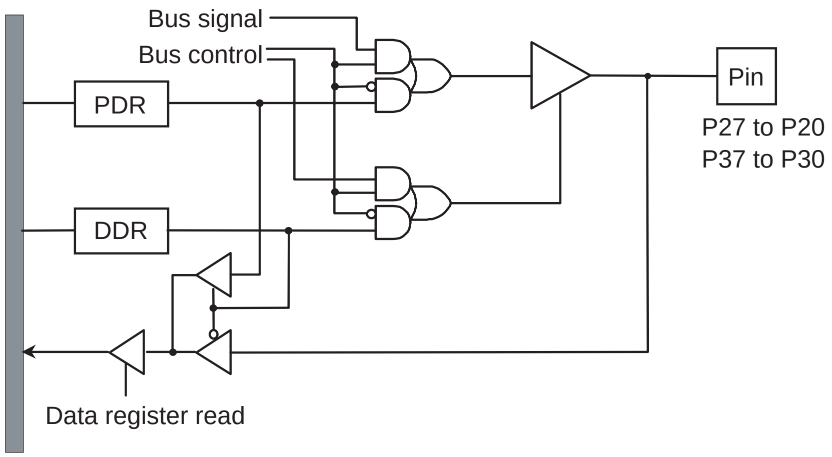
The PDR is undefined by a reset the DDR is cleared to "0" and set to Input.

In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of the input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 2, 3

Figure 5.4-1 Block Diagram of Port 2, 3



■ Registers for Port 2, 3

- Port 2, 3 data register (PDR2, 3)

Figure 5.4-2 Port 2 data register (PDR2)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000001H	PD27	PD26	PD25	PD24	PD23	PD22	PD21	PD20	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.4-3 Port 3 data register (PDR3)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000000H	PD37	PD36	PD35	PD34	PD33	PD32	PD31	PD30	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Port 2, 3 direction register (DDR2, 3)

Figure 5.4-4 Port 2 direction register (DDR2)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000009H	DR27	DR26	DR25	DR24	DR23	DR22	DR21	DR20	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.4-5 Port 3 direction register (DDR3)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000008H	DR37	DR36	DR35	DR34	DR33	DR32	DR31	DR30	0000 0000B
Access →	W	W	W	W	W	W	W	W	

5.5 Port 5

Port 5 is the input/output port and is also used for the external bus function.

■ Functions of Port 5

The port has three registers per bit, namely DDR, PDR, and PFS, and the port input/output setup and function selection can be executed independently per bit. The external bus function is selected for pins whose PFS is "1", whereas the port function is selected for those whose PFS is "0". Pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input.

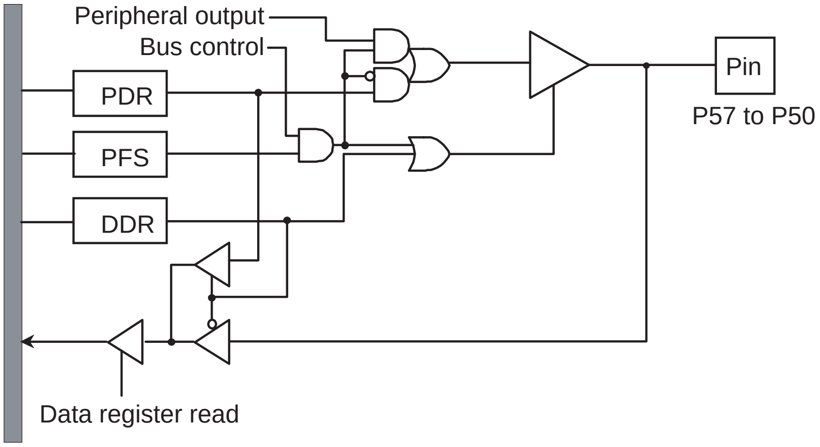
The PFS is initialized by a reset and the port function is selected. Also, the PDR is undefined and the DDR is cleared to "0" and the port is specified to Input.

In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of the input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 5

Figure 5.5-1 Block Diagram of Port 5



■ Registers for Port 5

- ● Port 5 data register (PDR5)

Figure 5.5-2 Port 5 data register (PDR5)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000006H	PD57	PD56	PD55	PD54	PD53	PD52	PD51	PD50	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- ● Port 5 direction register (DDR5)

Figure 5.5-3 Port 5 direction register (DDR5)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000EH	DR57	DR56	DR55	DR54	DR53	DR52	DR51	DR50	0000 0000B
Access →	W	W	W	W	W	W	W	W	

- ● Port 5 function selection register (PFS5)

Figure 5.5-4 Port 5 function selection register (PFS5)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000606H	A15E	A14E	A13E	A12E	A11E	A10E	A09E	A08E	1111 1111B
Access →	W	W	W	W	W	W	W	W	

Table 5.5-1 Operation of function selection bits

Function selection bit	Register setting value	
	0	1
[bit7]: A15E	General-purpose port	Address (A15) output *
[bit6]: A14E	General-purpose port	Address (A14) output *
[bit5]: A13E	General-purpose port	Address (A13) output *
[bit4]: A12E	General-purpose port	Address (A12) output *
[bit3]: A11E	General-purpose port	Address (A11) output *
[bit2]: A10E	General-purpose port	Address (A10) output *
[bit1]: A09E	General-purpose port	Address (A09) output *
[bit0]: A08E	General-purpose port	Address (A08) output *

Note:

The * setting is only valid when the external bus mode is specified. The general-purpose port function is always selected for single-chip mode.

5.6 Port 6, 7

Ports 6 and 7 are input/output ports, and are also used for the external bus function and timer output function.

■ Port 6, 7

The port has three register per bit, namely DDR, PDR and PFS, and port input/output setup and function selection can be executed independently per bit.

The external bus function or peripheral function is selected for pins whose PFS is "1", whereas the port function is selected for those whose PFS is "0". Pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input.

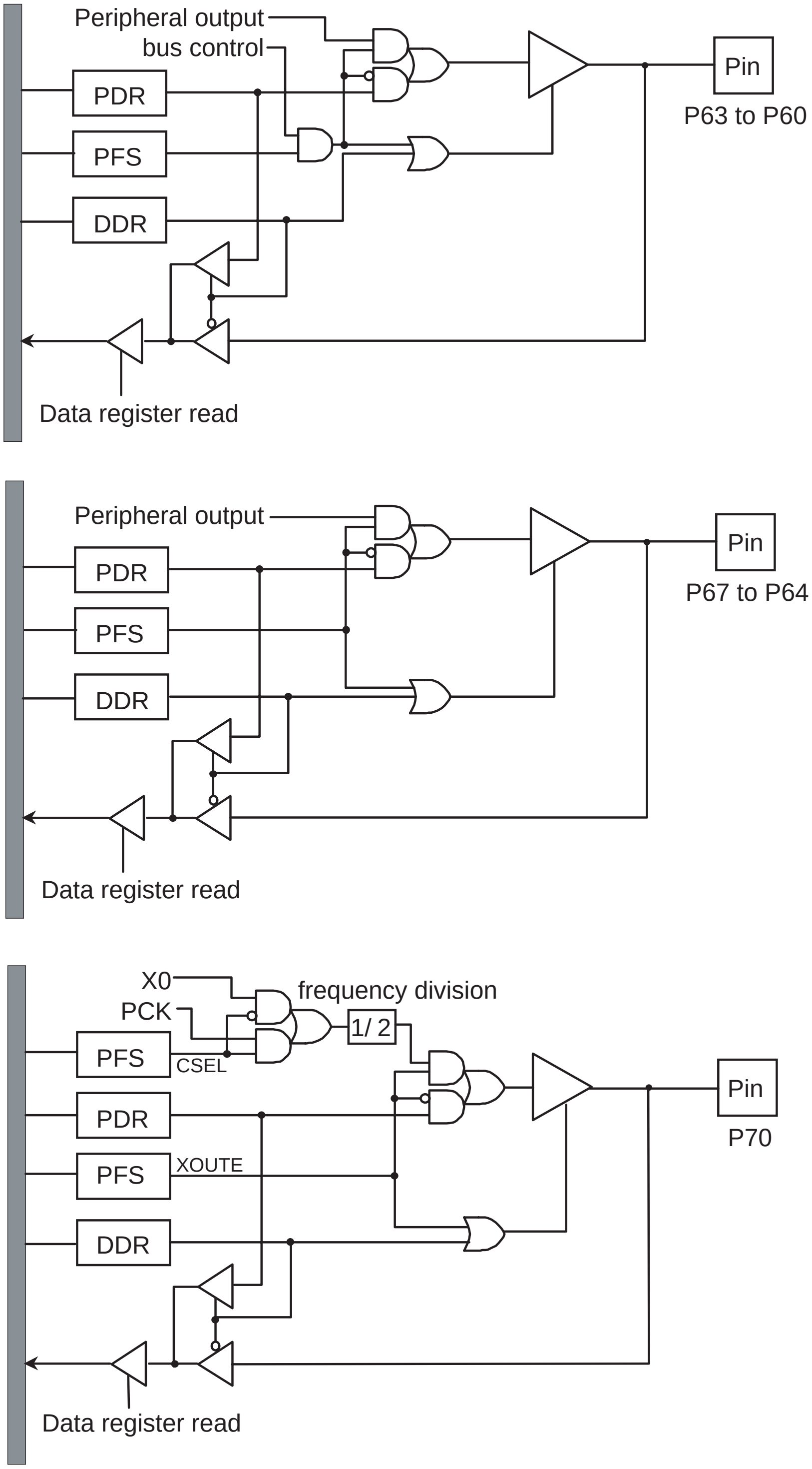
The PFS is initialized by a reset and the port function is selected. Also, the PDR is undefined and the DDR is cleared to "0" and the port is specified to input.

In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 6, 7

Figure 5.6-1 Block Diagram of Port 6, 7



■ Registers for Port 6, 7

- ● Port 6, 7 data register (PDR6, 7)

Figure 5.6-2 Port 6 data register (PDR6)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000005H	PD76	PD66	PD65	PD64	PD63	PD62	PD61	PD60	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.6-3 Port 7 data register (PDR7)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000004H	–	–	–	–	–	–	–	PD70	---- ---XB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- ● Port 6, 7 direction register (DDR6, 7)

Figure 5.6-4 Port 6 direction register (DDR6)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000DH	DR67	DR66	DR65	DR64	DR63	DR62	DR61	DR60	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.6-5 Port 7 direction register (DDR7)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000CH	–	–	–	–	–	–	–	DR70	---- ---0B
Access →	W	W	W	W	W	W	W	W	

- ● Port 6, 7 function selection register (PFS6, 7)

Figure 5.6-6 Port 6 function selection register (PFS6)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000605H	T4OE	T5O1E	–	–	ALEE	WR1XE	WR0XE	RDXE	0000 1111B
Access →	W	W	W	W	W	W	W	W	

Table 5.6-1 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]: T4OE	General-purpose port	T4O output
[bit6]: T5O1E	General-purpose port	T5O output
[bit5]:	General-purpose port	Setting disabled
[bit4]:	General-purpose port	Setting disabled
[bit3]: ALEE	General-purpose port	ALE output *
[bit2]: WR1XE	General-purpose port	$\overline{\text{WR1}}$ output *
[bit1]: WR0XE	General-purpose port	$\overline{\text{WR0}}$ output *
[bit0]: RDXE	General-purpose port	$\overline{\text{RD}}$ output *

Note:

The * setting is only valid when the external bus mode is specified. The general-purpose port function is always selected for single-chip mode.

Figure 5.6-7 Port 7 function selection register (PFS7)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000604H	–	–	–	–	–	–	CSEL	XOUTE	---- --00B
Access →	–	–	–	–	–	–	W	W	

Table 5.6-2 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]:		
[bit6]:		
[bit5]:		
[bit4]:		
[bit3]:		
[bit2]:		
[bit1]: CSEL	fch/2 output (X0)	PCK/2 output (OSCI)
[bit0]: XOUTE	General-purpose port	Clock output enabled

5.7 Port 4, 8, 9

Ports 4, 8 and 9 are input/output ports, and are also used for PPG output and general-purpose prescaler output.

■ Functions of Port 4, 8, 9

The port has three registers per bit, namely DDR, PDR, and PFS, and port input/output setup and function selection can be executed independently per bit. The peripheral function is selected for pins whose PFS is "1", whereas the port function is selected for those whose PFS is "0". Pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are se to Input.

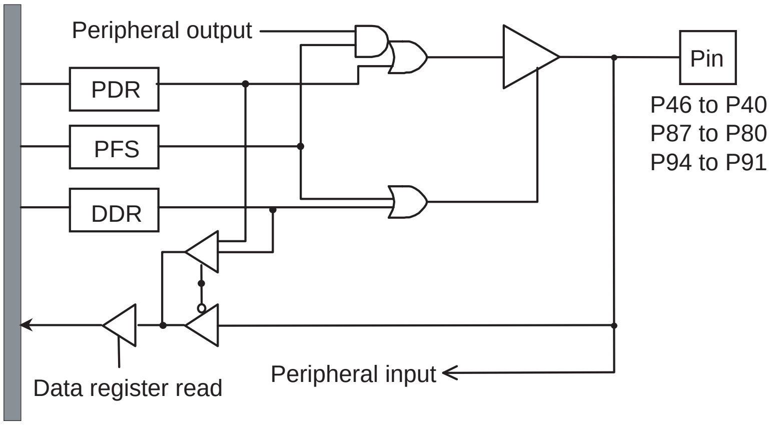
The PFS is cleared to "0" by reset, and the port function is selected, the PDR is undefined, and the DDR is cleared to "0" and set to Input.

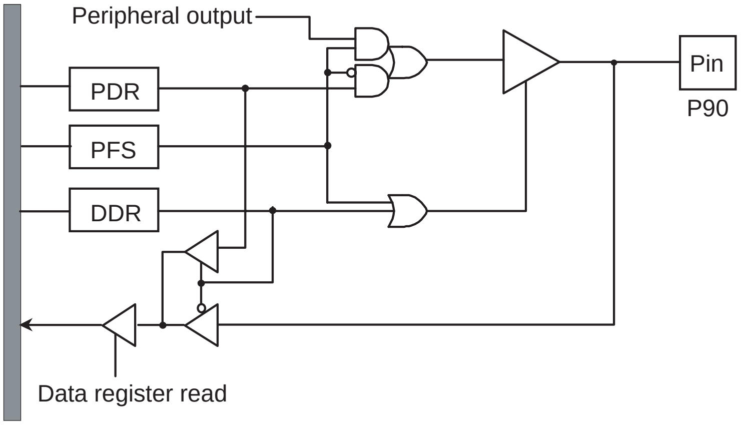
In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port 4, 8, 9

Figure 5.7-1 Block Diagram of Port 4, 8, 9





■ Registers for Port 4, 8, 9

- Port 4, 8, 9 data register (PDR4, 8, 9)

Figure 5.7-2 Port 4 data register (PDR4)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000007H	PD47	PD46	PD45	PD44	PD43	PD42	PD41	PD40	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.7-3 Port 8 data register (PDR8)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000013H	PD87	PD86	PD85	PD84	PD83	PD82	PD81	PD80	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.7-4 Port 9 data register (PDR9)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000012H	–	–	–	PD94	PD93	PD92	PD91	PD90	---X XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Port 4, 8, 9 direction register (DDR4, 8, 9)

Figure 5.7-5 Port 4 direction register (DDR4)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00000FH	DR47	DR46	DR45	DR44	DR43	DR42	DR41	DR40	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.7-6 Port 8 direction register (DDR8)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00001BH	DR87	DR86	DR85	DR84	DR83	DR82	DR81	DR80	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.7-7 Port 9 direction register (DDR9)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00001AH	–	–	–	DR94	DR93	DR92	DR91	DR90	---0 0000B
Access →	W	W	W	W	W	W	W	W	

- Port 4, 8, 9 function selection register (PFS4, 8, 9)

Figure 5.7-8 Port 4 function selection register (PFS4)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000607H	–	PG18E	PG17E	PG16E	PG15E	PG14E	PG13E	PG12E	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Table 5.7-1 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]:	General-purpose port	Setting disabled
[bit6]: PG18E	General-purpose port	PPG output 18
[bit5]: PG17E	General-purpose port	PPG output 17
[bit4]: PG16E	General-purpose port	PPG output 16
[bit3]: PG15E	General-purpose port	PPG output 15
[bit2]: PG14E	General-purpose port	PPG output 14
[bit1]: PG13E	General-purpose port	PPG output 13
[bit0]: PG12E	General-purpose port	PPG output 12

Figure 5.7-9 Port 8 function selection register (PFS8)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000023H	PG11E	PG10E	PG09E	PG08E	PG07E	PG06E	PG05E	PG04E	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Table 5.7-2 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]: PG11E	General-purpose port	PPG output 11
[bit6]: PG10E	General-purpose port	PPG output 10
[bit5]: PG09E	General-purpose port	PPG output 09
[bit4]: PG08E	General-purpose port	PPG output 08
[bit3]: PG07E	General-purpose port	PPG output 07
[bit2]: PG06E	General-purpose port	PPG output 06
[bit1]: PG05E	General-purpose port	PPG output 05
[bit0]: PG04E	General-purpose port	PPG output 04

Figure 5.7-10 Port 9 function selection register (PFS9)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000022H	–	–	–	PG03E	PG02E	PG01E	PG00E	POE	---0 0000B
Access →	–	–	–	W	W	W	W	W	

Table 5.7-3 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]:		
[bit6]:		
[bit5]:		
[bit4]: PG03E	General-purpose port	PPG output 03
[bit3]: PG02E	General-purpose port	PPG output 02
[bit2]: PG01E	General-purpose port	PPG output 01
[bit1]: PG00E	General-purpose port	PPG output 00
[bit0]: POE	General-purpose port	General-purpose prescaler output

5.8 Port A, B

Ports A and B are the input/output ports, and analog input and key input (for port A only) are shared.

■ Functions of Port A, B

The port has three registers per bit, namely, DDR, PDR, and PIE (port input enable register), and input/output can be selected independently per bit. In terms of port input/output, pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input. To be used as a general-purpose port, port input is enabled by setting "1" to the PIE.

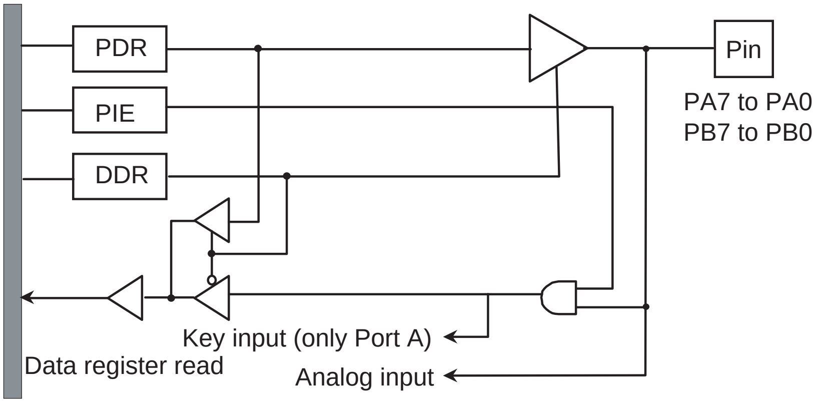
The PIE is cleared to "0" by reset, and the port function is disabled. The PDR is undefined, and the DDR is cleared to "0" and set to Input.

In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port A, B

Figure 5.8-1 Block Diagram of Port A, B



■ Registers for Port A, B

● Port A, B data register (PDRA, B)

Figure 5.8-2 Port A data register (PDRA)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000011H	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.8-3 Port B data register (PDRB)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000010H	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0	XXXX XXXX
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

● Port A, B direction register (DDRA, B)

Figure 5.8-4 Port A direction register (DDRA)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000019H	DRA7	DRA6	DRA5	DRA4	DRA3	DRA2	DRA1	DRA0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.8-5 Port B direction register (DDRB)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000018H	DRB7	DRB6	DRB5	DRB4	DRB3	DRB2	DRB1	DRB0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

● Port A, B input enable register (PIEA, B)

Figure 5.8-6 Port A input enable register (PIEA)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000021H	PIA7	PIA6	PIA5	PIA4	PIA3	PIA2	PIA1	PIA0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.8-7 Port B input enable register (PIEB)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000020H	PIB7	PIB6	PIB5	PIB4	PIB3	PIB2	PIB1	PIB0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

When the port is used for input functions, input is enabled by setting "1" to the supported bit.

Set "0" for the port is to be used as the analog input.

5.9 Port C, D

Ports C and D are input/output ports, and are also used as the PWM output, serial I/O, and external interrupt.

■ Functions of Port C, D

The port has three registers per bit, namely, DDR, PDR, and PFS and input/output, setup and function selection can be executed independently per bit. The peripheral function is selected for pin whose PFS is "1", whereas the port function is selected for these whose PFS is "0".

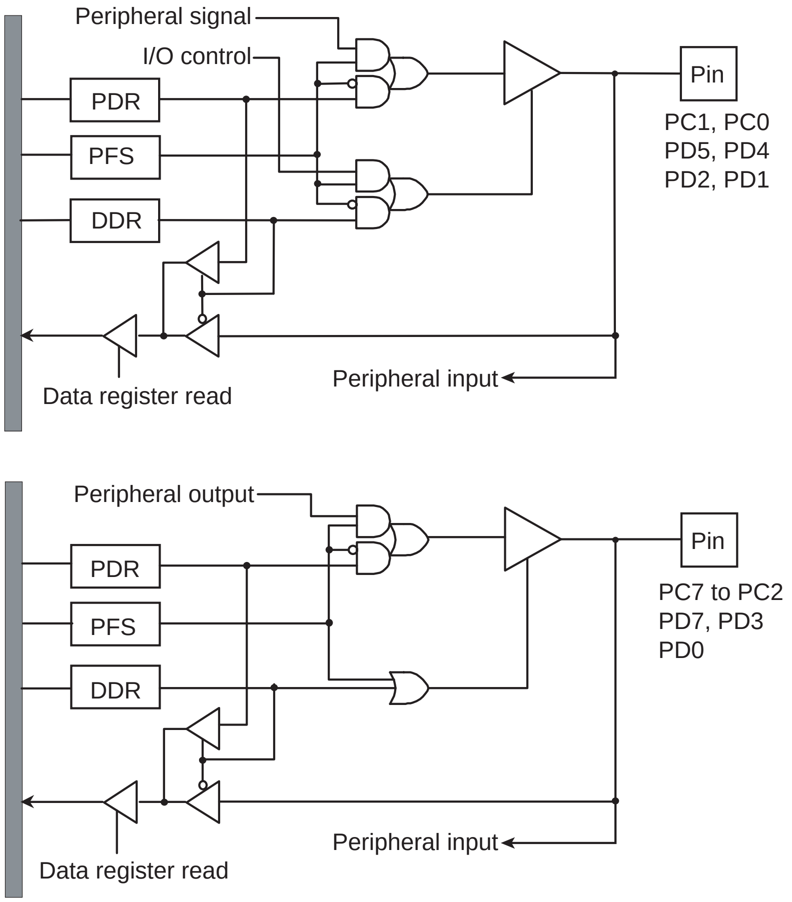
The pins whose DDR is "1" are set to Output, whereas pins whose DDR is "0" are set to Input. The PFS is cleared to "0" by reset, and the port function is selected. The PDR is undefined, and the DDR is cleared to "0" and set to Input.

In terms of the DDR specification per bit, for output setup, the value written to the PDR is output to the pin, and the PDR contents are read as the read value of the PDR in this case.

At the time of input setup, the pin will have high impedance status. At this time, the pin level is read from the read value of the PDR.

■ Block Diagram of Port C, D

Figure 5.9-1 Block Diagram of Port C, D



■ Registers for Port C, D

- ● Port C, D data register (PDRC, D)

Figure 5.9-2 Port C data register (PDRC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000017H	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	XXXX XXXX
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Figure 5.9-3 Port D data register (PDRD)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000016H	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- ● Port C, D direction register (DDRC, D)

Figure 5.9-4 Port C direction register (DDRC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00001FH	DRC7	DRC6	DRC5	DRC4	DRC3	DRC2	DRC1	DRC0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Figure 5.9-5 Port D direction register (DDRD)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00001EH	DRD7	DRD6	DRD5	DRD4	DRD3	DRD2	DRD1	DRD0	0000 0000B
Access →	W	W	W	W	W	W	W	W	

- ● Port C, D function selection register (PFSC, D)

Figure 5.9-6 Port C input enable register (PFSC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000027H	PWM00E	PWM01E	PWM02E	PWM10E	PWM11E	PWM12E	SCK0E	S00E	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Table 5.9-1 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]: PWM00E	General-purpose port	PWM00 output
[bit6]: PWM01E	General-purpose port	PWM01 output
[bit5]: PWM02E	General-purpose port	PWM02 output
[bit4]: PWM10E	General-purpose port	PWM10 output
[bit3]: PWM11E	General-purpose port/SCS2 input	PWM11 output
[bit2]: PWM12E	General-purpose port/SCS1 input	PWM12 output
[bit1]: SCK0E	General-purpose port	SCK0 input/output
[bit0]: SO0E	General-purpose port	SO0 output

Figure 5.9-7 Port D function selection register (PFSD)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000026H	−	−	SCK1E	SO1E	−	SCK2E	SO2E	−	0000 0000B
Access →	W	W	W	W	W	W	W	W	

Table 5.9-2 Operation of function selection bits

Function selection bits	Register setting value	
	0	1
[bit7]:	General-purpose port/SI0 input	Setting prohibited
[bit6]:	General-purpose port/SCS0 input	Setting prohibited
[bit5]: SCK1E	General-purpose port	SCK1 input/output
[bit4]: SO1E	General-purpose port	SO1 output
[bit3]:	General-purpose port/SI1, INT2 input	Setting prohibited
[bit2]: SCK2E	General-purpose port	SCK2 input/output
[bit1]: SO2E	General-purpose port	SO2 output
[bit0]:	General-purpose port/SI2 input	Setting prohibited

CHAPTER 6

FG Input

This chapter describes an outline of the FG input section, the register configuration/functions, and their operation.

6.1 Overview of FG Input

The Frequency Generate (FG) input section comprises of the capstan input, drum input, and reel input sections. Each input section performs division of the input FG signal, generation of signals to be input to the FRC capture section and masking of the predetermined period.

■ Capstan Input

The capstan input consists of the following 6 items.

- Capstan control register
- Capstan input control register
- Capstan mask timer control register
- Multiplying circuit
- 8 -bit programmable divider
- Mask timer

■ Drum Input

The drum input consists of the following 5 items.

- Drum control register
- Drum input control register
- Drum mask timer control register
- 4-bit programmable divider
- Mask timer

■ Reel Input

The reel input consists of the following 7 items.

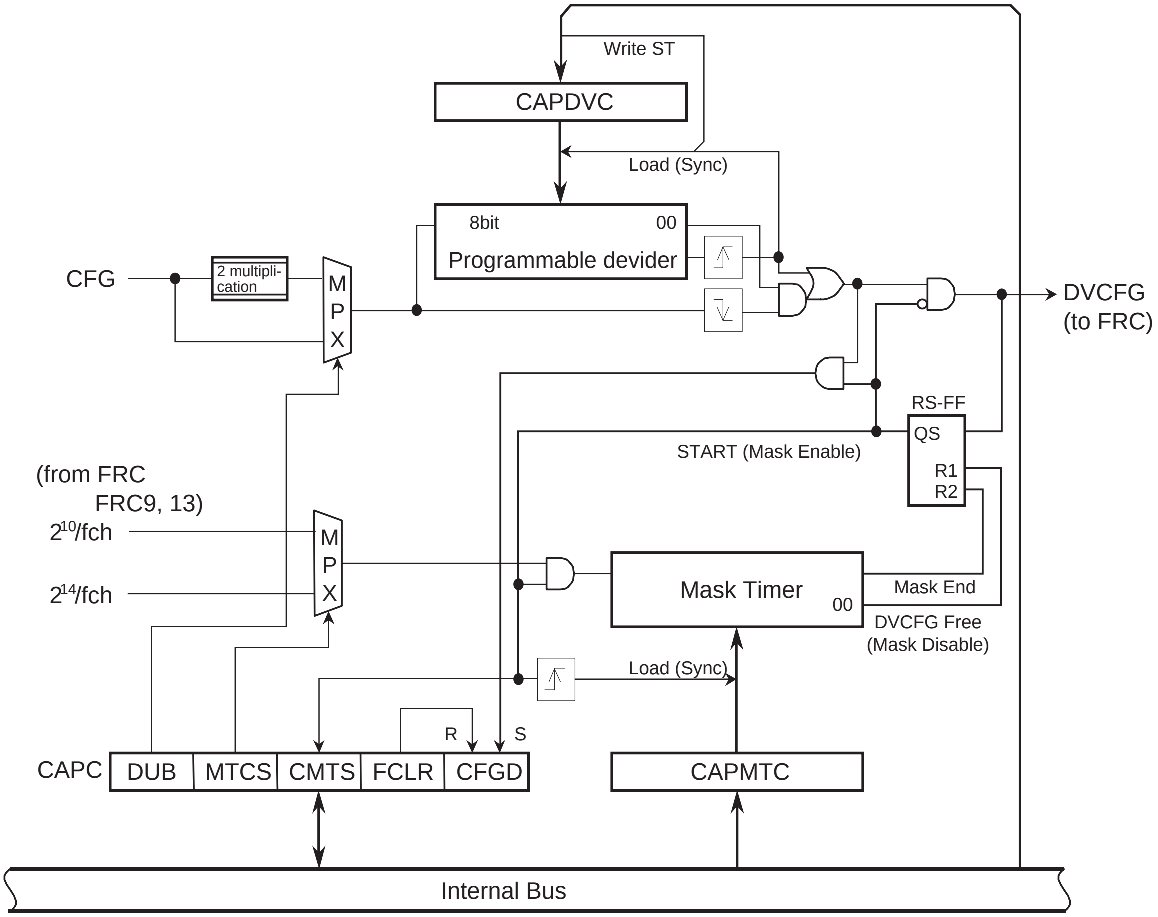
- Reel control register
- Reel 0 input control register
- Reel 0 mask timer control register
- Reel 1 input control register
- Reel 1 mask timer control register
- 8-bit programmable divider
- Mask timer

6.2 Capstan Input

The capstan input section is comprised of a multiplying circuit, 8-bit programmable divider, and mask timer. This section explains the operation of each section and control register.

■ Block Diagram of Capstan Input

Figure 6.2-1 Block Diagram of Capstan Input



■ Register List of Capstan Input

Figure 6.2-2 Register list of Capstan Input

bit 7 ←→ 0

Address:	000050H	CAPDVC	Capstan input control register
	000051H	CAPMTC	Capstan timer control register
	000052H	CAPC	Capstan control register

■ Capstan Control Register (CAPC)

Figure 6.2-3 Capstan Control Register (CAPC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000052H	CFGD	FCLR	CMTS	MTCS	–	–	–	DUB	X1X0 ---0B
Access →	R	W	R	R/W					

[bit7]:CFGD

This is the capstan edge detection flag during capstan masking.

0	Without edge detection
1	With edge detection

[bit6]:FCLR

This is the capstan input edge detection flag clear bit.

0	Clear the CFGD flag.
1	None

The read value of this bit is always "1".

[bit5]:CMTS

This is the capstan mask timer status flag.

0	Mask released
1	Masking

[bit4]:MTCS

This is clock source selection bit of mask timer.

	Selection Clock	in fch:@20MHz
0	2^{10}/fch (FRC9)	51.2 µs
1	2^{14}/fch (FRC13)	819.2 µs

[bit3 to 1]:

It is an unused bit.

[bit0]:DUB

This is the CFG input multiplication selection bit.

0	None
1	2 multiplication

■ Capstan Input Control Register (CAPDVC)

Figure 6.2-4 Capstan Input Control Register (CAPDVC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000050_H	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0	XXXX $XXXX_B$
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:DIV7 to 0

Executes division control of the capstan input and edge detection control depending on the set value.

Set value	Division control	Edge detection
00_H	None	Both edge detection
01_H	1-frequency division	Rising edge detection
02_H	2-frequency division	
03_H	3-frequency division to	
to	to	
FD_H	253-frequency division	
FE_H	254-frequency division	
FF_H	255-frequency division	

■ Capstan Mask Timer Control Register (CAPMTC)

Figure 6.2-5 Capstan mask timer control register (CAPMTC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000051_H	D7	D6	D5	D4	D3	D2	D1	D0	XXXX $XXXX_B$
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:D7 to 0

Executes masking period control for the capstan input depending on the set value. When ΦMT is specified as the clock cycle time selected by the mask timer clock select (CS) bit of the capstan control register, and N is specified as the set value, the capstan input masking period TM is as follow.

$TM = \Phi MT \times N \pm \Phi MT/2$

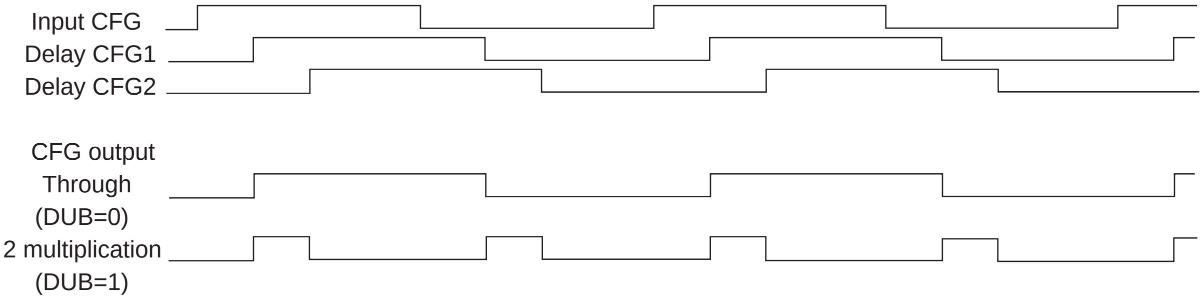
However, N is set to 0, the mask processing does not perform.

■ Operation of Capstan Input

● Operation of Multiplying circuit

The capstan FG (CFG) input is input to the 8-bit programmable divider via "Slew" or double circuit by setting the multiplication select (DUB) bit of the capstan control register (CAPC). The input CFG signal is edge detected after division that was set by the capstan control register (CAPDVC) is performed, and then passes through the capstan mask timer, and is connected to the FRC capture unit as the DVCFG signal.

Figure 6.2-6 CFG output by multiplication selection



● Operation of 8-bit Programmable divider

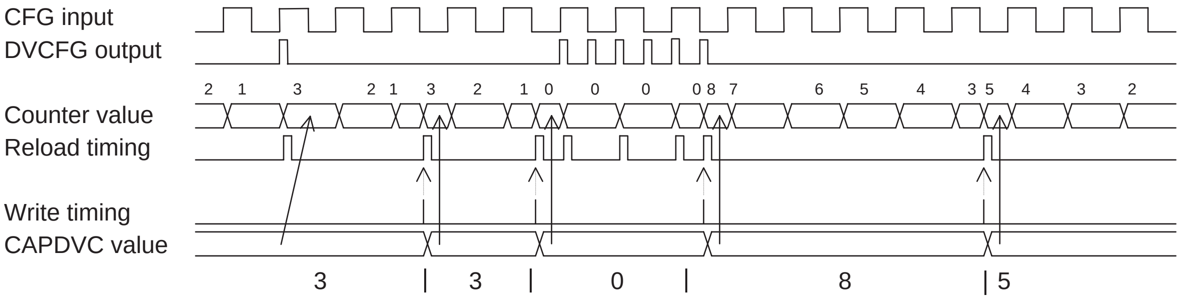
The programmable divider can be divided within the range between 1 to 255 by the value set to the capstan input control register (CAPDVC).

When the value set to CAPDVC is "00_H", division is not executed, and both edges of the CFG input are directly output as the DVCFG signal. When values other than "00_H" are set to CAPDVC, the output divided by the rising edge of the CFG input is output as the DVCFG signal.

When the value set to CAPDVC is updated during division, the counter value is updated immediately after the update. In this case, the DVCFG output is masked if the counter is updated simultaneously with the DVCFG output (reload operation).

Figure 6.2-7 shows the sequence of the programmable divider.

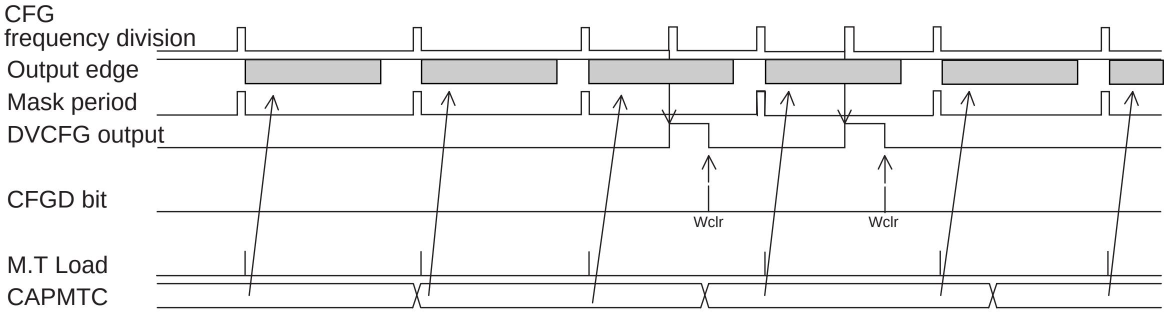
Figure 6.2-7 Operation timing diagram of programmable divider



● Operation of Mask Timer

The mask timer loads the CAPMTC value and starts counting, and also masks the DVCFG output after detecting the CFG division output edge. When the DVCFG output is detected during the masking period, the CFGD flag is set. Even if the CAPMTC value is updated during the masking period, load operation to the mask timer will not be executed. This updated value will be valid from the next masking period.

Figure 6.2-8 Operation timing diagram of Mask Timer



● Limitation on using

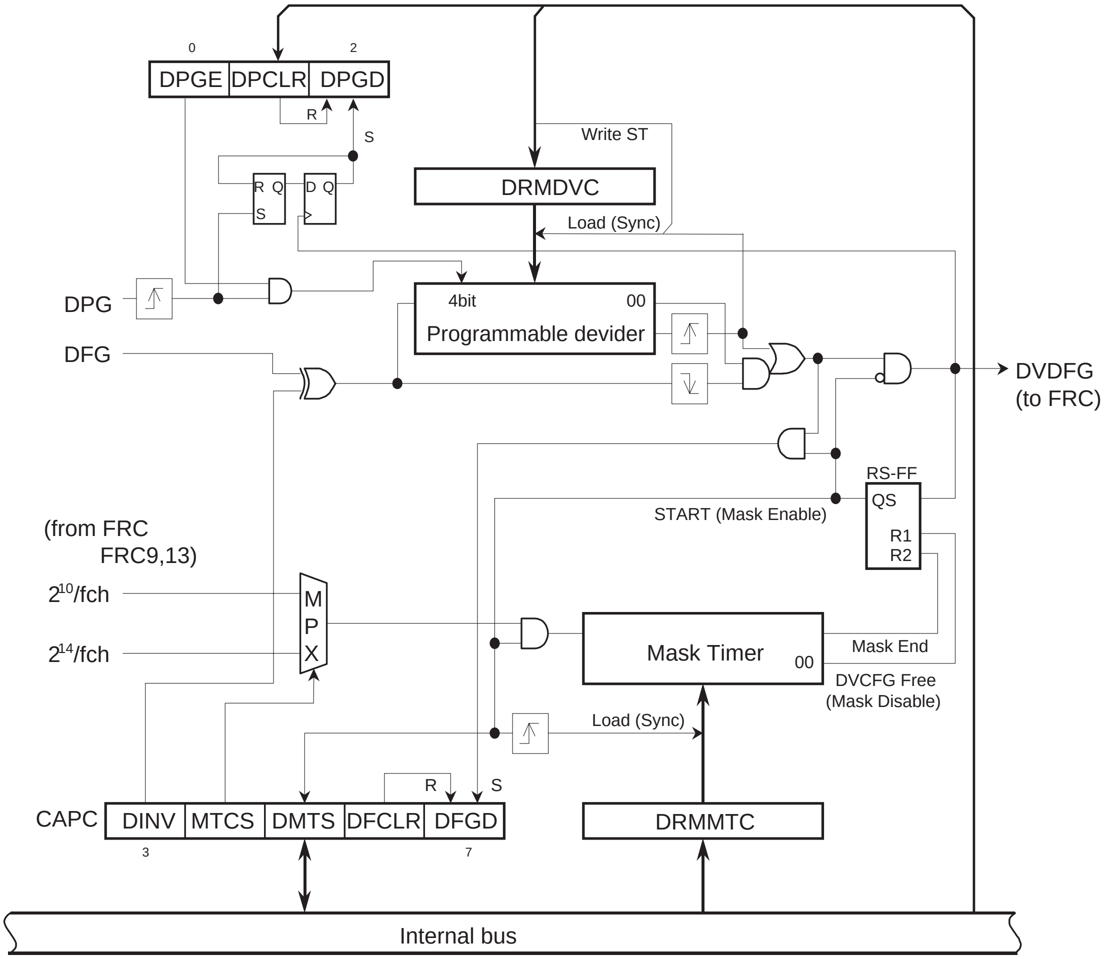
When double mode is specified for the capstan input, do not set "00_H" to the capstan input control register. It may cause the malfunction.

6.3 Drum Input

The drum input section comprises of the 4-bit programmable divider and mask timer. This section explains the operation of each section and control register.

■ Block Diagram of Drum Input

Figure 6.3-1 Block diagram of Drum Input



■ Register List of Drum Input

Figure 6.3-2 Register list of Drum Input

bit 7 ←→ 0

Address:	000053H	DRMC	Drum control register
	000054H	DRMDVC	Drum input control register
	000055H	DRMMTC	Drum mask timer control register

■ Drum Control Register (DRMC)

Figure 6.3-3 Drum Control Register (DRMC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000053_H	DFCG	DFCLR	DMTS	MTCS	DINV	DPGD	DPCLR	DPGE	$X1X0\ 0X10_B$
Access →	R	W	R	R/W	R/W	R	W	R/W	

[bit7]:DFGD

This is the division FG edge detection flag during drum masking.

0	Without edge detection
1	With edge detection

[bit6]:DFCLR

This is the edge detection flag clear bit during drum masking.

0	Clear the DFGD flag.
1	None

The read value of this bit is always "1".

[bit5]:DMTS

This is the drum mask timer status flag.

0	Mask released
1	Masking

[bit4]:MTCS

This is clock source selection bit of mask timer.

	Selection Clock	in fch:@20MHz
0	2^{10}/fch (FRC9)	51.2 µs
1	2 ^{14}/fch (FRC13)	819.2 µs

[bit3]:DINV

This is the polarity control bit of the DFG input signal.

0	Input through
1	Input inversion

[bit2]:DPGD

It is DPG input edge detection flag.

0	DPG Input None
1	DPG input

[bit1]:DPCLR

This is the clear bit of DPG input edge detection flag.

0	Clear the DPGD flag.
1	None

The read value of this bit is always "1".

[bit0]:DPGE

This is the initialization control bit for the programmable divider using DPG input.

0	Not initialize by DPG input
1	Initialize by DPG input

■ Drum Input Control Register (DRMDVC)

Figure 6.3-4 Drum Input Control Register (DRMDVC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000054_H	–	–	–	–	DIV3	DIV2	DIV1	DIV0	---- $XXXX_B$
Access →	–	–	–	–	–	R/W	R/W	R/W	

[bit7 to 4]:

It is an unused bit.

[bit3 to 0]:DIV3 to 0

Division control of the drum input and edge detection control are performed by the set value.

Set value	Division control	Edge detection
00_H	None	Both edge detection
01_H	1-frequency division	Rising edge detection
02_H	2-frequency division	
03_H	3-frequency division	
to	to	
$0D_H$	13-frequency division	
$0E_H$	14-frequency division	
$0F_H$	15-frequency division	

■ Drum Mask Timer Control Register (DRMMTC)

Figure 6.3-5 Drum mask timer control register (DRMMTC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000055H	D7	D6	D5	D4	D3	D2	D1	D0	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:D7 to 0

The masking period control of the drum input is performed by the set value.

When Φ is specified as the clock cycle time selected by the mask timer clock select (CS) bit of the drum control register, and N is specified as the set value, the drum input masking period TM is as follow.

TM= ΦMT × N ± ΦMT/2. However, N is set to 0, the mask processing does not perform.

■ Operation of Drum Input

● 4-bit Programmable divider

The programmable divider can perform division within the range 1 to 15 using the value set to the drum input control register (DRMDVC).

Refer to the capstan input section for operation of the programmable divider.

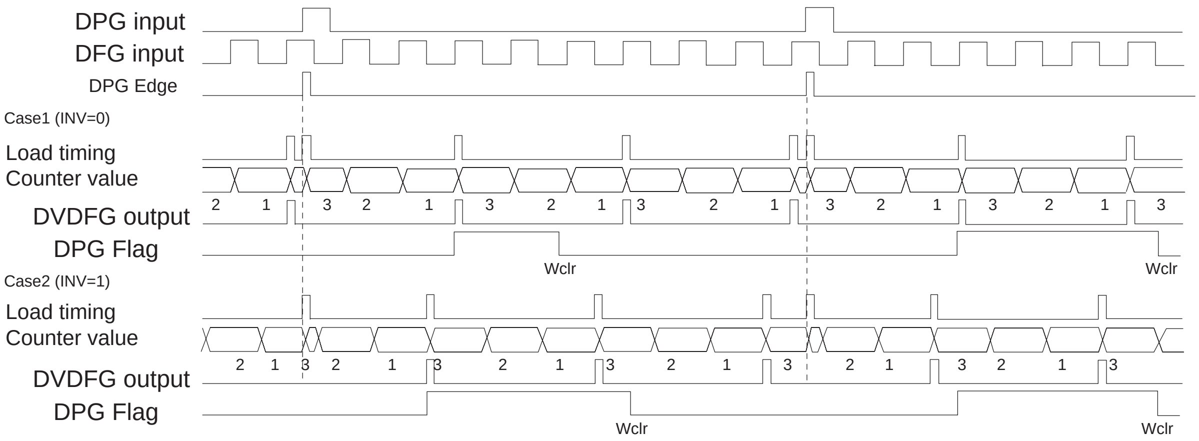
Rewriting the DINV bit affects the programmable divider operation. Prior to perform rewriting, some kind of action should be performed, such as masking the capture input.

● Initialization of programmable divider by DPG input

The programmable divider is initialized by the rising edge of the DPG input if DPG input is enabled.

Figure 6.3-6 shows the initialization operation by DPG input and the set timing of DPG detection flag.

Figure 6.3-6 Timing diagram of initialization operation by DPG input

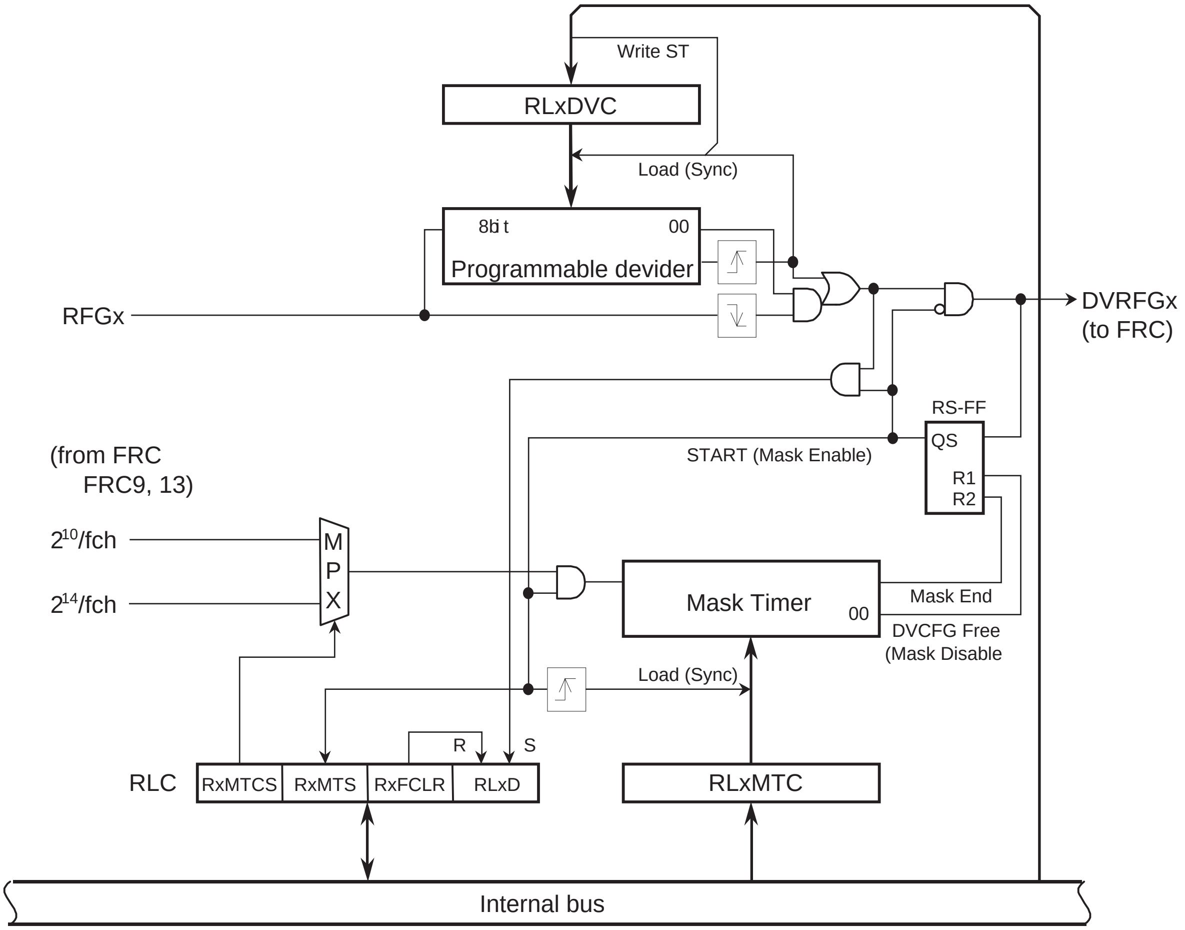


6.4 Reel Input

The reel input section comprises of the 8-bit programmable divider and mask timer. This section explains the register which controls the operation of each section.

■ Block Diagram of Reel Input

Figure 6.4-1 Block Diagram of Reel Input



■ Register List of Reel Input

Figure 6.4-2 Register list of Reel Input

bit	7 ←→ 0	
Address: 000057H	RLC	Reel control register
000058H	RL0DVC	Reel 0 input control register
000059H	RL0MTC	Reel 0 mask timer control register
00005AH	RL1DVC	Reel 1 input control register
00005BH	RL1MTC	Reel 1 mask timer control register

■ Reel Control Register (RLC)

Figure 6.4-3 Reel Control Register (RLC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000057H	RL1D	R1FCLR	R1MTS	R1MTCS	RL0D	R0FCLR	R0MTS	R0MTCS	X1X0 X1X0B
Access →	R	W	R	R/W	R	W	R	R/W	

[bit7]:RL1D

This is the division FG edge detection flag during reel 1 masking.

0	Without edge detection
1	With edge detection

[bit6]:R1FCLR

This is the edge detection flag clear bit during the reel 1 masking.

0	Clear RL1D flag.
1	None

The read value of this bit is always "1".

[bit5]:R1MTS

This is the reel 1 mask timer status flag.

0	Mask released
1	Masking

[bit4]:R1MTCS

This is clock source selection bit of reel 1 mask timer.

	Selection Clock	in fch:@20MHz
0	2^{10}/fch (FRC9)	51.2 µs
1	2^{14}/fch (FRC13)	819.2 µs

[bit3]:RL0D

This is the division FG edge detection flag during the reel 0 masking.

0	Without edge detection
1	With edge detection

[bit2]:R0FCLR

This is the edge detection flag clear bit during the reel 0 masking.

0	Clear RL0D flag
1	None

The read value of this bit is always "1".

[bit1]:R0MTS

This is the reel 0 mask timer status flag.

0	Mask released
1	Masking

[bit0]:R0MTCS

This is clock source selection bit of reel 0 mask timer.

	Selection Clock	in fch:@20MHz
0	2^{10}/fch (FRC9)	51.2 µs
1	2^{14}/fch (FRC13)	819.2 µs

■ Reel Input Control Register (RLxDVC)

Figure 6.4-4 Reel input control register (RLxDVC)

bit	7	6	5	4	3	2	1	0	Initial value
	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0	XXXX XXXX$_B$
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:DIV7 to 0

Division control of the drum input and edge detection control are performed by the set value.

Set value	Division control	Edge detection
00_H	None	Both edge detection
01_H	1-frequency division	Rising edge detection
02_H	2-frequency division	
03_H	3-frequency division	
to	to	
FD_H	253-frequency division	
FE_H	254-frequency division	
FF_H	255-frequency division	

■ Reel Mask Timer Control Register (RLxMTC)

Figure 6.4-5 Reel mask timer control register (RLxMTC)

bit	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	XXXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:D7 to 0

The masking period control of the reel input is performed by the set value. When ΦMT is specified as the clock cycle time selected by the mask timer clock select (CS) bit of the reel control register, and N is specified as the set value, the drum input masking period TM is as follow.

TM = ΦMT × N ± ΦMT/2. However, N is set to 0, the mask processing does not perform.

■ Operation of Reel Input

Refer to "6.2 Capstan Input" for the 8-bit programmable divider and mask timer operations.

CHAPTER 7
FRC Capture

This chapter describes an outline of the FRC capture section, the register configuration/functions, and each input section operation.

7.1 Overview of FRC Capture

The FRC capture section has built-in 24-bit free-run counter and uses FIFO format.

■ Feature of FRC Capture

- Built-in 24-bit free-run counter (Minimum resolution 50 ns:@20 MHz)
- Built-in FIFO (Data 21-bit x 8, factor 8-bit x 8)

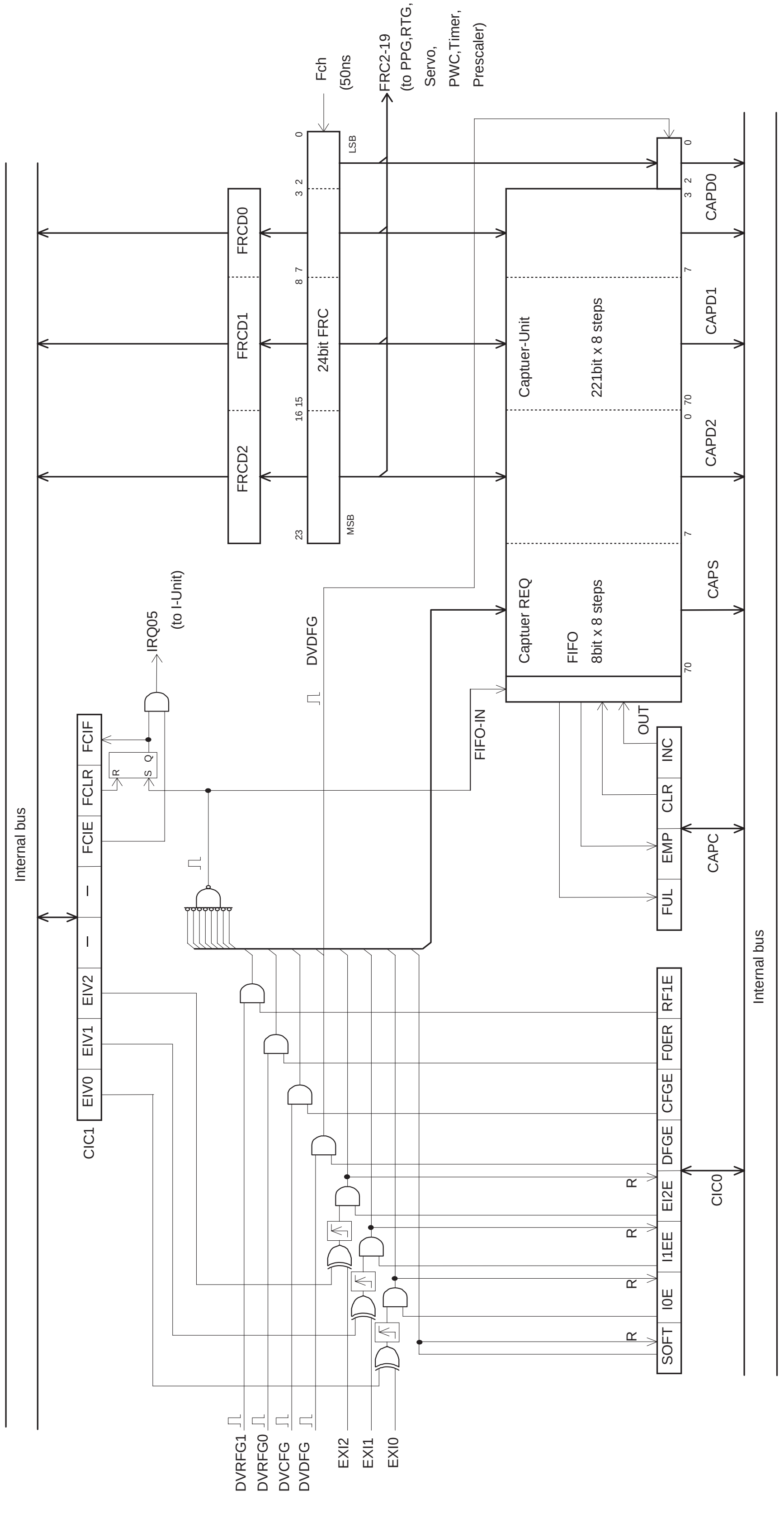
■ Register List of FRC Capture

Figure 7.1-1 Register list of FRC capture

Address:	bit 7 ↔ 0	
000060H	――――	
000061H	FRCD2	FRC count data register
000062H	FRCD1	
000063H	FRCD0	
000064H	CIC1	Capture input control register
000065H	CIC0	
000066H	――――	
000067H	CAPC	Capture control register
000068H	CAPS	Capture source register
000069H	CAPD2	Capture data register
00006AH	CAPD1	
00006BH	CAPD0	

■ Block Diagram of FRC Capture

Figure 7.1-2 Block diagram of FRC capture



7.2 Register of FRC Capture

This section shows the register configuration/functions of the FRC capture.

■ Capture Input Control Register (CIC1, CIC0)

- Capture input control register (CIC1)

Figure 7.2-1 Capture input control register (CIC1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000064H	FCIF	FCLR	FCIE	−	−	ICR2	ICR1	ICR0	X10- -000B
Access →	R	W	R/W	−	−	R/W	R/W	R/W	

[bit7]:FCIF

It is capture request detection and interrupt request flag.

0	No capture request
1	Detect capture request

[bit6]:FCLR

It is capture request detection flag clear bit.

0	Clear CIF flag.
1	None

The read value of this bit is always "1".

[bit5]:FCIE

It is interrupt request enable bit.

0	Interdiction
1	Permission

[bit4, 3]:

It is an unused bit.

[bit2]:EIV2

It is EXI2 input edge detection polarity selection bit.

0	Falling edge detection
1	Rising edge detection

[bit1]:EIV1

It is EXI1 input edge detection polarity selection bit.

0	Falling edge detection
1	Rising edge detection

[bit0]:EIV0

It is EXI0 input edge detection polarity selection bit.

0	Falling edge detection
1	Rising edge detection

- Capture input control register (CIC0)

Figure 7.2-2 Capture input control register (CIC0)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000065H	RF1E	RF0E	CFGE	DFGE	EI2E	EI1E	EI0E	S0FT	0000 0000B
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	

Note:

Do not use the read modify write command on this register.

[bit7]:RF1E

It is DVRFG1 input control bit.

0	Input interdiction
1	Input permission

[bit6]:RF0E

It is DVRFG0 input control bit.

0	Input interdiction
1	Input permission

[bit5]:CFGE

It is DVCFG input control bit.

0	Input interdiction
1	Input permission

[bit4]:DFGE

It is DVDFG input control bit.

0	Input interdiction
1	Input permission

[bit3]:EI2E

It is EXI2 input control bit.

0	Input interdiction
1	Input permission. It is automatically to "0" by acceptance of capture request.

[bit2]:EI1E

It is EXI1 input control bit.

0	Input interdiction
1	Input permission. It is automatically to "0" by acceptance of capture request.

[bit1]:EI0E

It is EXI0 input control bit.

0	Input interdiction
1	Input permission. It is automatically to "0" by acceptance of capture request.

[bit0]:SOFT

It is software trigger generation bit.

0	None
1	Generate capture request.

The read value of this bit is always "0".

■ Capture Control Register (CAPC)

Figure 7.2-3 Capture control register (CAPC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000067H	–	–	–	–	FUL	EMP	CLR	INC	---- 0100B
Access →	–	–	–	–	R	R	W	W	

[bit7, 6, 5, 4]:

It is an unused bit.

[bit3]:FUL

It is FIFO full flag.

0	Status which affords to input the data to FIFO
1	Status of FIFO full

[bit2]:EMP

It is FIFO empty flag.

0	Status which data is remained in FIFO
1	Status of FIFO empty

[bit1]:CLR

It is FIFO clear control bit.

0	None
1	Clear FIFO.

The read value of this bit is always "0".

[bit0]:INC

It is FIFO output control bit.

0	None
1	Output next data on FIFO.

The read value of this bit is always "0".

■ Capture Source Register (CAPS)

Figure 7.2-4 Capture source register (CAPS)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000068H	RFG1	RFG0	CFG	DFG	EXI2	EXI1	EXI0	S0FT	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

These bits indicate the capture factors of the captured data.

When each bit is "1", it has a capture request.

When it is "0", it does not have a capture request.

■ Capture Data Registers (CAPD2 to 0)

Figure 7.2-5 Capture data register (CAPD2)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000069H	CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

Figure 7.2-6 Capture data register (CAPD1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00006AH	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

Figure 7.2-7 Capture data register (CAPD0)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00006BH	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT0	CNT0	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

These indicate the time that the capture request is generated.

■ FRC Count Data Register (FRCD2 to 0)

Figure 7.2-8 FRC count data register (FRCD2)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000061H	CNT23	CNT22	CNT21	CNT20	CNT19	CNT18	CNT17	CNT16	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

Figure 7.2-9 FRC count data register (FRCD1)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000062H	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	XXXX XXXX
Access →	R	R	R	R	R	R	R	R	

Figure 7.2-10 FRC count data register (FRCD0)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000063H	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT0	CNT0	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

Reading this register enables the free run counter value (time) to be known.

7.3 Operation of FRC Capture

Up to eight capture data can be fetched in FIFO. If a new capture request is generated while data is full, the former data will be updated with the new data.
If a capture request is generated while the FIFO storage is empty, overhead for a maximum of "fch x 18 cycles" (10 cycles from the CIF set) will be generated until data setup (readable status).

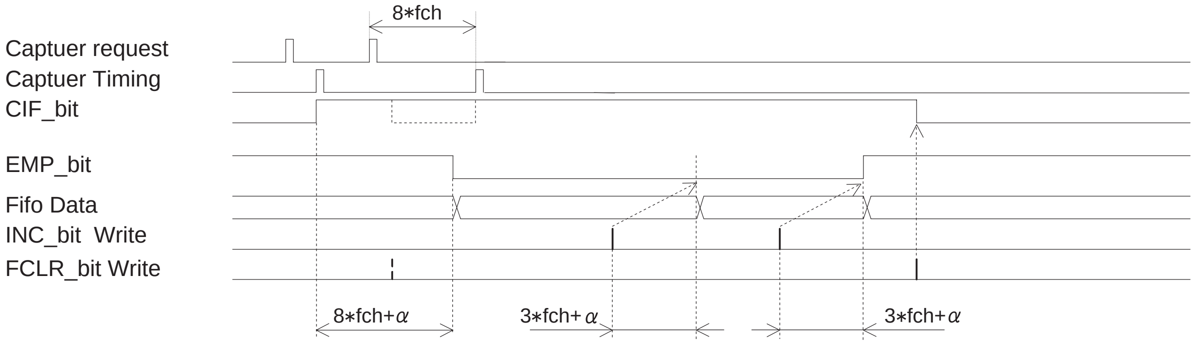
■ Controlling Method of FIFO

Latching the data in FIFO is performed the following procedure.

1. When EMP bit is "0", read the data.
2. Writes the INC bit to "1". Therefore, next data in FIFO is ready. (Maximum of fch x 5 cycles)
3. Checks the EMP bit and EMP=0: Returns to 2) to extract the following data. EMP=1: Completes all data reading.

To initialize FIFO and dispose of data within FIFO storage, writing "1" to the FIFO clear bit initializes the FIFO.

Figure 7.3-1 Operation timing diagram of FIFO



■ Capture Data

The lowest 3 bits of the capture data will be valid only when the DFG bit of the capture source register is "1". When the DFG bit is "0", masks the lowest 3 bits after reading the capture data.

■ Reading of FRC Count Data

Use the word access command to read the counter data. When the half-word access command is read, after reading the counter data subordinate, read the superior data.

Correct values cannot be read using byte access commands. The top byte read by the word access command is invalid data, so execute mask processing.

CHAPTER 8

Programmable Pulse Generator (PPG0, 1)

This chapter describes an outline of the programmable pulse generator (PPG0, 1), the register configuration/ functions, and their operation.

8.1 Overview of Programmable Pulse Generator (PPG0, 1)

The programmable pulse generators (PPG0, 1) have built-in buffer RAM (PPG0: 256 bytes, PPG1: 64 bytes) and A/DC hard start function.

■ Feature of Programmable Pulse Generator (PPG0, 1)

- Built-in buffer RAM (PPG0: 256 bytes, PPG1: 64 bytes)
- Output timing precision 800 ns(@20 MHz)
- Built-in A/D converter hard start function

■ Block Diagram of Programmable Pulse Generator (PPG0, 1)

Figure 8.1-1 Block diagram of programmable pulse generator (PPG0, 1)

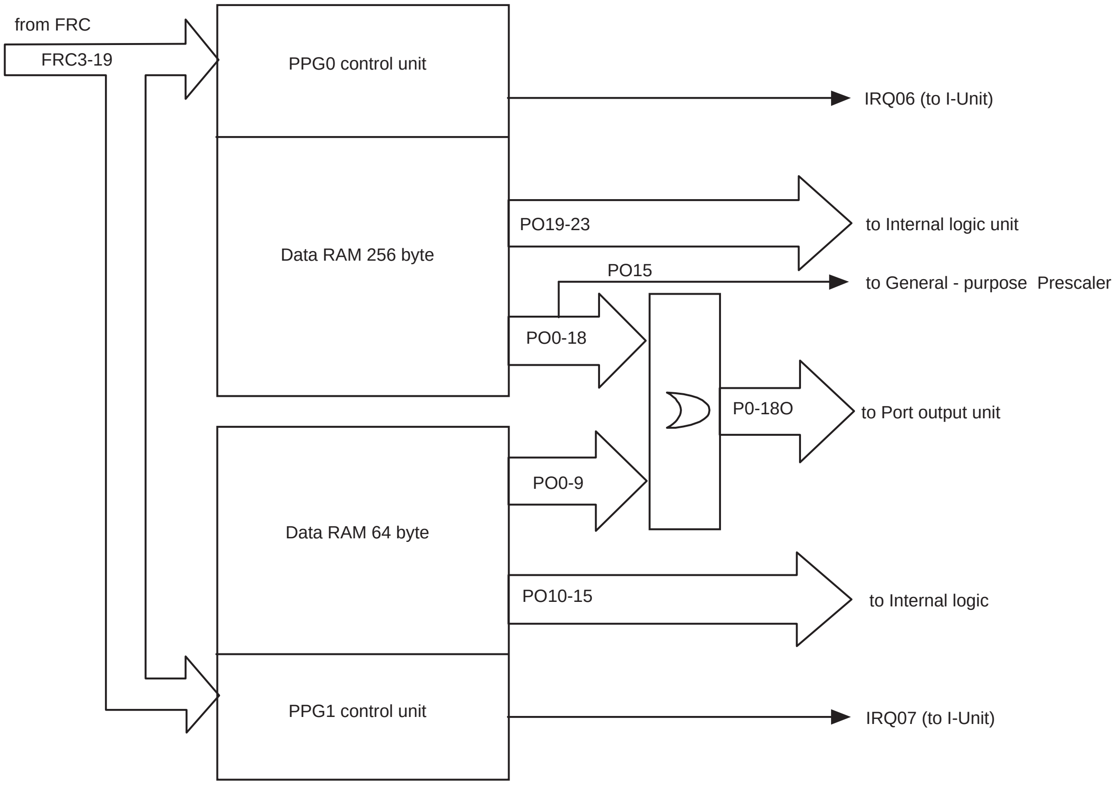
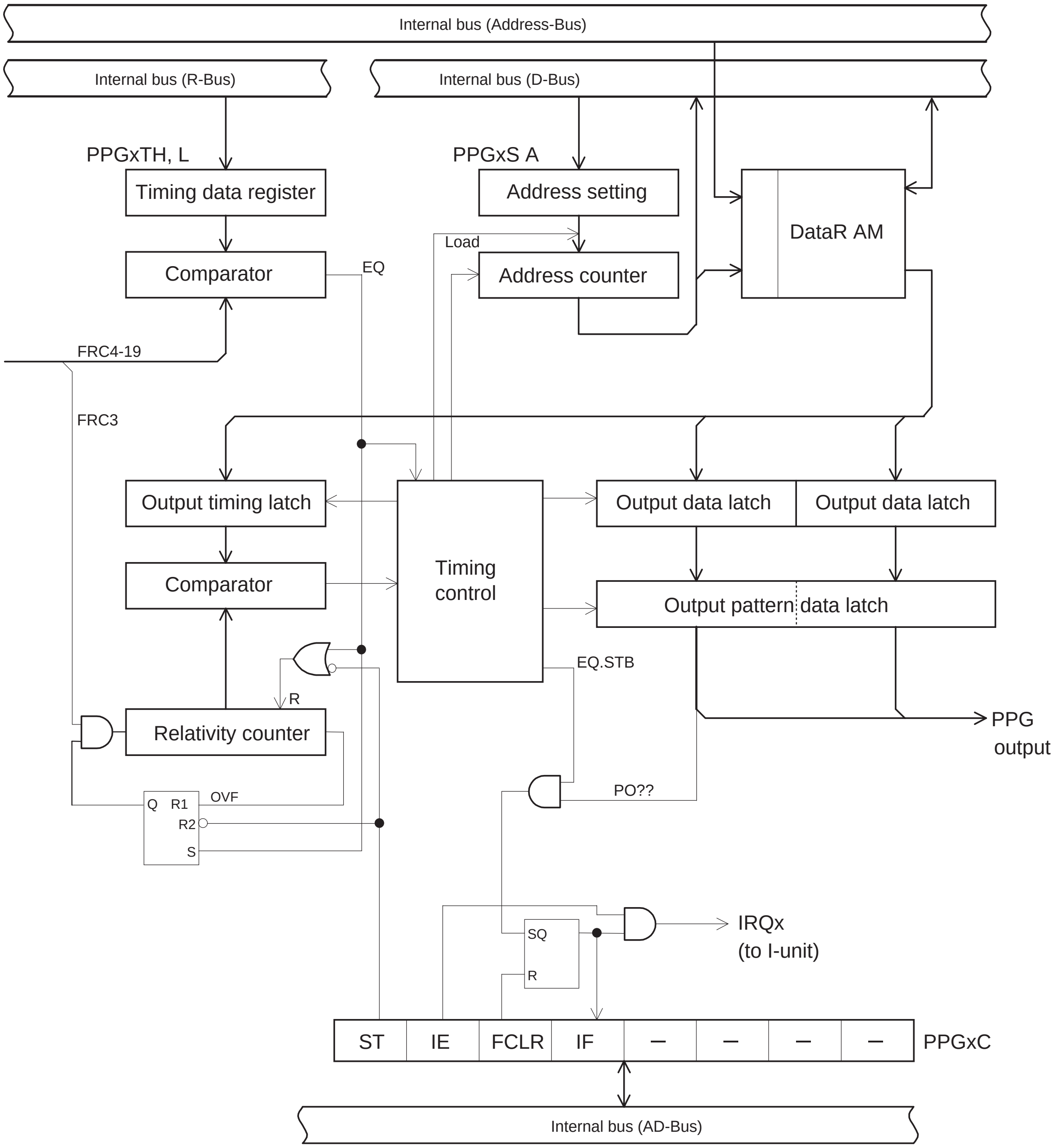
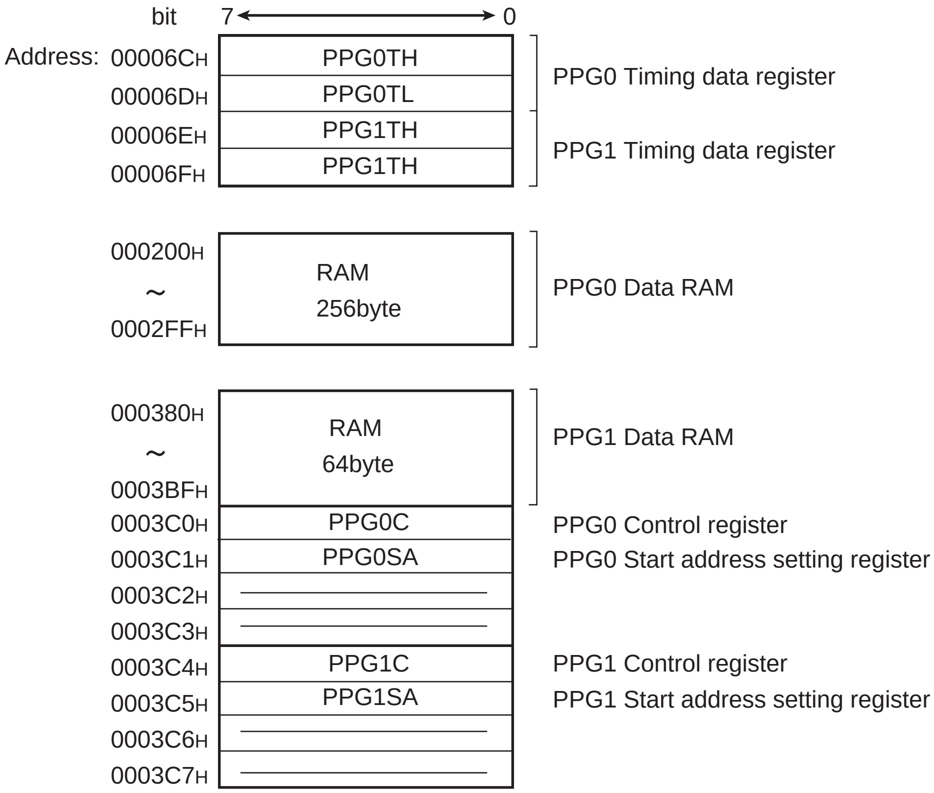


Figure 8.1-2 Block diagram of programmable pulse generator (PPG0, 1)



■ Register List of Programmable Pulse Generator (PPG0, 1)

Figure 8.1-3 Register list of programmable pulse generator (PPG0, 1)



8.2 Register of Programmable Pulse Generator (PPG0, 1)

The register configuration/functions of the programmable pulse generator (PPG0, 1) are described.

■ PPGx Control Register (PPGxC)

Figure 8.2-1 PPGx control register (PPGxC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0003C0$_H$ (PPG0C) 0003C4$_H$ (PPG1C)	Test	Test	Test	Test	IF	FCLR	IE	ST	0000 X100$_B$
Access →	R/W	R/W	R/W	R/W	R	W	R/W	R/W	

[bit7 to 4]:Test

Please set "0".

[bit3]:IF

It is output timing data match interrupt request flag.

0	Not match
1	Match

[bit2]:FCLR

It is interrupt request flag clear bit.

0	Clear IF.
1	None

The read value of this bit is always "1".

[bit1]:IE

It is interrupt enable bit.

0	Interrupt interdiction
1	Interrupt permission

[bit0]:ST

It is PPG start bit.

0	PPG stop
1	PPG start

■ Timing Data Register (PPGxT)

Figure 8.2-2 Timing data register (PPGxTH)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00006C$_H$ (PPG0TH)~ 00006E$_H$ (PPG1TH)	D15	D14	D13	D12	D11	D10	D09	D08	XXXX XXXX$_B$
Access →	W	W	W	W	W	W	W	W	

Figure 8.2-3 Timing data register (PPGxTL)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00006D$_H$ (PPG0TL)~ 00006F$_H$ (PPG1TL)	D07	D06	D05	D04	D03	D02	D01	D00	XXXX XXXX$_B$
Access →	W	W	W	W	W	W	W	W	

When the timing data register is updated continuously, write new data after 18/fch or more has passed after writing.

Do not use the byte access command to access the register.

■ Start Address Setting Register (PPGxSA)

Figure 8.2-4 Start address setting register (PPGxSA)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0003C1$_H$ (PPG0SA) 0003C5$_H$ (PPG1SA)	A7/SA7	A6/SA6	A5/SA5	A4/SA4	A3/SA3	A2/SA2	A1/SA1	A0/SA0	XXXX XXXX$_B$
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:A7 to 0

The set address will be the PPG output starting frame data address, and the read value will be the frame data address to be output next.

Table 8.2-1 Relationship between start address and data RAM

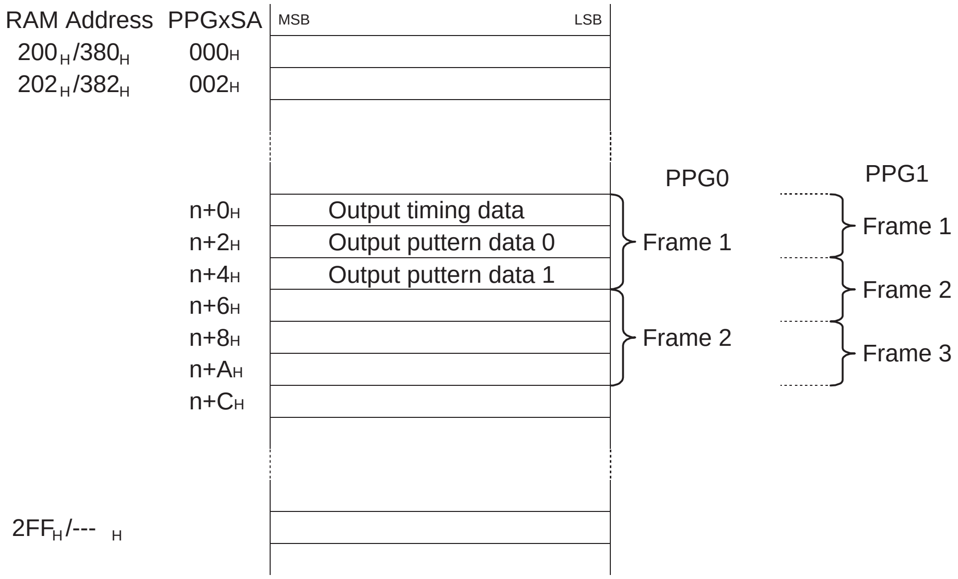
Set value	Data RAM Address	
	PPG0	PPG1
00$_H$	000200$_H$	000380$_H$
02$_H$	000202$_H$	000382$_H$
04$_H$ to 3C$_H$	000204$_H$ to 00023C$_H$	000384$_H$ 0003BC$_H$
3E$_H$	00023E$_H$	0003BE$_H$
40$_H$	000240$_H$	-
42$_H$	000242$_H$	-
44$_H$ to FC$_H$	000244$_H$ to 0002FC$_H$	-
FE$_H$	0002FE$_H$	-

8.3 PPG Data RAM

This section shows the relationship between the PPG data RAM and frame.

■ Relationship between PPG Data RAM and Frame

Figure 8.3-1 Relationship between PPG data RAM and frame



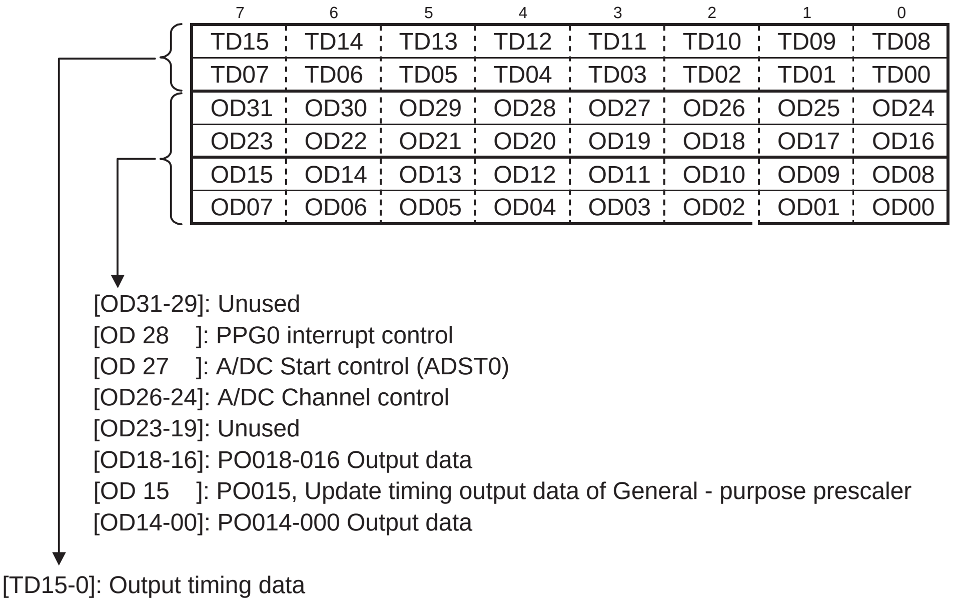
8.4 Configuration of Frame Data

Each frame is made up of 2-byte length output timing data (OTD) that specifies the output time and n x 2-byte length output pattern data (OPDx) that specifies the output value.

In terms of the PPG0 frame configuration, 4 bytes (32 bits) of output pattern data exist, which is a 6 bytes per frame configuration; and for PPG1, 2 bytes (16 bits) of output pattern data exist, which is a 4 bytes per frame configuration.

■ PPG0 Frame Data

Figure 8.4-1 PPG0 Frame data



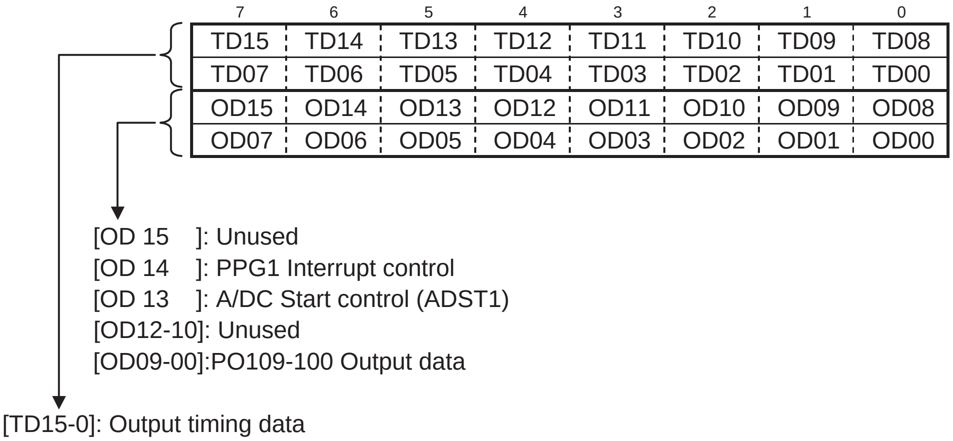
7	6	5	4	3	2	1	0
TD15	TD14	TD13	TD12	TD11	TD10	TD09	TD08
TD07	TD06	TD05	TD04	TD03	TD02	TD01	TD00
OD31	OD30	OD29	OD28	OD27	OD26	OD25	OD24
OD23	OD22	OD21	OD20	OD19	OD18	OD17	OD16
OD15	OD14	OD13	OD12	OD11	OD10	OD09	OD08
OD07	OD06	OD05	OD04	OD03	OD02	OD01	OD00

[OD31-29]: Unused
[OD 28]: PPG0 interrupt control
[OD 27]: A/DC Start control (ADST0)
[OD26-24]: A/DC Channel control
[OD23-19]: Unused
[OD18-16]: PO018-016 Output data
[OD 15]: PO015, Update timing output data of General - purpose prescaler
[OD14-00]: PO014-000 Output data

[TD15-0]: Output timing data

■ PPG1 Frame Data

Figure 8.4-2 PPG1 Frame data



7	6	5	4	3	2	1	0
TD15	TD14	TD13	TD12	TD11	TD10	TD09	TD08
TD07	TD06	TD05	TD04	TD03	TD02	TD01	TD00
OD15	OD14	OD13	OD12	OD11	OD10	OD09	OD08
OD07	OD06	OD05	OD04	OD03	OD02	OD01	OD00

[OD 15]: Unused
[OD 14]: PPG1 Interrupt control
[OD 13]: A/DC Start control (ADST1)
[OD12-10]: Unused
[OD09-00]:PO109-100 Output data

[TD15-0]: Output timing data

8.5 Operation of PPG

The operation of PPG has output and start operation.

■ Output Operation of PPG

Clears the relative counter when the values set to the timing data register (PPGxTH, PPGxTL) and the FRC value match, and after loading the value of the address set register to the memory address counter, retains the 1st field data of data RAM (output timing data, output data) in the output timing data latch and output data latch.

The retained output data is transferred to the output latch by the rising edge of the relative counter clock under the cycle whose relative counter and output timing data match, and the 2nd field data is newly retained in each latch.

Figure 8.5-1 Output operation of PPG

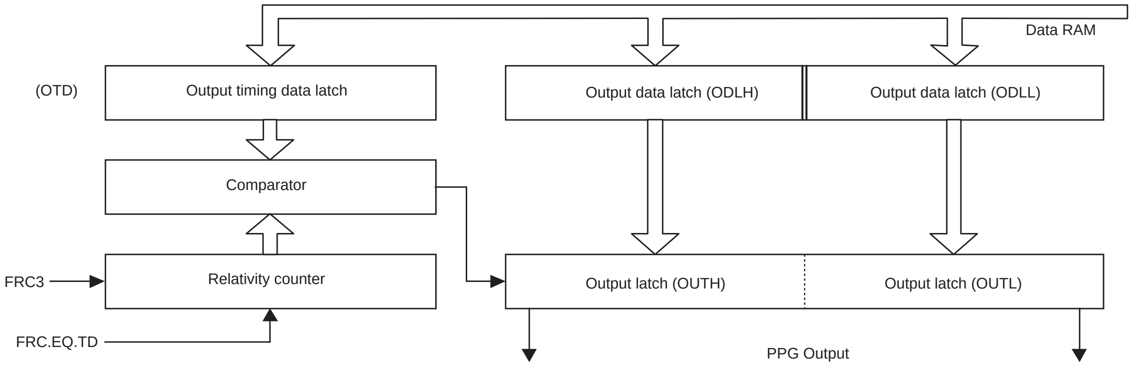


Figure 8.5-2 PPG output detail timing

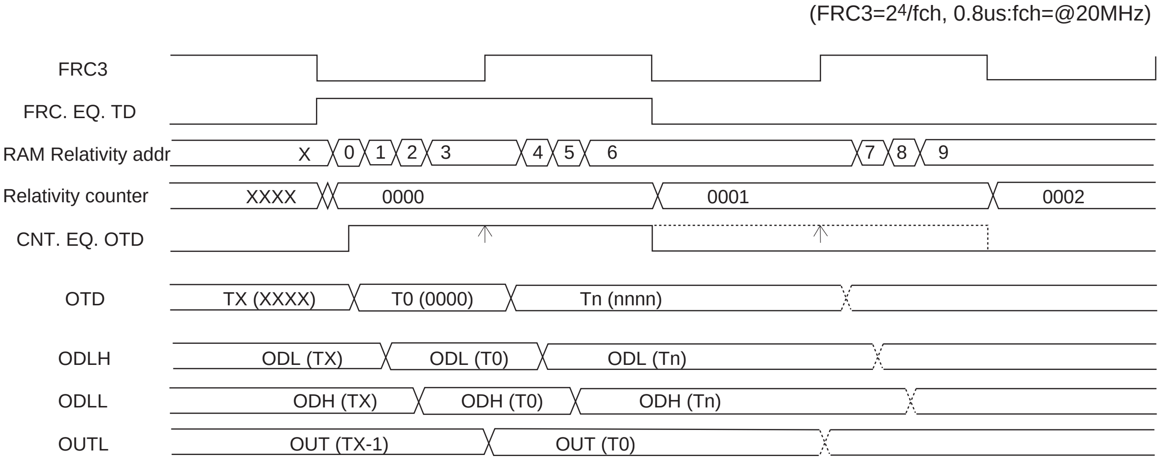
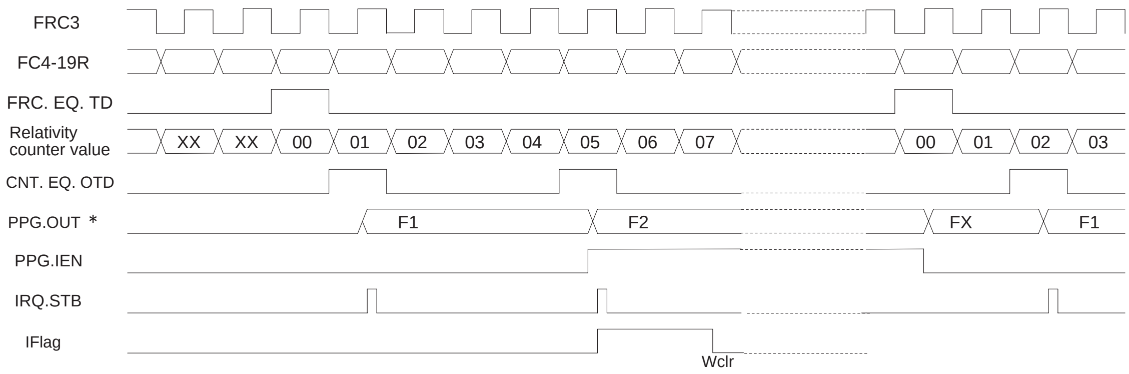


Figure 8.5-3 Operation timing of PPG



■ Start Operation of PPG

The PPG starts operation by setting "1" to the start bit. The initial operation is performed as follow.

- Clears the relative counter and loads the initial address in synchronization with the falling edge of the FRC clock after starting.
- Reads the frame data indicated by the initial address, sets data to each latch of OTD and ODL, and the status will be compare match wait for the OTD value and relative counter value.
- When compare match is detected, transfers the ODL value to the output latch (OUTL) in synchronization with rising edge of the FRC clock, reads the next frame data, then updates each latch value of OTD and ODL, while the status is compare match wait. (This operation is performed repeatedly.)

■ Update Timing Data Register

When timing data is updated during PPG operation, the next data should be written after FRC3 x 2 or longer has passed after writing the register.

■ Precaution when Clear IF Flag

Do not perform clearing of IF flag and interrupt enable (IE=1) concurrently. Enables interrupts after clearing the IF flag.

CHAPTER 9

Real Timing Generator (RTG)

This chapter describes an outline of the real timing generator (RTG), the register configuration/functions, and their operation.

9.1 Overview of Real Timing Generator (RTG)

The real timing generator (RTG) has 3 built-in circuits, namely real timing generators 0 to 2 (RTG0 to 2).

■ Feature of Real Timing Generator (RTG)

- Contain 3 RTG circuit
- Output timing accuracy selectable 400 ns/800 ns
- 5 timing outputs

■ Block Diagram of Real Timing Generator (RTG)

Figure 9.1-1 Block diagram of real timing generator (RTG)

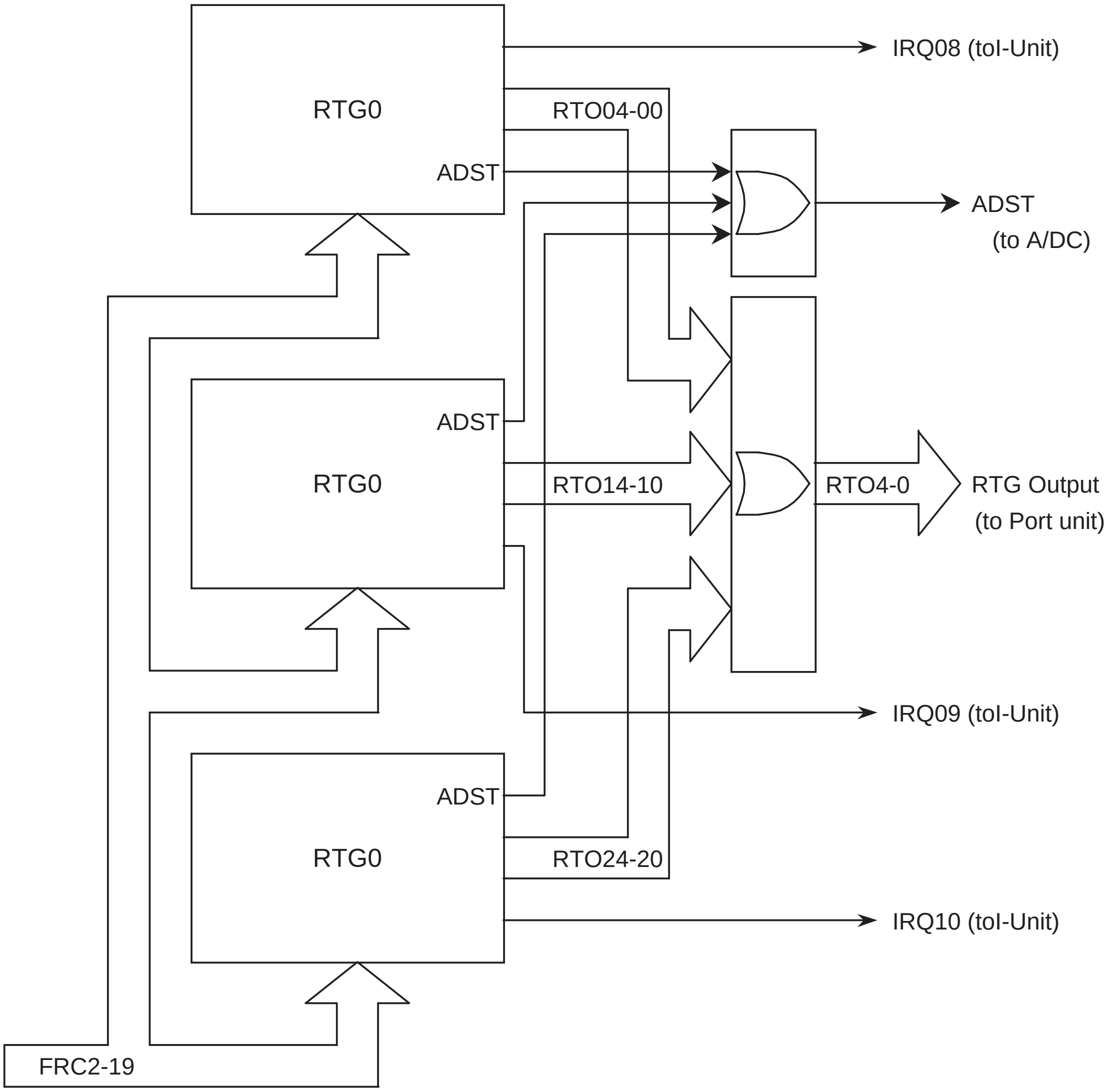
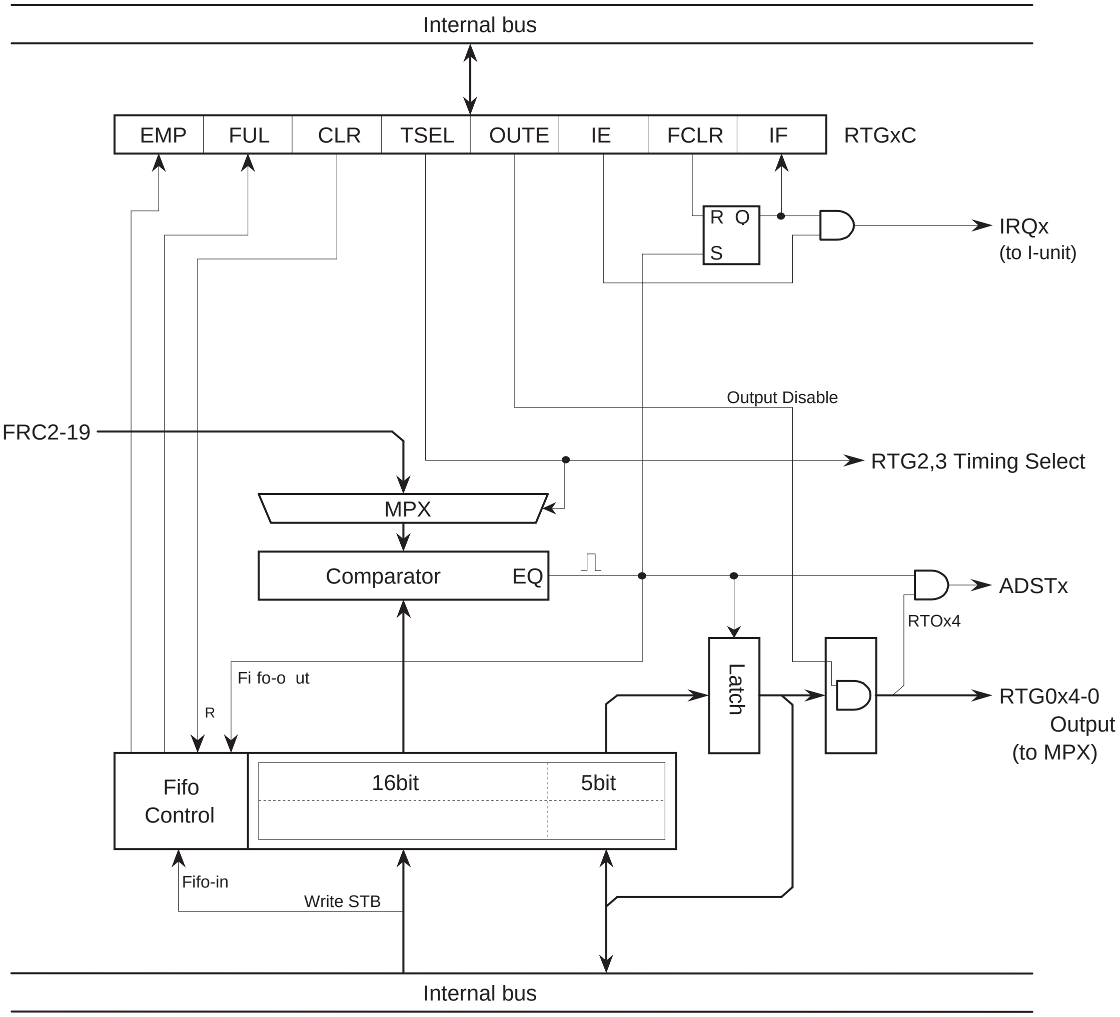


Figure 9.1-2 Block diagram of real timing generator (RTG)



■ Register List of Real Timing Generator (RTG)

Figure 9.1-3 Register list of Real Timing Generator (RTG)

Address:	bit 7 ←→ 0	
000034H	RTG0C	RTG0 Control register
000035H	RTG0D	RTG0 Output data register
000036H	RTG0TH	RTG0 Timing data register
000037H	RTG0TL	
000038H	RTG1C	RTG1 Control registe
000039H	RTG1D	RTG1 Output data register
00003AH	RTG1TH	RTG1 Timing data register
00003BH	RTG1TL	
00003CH	RTG2C	RTG2 Control registe
00003DH	RTG2D	RTG2 Output data register
00003EH	RTG2TH	RTG2 Timing data register
00003FH	RTG2TL	

9.2 Register of Real Timing Generator (RTG)

The register configuration/functions of the real timing generator (RTG) are described.

■ RTGx Control Register (RTGxC)

Figure 9.2-1 RTGx Control Register (RTGxC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000034H (RTG0C) 000038H (RTG1C) 00003CH (RTG2C)	IF	FCLR	IE	OUTE	TSEL*	CLR	FUL	EMP	X100 0001B
Access →	R	W	R/W	R/W	R/W	W	R	R	

[bit7]:IF

It is output timing data match interrupt request flag.

0	Not match
1	Match

[bit6]:FCLR

It is interrupt request flag clear bit.

0	Clear IF.
1	None

The read value of this bit is always "1".

[bit5]:IE

It is interrupt enable bit.

0	Interrupt interdiction
1	Interrupt permission

[bit4]:OUTE

It is output enable bit.

0	RTG output interdiction (Low output fixed)
1	RTG output enable

[bit3]:TSEL

It is timing accuracy selection bit of RTG section (RTG0 to 2).

0	400 ns accuracy (in fch=@20 MHz)
1	800 ns accuracy (in fch=@20 MHz)

This bit exists with the RTG0 only, and it affects the timing accuracy of all RTGs.

[bit2]:CLR

It is FIFO initialization bit.

0	None
1	Clear FIFO.

The read value of this bit is always "0".

[bit1]:FUL

It is FIFO full flag.

0	FIFO empty status
1	FIFO full status

[bit0]:EMP

It is FIFO empty flag.

0	Data is remained in FIFO
1	FIFO empty status

■ Output Data Register (RTGxD)

Figure 9.2-2 Output data register (RTGxD)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000035H (RTG0D) 000039H (RTG1D) 00003DH (RTG2D)	–	–	–	OD4	OD3	OD2	OD1	OD0	---X XXXXB
Access →	–	–	–	R/W	R/W	R/W	R/W	R/W	

Data written to this register will be the RTG output data. Reading value is read that the RTG is current output data. In terms of the RTG output, the value logically added (OR) per bit of each channel is output.

■ Timing Data Register (RTGxT)

Figure 9.2-3 Timing data register (RTGxT)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000036H (RTG0TH) 00003AH (RTG1TH) 00003EH (RTG2TH)	D15	D14	D13	D12	D11	D10	D09	D08	XXXX XXXXB
Access →	W	W	W	W	W	W	W	W	

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000037H (RTG0TL) 00003BH (RTG1TL) 00003FH (RTG2TL)	D07	D06	D05	D04	D03	D02	D01	D00	XXXX XXXXB
Access →	W	W	W	W	W	W	W	W	

It is a register to set the output time of the RTG output. Writing this register sets the value set by the output data register in FIFO format.

9.3 Operation of Real Timing Generator (RTG)

Initiation procedure of the real timing generator (RTG) and RTG output timing are described.

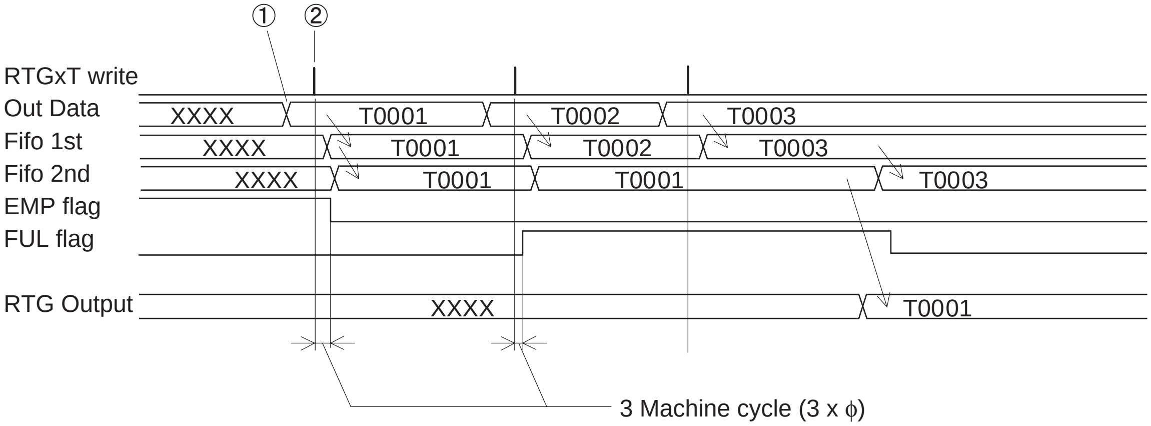
■ Initiation Procedure of Real Timing Generator (RTG)

1. Performs the initialization of control register.
2. Set the output data to RTGxD. In this case, do not use the command with RMW.
3. Set the output time to RTGxTH, L. In this case, the output data set by 2) is transferred to FIFO and EMP flag becomes 0. 3 machine cycles are required for the EMP flag to be determined after setting the output time. So if the flag is to be checked immediately after setting the output time, test should be carried out after executing dummy read, etc.

Notes:

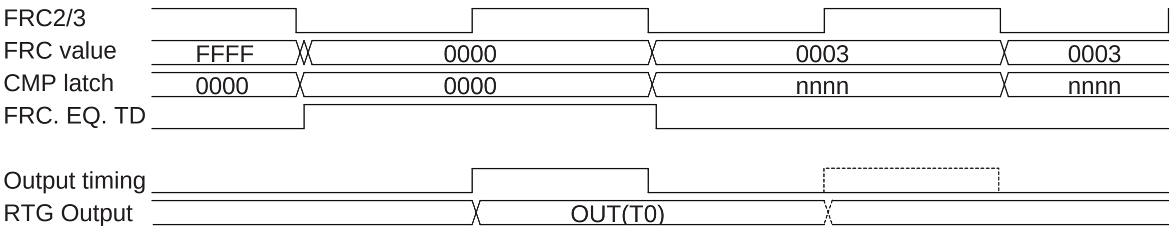
- Do not use word access commands to set the output data and output time.
- Use half-word access commands to set the output time.
- Prior to setting up the output time, set up the output data.

Figure 9.3-1 Operation timing of FIFO



■ RTG Output Timing

Figure 9.3-2 Output timing of RTG



CHAPTER 10
Timer

This chapter describes an outline of the timer section, the register configuration/functions, and timer section operation.

10.1 Overview of Timer

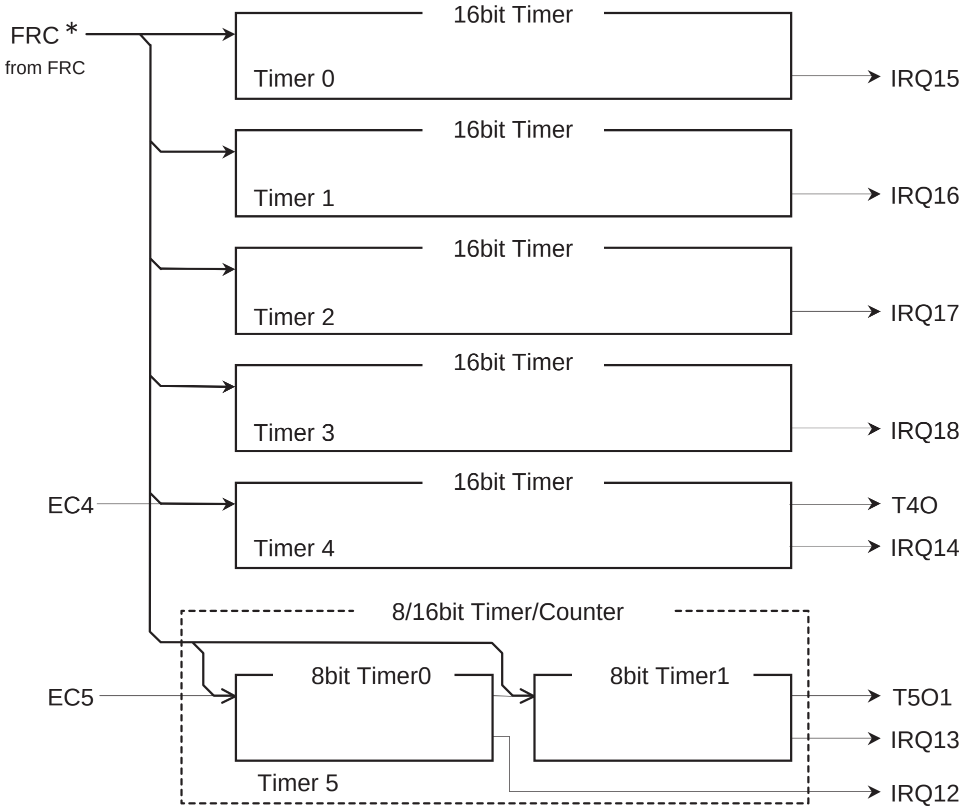
The timer section comprises of a 16-bit timer and 8-/16-bit timer/counter.

■ Feature of Timer

- 16-bit x 4ch
- 16-bit timer/counter x 1ch (with square wave output)
- 8-/16-bit timer/counter x 1ch (with square wave output)

■ Configuration of Timer

Figure 10.1-1 Block diagram of Timer



■ Register List of Timer

Figure 10.1-2 Register list of Timer

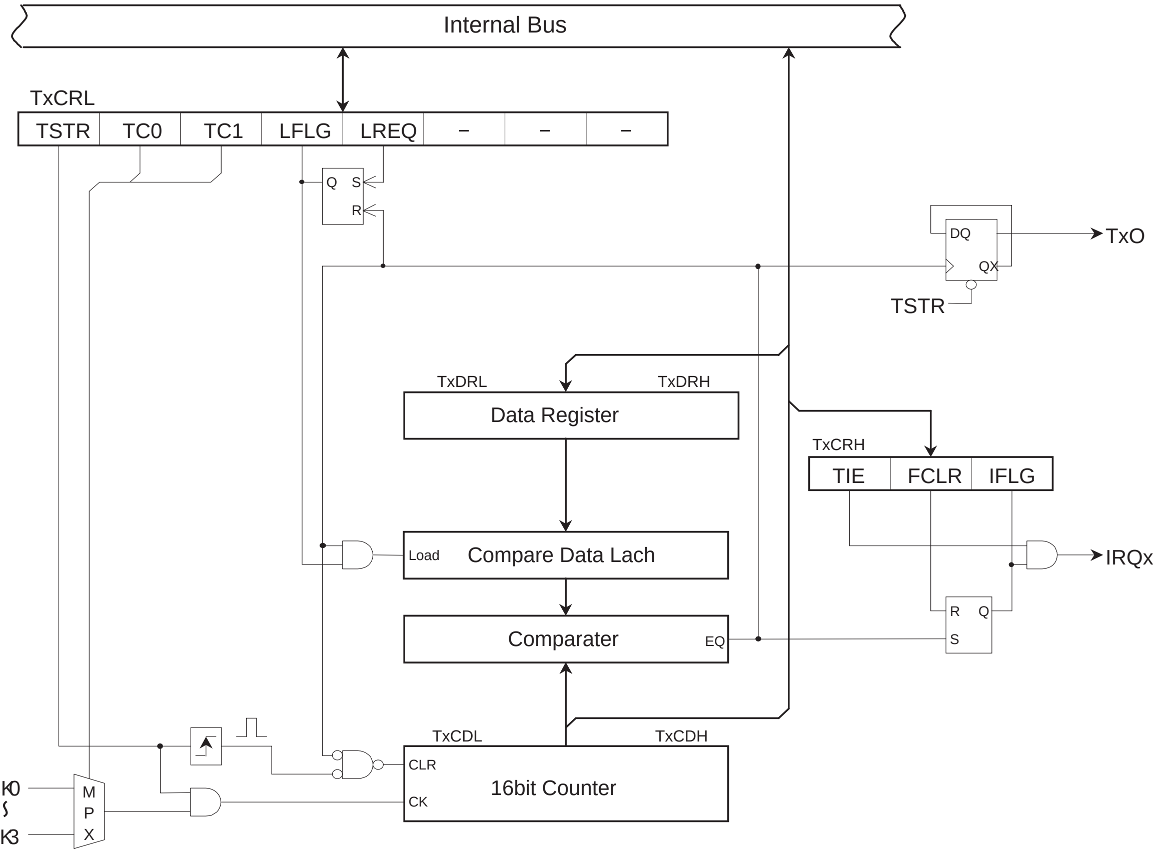
Address:	bit 15 ↔ 8	7 ↔ 0	
000070H	T0CD		Timer 0 (16 bit Timer)
000072H	T0DR		
000074H	T0CR		
000076H	T1CR		Timer 1 (16 bit Timer)
000078H	T1CD		
00007AH	T1DR		
00007CH	T2CD		Timer 2 (16 bit Timer)
00007EH	T2DR		
000080H	T2CR		
000082H	T3CR		Timer 3 (16 bit Timer)
000084H	T3CD		
000086H	T3DR		
000088H	T4CD		Timer 4 (16 bit Timer)
00008AH	T4DR		
00008CH	T4CR		
00008EH	T5CR1	T5CR0	Timer 5 (16 bit Timer)
000090H	T5DR1	T5DR0	
000092H	T5CD1	T5CD0	

10.2 Overview of 16-bit Timer (Timer 0 to 4)

16-bit timer (timer 0 to 4) can be select from 4 clocks.

■ Block Diagram of 16-bit Timer (Timer 0 to 4)

Figure 10.2-1 Block diagram of 16-bit timer (Timer 0 to 4)



■ Register List of 16-bit Timer (Timer 0 to 4)

Figure 10.2-2 Register list of 16-bit timer (timer 0 to 4)

bit	15 ←→ 8	7 ←→ 0	
Address 000070_H	T0CDH	T0CDL	Timer 0 Count Data Register
000072_H	T0DRH	T0DRL	Timer 0 Data Register
000074_H	T0CRH	T0CRL	Timer 0 Control Register
000076_H	T1CRH	T1CRL	Timer 1 Control Register
000078_H	T1CDH	T1CDL	Timer 1 Count Data Register
$00007A_H$	T1DRH	T1DRL	Timer 1 Data Register
$00007C_H$	T2CDH	T2CDL	Timer 2 Count Data Register
$00007E_H$	T2DRH	T2DRL	Timer 2 Data Register
000080_H	T2CRH	T2CRL	Timer 2 Control Register
000082_H	T3CRH	T3CRL	Timer 3 Control Register
000084_H	T3CDH	T3CDL	Timer 3 Count Data Register
000086_H	T3DRH	T3DRL	Timer 3 Data Register
000088_H	T4CDH	T4CDL	Timer 4 Count Data Register
$00008A_H$	T4DRH	T4DRL	Timer 4 Data Register
$00008C_H$	T4CRH	T4CRL	Timer 4 Control Register

10.3 Register of 16-bit Timer (Timer 0 to 4)

Register configuration/functions of 16-bit timer (timer 0 to 4) is shown.

■ Timer Control Register L (TxCRL)

Figure 10.3-1 Timer control register L (TxCRL)

bit	7	6	5	4	3	2	1	0	Initial value
	−	−	−	LREQ	LFLG	TC1	TC0	TSTR	---0 0000B
Access→	−	−	−	R/W	R	R/W	R/W	R/W	

[bit7 to 5]:

It is an unused bit.

[bit4]:LREQ

It is timer data load request bit.

0	None
1	Load request is performed timer data to compare latch and output compare data to output latch.

Always read "0" at reading.

[bit3]:LFLG

It is timer data load complete flag bit.

0	Timer data load complete.
1	Timer data load does not complete.

[bit2]:TC1

[bit1]:TC0

There are clock source selection bit. Timer 0 to 4 is as follow.

Table 10.3-1 Timer 0 to 3

TC1	TC0	Selection Clock		Clock cycle time (fch:@20MHz)
0	0	Internal clock	2^4/fch (FRC3)	0.8 µs
0	1		2^6/fch (FRC5)	3.2 µs
1	0		2^8/fch (FRC7)	12.8 µs
1	1		2^{10}/fch (FRC9)	51.2 µs

Table 10.3-2 Timer 4

TC1	TC0	Selection Clock		Clock cycle time (fch:@20 MHz)
0	0	Internal clock	2^4/fch (FRC3)	0.8 µs
0	1		2^6/fch (FRC5)	3.2 µs
1	0		2^8/fch (FRC7)	12.8 µs
1	1	External clock	EC4	Min ***** µs

*:fch: source oscillation frequency

[bit0]:TSTR

Table 10.3-3 Timer operation enable bit

0	Timer operation stop
1	Enabling timer Operation

■ Timer Control Register H (TxCRH)

Figure 10.3-2 Timer control register H (TxCRH)

bit	7	6	5	4	3	2	1	0	Initial value
	Test	–	–	–	–	FCLR	IFLG	TIE	0--- -1X0$_B$
Access →	R/W	–	–	–	–	R/W	R	R/W	

[bit7]:Test

It is for a test.

Always write "0" to this bit.

[bit6-3]:

It is an unused bit.

[bit2]:FCLR

It is match detection flag clear bit.

0	Clear match detection flag.
1	None

Always read "1" at reading.

[bit1]:IFLG

It is match detection flag.

0	No match detection
1	Match detection

[bit0]:TIE

It is interrupt enable bit.

0	Interrupt interdiction
1	Interrupt permission

■ Timer Data Register H, L (TxDRH, TxDRL)

Figure 10.3-3 Timer data register H, L (TxDRH, TxDRL)

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	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
									XXXX XXXX
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The counter is cleared by the register that sets the value to be compared with the counter value when matched with the counter value.

■ Timer Count Register H, L (TxCDH, TxCDL)

Figure 10.3-4 Timer count register H, L (TxCDH, TxCDL)

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	R	R	R	R	R	R	R	R	

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	R	R	R	R	R	R	R	R	

10.4 Operation of 16-bit Timer (Timer 0 to 4)

In terms of the 16-bit timer, the internal clock can be selected from four types by setting the clock source selection bit (TC1, TC0) of the timer control register (TxCRL). The timer data register (TxDRH, TxDRL) will be the one used for setting the interval time, and reading the timer count data register (TxCDH, TxCDL) enables the timer count value to be known.

■ Operation of 16-bit Timer (Timer 0 to 4)

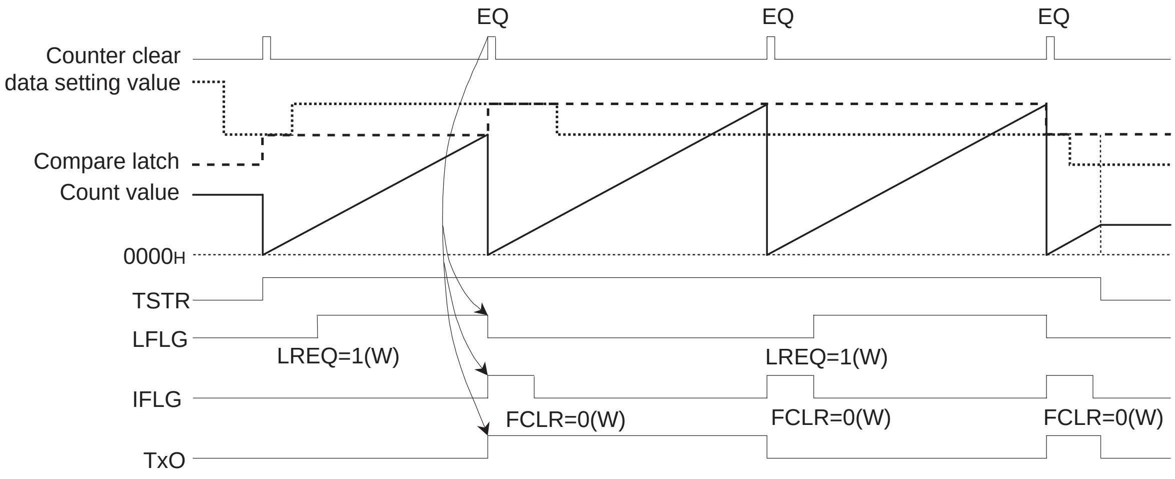
When the timer starts up, counting begins after the counter is cleared to "0000_H" by writing "1" to the timer start bit (TSTR) of the timer control register "L" after setting the interval time to the timer data register. The compare latch of the timer data register is loaded immediately after writing if the timer is suspended.

Matching the counter value with the set value of the compare latch sets the match detection flag (IFLG) to "1". In this case, the counter is cleared to "0000_H" if the load request flag is set, the data register value is loaded to the compare latch, and counting continues. In terms of the interval time "T", when "n" is selected as the data register set value, and "ϕ" is selected as the selection clock, the interval time "T" becomes below.

$T= \Phi \times (n+1)$ The toggle output frequency fTO is as follow.

$fTO=1/\{\Phi \times (n+1) \times 2\}$

Figure 10.4-1 Operation of 16-bit timer

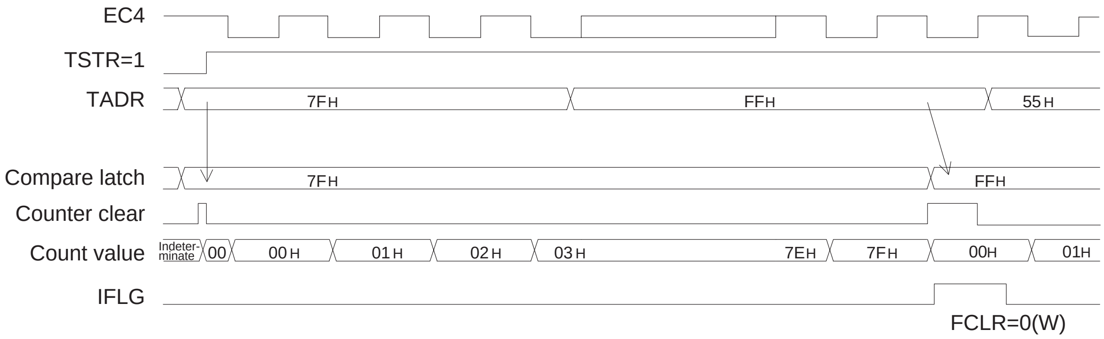


■ External Clock Mode (Only Timer 4)

The external clock input is selected as the external clock mode by the clock source selection bit (TC1, TC0) of the timer control register (T4CR).

In order to start up the timer, writing "1" to the T4CR timer start bit (TSTR) clears the counter, and then counting starts. The number of events can be found by reading the count data register (T4CD).

Figure 10.4-2 External Clock Mode operation



10.5 Overview of 8-/16-bit Timer/Counter

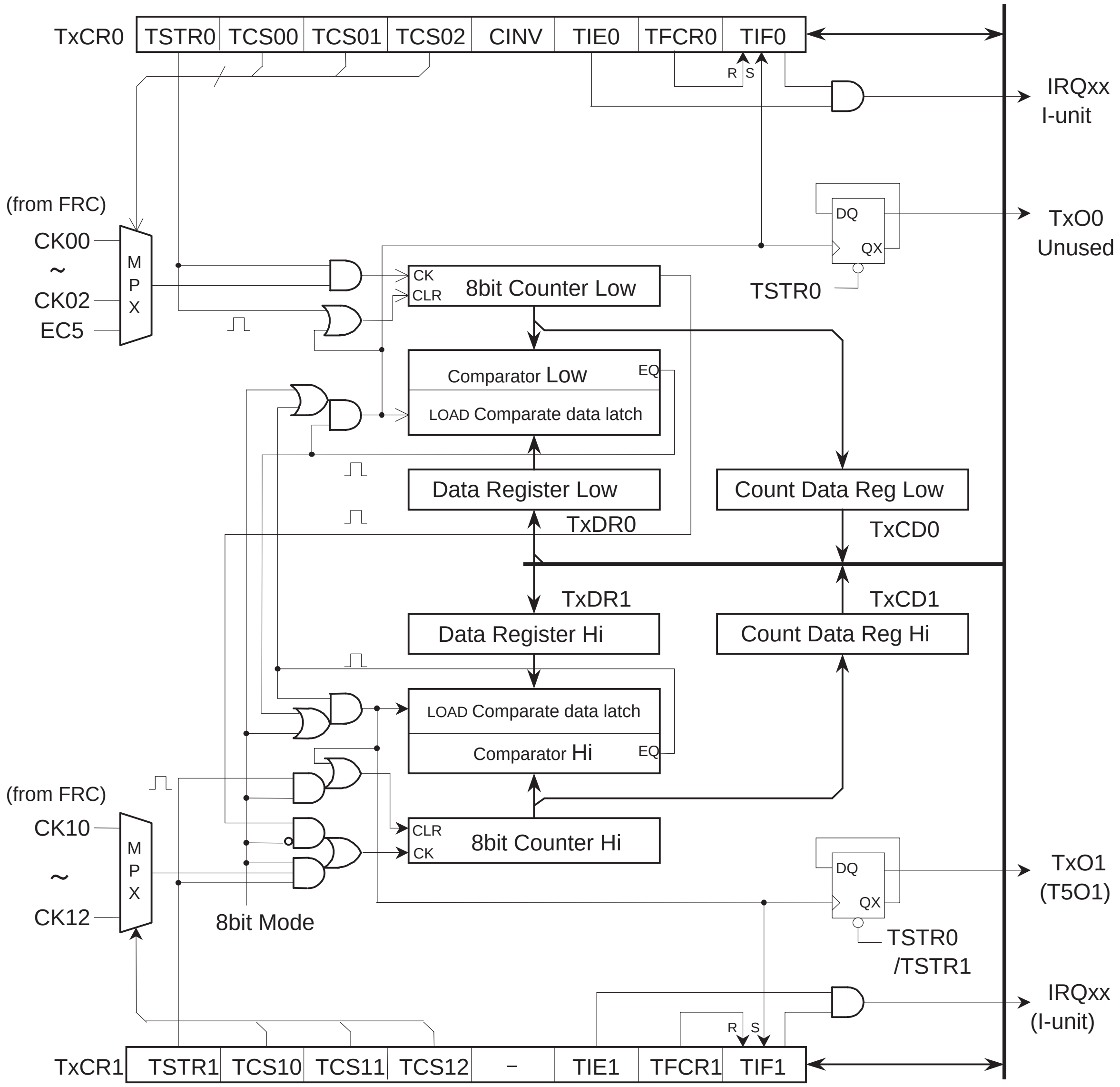
3 internal and 1 external clock input can be selected.

■ Feature of 8-/16-bit Timer/Counter

- 3 internal and 1 external clock input can be selected.
- Usable as an 8-bit timer/counter and 8-bit timer under 8-bit 2ch mode.
- Usable as 16-bit timer/counter under 16-bit mode.
- Square wave output function

■ Block Diagram of 8-/16-bit Timer/Counter

Figure 10.5-1 Block diagram of 8-/16-bit Timer/Counter



■ Register List of 8-/16-bit Timer/Counter

Figure 10.5-2 Register list of 8-/16-bit Timer/Counter

Address:	bit 15 ←→ 8	7 ←→ 0	
00008EH	TxCR1	TxCR0	Timer Control Register 1,0
000090H	TxDR1	TxDR0	Timer Data Register 1,0
000092H	TxCD1	TxCD0	Count Data Register 1,0

10.6 Register of 8-/16-bit Timer/Counter

Register configuration/functions of 8-/16-bit timer/counter is shown.

■ Timer Control Register 1 (TxCR1)

Figure 10.6-1 Timer control register 1(TxCR1)

bit	7	6	5	4	3	2	1	0	Initial value
	TIF1	TFCR1	TIE1	−	TCS12	TCS11	TCS10	TSTR1	010- 0000B
Access→	R	W	R/W	−	R/W	R/W	R/W	R/W	

[bit7]:TIF1

It is compare match detection flag bit.

0	Compare match does not generate.
1	Compare match generates.

[bit6]:TFCR1

It is compare match detection flag clear bit.

0	Clear compare match detection flag.
1	None

The read value of this bit is always "1".

[bit5]:TIE1

It is interrupt enable bit of timer 1.

0	Interdiction
1	Permission

[bit4]:

It is an unused bit.

[bit3]:TCS12

[bit2]:TCS11

[bit1]:TCS10

It is clock source selection bit of timer.

TCS12	TCS11	TCS10	Selection Clock source	in fch = @20 MHz
0	0	0	2^{4}/fch (FRC3)	0.8 µs
0	0	1	2^{6}/fch (FRC5)	3.2 µs
0	1	0	2^{8}/fch (FRC7)	12.8 µs
0	1	1	Setting disabled	
1	0	0		
1	0	1		
1	1	0		
1	1	1	16-bit mode	-

[bit0]:TSTR1

It is timer start bit.

0	Operation stop
1	Clear counter and start operation.

■ Timer Control Register 0 (TxCR0)

Figure 10.6-2 Timer control register 0 (TxCR0)

bit	7	6	5	4	3	2	1	0	Initial value
	TIF0	TFCR0	TIE0	CINV	TCS02	TCS01	TCS00	TSTR0	0100 0000$_B$
Access →	R	W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7]:TIF0

It is compare match detection flag bit.

0	Compare match does not generate.
1	Compare match generates.

[bit6]:TFCR0

It is compare match detection flag clear bit.

0	Clear compare match detection flag.
1	None

The read value of this bit is always "1".

[bit5]:TIE0

It is interrupt enable bit of timer 0.

0	Interdiction
1	Permission

[bit4]:CINV

It is count clock reverse bit.

0	Count up by falling of selection clock source.
1	Count up rising of selection clock source.

[bit3]:TCS02

[bit2]:TCS01

[bit0]:TCS00

It is clock source selection bit.

TCS02	TCS01	TCS00	Selection Clock source	in fch = @20 MHz
0	0	0	2 ^{4}/fch (FRC3)	0.8 µs
0	0	1	2 ^{6}/fch (FRC5)	3.2 µs
0	1	0	2 ^{8}/fch (FRC7)	12.8 µs
0	1	1	Setting disabled	
1	0	0		
1	0	1		
1	1	0		
1	1	1	External Clock Mode	-

[bit0]:TSTR0

It is timer start bit.

0	Operation stop
1	Clear counter and start operation.

■ Timer Data Register 1, 0 (TxDR1, TxDR0)

Figure 10.6-3 Timer data register 1, 0 (TxDR1, TxDR0)

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	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
									XXXX XXXX
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The timer data register value is loaded to the compare latch, and used as data compared to the counter value. Use the setting of the interval time under the timer mode. Also used to set the number of events to be detected under the external clock mode.

■ Count Data Register (TxCD1, TxCD0)

Figure 10.6-4 Count data register (TxCD1, TxCD0)

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	R	R	R	R	R	R	R	R	

bit	7	6	5	4	3	2	1	0	Initial value
									XXXX XXXX
Access →	R	R	R	R	R	R	R	R	

Read operation under the timer/counter mode reads the counter value.

10.7 Operation of 8-/16-bit Timer/Counter

In terms of 8-/16-bit timer/counter operations, there are controls for the 8-bit internal clock mode, 8-bit external clock mode (event counter), and 16-bit mode.

■ Operation in 8-bit Internal Clock Mode

In terms of the 8-bit internal clock mode, the internal clock can be selected from seven types by setting the clock source selection bit (TCS02, TCS01, TCS00) (TCS12, TCS11, TCS10) of the timer control register 0, 1 (TxCR0, TxCR1). The timer data register 0, 1 (TxDR0, TxDR1) will be the register for setting the interval time.

When the timer starts up, writing "1" to the timer start bit (TSTR0, TSTR1) of the timer control register after setting the interval time to the data register clears the counter to "00_H", and loads the data register value to the compare latch. Then counting starts.

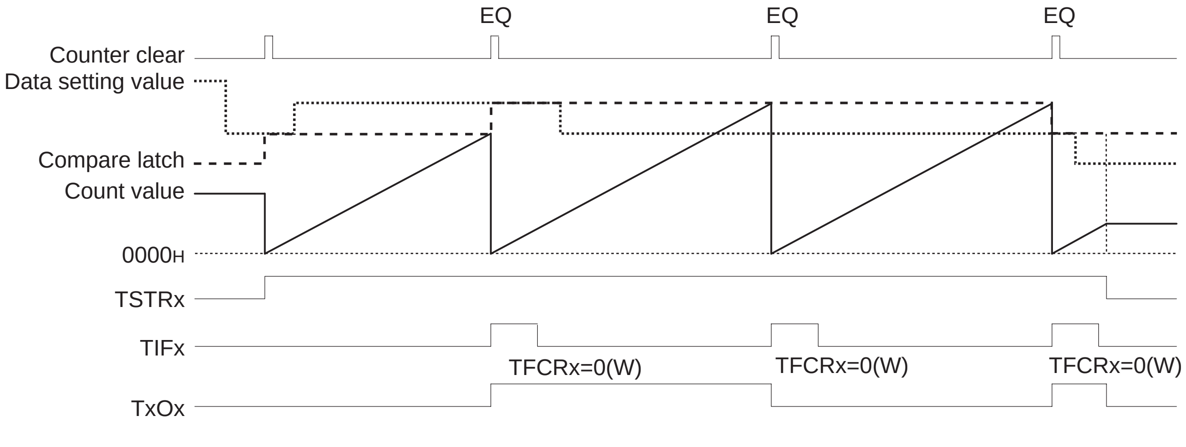
When the counter value matches the compare latch value, "1" is set as the compare match detection flag (TIF0, TIF1). In this case, the counter is cleared to "00_H", and the data register value is reloaded to the compare latch, and counting continues. Also, the toggle output (TxOx) reverses the output per compare match.

When "n" is selected as the data register set value, and "Φ" is selected as the selection clock, the interval time "T" becomes below.

$T = \Phi \times (n+1)$ The toggle output frequency fTO is as follow.

$fTO=1/\{\Phi \times (n+1) \times 2\}$

Figure 10.7-1 Operation of Internal clock mode

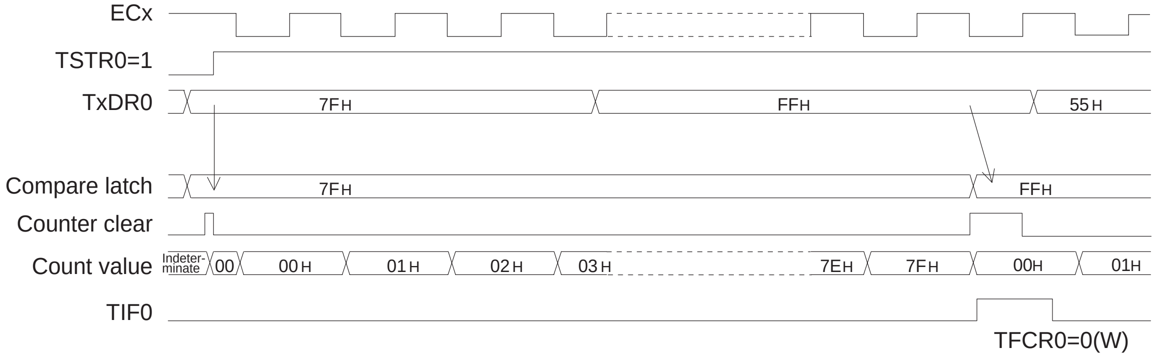


■ Operation of 8-bit External Clock Mode (Event Counter)

The external clock mode is used to select the external clock input using the clock source selection bit (TCS02, TCS01, TCS00) of the timer control register 0 (TxCR0).

In order to start up the timer, writing "1" to the TxCR0 timer start bit (TSTR0) clears the counter, and then counting starts. The number of events can be found by reading the count data register (TxCDL).

Figure 10.7-2 Operation of external Clock Mode



■ Control in 16-bit Mode

The 16-bit mode can be used as the 16-bit timer/counter in the same way as the 8-bit mode by setting each bit of the timer control register (TxCR1, TxCR1) as Figure 10.7-3 .

Figure 10.7-3 Control in 16-bit mode

bit	7	6	5	4	3	2	1	0
TxCR1	TIF1	TFCLR1	TIE1	——	TCS12	TCS11	TCS10	TSTR1
	No relation		Set ot 0		Set to 111			Set ot 0

bit	7	6	5	4	3	2	1	0
TxCR0	TIF0	TFCLR0	TIE0	CINV	TCS02	TCS01	TCS00	TSTR0
	Flag			Clock polling	Clock select			Start/Stop

In terms of the 16-bit mode setting, set "111" to the TCS12, TCS11, and TCS10 bits of timer control register 1 (TxCR1).

In the 16-bit mode, timers are controlled by the timer control register 0 (TxCR0). As regards the data register in this case, TxDR1 is the superior byte and TxDR0 is the subordinate byte. For the count data register, TxCD1 is the superior byte and TxCD0 is the subordinate byte.

The selection of the clock source is performed in the TCS02, TCS01, and TCS00 bits of TxCR0. In order to start the timer, writing "1" to the TSTR0 bit of TxCR0 clears the counter, and counting starts.

When the counter value matches the compare latch value, the TIF0 bit of TxCR0 is set to "1". As above, under 16-bit mode, the timer 0 side function can be used as it is in the extended 16-bit format.

Refer to the "■Operation in 8-bit Internal Clock Mode, Operation of 8-bit external clock mode (Event counter)" for operations under 16-bit mode.

CHAPTER 11
12-bit PWM

This chapter describes an outline of the PWM, the register configuration/functions, and the PWM operations.

11.1 Overview of 12-bit PWM

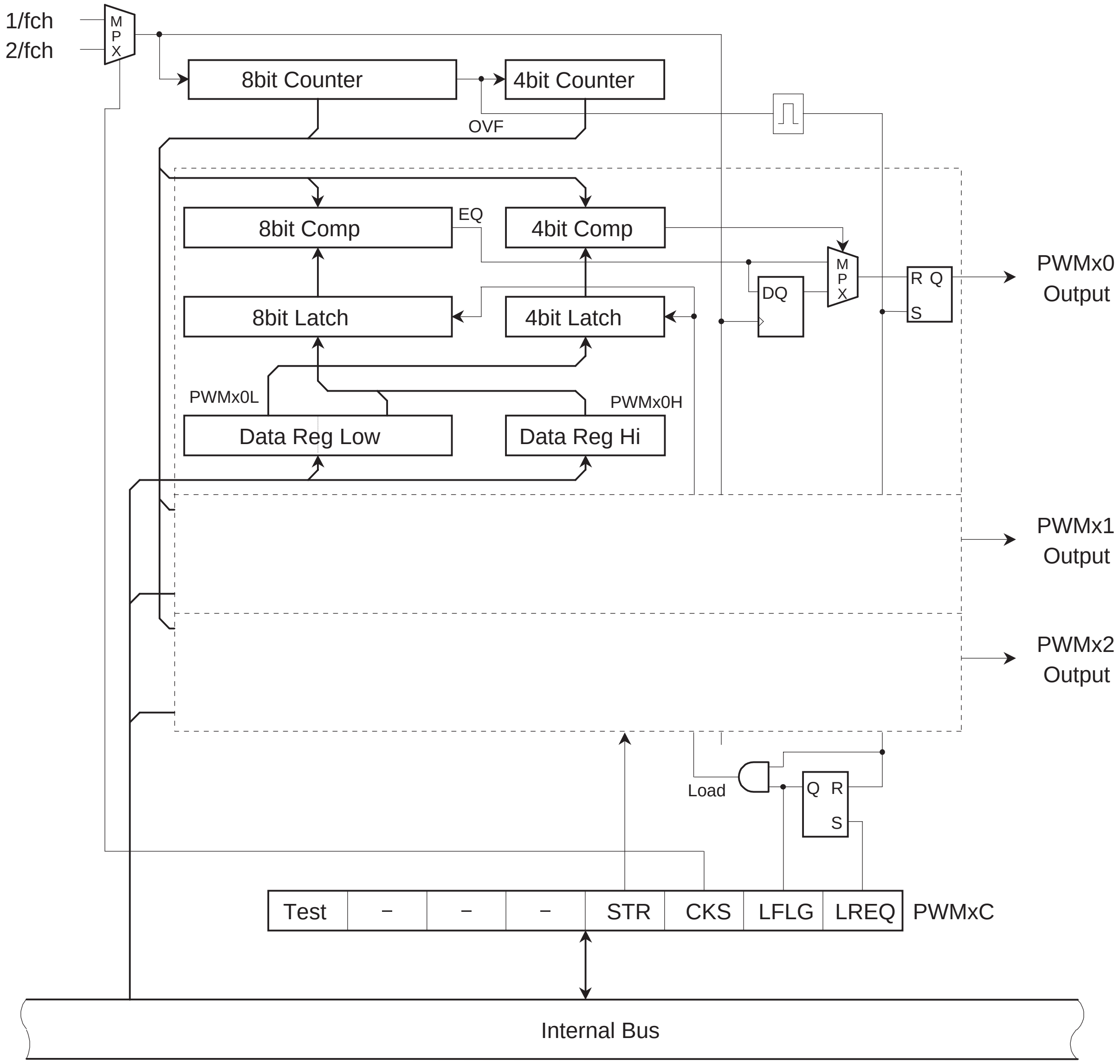
The 12-bit PWM has a 12-bit resolution using the duty control and added pulse based on the selectable basic frequency.

■ Feature of 12-bit PWM

- 12-bit resolution (rate, multi-type)
- The conversion cycle can be selected: 0.2 ms/basic frequency 78.1 kHz (in fch:@20 MHz) and 0.4 ms/basic frequency 39.0 kHz (in fch:@20 MHz).

■ Block Diagram of 12-bit PWM

Figure 11.1-1 Block Diagram of 12-bit PWM



■ Register list of 12-bit PWM

Figure 11.1-2 Register list of 12-bit PWM

Address: bit	15 ←→ 8	7 ←→ 0	
000040H	–	PWM0C	PWM0 Control Register
000042H	PWM00H	PWM00L	PWM00 Data Register
000044H	PWM01H	PWM01L	PWM01 Data Register
000046H	PWM02H	PWM02L	PWM02 Data Register
000048H	–	PWM1C	PWM1 Control Register
00004AH	PWM10H	PWM10L	PWM10 Data Register
00004CH	PWM11H	PWM11L	PWM11 Data Register
00004EH	PWM12H	PWM12L	PWM12 Data Register

11.2 Register of 12-bit PWM

Register configuration/functions of 12-bit PWM is shown.

■ PWMx Control Register (PWMxC)

Figure 11.2-1 PWMx Control Register (PWMxC)

bit	7	6	5	4	3	2	1	0	Initial value
	Test	–	–	–	STR	CKS	LFLG	LREQ	0--- 00X0B
Access→	R/W	–	–	–	R/W	R/W	R	W	

[bit7]:Test

Please set "0".

[bit6 to 4]:

It is an unused bit.

[bit3]:STR

It is PWM operation enable bit.

0	Operation disabled
1	Operation enabled

[bit2]:CKS

It is clock select bit.

0	1/fch (fch: 50 ns in @20 MHz)
1	2/fch (fch: 100 ns in @20 MHz)

[bit1]:LFLG

It is load request flag to compare latch.

0	No load request and complete load.
1	Load request

[bit0]:LREQ

It is load request bit.

0	None
1	Load data register value to compare latch.

The read value of this bit is always "0".

■ PWMx0 to 2 Data Register (PWMDxx)

Figure 11.2-2 PWMx0 to 2 Data Register (PWMDxx)

bit	7	6	5	4	3	2	1	0	Initial value
	–	–	–	–	D11	D10	D9	D8	---- XXXXB
Access →	–	–	–	–	W	W	W	W	

bit	7	6	5	4	3	2	1	0	Initial value
	D7	D6	D5	D4	D3	D2	D1	D0	XXXX XXXXB
Access →	W	W	W	W	W	W	W	W	

[bit11 to 4]:D11 to 4

It is duty control bit.

Control duty of basic frequency.

[bit3 to 0]:D3 to 0

It is added pulse control bit.

Control position of added pulse.

Use the half-word access command to write to this register.

(Byte access command cannot be write.)

11.3 Operation of 12-bit PWM

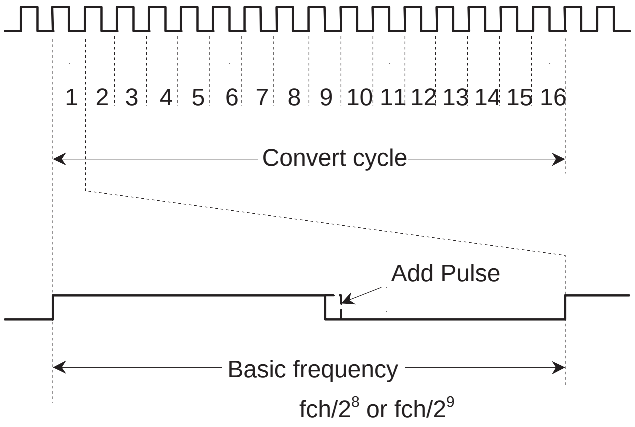
The 12-bit PWM acquires a 12-bit resolution by inserting added pulse to the output waveform using the 8-bit and 4-bit counters.

■ Operation of 12-bit PWM

The 12-bit PWM is operated by the 8-bit and 4-bit counters. The 8-bit counter will be the basic frequency at the time of PWM conversion, and output waveform duty is specified by the duty control-bit of the data register. In terms of the conversion operation, this is repeated 16 times, and 12-bit resolution is acquired by inserting an added pulse to the specific position during repeated operation. Insertion position of the added pulse is specified by the added pulse control bit.

The PWM output waveform will be as Figure 11.3-1 .

Figure 11.3-1 PWM operation waveform



A pulse is added when "1" is specified in accordance with the added pulse control bit value.

Table 11.3-1 Pulse Insert Position

DATA	Insertion position of added pulse															
0000_B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001_B	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-
0010_B	-	-	-	-	1	-	-	-	-	-	-	-	1	-	-	-
0011_B	-	-	-	-	1	-	-	-	1	-	-	-	1	-	-	-
0100_B	-	-	1	-	-	-	1	-	-	-	1	-	-	-	1	-
0101_B	-	-	1	-	-	-	1	-	1	-	1	-	-	-	1	-
0110_B	-	-	1	-	1	-	1	-	-	-	1	-	1	-	1	-
0111_B	-	-	1	-	1	-	1	-	1	-	1	-	1	-	1	-
1000_B	-	1	-	1	-	1	-	1	-	1	-	1	-	1	-	1
1001_B	-	1	-	1	-	1	-	1	1	1	-	1	-	1	-	1
1010_B	-	1	1	1	-	1	-	1	-	1	-	1	1	1	-	1
1011_B	-	1	-	1	1	1	-	1	1	1	-	1	1	1	-	1
1100_B	-	1	1	1	-	1	1	1	-	1	1	1	-	1	1	1
1101_B	-	1	1	1	-	1	1	1	1	1	1	1	-	1	1	1
1110_B	-	1	1	1	1	1	1	1	-	1	1	1	1	1	1	1
1111_B	-	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

■ Update procedure of PWM data

1. Waits for load completion when the load request flag (LFLG) is set.
2. Sets the set value to each data register and writes "1" to the load request bit (LREQ). →The load request flag (LFLG) is set. Do not update the data when LFLG is 1. It may cause the malfunction.
3. When the load is completed, and the LFLG is cleared. (The load operation generates 1/interval period of basic frequency.)

CHAPTER 12
8-bit Pulse Width Counter

This chapter describes an outline of the 8-bit pulse width counter, the register configuration/functions, and 8-bit pulse width counter operations.

12.1 Overview of 8-bit Pulse Width Counter

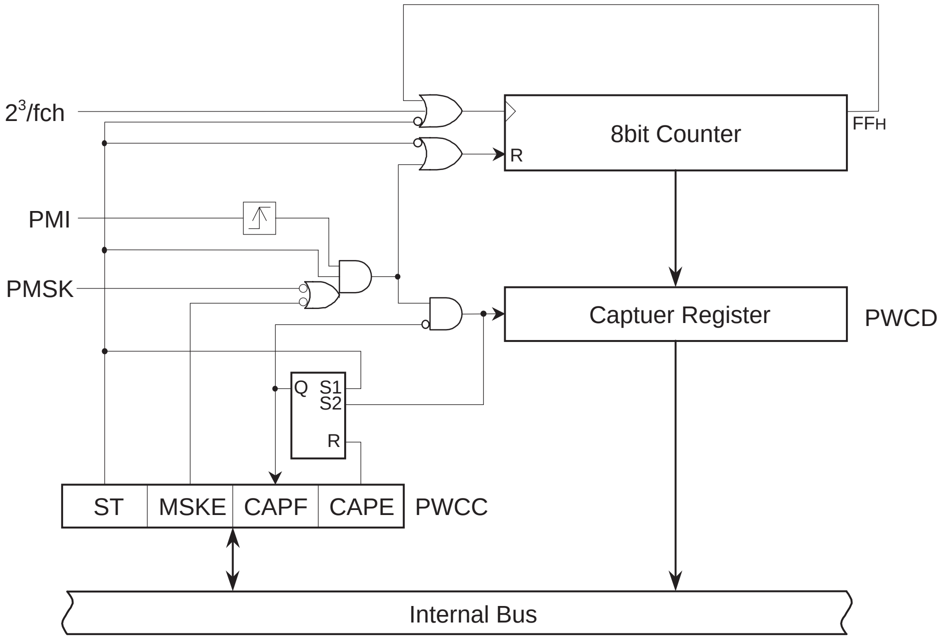
The 8-bit pulse width counter measures the pulse width by 400 ns accuracy.

■ Feature of 8-bit Pulse Width Counter

- Pulse width measurement accuracy 400 ns (in fch = @20 MHz)
- Mask input function

■ Block Diagram of 8-bit Pulse Width Counter

Figure 12.1-1 Block Diagram of 8-bit pulse width counter



■ Register List of 8-bit Pulse Width Counter

Figure 12.1-2 Register list of 8-bit pulse width counter

bit	7 ←→ 0	
Address: 000094H	PWCC	PWC Control Register
000095H	PWCD	PWC Data Register

12.2 Register of 8-bit Pulse Width Counter

The register configuration/functions of the 8-bit pulse width counter are mentioned.

■ PWC Control Register (PWCC)

Figure 12.2-1 PWC Control Register (PWCC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000094$_H$	Test	–	–	–	CAPE	CAPF	MSKE	ST	0--- 0X00$_B$
Access →	R/W	–	–	–	W	R	R/W	R/W	

[bit7]:Test

It is test bit.

Please set "0".

[bit6 to 4]:

It is an unused bit.

[bit3]:CAPE

It is capture enable bit.

0	None
1	Clear CAPF bit and become capture enable status.

The read value of this bit is always "0".

[bit2]:CAPF

It is capture flag.

0	No capture data. Capture enable status
1	Capture data. Capture disable status

[bit1]:MSKE

It is mask (PMSK) input enable bit.

0	PMSK input interdiction
1	PMSK input permission

[bit0]:ST

It is PWC start bit.

0	PWC stop. CAPF is fixed "1".
1	PWC Operation

■ PWC Data Register (PWCD)

Figure 12.2-2 PWC data register (PWCD)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000095H									XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

It is a register to store the measurement value of the pulse width.

12.3 Operation of 8-bit Pulse Width Counter

The pulse width counting operation of the 8-bit pulse width counter is described.

■ Pulse Input Mask Function

Pulse input (PMI) can be directly masked by the pulse mask input signal (PMSK).

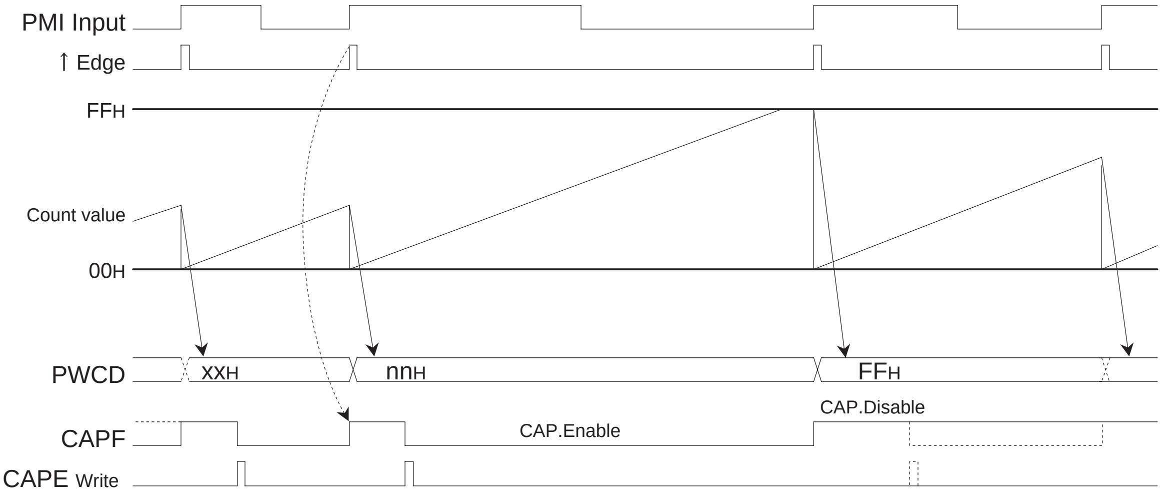
In order to control PMI input by PMSK input, set "1" to the mask input enabled (MSKE) bit of the PWC control register (PWCC) for PMSK input enabled status. Therefore, the PMI input can be controlled by the PMSK input signal.

■ Pulse Width Count Operation

The PWC clears the counter value capture and counter at the rising edge of the PMI signal. The cleared counter continues the counting operation as it is, and when the counter value becomes "FF_H", counting stops, and retains that status until detecting the rising edge of the next PMI input.

In terms of the capture operation, when the capture flag (CAPF) is "0" (enabled status), capture is executed, whereas when CAPF is "1" (disabled status), capture is not executed. These operation shows in Figure 12.3-1 .

Figure 12.3-1 PWC operation



■ CAPF Bit

The PWC initializes the CAPF bit to "1", when the ST bit is "0". In this case, do not clear the CAPF bit during the operation is stopped.

Clearing the CAPF bit must be executed after enabling the PWC operation.

CHAPTER 13
External Interrupt

External interrupt comprises of the key input interrupt and external interrupt sections. This chapter describes an outline of the external interrupt 1 (key input circuit) and external interrupt (INT0 to 2), and the register configuration/functions, and their operation.

13.1 Overview of External Interrupt

External interrupt comprises of the key input interrupt and external interrupt sections, and a total of 11 factors can be received.

■ Feature of External Interrupt

External interrupts comprise of the key input interrupt and external interrupt sections.

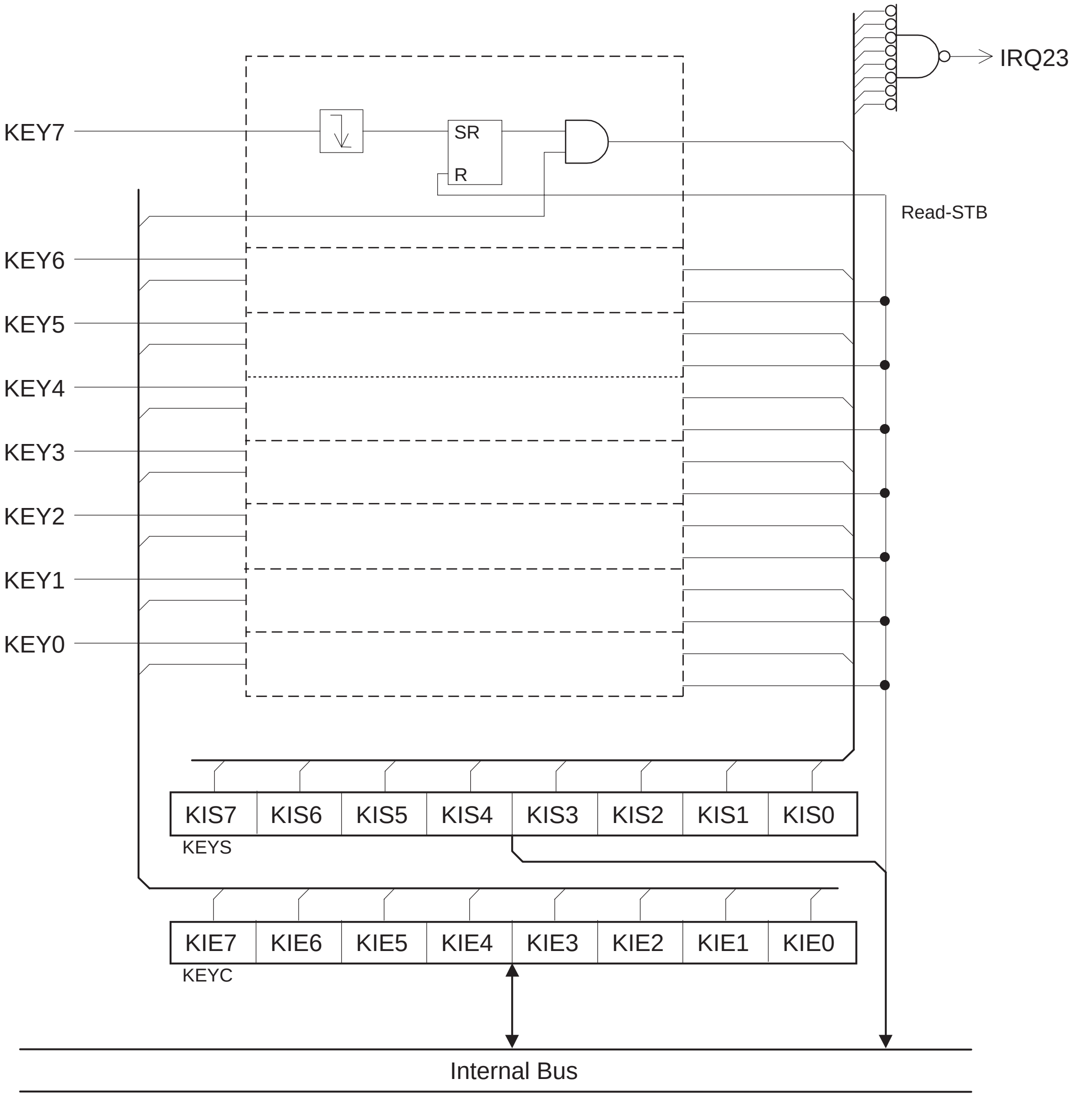
- External interrupt3 input
- Key input interrupt 8 input

13.2 External Interrupt 1 (Key Input Circuit)

The external interrupt 1 (key input circuit) has 8 inputs.

■ Block Diagram of External Interrupt 1 (Key Input Circuit)

Figure 13.2-1 Block Diagram of Key input circuit



■ Register List of External Interrupt 1 (Key Input Circuit)

Figure 13.2-2 Register list of External interrupt 1 (Key input circuit)

	bit 7 ← → 0	
Address: 000096H	KEYC	Key Input Control Register
000097H	KEYS	Key Input Status Register

■ Key Input Control Register (KEYC)

Figure 13.2-3 Key input control register (KEYC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000096H	KIE7	KIE6	KIE5	KIE4	KIE3	KIE2	KIE1	KIE0	0000 0000B
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7 to 0]:KIE7 to 0

There are input enable bit of KEY7 to 0.

0	KEY input interdiction
1	KEY input permission

■ Key Input Status Register (KEYS)

Figure 13.2-4 Key input status register (KEYS)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000097H	KIS7	KIS6	KIS5	KIS4	KIS3	KIS2	KIS1	KIS0	XXXX XXXXB
Access →	R	R	R	R	R	R	R	R	

[bit7 to 0]:KIS7 to 0

There are edge detection flag of KEY7 to 0.

0	None
1	Falling edge detection

These flags are cleared by the read operation.

■ Operation of External Interrupt 1 (Key Input Circuit)

In order to use the key input, Port A input must be enabled. If Port A input is disabled while key input is enabled, a key input edge detection interrupt may be generated.

The key input sets the flip-flop when the input signal falling edge is detected. In this case, if the supported key input is enabled, an interrupt request is generated. The generated interrupt request can be also used to return from standby status, etc.

Do not use the bit operation command to read the key input status register.

13.3 External Interrupt (INT0 to 2)

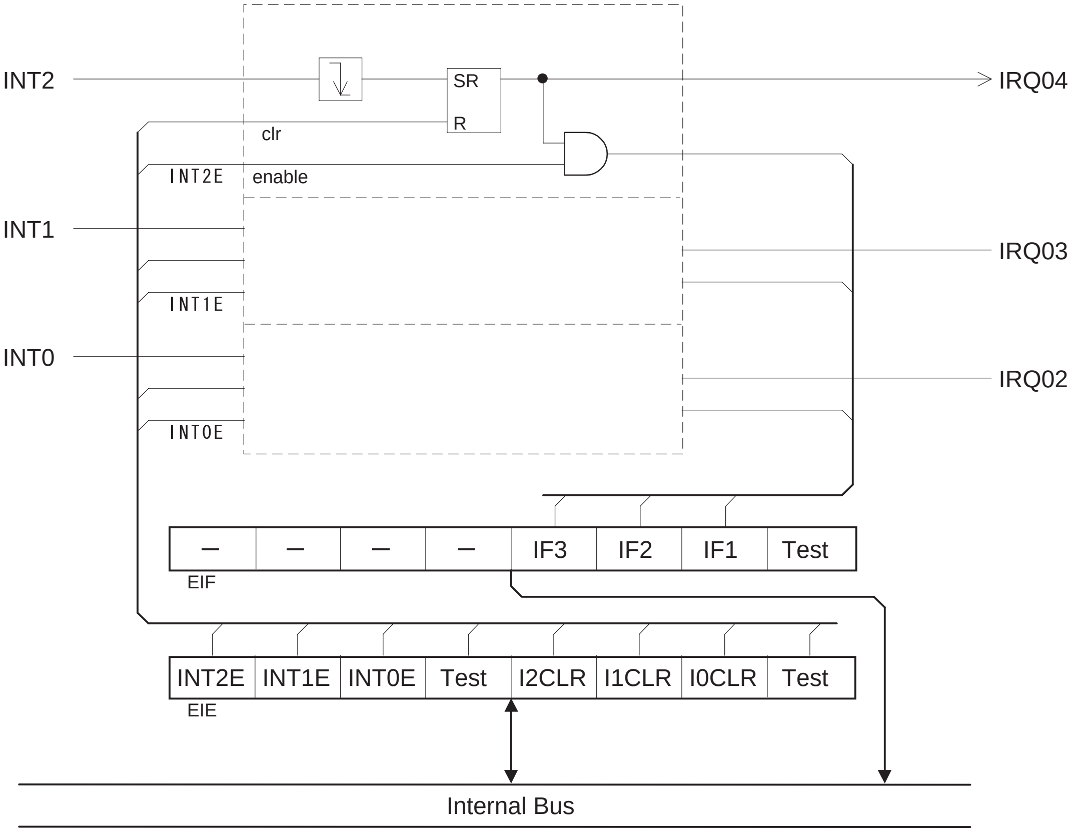
The external interrupt (INT0 to 2) has 3 inputs.

■ Feature of External Interrupt (INT0 to 2)

- Each interrupt is independent vector.
- Usable for returning from standby.

■ Block Diagram of External Interrupt (INT0 to 2)

Figure 13.3-1 Block diagram of external interrupt (INT0 to 2)



■ Register List of External Interrupt (INT0 to 2)

Figure 13.3-2 Register list of external interrupt (INT0 to 2)

Address:	bit 7 ←→ 0	
000098H	EIE	External Interupt Enable R
000099H	EIF	External Interupt Request

■ External Interrupt Enable Register (EIE)

Figure 13.3-3 External interrupt enable register (EIE)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000098H	INT2E	INT1E	INT0E	Test	I2CLR	I1CLR	I0CLR	−	0000 111-B
Access →	R/W	R/W	R/W	R/W	W	W	W	W	

[bit7 to 5]:INT2E to 0E

There are interrupt enable bits of INT2 to 0.

0	Interrupt interdiction
1	Interruption permission

[bit4]:Test

It is test control bit.

Please set "0".

[bit3 to 1]:I2CLR to I0CLR

There are interrupt flag clear bits of INT2 to 0.

0	The flag is clear.
1	None

The read value of this bit is always "1".

[bit0]:

It is an unused bit.

■ External Interrupt Request Flag (EIF)

Figure 13.3-4 External interrupt request flag (EIF)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000099H	−	−	−	−	IF2	IF1	IF0	Test	---- XXXXB
Access →	−	−	−	−	R	R	R	R	

[bit7 to 4]:

It is an unused bit.

[bit3 to 1]:IF2 to 0

There are input edge detection flag of INT2 to 0.

0	None
1	Falling edge detection

[bit0]:Test

It is test bit.

■ Operation of External Interrupt (INT0 to 2)

The external interrupt sets the interrupt request flag (IF2 to 0) when the falling edge of the input signal is detected. In this case, if the supported interrupt is enabled (INTxE=1), an interrupt request is generated to the interrupt controller, whereas if it is disabled, no interrupt request is generated.

The external interrupt can be used for returning the standby status.

Before enabling interrupts, clear the interrupt request flag.

CHAPTER 14

Delayed Interrupt Module

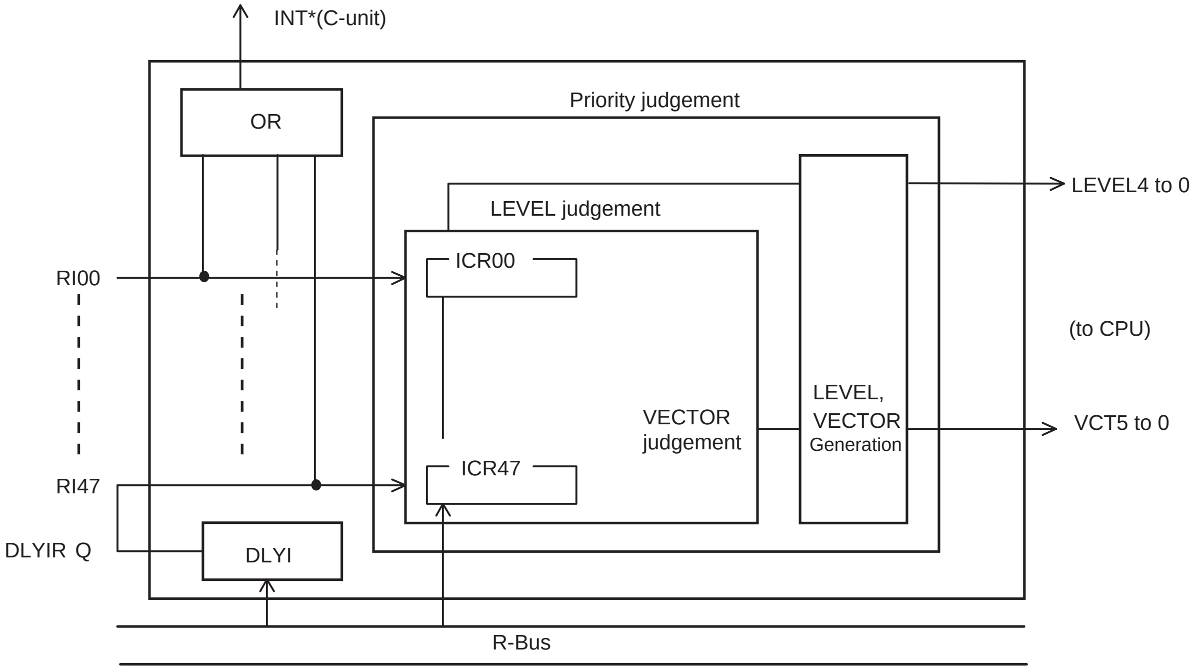
This chapter describes an outline of the delayed interrupt module, the register configuration/functions, and delayed interrupt module operations.

14.1 Overview of Delayed Interrupt Module

The delay interrupt module is used to generate interrupts for switching tasks. Using this module enables interrupt requests to the CPU to be generated or cancelled by the software.

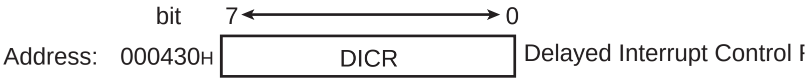
■ Block Diagram of Delayed Interrupt Module

Figure 14.1-1 Block diagram of delayed interrupt module



■ Register List of Delayed Interrupt Module

Figure 14.1-2 Register list of Delayed Interrupt Module



14.2 Delayed Interrupt Control Register (DICR)

The delay interrupt control register (DICR) controls delay interrupts.

■ Delayed Interrupt Control Register (DICR)

Figure 14.2-1 Delayed interrupt control register (DICR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 00430H	–	–	–	–	–	–	–	DLYI	---- ---0B
Access→	–	–	–	–	–	–	–	R/W	

[bit7 to 1]:

It is an unused bit.

[bit0]:DLYI

It is EINT input edge detection flag.

Table 14.2-1

0	Delayed interrupt factor cancellation and no request
1	Delayed interrupt factor generation

This bit is controlled the generation or cancellation of the corresponding interrupt factor.

14.3 Operation of Delayed Interrupt Module

Delay interrupts are used to generate interrupts for switching tasks. Using this function enables interrupt requests to the CPU to be generated or cancelled by the software.

■ Interrupt Number

The delay interrupt is allocated to the interrupt factor supporting the largest interrupt number. It will be the interrupt factor that supports the interrupt number 63 (3FH) for this product comprising of 48 factors.

■ DLYI Bit of DICR

Writing "1" to this bit generates a delay interrupt factor. Also, writing "0" cancels a delay interrupt factor.

This bit is the same as the interrupt factor flag under general interrupts. Clear this bit within the interrupt routine, and simultaneously switch the task.

CHAPTER 15

Interrupt Controller

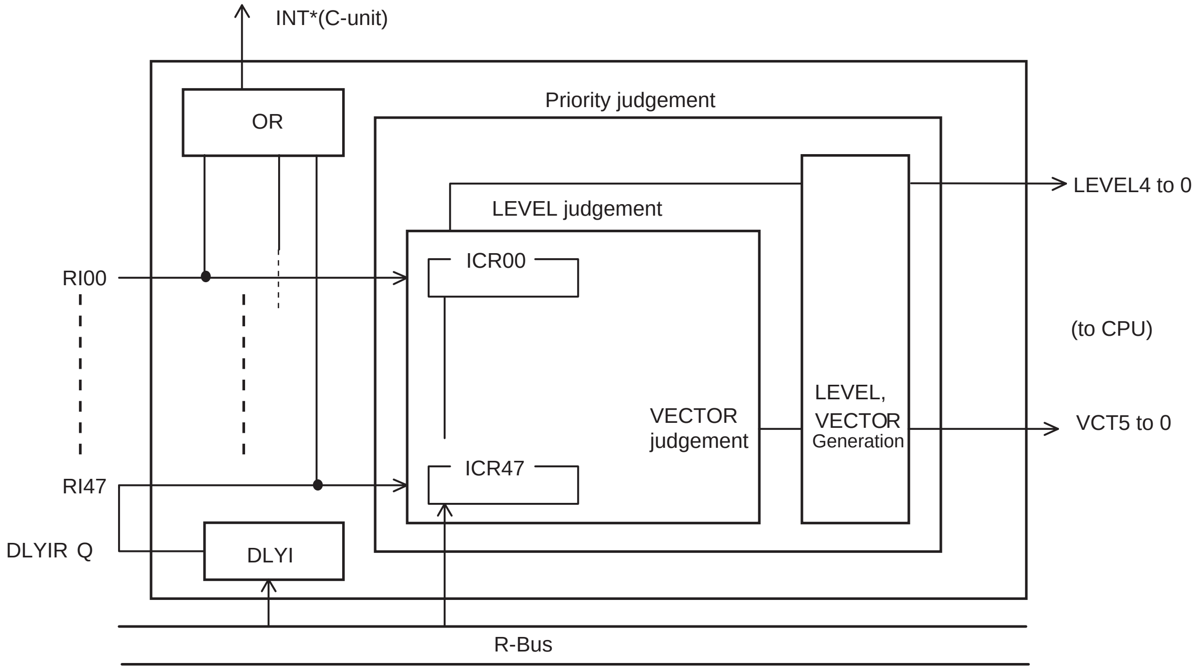
This chapter describes an outline of the interrupt controller, the register configuration/functions, and the interrupt controller operations.

15.1 Overview of Interrupt Controller

The interrupt controller comprises of the interrupt control register, interrupt priority decision circuit, interrupt level, and interrupt number generation sections, and controls interrupt reception and adjustment.

■ Block Diagram of Interrupt Controller

Figure 15.1-1 Block Diagram of Interrupt controller



Notes:

- The DLYI in the figure means the delayed interrupt section. (For details, see "CHAPTER 14 ".)
- INT* is the wake-up signal to the clock control sections during sleep/stop status.

■ Register List of Interrupt Controller

Figure 15.1-2 Register list of interrupt controller

	bit	7 ←→ 0	
Address:	000400H	ICR00	Interrupt Control Register 00
	000401H	ICR01	Interrupt Control Register 01
	000402H	ICR02	Interrupt Control Register 02
	000403H	ICR03	Interrupt Control Register 03
	000404H	ICR04	Interrupt Control Register 04
	∫	∫	∫
	00042BH	ICR43	Interrupt Control Register 43
	00042CH	ICR44	Interrupt Control Register 44
	00042DH	ICR45	Interrupt Control Register 45
	00042EH	ICR46	Interrupt Control Register 46
	00042FH	ICR47	Interrupt Control Register 47

15.2 Interrupt Control Register (ICRxx)

This is interrupt control register. One is set per interrupt input, and sets the interrupt level for the interrupt request to be supported.

■ Interrupt Control Register (ICRxx)

Figure 15.2-1 Interrupt control register (ICRxx)

bit	7	6	5	4	3	2	1	0	Initial value
	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	---1 1111$_B$
Access →	–	–	–	R	R/W	R/W	R/W	R/W	

[bit4 to 0] ICR4 to 0

The interrupt level of the interrupt request to be supported is specified by the interrupt level setting bit.

When the interrupt level set to this register is the level mask value that has been set to the ILM register of the CPU or higher, the interrupt request is masked by the CPU side.

Initialized to 11111$_B$ by reset.

Table 15.2-1 is shown the interrupt-level setting bits that can be set and the corresponding interrupt levels.

Table 15.2-1 Level setting and corresponding interrupt levels

ICR4	ICR3	ICR2	ICR1	ICR0	Interrupt level	
0	0	0	0	0	0	↕ System reservation
0	1	1	1	0	14	
0	1	1	1	1	15	
1	0	0	0	0	16	Highest level that can be set
1	0	0	0	1	17	↑ (High)
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	↓ (low)
1	1	1	1	1	31	Disables the interrupt.

*: ICR4 is fixed "1" and cannot be "0".

15.3 Operation of Interrupt Controller

The interrupt request generated at the resource is controlled by the interrupt-enabled bit that has been set per resource. If an interrupt is enabled at each resource, the generated interrupt request generates an interrupt request signal to the interrupt controller.

■ Priority Order Evaluation

The interrupt controller selects the highest priority factor from those that have been generated simultaneously, and outputs the factor's interrupt level and interrupt number to the CPU.

The criteria for evaluating the priority order of the interrupt causes are shown below.

● The interrupt cause meets the following conditions:

- The interrupt cause has an interrupt level other than 31 (A value of 31 represents interrupt disable.)
- The interrupt cause has the lowest interrupt level
- The interrupt cause has the smallest interrupt number

Table 15.3-1 shows the relationship between interrupt causes, interrupt numbers, and interrupt levels.

Figure 15.3-1 Flowchart of interrupt cause

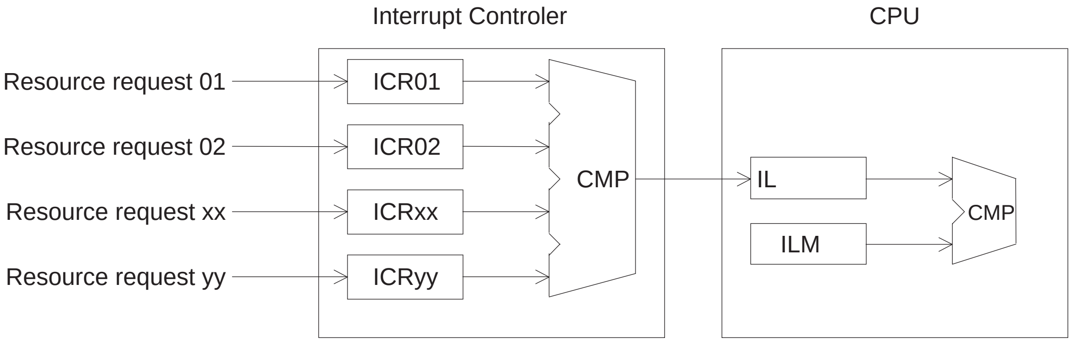


Table 15.3-1 Relationship between interrupt causes, interrupt numbers, and interrupt levels

Interrupt cause		Interrupt number		Interrupt level	Offset	TBR default address
		10 decimal	16 decimal			
IRQ0	Reserved	16	10_H	ICR00	$03BC_H$	$000FFFBC_H$
IRQ1	Reserved	17	11_H	ICR01	$03B8_H$	$000FFFB8_H$
IRQ2	External Interrupt 0 (INT0)	18	12_H	ICR02	$03B4_H$	$000FFFB4_H$
IRQ3	External Interrupt 1 (INT1)	19	13_H	ICR03	$03B0_H$	$000FFFB0_H$
IRQ4	External Interrupt 2 (INT2)	20	14_H	ICR04	$03AC_H$	$000FFFAC_H$
IRQ5	24-bit Free Run Counter/Capture	21	15_H	ICR05	$03A8_H$	$000FFFA8_H$
IRQ6	Programmable Pulse Generator ch0	22	16_H	ICR06	$03A4_H$	$000FFFA4_H$
IRQ7	Programmable Pulse Generator ch1	23	17_H	ICR07	$03A0_H$	$000FFFA0_H$
IRQ8	Real timing Generator ch0	24	18_H	ICR08	$039C_H$	$000FFF9C_H$
IRQ9	Real timing Generator ch1	25	19_H	ICR09	0398_H	$000FFF98_H$
IRQ10	Real timing Generator ch2	26	$1A_H$	ICR10	0394_H	$000FFF94_H$
IRQ11	10-bit A/D Converter (Hard)	27	$1B_H$	ICR11	0390_H	$000FFF90_H$
IRQ12	8-/16-bit Timer/Counter(Timer5 to 0)	28	$1C_H$	ICR12	$038C_H$	$000FFF8C_H$
IRQ13	8-/16-bit Timer/Counter(Timer5 to 1)	29	$1D_H$	ICR13	0388_H	$000FFF88_H$
IRQ14	16-bit Timer/Counter (Timer4)	30	$1E_H$	ICR14	0384_H	$000FFF84_H$
IRQ15	16-bit Timer (Timer0)	31	$1F_H$	ICR15	0380_H	$000FFF80_H$
IRQ16	16-bit Timer (Timer1)	32	20_H	ICR16	$037C_H$	$000FFF7C_H$
IRQ17	16-bit Timer (Timer2)	33	21_H	ICR17	0378_H	$000FFF78_H$
IRQ18	16-bit Timer (Timer3)	34	22_H	ICR18	0374_H	$000FFF74_H$
IRQ19	Serial I/O ch0 (SIO0)	35	23_H	ICR19	0370_H	$000FFF70_H$
IRQ20	Serial I/O ch1 (SIO1)	36	24_H	ICR20	$036C_H$	$000FFF6C_H$
IRQ21	Serial I/O ch2 (SIO2)	37	25_H	ICR21	0368_H	$000FFF68_H$
IRQ22	10-bit A/D Converter (Soft)	38	26_H	ICR22	0364_H	$000FFF64_H$
IRQ23	Key-Input Interrupt	39	27_H	ICR23	0360_H	$000FFF60_H$
IRQ24	Reserved	40	28_H	ICR24	$035C_H$	$000FFF5C_H$
IRQ25	Reserved	41	29_H	ICR25	0358_H	$000FFF58_H$
IRQ26	Reserved	42	$2A_H$	ICR26	0354_H	$000FFF54_H$
IRQ27	Reserved	43	$2B_H$	ICR27	0350_H	$000FFF50_H$
IRQ28	Reserved	44	$2C_H$	ICR28	$034C_H$	$000FFF4C_H$
IRQ29	Reserved	45	$2D_H$	ICR29	0348_H	$000FFF48_H$
IRQ30	Reserved	46	$2E_H$	ICR30	0344_H	$000FFF44_H$
IRQ31	Reserved	47	$2F_H$	ICR31	0340_H	$000FFF40_H$
IRQ32	Reserved	48	30_H	ICR32	$033C_H$	$000FFF3C_H$
IRQ33	Reserved	49	31_H	ICR33	0338_H	$000FFF38_H$
IRQ34	Reserved	50	32_H	ICR34	0334_H	$000FFF34_H$
IRQ35	Reserved	51	33_H	ICR35	0330_H	$000FFF30_H$
IRQ36	Reserved	52	34_H	ICR36	$032C_H$	$000FFF2C_H$
IRQ37	Reserved	53	35_H	ICR37	0328_H	$000FFF28_H$
IRQ38	Reserved	54	36_H	ICR38	0324_H	$000FFF24_H$
IRQ39	Reserved	55	37_H	ICR39	0320_H	$000FFF20_H$
IRQ40	Reserved	56	38_H	ICR40	$031C_H$	$000FFF1C_H$
IRQ41	Reserved	57	39_H	ICR41	0318_H	$000FFF18_H$
IRQ42	Reserved	58	$3A_H$	ICR42	0314_H	$000FFF14_H$
IRQ43	Reserved	59	$3B_H$	ICR43	0310_H	$000FFF10_H$
IRQ44	Reserved	60	$3C_H$	ICR44	$030C_H$	$000FFF0C_H$
IRQ45	Reserved	61	$3D_H$	ICR45	0308_H	$000FFF08_H$
IRQ46	Flash	62	$3E_H$	ICR46	0304_H	$000FFF04_H$
IRQ47	Delayed Interrupt (REALOS)	63	$3F_H$	ICR47	0300_H	$000FFF00_H$

■ Return by Standby Mode (Stop/Sleep)

The function returned from the stop mode is realized in the interrupt controller by generating an interrupt request.

Even if an interrupt request is generated from a peripheral, a return request is generated from the standby mode to the clock control section.

Operation of the priority decision section restarts after the clock is supplied upon returning from the stop status. The CPU continues executing commands until the result is output from the priority decision section.

The same operation is performed when returning from the sleep status.

Note:

Interrupt request output should be disabled from the stop and sleep mode by the supported peripheral control register in order that the interrupt factors are not return factors. As the return request signal from the standby mode is just a logical sum output of all interrupt factor, the contents of the interrupt level that has been set to the ICR are not taken into account.

■ Cancellation of Interrupt Cause

There is a limit between the command for releasing the interrupt factor and the RETI command under the interrupt routine. Refer to "3.9 EIT (Exception, Interruption, and Trap)" for details.

CHAPTER 16

10-bit A/D Converter

This chapter describes an outline of the 10-bit A/D converter, the register configuration/functions, and the 10-bit A/D converter operations.

16.1 Overview of 10-bit A/D Converter

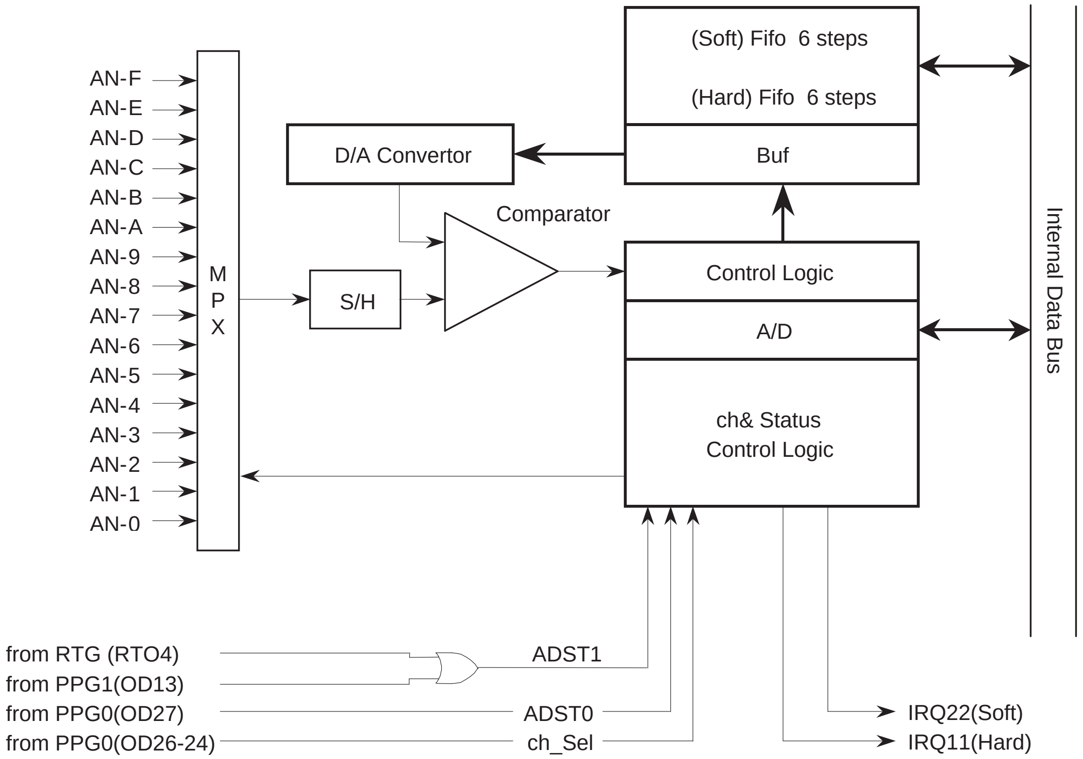
The 10-bit successive approximation type A/D converter retains the conversion initiation modes using software and hardware.

■ Feature of 10-bit A/D Converter

- Conversion time 8.4 µs (sampling 6.3 µs, conversion 2.2 µs) in fch:@20 MHz
- Corporate 6 stages of FIFO soft conversion result (4-bit + 10-bit)
- Corporate 6 stages of FIFO hard conversion result (4-bit + 10-bit)
- Channel scan function

■ Block Diagram of 10-bit A/D Converter

Figure 16.1-1 Block Diagram of 10-bit A/D Converter



■ Register List of 10-bit A/D Converter

Figure 16.1-2 Register list of 10-bit A/D Converter

Address:	bit 15 ←→ 8	7 ←→ 0	
0000A0$_H$	ADCH	ADCL	A/DC Control Register
0000A2$_H$	SCIS		Soft Conversion Analog Input Selection Register
0000A4$_H$	——	SCSR	Soft Conversion Status Register
0000A6$_H$	SCFD		Soft Conversion FIFO Data Register
0000A8$_H$	——	HCSR	Hard Conversion Status Register
0000AA$_H$	HCFD		Hard Conversion FIFO Data Register

16.2 Register of 10-bit A/D Converter

The register configuration/functions of the 10-bit A/D converter is shown.

■ A/D Converter Control Register (ADCH, ADCL)

● ADCH

Figure 16.2-1 A/D converter control register (ADCH)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000A0H	Test	Test	Test	ADMV	HCNS	HCS	SCS	SSTR	XXX0 0000B
Access →	R	R	R	R	R	R	R	R	

[bit7 to 5]:Test

It is test bit.

[bit4]:ADMV

It is flag to indicate under A/D conversion.

0	Not Under the conversion
1	Under the conversion

[bit3]:HCNS

It is hard start hold flag.

0	No conversion hold by hard start (without nest)
1	Conversion hold by hard start (with nest)

[bit2]:HCS

It is hard conversion status flag.

0	Conversion complete by hard start
1	Under the conversion by hard start

[bit1]:SCS

It is soft conversion status flag.

0	Conversion complete by soft start
1	Under the conversion by soft start

[bit0]:SSTR

It is soft conversion start bit.

0	None
1	Start/restart soft conversion (writing under conversion)

When this bit is read, "0" is always read.

● ADCL

Figure 16.2-2 A/D converter control register (ADCL)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000A1H	Test	Test	Test	−	−	Hi2	Hi1	Hi0	000- -XXXB
Access →	R/W	R/W	R	−	−	R/W	R/W	R/W	

[bit7 to 6]:Test

Please set "0".

[bit5]:Test

It is test bit.

[bit4, 3]:

There are unused bit.

[bit2 to 0]:Hi2 to 0

There are analog input selection bits for hard conversion.

Hi2	Hi1	Hi0	Selection analog input
0	0	0	AN-0
0	0	1	AN-1
0	1	0	AN-2
0	1	1	AN-3
1	0	0	AN-4
1	0	1	AN-5
1	1	0	AN-6
1	1	1	AN-7

■ Soft Conversion Analog Input Selection Register (SCIS)

Figure 16.2-3 Soft conversion analog input selection register (SCIS)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
Address: 0000A2H	iF	iE	iD	iC	iB	iA	i9	i8	i7	i6	i5	i4	i3	i2	i1	i0	0000 0000H
Access →	R/W																

[bit15 to 0]:iF to 0

There are analog input selection bits for soft conversion.

0	Input non-selection
1	Input selection

When a number of inputs are selected, all selected inputs are converted sequentially.

■ Soft Conversion Status Register (SCSR)

Figure 16.2-4 Soft conversion status register (SCSR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000A5$_H$	SCEF	SECR	SCIE	−	−	SFCR	SFUL	SEMP	X10- -001$_B$
Access →	R	W	R/W	−	−	W	R	R	

[bit7]:SCEF

It is soft conversion end flag.

0	None or Under the conversion
1	Conversion complete

[bit6]:SECR

It is SCEF clear bit.

0	Clear soft conversion end flag.
1	None

The read value of this bit is always "1".

[bit5]:SCIE

It is soft conversion interrupt enable bit.

0	Interrupt interdiction
1	Interruption permission

The interrupt request is generated when SCEF=1 is SCIE=1.

[bit4, 3]:

It is an unused bit.

[bit2]:SFCR

It is soft conversion FIFO clear bit.

0	None
1	Clear FIFO.

The read value of this bit is always "0".

[bit1]:SFUL

It is soft conversion FIFO full bit.

0	Status that data can input to FIFO
1	FIFO full

[bit0]:SEMP

It is soft conversion FIFO empty bit.

0	Status that data is retained in FIFO
1	FIFO empty

■ Soft Conversion FIFO Data Register (SCFD)

Figure 16.2-5 Soft conversion FIFO data register (SCFD)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
Address: 0000A6H	C3	C2	C1	C0	−	−	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	XXXX --XXH
Access →	R																XXXX XXXXH

It is conversion result register for soft start. Reading this register enables data to be fetched sequentially.

[bit11, 10]:

It is an unused bit.

[bit15 to 12]:C3 to 0

It is input channel of conversion result.

C3	C2	C1	C0	Channel supported by conversion result
0	0	0	0	AN-0
0	0	0	1	AN-1
0 to 1	0 to 1	1 to 1	0 to 0	AN-2 to AN-E
1	1	1	1	AN-F

[bit9 to 0]:D9 to 0

It is soft conversion data.

■ Hard Conversion Status Register (HCSR)

Figure 16.2-6 Hard Conversion Status Register (HCSR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0000A9H	HCEF	HECR	HCIE	−	−	HFCR	HFUL	HEMP	X10- -001B
Access →	R	W	R/W	−	−	W	R	R	

[bit7]:HCEF

It is hard conversion end flag.

0	None or Under the conversion
1	Conversion complete

[bit6]:HECR

It is HCEF clear bit.

0	Clear hard conversion end flag.
1	None

The read value of this bit is always "1".

[bit5]:HCIE

It is hard conversion interrupt enable bit.

0	Interruption interdiction
1	Interruption permission

The interrupt request is generated when HCEF=1 is HCIE=1.

[bit2]:HFCR

It is hard conversion FIFO clear bit.

0	None
1	Clear FIFO.

The read value of this bit is always "0".

[bit1]:HFUL

It is hard conversion FIFO full bit.

0	Status that data can input to FIFO
1	Status of FIFO full

[bit0]:HEMP

It is hard conversion FIFO empty bit.

0	Status that data is retained in FIFO
1	Status of FIFO empty

■ Hard Conversion FIFO Data Register (HCFD)

Figure 16.2-7 Hard conversion FIFO data register (HCFD)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Initial value
Address: 0000AAH	RS	C2	C1	C0	−	−	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	XXXX --XXH
Access →	R																XXXX XXXXH

It is conversion result register for hard start. Reading this register enables data to be fetched sequentially.

[bit15]:RS

It is hard conversion request status.

0	ADST0 (PPG0)
1	ADST1 (PPG1, RTG)

[bit14 to 12]:C2 to 0

It is input channel of hard conversion result.

C2	C1	C0	Channel supported by conversion result
0	0	0	AN-0
0	0	1	AN-1
0 to 1	1 to 1	0 to 0	AN-2 to AN-6
1	1	1	AN-7

[bit9 to 0]:D9 to 0

There are hard conversion data.

16.3 Operation of 10-bit A/D Converter

In terms of 10-bit A/D converter operations, A/D operation is possible using both software and hardware conversions.

■ A/D Operation by Soft Conversion

In order to carry out A/D conversion using software conversion, first select the required channel from the 16 analog input pins AN0 to ANF. Enabled by writing "1" to the bit supported by the SCIS register.

● For one channel

When only one channel is selected as the analog input pin for software conversion: Writing "1" to the SSTR bit of the ADCH register starts the software conversion, and the SCS bit of the ADCH register will be set to "1".

Writing "1" to the SSTR bit again during conversion initializes the conversion operation, and conversion restarts.

When A/D conversion ends, the SCS bit of the ADCH register is reset to "0", and the SCEF bit of the SCSR register is set to "1". Reading these status bits enables the end of the conversion to be determined. If the interrupt for conversion completion needs to be generated, the SCIS bit of the SCSR register shall be set to "1".

● For multiple channel

When a number of channels are selected as the analog input pin for software conversion, automatically checks whether or not each channel is selectable, then switches channels sequentially, starts up A/D conversion, and stores the conversion results on a FIFO basis.

For the channel to be converted, writing "1" to the bit supported by the SCIS register and writing "1" to the SSTR bit of the ADCH register starts software conversion, and the SCS bit of the ADCH register will be set to "1". The conversion channel is selected from 0 to 15 sequentially, any channels that are not selected by the SCIS register are not converted, and the next selected channel will be converted.

Writing "1" to the SSTR bit during conversion initializes the conversion operation, and conversion restarts from channel 0.

When A/D conversion of all selected channels ends, the SCS bit of the ADCH register is reset to "0", and the SCEF bit of the SCSR register is set to "1". If an interrupt for conversion completion needs to be generated, the SCIS bit of the SCSR register shall be set to "1".

The A/D converted results can be stored FIFO up to six times, and they are automatically input on FIFO basis at the end of A/D conversion. FIFO data can be retrieved by reading the SCFD register, then after reading is automatically incremented in FIFO format, and the next data is output.

However, when the SCFD register is read through byte access, it is not incremented in FIFO format. When the next conversion is carried out while FIFO basis is full (SCSR:SFUL_bit=1), the conversion results will be overwritten at the sixth FIFO column.

■ A/D Operation by Hard Conversion

A/D conversion using hardware conversion can be operated by the ADST0 (PPG0) or ADST1 (PPG1, RTG) factors, and operation starts when the ADST0 or ADST1 rising edge is detected. Both ADST0 and ADST1 can select eight analog input pins (AN0 to AN7), with ADST0 being selected by the OD26 to 24 signals from the PPG0, and ADST1 being selected by the ADCL register Hi2 to 0.

When the A/D conversion is performed, the HCS bit of the ADCH register is set to "1". When conversion ends, the HCS bit of the ADCH register is reset to "0", and the HCEF bit of the HCSR register is set to "1". If the conversion completion interrupt needs to be generated, the HCIS bit of the HCSR register shall be set to "1".

The A/D converted results can be stored FIFO up to six times, and they are automatically input on FIFO basis at the end of A/D conversion. FIFO data can be retrieved by reading the HCFD register, then after reading is automatically incremented in FIFO format, and the next data is output.

However, when the HCFD register is read through byte access, it is not incremented in FIFO format.

● Generation condition of hard start request signal

When OD027(PPG0) & EQ(PPG0)=1, the ADST0 signal generates.

When OD113 & EQ(PPG1)=1, the ADST1 signal generates.

When RTO04 & EQ0(RTG0)=1, the ADST1 signal generates.

When RTO14 & EQ1(RTG1)=1, the ADST1 signal generates.

When RTO24 & EQ2(RTG2)=1, the ADST1 signal generates.

■ Priority Order of A/D Conversion

In terms of the A/D conversion initiation factors, three types of initiation can be executed, namely initiation by software, and initiation by two types of hardware (ADST0 & ADST1), and their priority ranking is as follows.

Priority order	High	ADST0 initiation by hardware
	to	ADST1 initiation by hardware
	Low	Initiation by software

● When initiation by hardware enters at software operation

Suspend the operation by software and perform the operation by hardware. When hardware operation ends, software operation automatically restarts.

● When ADST0 initiation enters at operation by ADST1 initiation

Hold the operation by ADST1 initiation and perform the operation by ADST0 initiation. When the operation initiated by the ADST0 ends, ADST1 automatically restarts.

Note:

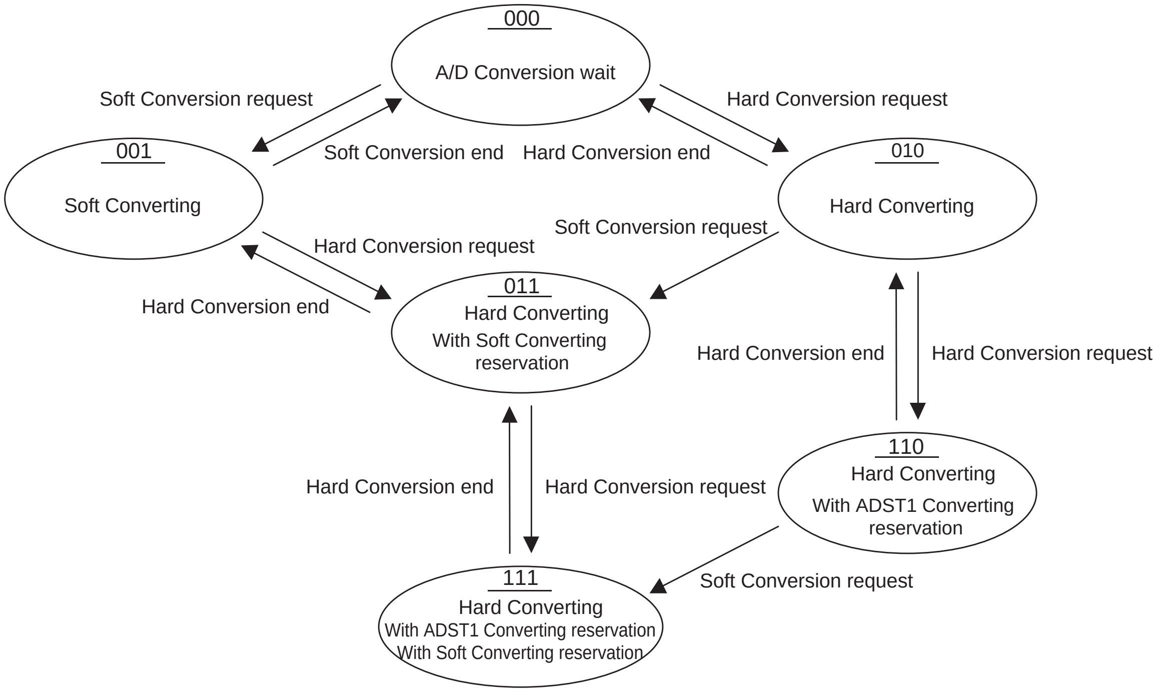
Conversion initiation requests with the same factor are masked during conversion by hardware.

16.4 State Transition of 10-bit A/D Converter

Figure 16.4-1 shows the state transition of 10-bit A/D converter.

■ State Transition of 10-bit A/D Converter

Figure 16.4-1 State transition of 10-bit A/D converter



The operation state can be known by the SCS, HCS, and HCNS bit of the ADCH register.

HCNS	HCS	SCS	Status Description
0	0	0	Analog to digital conversion wait
0	0	1	Analog to digital conversion by soft start
0	1	0	Analog to digital conversion by hard start (0 and 1)
0	1	1	Analog to digital conversion by hard start (0 and 1) Retention of soft start
1	1	0	Analog to digital conversion by hard start 0 Retention of soft start 1
1	1	1	Analog to digital conversion by hard start 0 Retention of soft start and hard start 1

CHAPTER 17

Serial I/O

This chapter describes an outline of the serial I/O, the register configuration/functions, and the serial data RAM and serial I/O operations.

17.1 Overview of Serial I/O

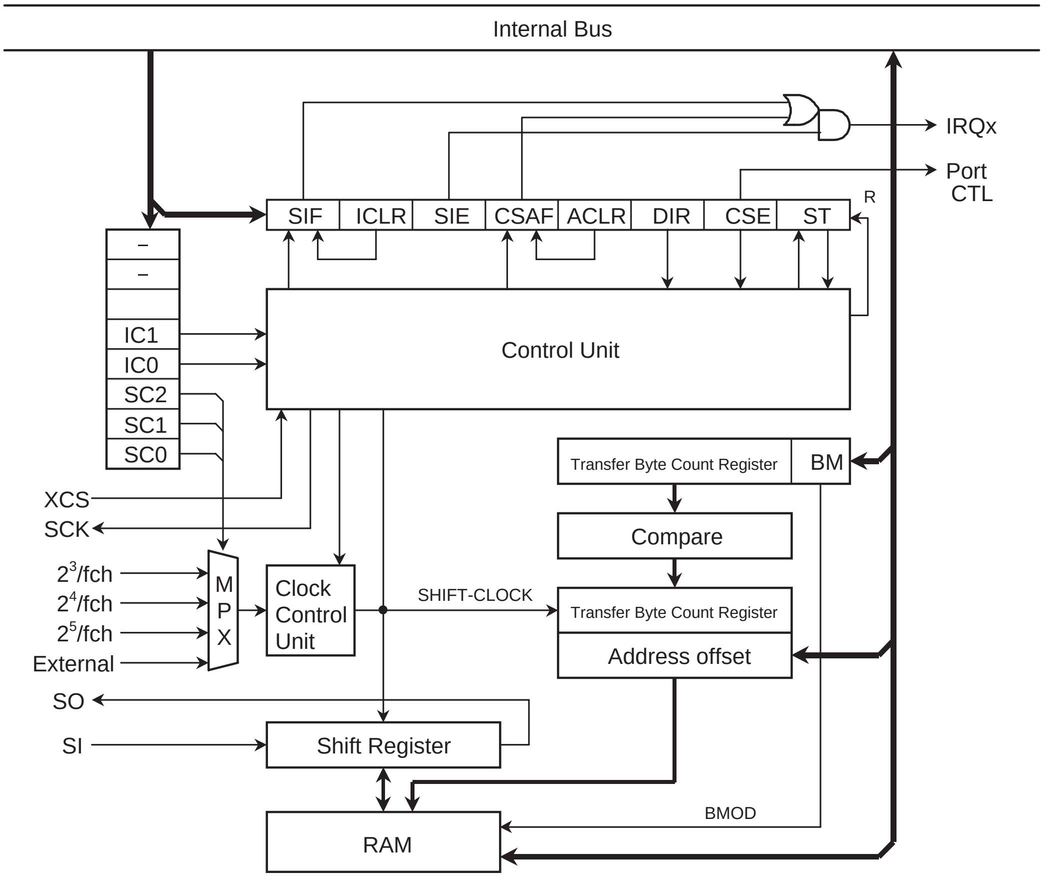
The serial I/O can automatically transfer 8-bit serial data and can select external initiation, multiple shift clock, and transfer interval time.

■ Feature of Serial I/O

- 8-bit serial data can be transferred using clock synchronization.
- Automatic transfer of up to 128 bytes
- External initiation by ship select input (XCS)
- 4-type shift clock (inner: 3, outer: 1) can be selected
- Enable/disable of transition interval time can be selected

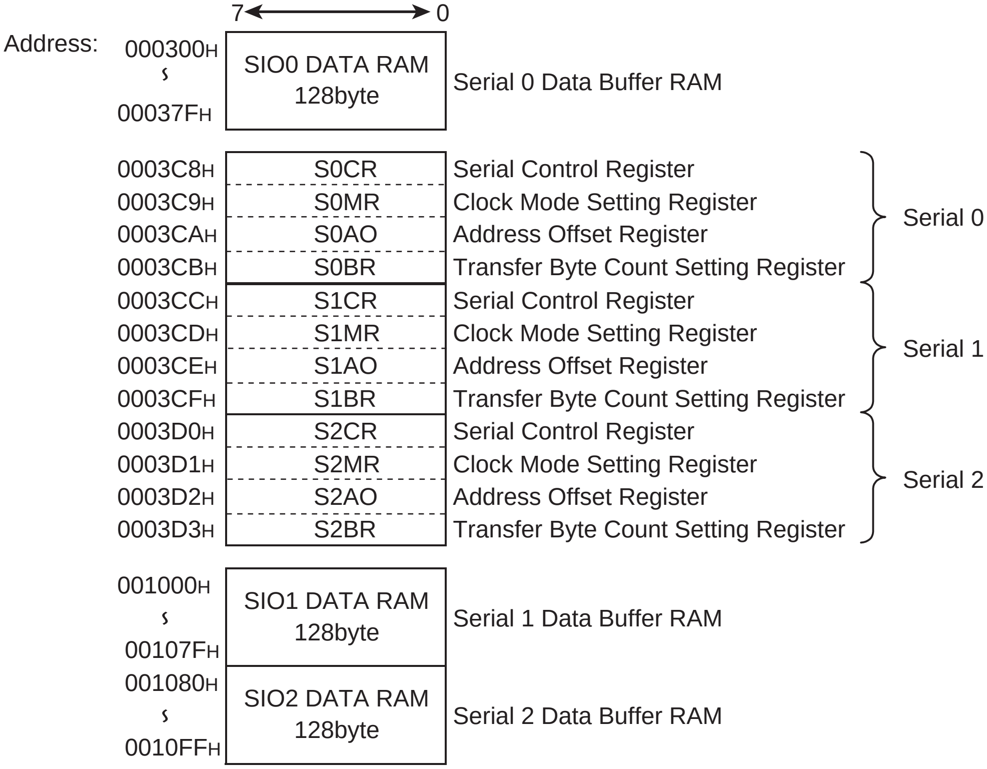
■ Block Diagram of Serial I/O

Figure 17.1-1 Block diagram of serial I/O



■ Register List of Serial I/O

Figure 17.1-2 Register list of Serial I/O



7 ⟷ 0

Address:			
000300H ~ 00037FH	SIO0 DATA RAM 128byte	Serial 0 Data Buffer RAM	
0003C8H	S0CR	Serial Control Register	Serial 0
0003C9H	S0MR	Clock Mode Setting Register	
0003CAH	S0AO	Address Offset Register	
0003CBH	S0BR	Transfer Byte Count Setting Register	
0003CCH	S1CR	Serial Control Register	Serial 1
0003CDH	S1MR	Clock Mode Setting Register	
0003CEH	S1AO	Address Offset Register	
0003CFH	S1BR	Transfer Byte Count Setting Register	
0003D0H	S2CR	Serial Control Register	Serial 2
0003D1H	S2MR	Clock Mode Setting Register	
0003D2H	S2AO	Address Offset Register	
0003D3H	S2BR	Transfer Byte Count Setting Register	
001000H ~ 00107FH	SIO1 DATA RAM 128byte	Serial 1 Data Buffer RAM	
001080H ~ 0010FFH	SIO2 DATA RAM 128byte	Serial 2 Data Buffer RAM	

17.2 Register of Serial I/O

The register configuration/functions of the serial I/O is shown.

■ Serial Control Register (SxCR)

Figure 17.2-1 Serial control register (SxCR)

bit	7	6	5	4	3	2	1	0	Initial value
	SIF	ICLR	SIE	CSAF	ACLR	CSE	DIR	ST	0100 1000B
Access→	R	W	R/W	R	W	R/W	R/W	R/W	

[bit7]:SIF

It is serial I/O transfer completion flag.

0	Serial data transfer does not complete.
1	Serial data transfer completes.

[bit6]:ICLR

It is transfer completion flag clear bit.

0	Clear transfer completion flag
1	None

The read value of this bit is always "1".

[bit5]:SIE

It is interrupt enable bit.

0	Interruption is disabled.
1	Interruption is enabled.

[bit4]:CSAF

It is forced end flag at chip select transfer.

0	Automatic transfer normally ends by chip select.
1	Forcibly ends by chip select under transfer.

[bit3]:ACLR

It is forced end flag clear bit.

0	Clear transfer completion flag.
1	None

The read value of this bit is always "1".

[bit2]:CSE

It is the chip select auto-transfer enable bit.

0	Automatic transfer is disabled.
1	Automatic transfer is enabled.

[bit1]:DIR

It is transfer direction control bit.

0	Transfer reception data to RAM per transmitting RAM data (transmission/reception mode)
1	Transmit RAM data (Transmission mode)

[bit0]:ST

It is serial transfer start bit.

0	Transfer Stop
1	Transfer start

This bit will be cleared to "0" when serial data transfer ends.

■ Clock Mode Setting Register (SxMR)

Figure 17.2-2 Clock mode setting register (SxMR)

bit	7	6	5	4	3	2	1	0	Initial value
	−	−		IC1	IC0	SC2	SC1	SC0	--0X XXXX$_B$
Access→	−	−	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7, 6]:

There are unused bits.

[bit5]:

It is undefined bit. (Reserved)

[bit4]:IC1

[bit3]:IC0

It is selection bit of interval. Setting the interval is as follows.

IC1	IC0	Interval: T int
0	0	Tclk × 16 µs
0	1	Tclk × 32 µs
1	0	Tclk × 64 µs
1	1	No interval time (selection is disabled at external clock)

*:Tclk= Shift clock cycle time

[bit2]:SC2

[bit1]:SC1

[bit0]:SC0

It is selection bit of shift clock.

SC2	SC1	SC0	Selection clock	Shift clock cycle time: Tclk	Cycle time (fch=@20MHz)
0	0	0	Specification prohibited	-	-
0	0	1	Internal clock	$2^3 \times \phi$	0.4 µs (1:1)
0	1	0		$2^4 \times \phi$	0.8 µs (1:1)
0	1	1		$2^5 \times \phi$	1.6 µs (1:1)
1	0	0	Specification prohibited	-	-
1	0	1	External clock	Min 8 x ϕ (duty 50%)	0.4 µs (ϕ:50 ns)
1	1	0	Specification prohibited	-	-
1	1	1		-	-

ϕ: Defined by peripheral clock gear (PCK1 and 0)

fch: Source oscillation frequency

Table 17.2-1 Shift clock cycle and interval at 20 MHz operation

SC2	SC1	SC0	Tclk	Tint		
				IC1, IC0=(0, 0)	IC1, IC0=(0, 1)	IC1, IC0=(1, 0)
0	0	1	0.4 µs	6.4 µs	12.8 µs	51.2 µs
0	1	0	0.8 µs	12.8 µs	25.6 µs	102.4 µs
0	1	1	1.6 µs	25.6 µs	51.2 µs	204.8 µs

■ Address Offset Register (SxAO)

Figure 17.2-3 Address offset register (SxAO)

bit	7	6	5	4	3	2	1	0	Initial value
	−	A6	A5	A4	A3	A2	A1	A0	-000 0000B
Access →	−	−	W	W	W	W	W	W	

Sets the start address offset of the serial data RAM.

■ Transfer Byte Number Setting Register (SxBR)

Figure 17.2-4 Transfer byte number setting register (SxBR)

bit	7	6	5	4	3	2	1	0	Initial value
	BM0D	BC6	BC5	BC4	BC3	BC2	BC1	BC0	0XXX XXXXB
Access →	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7]:BMOD

It is serial buffer mode control bit.

0	Buffer overwrite mode
1	Transmission/reception data buffer independent mode

[bit6 to 0]:BC6 to 0

It is transfer byte number setting bit. The transfer byte number becomes the set value +1. Up to 128 bytes can be transferred. However, when transmission/reception data buffer independent mode is specified by the BMOD bit, specification of BC6 will be invalid, and a maximum of 64 bytes will be transferred.

17.3 Serial Data RAM

The serial data RAM has 128 bytes per channel. In this section, explanations are given based on serial 0.

■ Serial Data RAM

For serial 0 of this RAM, the byte number set by the transfer byte number setting register from the address "300_H" is used as the serial data buffer.

When BMOD=0, the transfer address is 1st byte: "300_H"; 2nd byte: "301_H". When BMOD=1, the transfer address is 1st byte: transmission data "300_H", reception data "340_H"; 2nd byte: transmission data "301_H", reception data "341_H". In both cases, any area exceeding the byte number that has been set can be used as data RAM. If other than 00_H is set as the offset address, the value whereby the set value is added to the address of 300_H will be the 1st byte address.

Figure 17.3-1 Data RAM and transfer order (BMOD=0)

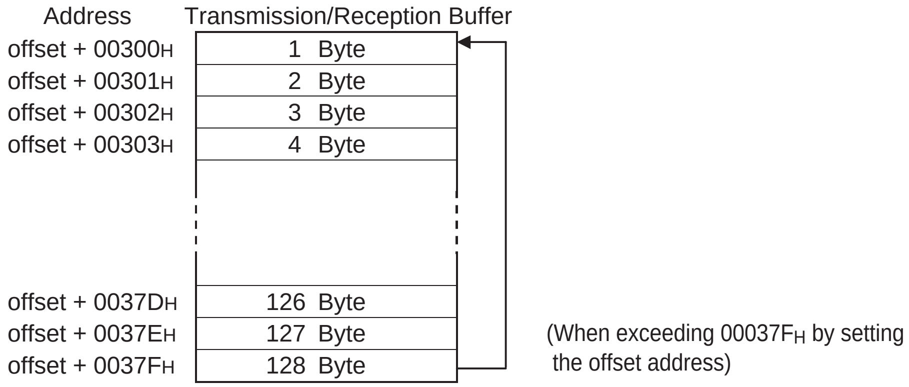
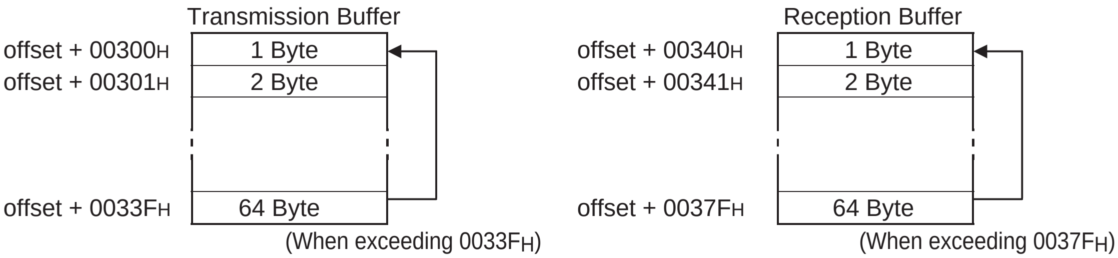


Figure 17.3-2 Data RAM and transfer order (BMOD=1)



17.4 Operation of Serial I/O

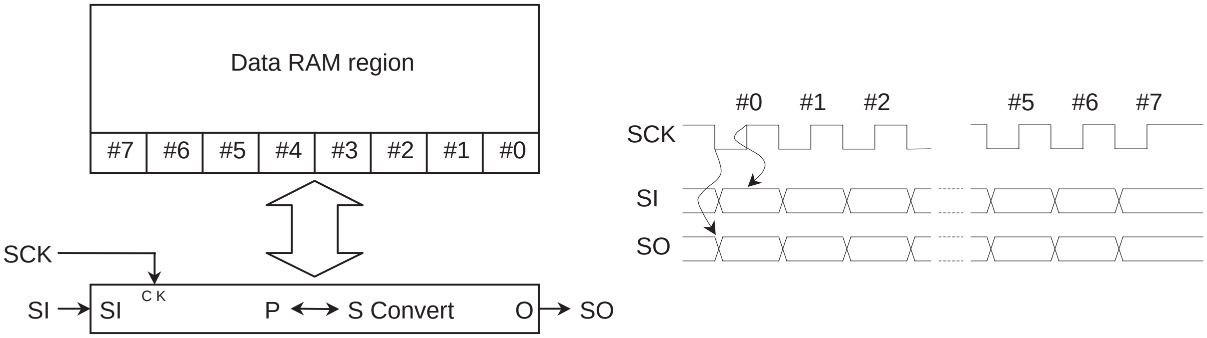
The serial I/O has two modes, namely, transmission mode and transmission/reception mode.

■ Operation of Serial I/O

The serial I/O has two modes, namely, transmission mode and transmission/reception mode. In the case of transmission mode (DIR=1), contents transferred to the P.S conversion buffer from the data RAM area are synchronized to the falling of the external clock or serial shift clock (SCK) generated internally and output to the serial output pin (SO) bit serially. In the case of transmission reception mode (DIR=0), contents transferred to the P.S conversion buffer from the data RAM area are synchronized to the falling of the external clock or serial shift clock (SCK) generated internally, output to the serial output pin (SO) bit serially, also retrieved into the P.S conversion buffer at the rising of the serial shift clock (SCK) bit serially from the serial input pin (SI), and transferred to the data RAM when 8-bit transfer ends.

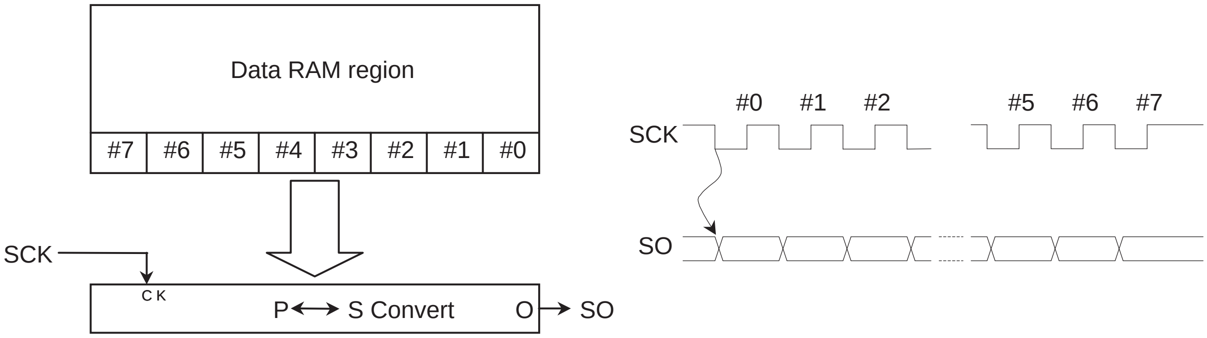
- ● Transmission/reception mode

Figure 17.4-1 Serial transfer mode at transmission/reception mode



- ● Transmission mode

Figure 17.4-2 Serial transfer mode at transmission mode



■ Operation Mode of Serial I/O

In terms of the serial I/O operation mode, there are two types, namely, internal shift clock mode and external shift clock mode depending on the shift clock type, and these are specified by the SCMR. Mode switching and clock selection should be executed while the serial I/O is stopped (ST bit of SCR = 0).

● Internal shift clock mode

Operated by the internal clock, and outputs the shift clock with 50% duty from the SCK pin as the synchronous timing output. In terms of data, 1-bit transfers are carried out per clock, and data for the number of bytes that have been set for the SCBR is transferred sequentially at set intervals.

Figure 17.4-3 Interval mode

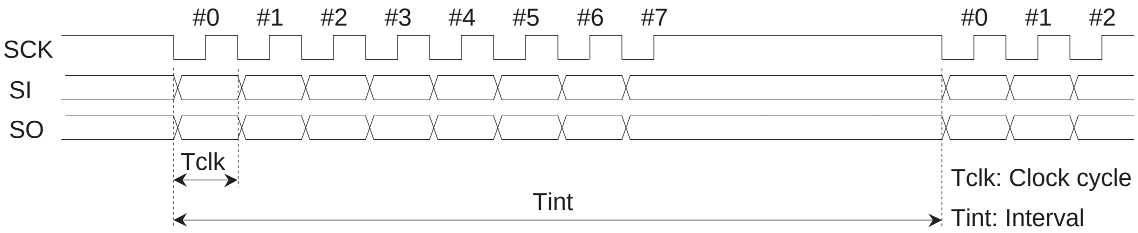
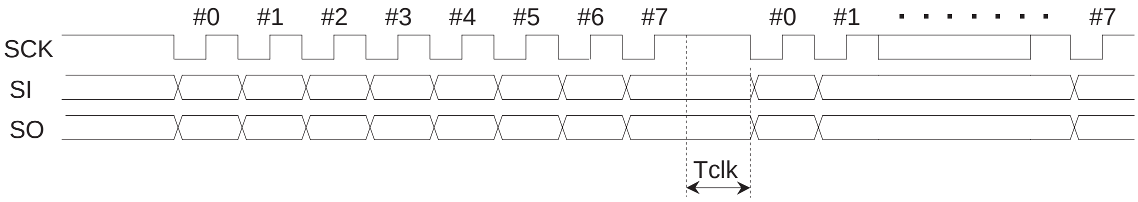


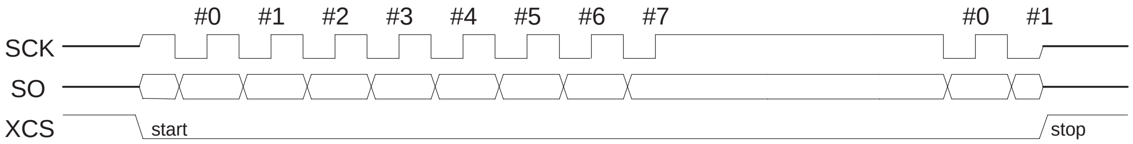
Figure 17.4-4 No interval mode



[Automatic transfer mode by chip select]

Setting "1" to chip select enable (CSE) bit enables control for the starting and stopping of transfers by the chip select input (XCS) pin. When serial transfer starts at the falling of the XCS pin, and the XCS pin rises before transfer for the number of bytes set has ended, serial transfer will be forcibly terminated. When the XCS pin is "H" level, both the SCK and SO pins will have high impedance.

Figure 17.4-5 Chip select transfer mode



● External shift clock mode

Transfers 1-bit data per clock in synchrony with the external shift clock input from the SCK pin. Select other than No Interval (IC1, 0=1, 1) as the interval specification in this case. When No Interval is selected, communication cannot be carried out correctly.

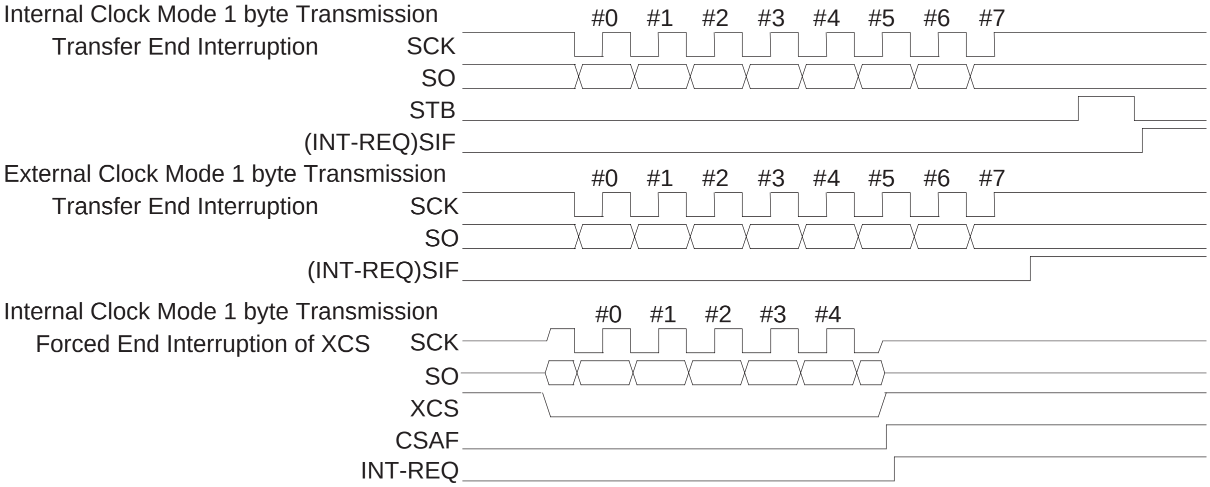
Note:

Do not write SCR, SCMR, or SCBR during serial I/O operation under either mode.

■ Interrupt Function

The serial I/O is set by the transfer end flag (SIF) bit when data transfer ends. In this case, the interrupt request is generated. When termination (suspension) is executed by inputting chip select, the chip select termination flag (CSAF) bit is set and an interrupt request is generated.

Figure 17.4-6 Interrupt generation timing



Internal Clock Mode 1 byte Transmission
Transfer End Interruption
#0 #1 #2 #3 #4 #5 #6 #7
SCK
SO
STB
(INT-REQ)SIF
External Clock Mode 1 byte Transmission
Transfer End Interruption
#0 #1 #2 #3 #4 #5 #6 #7
SCK
SO
(INT-REQ)SIF
Internal Clock Mode 1 byte Transmission
Forced End Interruption of XCS
#0 #1 #2 #3 #4
SCK
SO
XCS
CSAF
INT-REQ

■ Start/Stop Timing of Shift Operation

Writing "1" to the ST bit of the SCR initiates transfer, whereas writing "0" stops transfer. (When XCS input is enabled, transfer does not start until the falling of the XCS pin is detected after writing "1" to the ST bit.) When data transfer ends, clears the ST bit automatically to "0" and stops operation.

● Internal shift clock mode

Figure 17.4-7 Shift operation start/stop timing

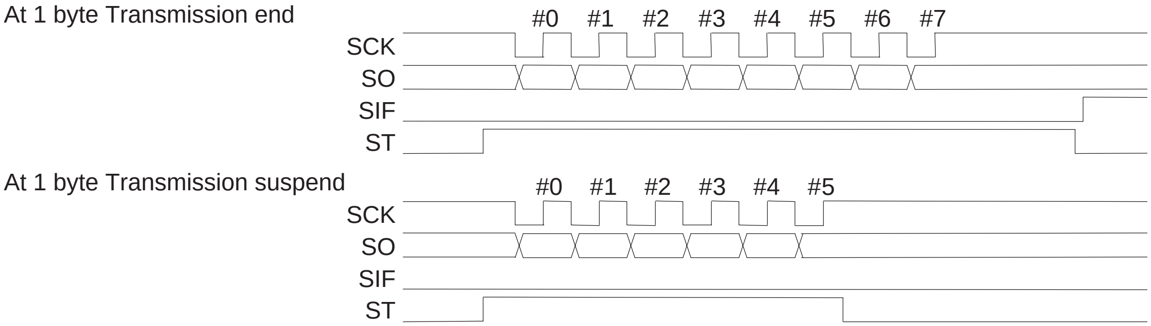
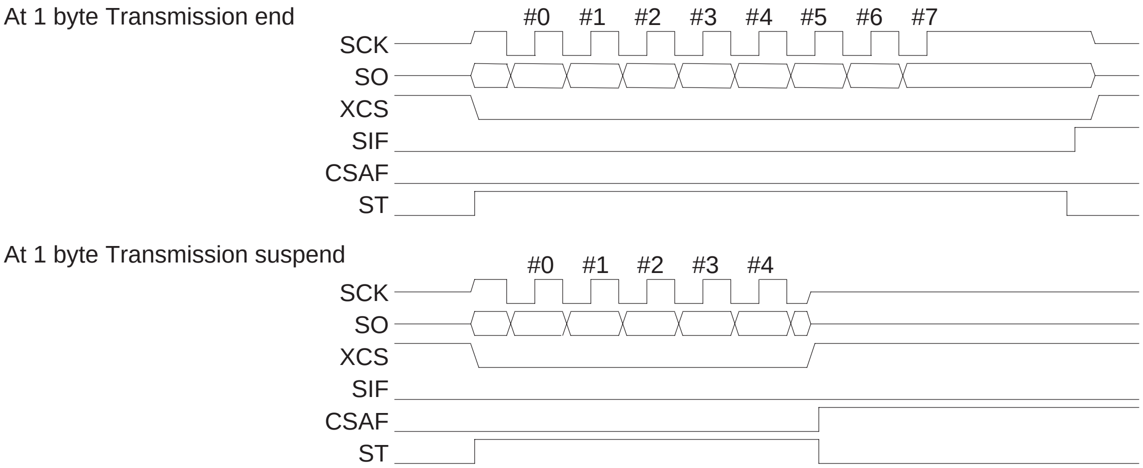
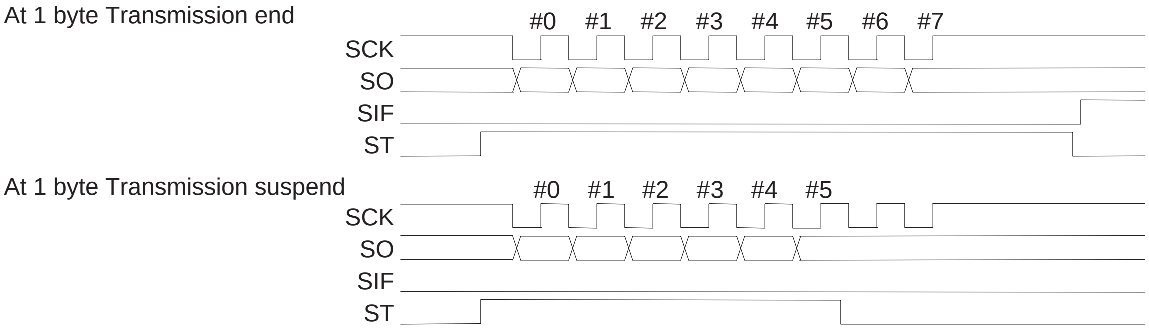


Figure 17.4-8 Shift operation start/stop timing by chip select



- External shift clock mode

Figure 17.4-9 Shift operation start/stop timing by external clock



CHAPTER 18
10-bit General-purpose Prescaler

This chapter describes an outline of the 10-bit general-purpose prescaler, the register configuration/functions, and the 10-bit general-purpose prescaler operations.

18.1 Overview of 10-bit General-purpose Prescaler

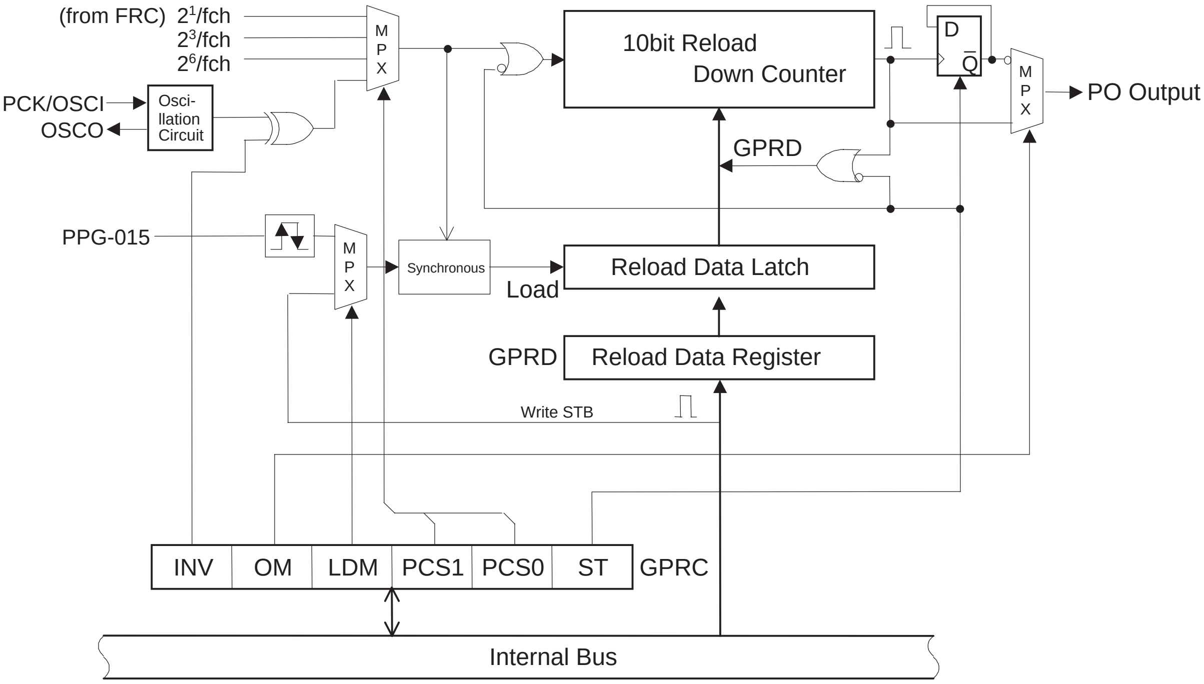
The 10-bit general-purpose prescaler has built-in dedicated oscillation circuit and a load function using the PPG output.

■ Feature of 10-bit General-purpose Prescaler

- 10-bit prescaler x 1ch (with square wave and pulse output)
- Dedicated internal oscillation circuit
- Includes load function driven by PPG output

■ Block Diagram of 10-bit General-purpose Prescaler

Figure 18.1-1 Block Diagram of 10-bit General-purpose Prescaler



■ Register List of 10-bit General-purpose Prescaler

Figure 18.1-2 Register list of 10-bit General-purpose Prescaler

bit	15 ←→ 8	7 ←→ 0	
Address: 000030H	——	GPRC	Prescaler Control Register
000032H	GPRDH	GPRDL	Data Register

18.2 Register of 10-bit General-purpose Prescaler

The register configuration/functions of 10-bit general-purpose prescaler is shown.

■ Prescaler Control Register (GPRC)

Figure 18.2-1 Prescaler control register (GPRC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000031H	–	–	INV	OM	LDM	PCS1	PCS0	ST	--XX 0XX0B
Access→	–	–	R/W	R/W	R/W	R/W	R/W	R/W	

[bit7, 6]:

There are unused bits.

[bit5]:INV

It is external clock input polarity selection bit.

0	Input through
1	Input inversion

[bit4]:OM

It is output mode selection bit.

0	Output reload pulse
1	Two-division output of reload pulse

[bit3]:LDM

It is update mode selection bit of reload data.

0	Update by writing data register
1	Update by both edges of PPG0 output (PO015)

[bit2, 1]:PCS1, 0

There are input clock select bits.

PCS1	PCS0	Selection Clock	Cycle (fch:@20MHz)
0	0	2^{1}/fch (FRC0)	100 ns
0	1	2^{3}/fch (FRC2)	400 ns
1	0	2^{6}/fch (FRC5)	3.2 us
1	1	External clock (PCK/OSCI)	-

[bit0]:ST

It is operation enable bit.

0	Operation stop
1	Enabling operations

■ Data Register (GPRD)

Figure 18.2-2 Data Register (GPRDH)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000032H	–	–	–	–	–	–	D9	D8	---- --XXB
Access→	–	–	–	–	–	–	W	W	

Figure 18.2-3 Data Register (GPRDL)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 000033H	D7	D6	D5	D4	D3	D2	D1	D0	XXXX XXXXB
Access→	W	W	W	W	W	W	W	W	

The division operation set to this register is performed. When "N" is set, the clock output frequency "fn" is below.

fn =1/{$\Phi \times$ (N+1)}

Use half-word access commands to write to this register. (Byte access command cannot be written.)

18.3 Operation of 10-bit General-purpose Prescaler

Division operation of the 10-bit general-purpose prescaler and updating operation of the reload data latch are described.

■ Division Operation and PO Output

Figure 18.3-1 shows the operation of general-purpose prescaler.

Figure 18.3-1 Operation of general-purpose prescaler

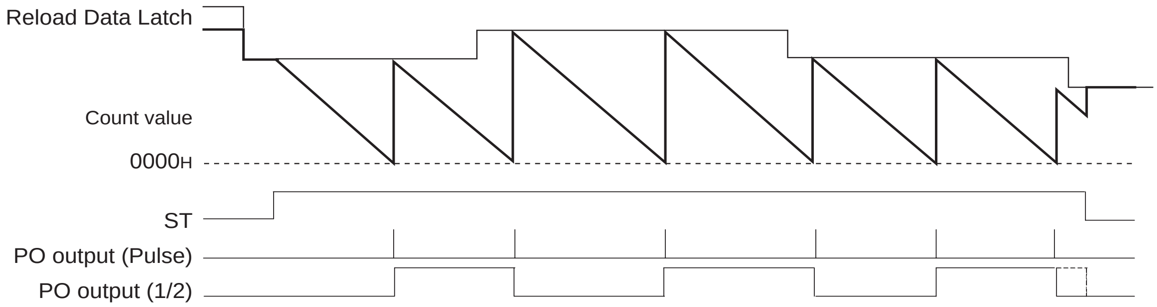
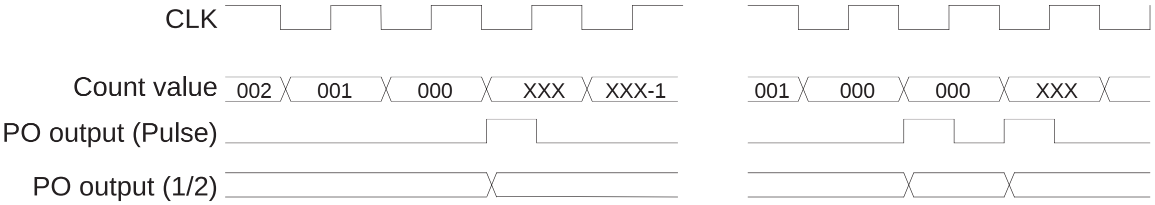


Figure 18.3-2 PO output timing



■ Update Operation of Reload Data Latch

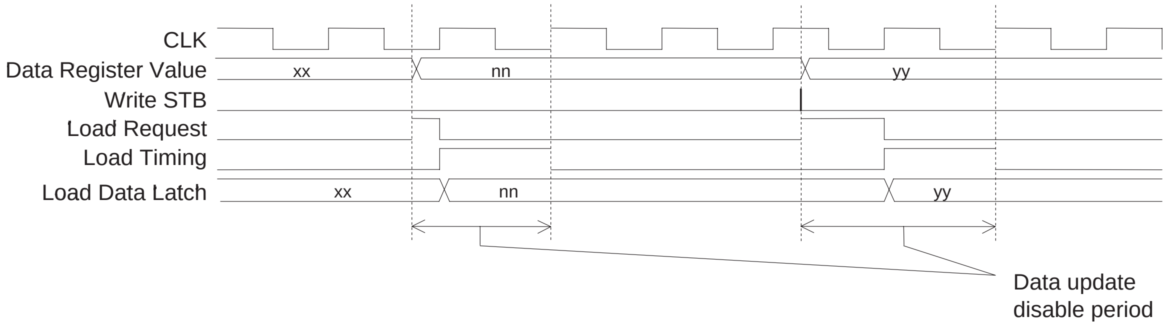
For updating operation of the reload data latch, there are two operation modes, namely updating by rewriting the data register and updating at both edges of the PPG output. Switch operation mode can be selected by the LDM bit of the GPRC.

● Update mode operation by rewriting data register

Under this mode, writing data register executes loading to the reload data latch. Newly loaded data (division value) is changed to operation with new division value as soon as the division operation using the previous value ends.

Update of the data register under this mode must be executed at an interval of "2 cycles of the clock selected by the clock selection bit + 1 machine cycle" or more.

Figure 18.3-3 Update timing of reload data latch



● Update mode operation by both edges of PPG output

Under this mode, exactly the same operation as the update mode by rewriting the data register is carried out except for detecting both edges of the PPG output and executing load requests per edge. When this mode is selected, no load request is generated by rewriting the data register.

CHAPTER 19

Bit Search Module

This chapter describes an outline of the bit search module, the register configuration/functions, the bit search module operations, and save/return processes.

19.1 Overview of Bit Search Module

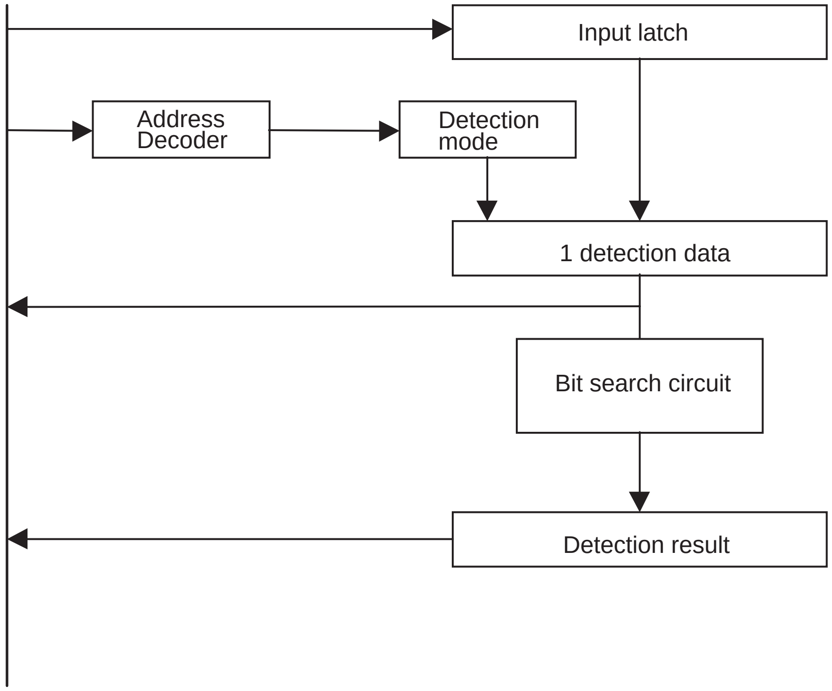
The bit search module detects 0, 1, or point of change for data written to the input register, and returns the detected bit position.

■ Feature of Bit Search Module

- Search for the bit position that first changes between 1 and 0 beginning from MSB or a word in one cycle.

■ Block Diagram of Bit Search Module

Figure 19.1-1 Block Diagram of Bit search module



■ Register List of Bit Search Module

Figure 19.1-2 Register list of Bit search module

	bit	31 ↔ 16	
Address:	000003F0H	BSD0	0 Detection Data Register
	000003F4H	BSD1	1 Detection Data Register
	000003F8H	BSDC	Change Point Detection Data Register
	000003FCH	BSRR	Detection Result Register

Note:

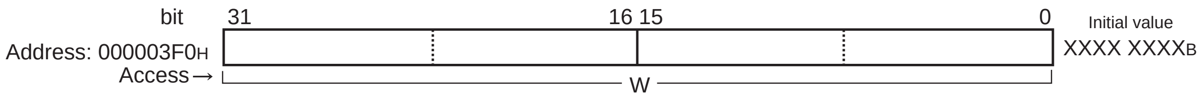
REALOS related hardware is used by the real time OS. Therefore, when REALOS is used, it cannot be used under the user program.

19.2 Register of Bit Search Module

The register configuration/functions of bit search module is shown.

■ 0 Detection Data Register (BSD0)

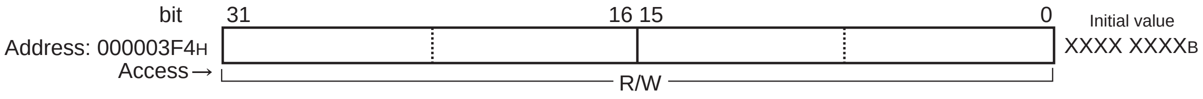
Figure 19.2-1 0 detection data register (BSD0)



- The module detects 0 for the value written to this register.
- The initial value by reset is irregular.
- The read value is indeterminate.
- Use 32-bit length data transfer command for data transfer (do not use 8-bit and 16-bit length data transfer commands).

■ 1 Detection Data Register (BSD1)

Figure 19.2-2 1 detection data register (BSD1)



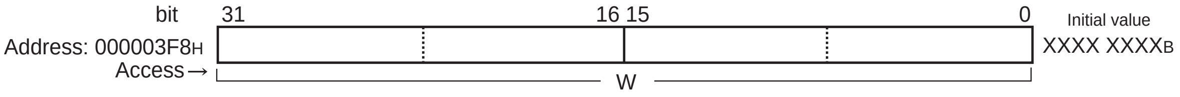
- Use 32-bit length data transfer command for data transfer (do not use 8-bit and 16-bit length data transfer commands).
- Write

The module detects 1 for the value written to this register.

- Read
  Saved data in the internal status of the bit search module is read. If the bit search module is used by the interrupt handler, etc., it is used when the original status is saved/returned. In case of 0 detection, change point detection, or even when data is written to the data register, save/return is possible by operating only data register for 1 detection.
- The initial value by reset is irregular.

■ Change Point Detection Data Register (BSDC)

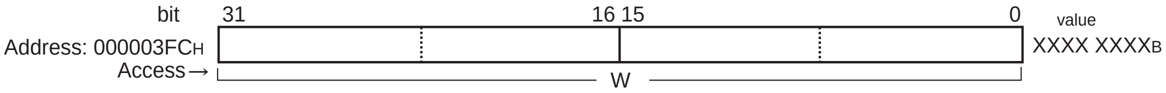
Figure 19.2-3 Change point detection data register (BSDC)



- The module detects a change point for the value written to this register.
- The initial value by reset is irregular.
- The read value is indeterminate.
- Use 32-bit length data transfer command for data transfer (do not use 8-bit and 16-bit length data transfer commands).

■ Detection Result Register (BSRR)

Figure 19.2-4 Detection result register (BSRR)



- The result of 0 detection, 1 detection or change detection is read from this register.
- Which detection results are to be read is decided by the data register that was written last.

19.3 Operation of Bit Search Module

0 detection and 1 detection by the bit search module, and detection operation are described.

■ 0 Detection

Scans data that was written to the data register for 0 detection from the MSB to LSB, and returns the position where the first "0" was detected.

The detection result can be obtained by reading the detection result register.

Relationship between the detected position and numeric value to be returned is as per Table 19.3-1 .

When "0" does not exist (in other words, when the numeric value is FFFFFFFF$_H$), a value of 32 is returned as the search result.

[Execution example]

Figure 19.3-1 Execution example of 0 detection

Program data	Read data (decimal)	
11111111111111111111000000000000$_B$	(FFFFF000$_H$)	→ 20
11111000010010011110000010101010$_B$	(F849E0AA$_H$)	→ 5
10000000000000101010101010101010$_B$	(8002AAAA$_H$)	→ 1
11111111111111111111111111111111$_B$	(FFFFFFFF$_H$)	→ 32

■ 1 Detection

Scans data that was written to the data register for 1 detection from the MSB to LSB, and returns the position where the first "1" was detected.

The detection result can be obtained by reading the detection result register.

Relationship between the detected position and value to be returned is as per Table 19.3-1 .

When "1" does not exist (in other words, when the numeric value is 00000000$_H$), a value of 32 is returned as the search result.

[Execution example]

Figure 19.3-2 Execution example of 1 detection

Program data	Read data (decimal)	
00100000000000000000000000000000$_B$	(20000000$_H$)	→ 2
00000001001000110100010101100111$_B$	(01234567$_H$)	→ 7
00000000000000111111111111111111$_B$	(0003FFFF$_H$)	→ 14
00000000000000000000000000000001$_B$	(00000001$_H$)	→ 31
00000000000000000000000000000000$_B$	(00000000$_H$)	→ 32

■ Change Point Detection

Scans data that was written to the data register for detecting the point of change from bit 30 to LSB, and compares it with the MSB value. Returns the position where the value different from the MSB was first detected.

The detection result can be obtained by reading the detection result register.

The detected position and value to be returned are as per Table 19.3-1 .

When the change point does not exist, value of 32 is returned.

In the change point detection, 0 is not returned as the detection result.

[Execution example]

Figure 19.3-3 Execution example of change point detection

```
            Program data                         Read data (decimal)

00100000000000000000000000000000H       (20000000B)   → 2
00000001001000110100010101100111H       (01234567B)   → 7
00000000000000111111111111111111H       (0003FFFFB)   → 1 4
00000000000000000000000000000001H       (00000001B)   → 3 1
00000000000000000000000000000000H       (00000000B)   → 3 2
11111111111111111111000000000000H       (FFFFF000B)   → 2 0
11111000010010011110000010101010H       (F849E0AAB)   → 5
10000000000000101010101010101010H       (8002AAAAB)   → 1
11111111111111111111111111111111H       (FFFFFFFFB)   → 3 2
```

Table 19.3-1 Bit position and value to be returned (decimal)

Detected bit position	Returned value	Detected bit position	Returned value	Detected bit position	Returned value	Detected bit position	Returned value
31	0	23	8	15	16	7	24
30	1	22	9	14	17	6	25
29	2	21	10	13	18	5	26
28	3	20	11	12	19	4	27
27	4	19	12	11	20	3	28
26	5	18	13	10	21	2	29
25	6	17	14	9	22	1	30
24	7	16	15	8	23	0	31
-						not exist	32

■ Save/Return Processes

When the internal status of the bit search module needs to be saved/returned, for example using the bit search module during interrupt handling, follow the procedure below.

1. Read the 1 detection data register and store the read data (Save).
2. Use the bit search module.
3. Write the data saved in step 1) to the 1 detection data register (Return).

As per the above operation, the value acquired when reading the detection result register for the next time is in accordance with the contents written to the bit search module before 1). Even if the data register that was written last is for 0 detection or change point detection, correctly returns by following the above procedure.

CHAPTER 20

Wait Controller

This chapter describes an outline of the wait control section, and the register configuration/functions.

20.1 Outline of Wait Control Section

The wait control section sets the access speed (wait number) for built-in memory.

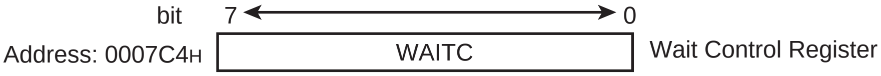
■ Internal Memory Area

The internal memory area is as follows.

- MB91191 series: address:0xC0000 to 0xfffff
- MB91192 series: address:0x80000 to 0xfffff (However, access is disabled 0x80800 to 0x9ffff.)

■ Register List of Wait Controller

Figure 20.1-1 Register list of Wait controller



20.2 Wait Control Register (WAITC)

The configuration/functions of the wait control register are shown.

■ Wait Control Register (WAITC)

Figure 20.2-1 Wait Control Register (WAITC)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 0007C4H	–	–	–	–	–	–	WTC1	WTC0	---- -000B
Access→	–	–	–	–	–	W	R/W	R/W	

[bit7 to 3]:

There are unused bits.

[bit2]:

Be sure to write "0".

[bit1, 0]:WTC1, 0

There are specification bits of wait count.

WTC1	WTC0	Wait count	at φ=fch/2	at φ=fch
0	0	0Wait(2cycle)	200 ns	Combination setting disable
0	1	1Wait(3cycle)	300 ns	150 ns
1	0	Reserved	-	-
1	1	Reserved	-	-

CHAPTER 21

Flash Memory

This chapter describes an outline of the flash memory, the register configuration/functions and the flash memory operations.

21.1 Overview of Flash Memory

The MB91F191A and MB91F192 devices have built-in flash memory that can erase all sectors in a block and erase per sector using a single +3V power source, and write per half-word (16-bit) using a FR-CPU with 254-Kbyte[*1] (2-Mbit) / 384-Kbyte (3-Mbit) capacities respectively.

■ Overview of Flash Memory

The function of this flash memory is the same as our 2Mbit (256K × 8 / 128K × 16) flash memory MBM29LV200T (excluding some sector configuration), and writing from outside the device using a[*2], ROM writer is also possible.

In addition to the MBM29LV200T equivalent function, if it is used as built-in ROM of the FR-CPU, command/data reading per word (32-bit) is possible, and high-speed operation can be realized.

It is built-in flash memory of 3V operation 254 KByte [*1] (MB91F191A)/384 KByte (MB91F192). Combining a flash memory macro and FR-CPU interface circuit realizes the following functions.

- CPU program/data storage memory function
  - When this flash memory is used as ROM, 32-bit bus width access is enabled.
  - Enables reading, writing, and erasing (auto program algorithm[*3]) using the CPU
- Functions equivalent to those of the MB29LV200T (a flash memory product)
  - Enables reading, writing, and erasing (auto program algorithm[*3]) using the ROM writer

This chapter describes use of this flash memory from the FR-CPU. Refer to the separate manual for the ROM writer User's Guide for details when using this flash memory from the ROM writer.

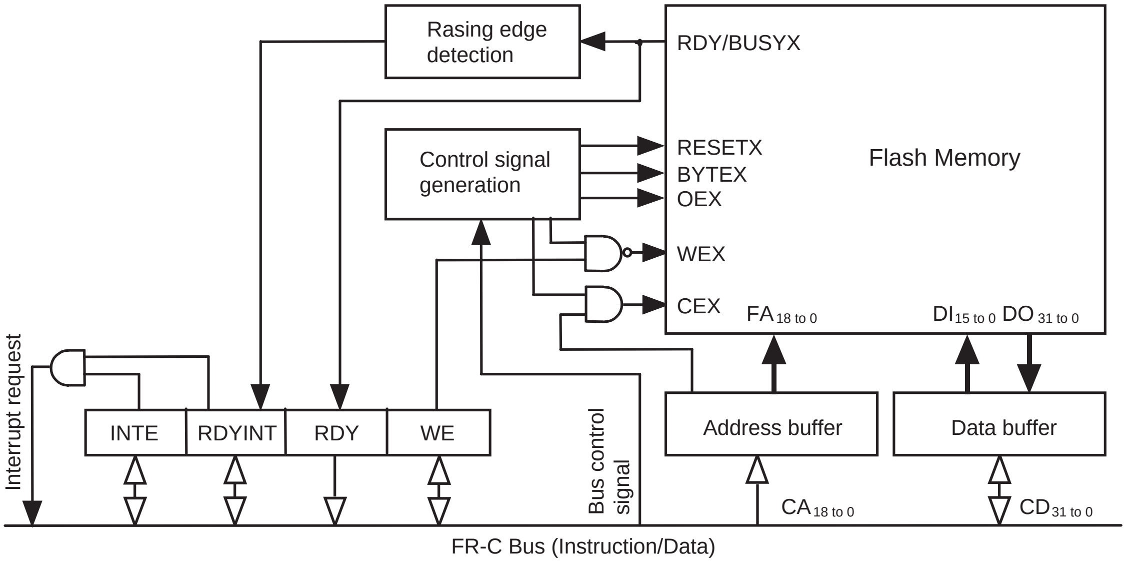
[*1]: The MBM29LV200T has 256 Kbytes, but a 2-Kbyte area overlaps with built-in RAM for the MB91F191A, so use is impossible.

[*2]: The memory capacity of the MB91F192 is 384 Kbytes, but it is extended to be based on the MBM29LV200T (256 Kbytes), so the function is the same.

[*3]: Automatic program algorithm=Embedded Algorithm TM
Embedded Algorithm TM is a registered trademark of Advanced Micro Devices, Inc.

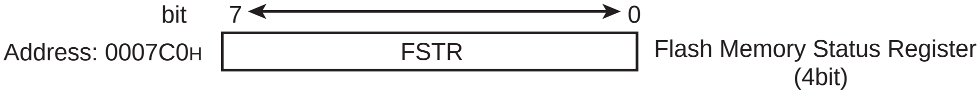
■ Block Diagram of Flash Memory

Figure 21.1-1 Block Diagram of Flash memory



■ Register List of Flash Memory

Figure 21.1-2 Register list of Flash memory



■ Memory Map and Sector Construction

Address mapping of the flash memory differs when accessed from the FR-CPU and by the ROM writer*1. This shows the mapping at accessing from the CPU.

Figure 21.1-3 Memory map and sector construction (MB91F191A)

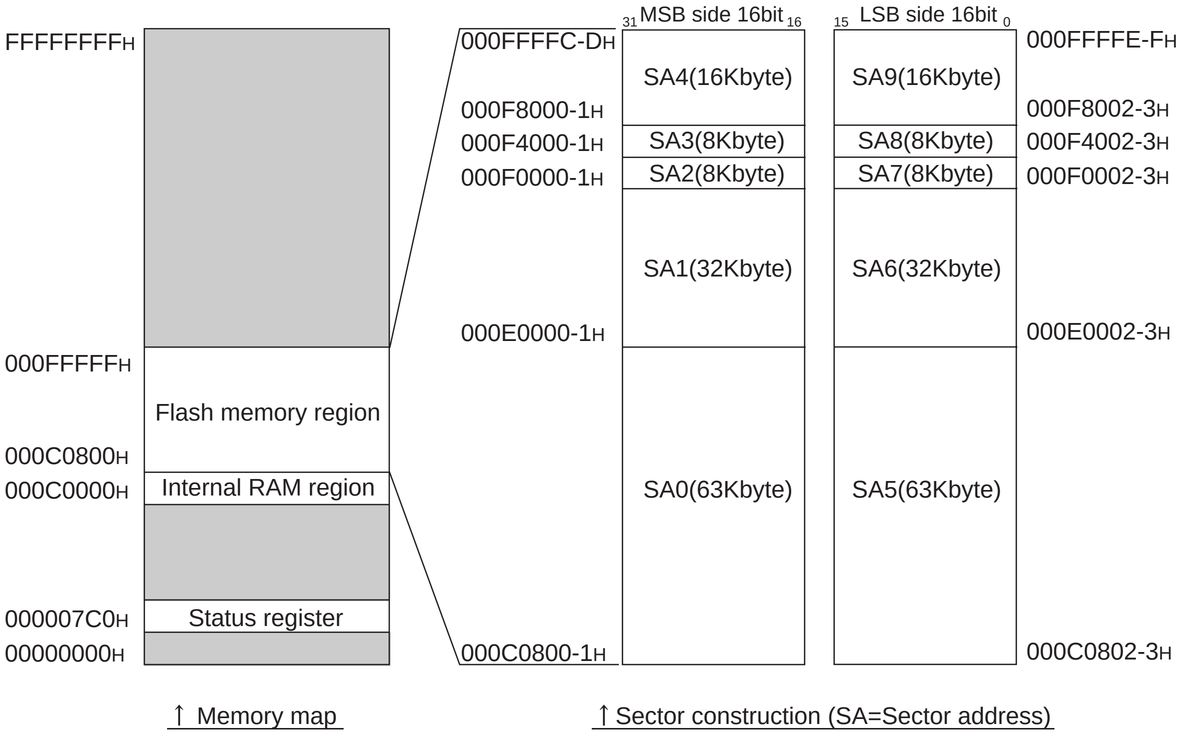


Table 21.1-1 Sector address table (MB91F191A)

Sector address	Address range	Corresponding bit position	Sector capacity
SA0	000A0000-1h to 000BFFFC-Dh (MSB side 16-bit)	Bit31 to 16	64 Kbytes
SA1	000C0000-1h to 000DFFFC-Dh (MSB side 16-bit)	Bit31 to 16	64 Kbytes
SA2	000E0000-1h to 000EFFFC-Dh (MSB side 16-bit)	Bit31 to 16	32 Kbytes
SA3	000F0000-1h to 000F3FFC-Dh (MSB side 16-bit)	Bit31 to 16	8 Kbytes
SA4	000F4000-1h to 000F7FFC-Dh (MSB side 16-bit)	Bit31 to 16	8 Kbytes
SA5	000F8000-1h to 000FFFFC-Dh (MSB side 16-bit)	Bit31 to 16	16 Kbytes
SA6	000A0002-3h to 000BFFFE-Fh (LSB side 16-bit)	Bit15 to 00	64 Kbytes
SA7	000C0002-3h to 000DFFFE-Fh (LSB side 16-bit)	Bit15 to 00	64 Kbytes
SA8	000E0002-3h to 000EFFFE-Fh (LSB side 16-bit)	Bit15 to 00	32 Kbytes
SA9	000F0002-3h to 000F3FFE-Fh (LSB side 16-bit)	Bit15 to 00	8 Kbytes

Figure 21.1-4 Memory Map and Sector construction (MB91F192)

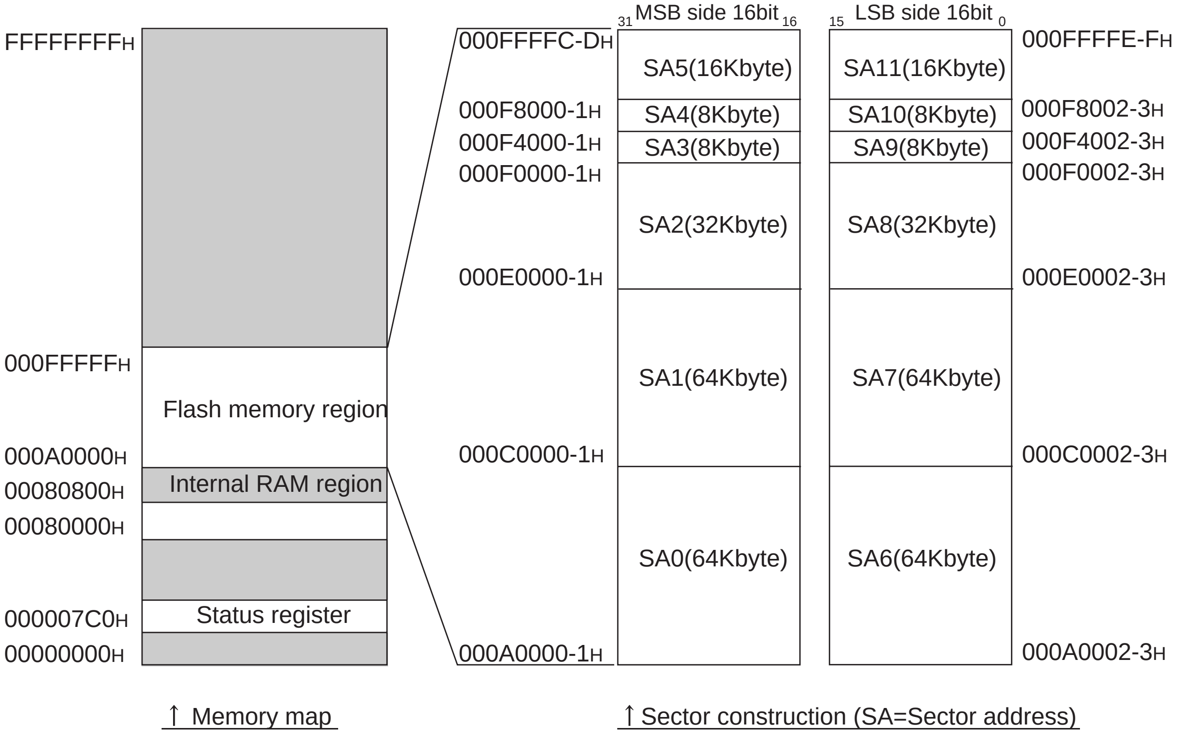


Table 21.1-2 Sector address table (MB91F192)

Sector address	Address range	Corresponding bit position	Sector capacity
SA0	000A0000-1h to 000BFFFC-Dh (MSB side 16-bit)	Bit31 to 16	64 Kbytes
SA1	000C0000-1h to 000DFFFC-Dh (MSB side 16-bit)	Bit31 to 16	64 Kbytes
SA2	000E0000-1h to 000EFFFC-Dh (MSB side 16-bit)	Bit31 to 16	32 Kbytes
SA3	000F0000-1h to 000F3FFC-Dh (MSB side 16-bit)	Bit31 to 16	8 Kbytes
SA4	000F4000-1h to 000F7FFC-Dh (MSB side 16-bit)	Bit31 to 16	8 Kbytes
SA5	000F8000-1h to 000FFFFC-Dh (MSB side 16-bit)	Bit31 to 16	16 Kbytes
SA6	000A0002-3h to 000BFFFE-Fh (LSB side 16-bit)	Bit15 to 00	64 Kbytes
SA7	000C0002-3h to 000DFFFE-Fh (LSB side 16-bit)	Bit15 to 00	64 Kbytes
SA8	000E0002-3h to 000EFFFE-Fh (LSB side 16-bit)	Bit15 to 00	32 Kbytes
SA9	000F0002-3h to 000F3FFE-Fh (LSB side 16-bit)	Bit15 to 00	8 Kbytes
SA10	000F4002-3h to 000F7FFE-Fh (LSB side 16-bit)	Bit15 to 00	8 Kbytes
SA11	000F8002-3h to 000FFFFE-Fh (LSB side 16-bit)	Bit15 to 00	16 Kbytes

*1: The mounted flash memory is little endian, but it is converted to big endian by the FR-CPU interface circuit. This conversion function does not work during access by the ROM writer, so mapping is different from CPU mode.

21.2 Flash Memory Status Register (FSTR)

The flash memory status register (FSTR) indicates the flash memory operation status.

■ Flash Memory Status Register (FSTR)

Figure 21.2-1 Flash memory status register (FSTR)

bit	7	6	5	4	3	2	1	0	Initial value
Address: 007C0H	INTE	RDYINT	WE	RDY	–	–	–	–	000- ---0B
Access→	R/W	R/W	R/W	R	–	–	–	R/W	

It is a register that indicates the operation status of flash memory.

Controls interrupts to the CPU and writing to the flash memory.

It can be accessed only by CPU. It cannot be accessed at mounting the writer.

Do not use the read modify write command to access this register.

[bit 7]: INTE

Controls generation of interrupts when auto algorithm (write, erase, etc.) of the flash memory ends.

0	Disables auto algorithm termination interrupt output. (Initial value)
1	Enables auto algorithm termination interrupt output.

Be initialized to "0" when resetting.

This bit can be read and written.

[bit 6]: RDYINT

It will be "1" when the auto algorithm (write, erase, etc.) of the flash memory ends.

The auto algorithm termination interrupt request is generated if this bit becomes "1" when interrupt generation is enabled by specifying bit7:INTE= "1".

Be initialized to "0" when resetting.

This bit can be read and written. However, "0" only is possible for writing, and even if "1" is written, the bit value will not be changed.

[bit 5]: WE

Controls the writing data and command to the flash memory under the CPU mode.

While this bit is "0", writing data and command to the flash memory will be invalid. Reading data from the flash memory will be 32-bit access.

While this bit is "1", writing data and command to the flash memory will be valid, and initiation of the auto algorithm is enabled. However, reading data from the flash memory will be 16-bit access, and 32-bit access is impossible. So during this period, it cannot be used as the program memory.

This bit must be re-written after confirming the auto algorithm (write/erase) is stopped by the RDY bit. While the RDY bit is "0", the value of this bit cannot be re-written.

0	Disables writing to the flash memory and enables 32-bit reading [Initial value]
1	Enables writing to the flash memory and disables 32-bit reading (programming mode)

Be initialized to "0" when resetting.

This bit can be read and written.

[bit 4]:RDY

This bit indicates the operation status of the auto algorithm (write/erase).

While this bit is "0", writing or erasing is carried out by the auto algorithm, and new write or erase commands cannot be received. Data cannot be read from the flash memory address. The read data indicates the flash memory status.

Refer to "21.5 Auto Algorithm Execute State" for details.

0	During writing/erasing, commands that read, write, and erase data cannot be received.
1	Commands that read, write, and erase data can be received.

This bit is not initialized at reset. (It conforms the status of flash memory at that time.)

Read is only possible. Writing does not affect the value of this bit.

[bit 3 to 1]:(reserved bit)

There are reserved bit. The read value is irregular, and write operation does not affect.

[bit 0]:(reserved bit)

It is reserved bit. These bits indicate "0" when read. Always write "0" to this bit. Writing "1" does not guarantee.

Be initialized to "0" when resetting.

Always write "0". These bits indicate "0" when read.

21.3 Operation of Flash Memory

When accessed by the FR-CPU, the following two types of access mode exist.

- **ROM mode: Word (32-bit) length data can be read in blocks, but not written.**
- **Programming mode: Word (32-bit) length access is disabled, but writing in half-word (16-bit) lengths is enabled.**

■ FR-CPU ROM Mode (32-bit, Read Only)

This mode functions as the internal ROM of FR-CPU. Enables word (32-bit) length data to be read in blocks, but cannot write to the flash memory or initiate auto algorithm.

● Mode specification method

- This mode is on when the "WE" bit of the flash memory status register is "0".
- When the CPU is in operation, it always enters this mode after the reset has been released.
- When the CPU is not in operation, it cannot enter this mode.

● Operation content

- When reading the flash memory area, word (32-bit) length data is read from the memory in blocks.
- The number of cycles required to read data is 2 cycles per word (1 wait). Using this enables commands to be supplied to the FR-CPU without waiting.

● Restrictions

- The address allocation method and endian type differ from writing the ROM writer.
- Under this mode, neither command nor data can be written to the flash memory.
- When setting the gear cycle of the CPU system to source oscillation x1, 1 Wait must be specified by the Wait control section before setting.

■ FR-CPU Programming Mode (16-bit, Read/Write)

This mode is enabled erasing/writing data. Access of word (32-bit) length data in blocks is impossible, so while operation is carried out under this mode, program execution on the flash memory is impossible.

● Mode specification method

- This mode is on when the "WE" bit of the flash memory status register is "1".
- When the CPU is in operation, the "WE" bit becomes "0" after the reset has been released. Write "1" to use this mode. Returns to ROM mode when the "WE" bit becomes "0" by rewriting "0" or generating a reset.
- The "WE" bit cannot be rewritten while the "RDY" bit of the flash memory status register is "0". Rewrite the "WE" bit after confirming that the "RDY" bit has changed to "1".

● Operation content

- When reading the flash memory area, half-word (16-bit) length data is read from the memory in blocks.

The number of cycles taken for reading is 2 cycles per half-word (1 wait).

- Writing command to the flash memory enables the auto algorithm to be initiated. Initiating auto algorithm enables the flash memory to be erased or written. Refer to "21.4 Flash Memory Auto Algorithm (Embedded Algorithm TM)" for details of the auto algorithm.

● Restrictions

- The address allocation method and endian type differ from writing the ROM writer.
- Under this mode, reading data of word (32-bit) length is disabled.
- When setting the gear cycle of the CPU system to source oscillation x 1, 1 Wait must be specified by the Wait control area before setting.

■ Auto Algorithm Execute State

When auto algorithm is initiated under CPU programming mode, the auto algorithm operation status can be identified by the internal ready/busy signal (RDY/BUSYX). This ready/busy signal level can be read as the "RDY" bit of the flash memory status register.

While the "RDY" bit is "0", writing or erasing is carried out by the auto algorithm, and no new write or erase commands can be received. Also, data cannot be read form the flash memory address.

The data read while the "RDY" bit is "0" is the hardware sequence flag indicating the flash memory status. (Refer to "21.5 Auto Algorithm Execute State ■Hardware sequence flag" for details.)

■ Interrupt Control

Interrupt request can be generated to the CPU by the exit sequence of the auto algorithm. By doing this, the end of prolonged auto algorithm sequences can be known immediately.

Auto algorithm exit interrupt is controlled by the "RDYINT" and "INTE" bit of the flash memory status register.

"RDYINT" bit is the auto algorithm termination interrupt flag. When detecting the rising edge of "0" from "1" of the internal ready/busy signal (RDY/BUSYX), sets to "1". When the "INTE" bit is "1", if the "RDYINT" bit is set, an interrupt request is output to the CPU.

Write "0" to the "RDYINT" bit or "INTE" bit to cancel the interrupt request.

21.4 Flash Memory Auto Algorithm (Embedded Algorithm ^{TM})

Writing or erasing of the flash memory cell is carried out by initiating the auto algorithm stored by the flash memory itself.

■ Command Operation

In order to initiate the auto algorithm, execute continuous writing of half-word (16-bit) data for 1 to 6 times to the flash memory. This is called a command.

If an invalid address or data is written, or the address and data are written in the wrong order, the flash memory is reset to read mode.

Table 21.4-1 lists the command.

Table 21.4-1 Command sequence table

Command sequence	Access numbers	1 Writing Cycle		2 Writing Cycle		3 Writing Cycle		4 Writing Cycle		5 Writing Cycle		6 Writing Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset	1	XXXXXXXX$_H$	F0F0$_H$	-		-		-		-		-	
Read/Reset	4	000F5556$_H$	AAAA$_H$	000EAAAA$_H$	5555$_H$	000F5556$_H$	F0F0$_H$	RA	RD	-		-	
Program	4	000F5556$_H$	AAAA$_H$	000EAAAA$_H$	5555$_H$	000F5556$_H$	A0A0$_H$	PA	PD	000EAAAA$_H$	5555$_H$	000E5556H	1010$_H$
Chip Erasing	6	000F5556$_H$	AAAA$_H$	000EAAAA$_H$	5555$_H$	000F5556$_H$	8080$_H$	000F5556$_H$	AAAA$_H$	000EAAAA$_H$	5555$_H$	SA	3030$_H$
Sector Erasing	6	000F5556$_H$	AAAA$_H$	000EAAAA$_H$	5555$_H$	000F5556$_H$	8080$_H$	000F5556$_H$	AAAA$_H$	-		-	
Sector Erasing being Suspended		XXXXXXXX$_H$	B0B0$_H$	-		-		-		-		-	
Sector Erasing being Restarted		XXXXXXXX$_H$	3030$_H$	-		-		-		-		-	

*: Commands must be issued by writing half-words under the FR-CPU programming mode.

RA: Read address /RD: Read data

PA: Write address/PD: Write data

SA: Sector address (specification of any address in the sector. Refer to Table 21.1-1 and Table 21.1-2)

Erase suspension command (B0$_H$) and erase restart command (30$_H$) are only valid while erasing the sector.

Two types of reset command can reset the flash memory to read mode.

● Read (Read)/Reset command

In order to return from timing limit excess to read mode, issue the read/reset command sequence. Reads data from the flash memory with the read cycle. The flash memory retains the read status until another command is input.

The flash memory is automatically read/reset when the power is turned on. In this case, the data read command in unnecessary.

● Program (Write)

Under CPU programming mode, writing is basically carried out per half-word. Writing requires four bus cycles. There are two "unlock" cycles for the command sequence, and the writing set up command and writing data cycle continue. Starts writing to the memory at the last writing cycle.

After executing the auto writing algorithm command sequence, the flash memory no longer requires external control. The flash memory generates the appropriate write pulse automatically created within it, and verifies the margin of the written cell. The auto writing operation ends when the bit 7 data matches the data written to this bit by the data polling function (refer to "21.5 Auto Algorithm Execute State - Hardware sequence flag"), takes this opportunity to return to read mode, and will not receive writing addresses any longer. As a result, the flash memory requests the next valid address at this stage. In this manner, data polling indicates that the memory is being written.

During writing, all commands written to the flash memory are ignored. If address data exists while hardware reset is initiated at writing, such data cannot be guaranteed.

Writing is enabled in any address order or even if the sector boundary is exceeded. Data "0" cannot be returned to data "1" by writing. If data "1" is written to data "0", the element is determined as defective by the data polling algorithm, or it appears to be written as data "1". However, when data is read under the reset/read mode, the data remains as "0". Data 0 can be changed to data 1 at the erase operation only.

● Chip Erasing

Chip erase (erasing all sectors in blocks) is carried out by accessing for six times. Firstly, two "unlock" cycles exist, and continuously, the "set up" command is written. Until the chip erase command, two more "unlock" cycles continue.

During a chip erase operation, the user does not need to write data to the flash memory prior to erasing. While executing the auto erase algorithm, the flash memory automatically writes "0" patterns and verifies before erasing all cells (pre-programmed). During this operation, the flash memory does not require external control.

Auto erase starts by writing within the command sequence, ends when bit 7 becomes "1", and the flash memory returns to read mode at this stage. The chip erasure time will be [time for sector erase x total number of sectors + time for writing chip (preprogrammed)].

● Sector Erasing

Sector erase requires six access cycles. Two "unlock" cycles exist, and the "set up" command is written continuously, then two more "unlock" cycles follow, and erasure of the sector starts by inputting the sector erase command at the sixth cycle. The next sector erase command can be received during the 50µs time-out from writing the last sector erase command.

Multiple sector erase can be simultaneously received by writing six bus cycles as mentioned before. This sequence is carried out by continuously writing the sector erase command (30_H) at the sector address to be erased simultaneously. Erasing the sector starts on exiting the 50 µs time-out period from writing the last sector erase command. In other words, when a number of sectors are erased simultaneously, the next sector to be erased must be input within 50 µs. If it is input after that, the command may not be received. Whether or not the sector erase command followed is valid can be monitored by bit 3. (Refer to "21.5 Auto Algorithm Execute State - Hardware sequence flag" for details). Resets the sector erase command during the time-out or any command other than erase suspension to read, and ignores the previous command sequence. In this case, erasure is completed by re-erasing the sector. The sector address can be input to the sector erase buffer from any combination and number (0 to 6).

For sector erasure, the user does not need to write to the flash memory before erasing. The flash memory writes to all cells within the sector to be erased automatically (preprogrammed). Other sectors not to be erased are unaffected while erasing the sector. During these operations, the flash memory does not require external control.

Auto sector erase starts after the 50µs time-out period from writing the last sector erase command. When the bit 7 data becomes "1" (refer to "21.5 Auto Algorithm Execute State - Hardware sequence flag" for details), it ends and the flash memory returns to read mode. Other commands are ignored. Data polling is activated at any address within the erased sector. Time for erasing multiple sectors will be [(time for erasing the sector + time for writing the sector (preprogrammed)) x number of sectors to be erased].

- Temporary deletion suspension

The erase suspension command suspends the auto algorithm of the flash memory while erasing the sector by user, and enables data from sectors not being erased to be read/written. This command is only valid while erasing sectors, and is ignored while erasing and writing chips. The erase suspension commands (BO_H) are only valid while erasing sectors that include sector erase time-out periods after sector erase commands (30_H). When this command is input during the time-out period, time-out ends immediately, and the erasure operation is suspended. When the erase restart command is written, erasure operation restarts. Any address is acceptable when inputting the command that suspends erase and the command that restarts erase.

If the erase suspension command is input while erasing the sector, it takes a maximum of 20 µs for the flash memory to stop the erasure operation. When the flash memory enters erase suspension mode, the ready/ busy output and bit 7 output are "1", and bit 6 stops toggling. Whether the erasure operation has stopped or not can be checked by inputting the address of the sector that is being erased and monitoring values read by bits 6 and 7. Writing of the erase suspension command is also ignored.

Once the erasure operation stops, the flash memory enters erase suspension/read mode. Under this mode, reading data is valid for sectors in which data erasure is not suspended, but it is same as standard reading for others. During reading erase suspension, bit 2 toggles to continuous reading from the sector whose erasure was suspended. (Refer to "21.5 Auto Algorithm Execute State - Hardware sequence flag" for details).

After entering erase suspension reading mode, the user can write to the flash memory by writing a write command sequence. This writing mode becomes the erase suspension writing mode. Writing under this mode is validated for sectors in which data erasure is not suspended, but for others, it is the same as writing normal bytes. Under erase suspension writing mode, bit 2 toggles for continuous reading from the sector whose erasure was suspended. The erase suspension bit can be used to detect this operation. Notes on use, bit 6 can read any address, but bit 7 must read the written address.

In order to restart the sector erase operation, the restart command (30_H) must be input. In this case, inputting of the restart command is ignored. On the other hand, the erase suspension command can be input after the flash memory restarts erasing.

21.5 Auto Algorithm Execute State

This flash memory has hardware that notifies the internal flash memory operation status and operation completion for outside of the flash memory to execute the write/ erase flow by auto algorithm. One is a hardware sequence flag, and the other is the ready/busy signal.

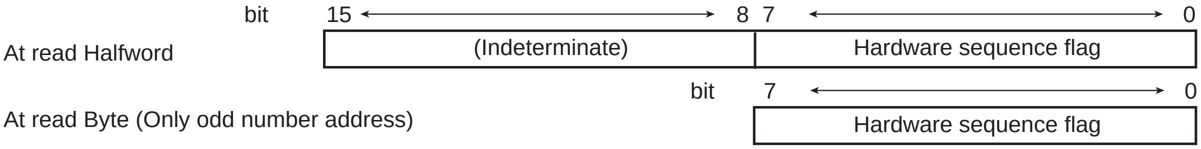
■ Ready/Busy Signal (RDY/BUSYX)

The flash memory has a ready/busy signal as well as a hardware sequence flag as a means to notify whether the internal auto algorithm is being executed or has finished. This ready/busy signal is connected to the flash memory interface circuit, and can be read as the "RDY" bit of the flash memory status register. Interrupt requests can be also generated to the CPU by the rising of the ready/busy signal. (Refer to "21.2 Flash Memory Status Register (FSTR)" for details).

When the read value of the "RDY" bit is "0", the flash memory is being written on or erased. In this case, neither write nor erase commands will be accepted. When the read value of the "RDY" bit is "1", the flash memory is on wait state to be read/written on or erased.

■ Hardware Sequence Flag

Figure 21.5-1 Hardware sequence flag



*: Word reading is disabled. (Only to be used under FR-CPU programming mode.)

The hardware sequence flag can be acquired as data by reading any address (odd addresses in the case of byte access) of the flash memory while executing auto algorithm. There are 5 valid bits of data, each of them expresses the auto algorithm status.

Figure 21.5-2 Hardware sequence flag (At Half word, Byte access)

bit	7	6	5	4	3	2	1	0
	DPOLL	TOGGLE	TLOVER	Indeterminate	SETIMR	TOGGL2	Indeterminate	Indeterminate

These flags are meaningless in FR-CPU ROM mode. Half-word or byte reading must be carried out under FR-CPU programming mode only.

Table 21.5-1 State list of hardware sequence flag

State			DPOLL	TOGGLE	TLOVER	SETIMR	TOGGL2
Executing	Automatic Writing operation		Reverse Data	Toggle	0	0	1
	Automatic Erasing operation		0	Toggle	0	1	Toggle
	Erase suspension mode	Erase suspension read (sectors in erase suspension)	1	1	0	0	Toggle [*1]
		Erase suspension read (sectors not in erase suspension)	Data	Data	Data	Data	Data
		Erase suspension write (sectors not in erase suspension)	Reverse Data	Toggle [*2]	0	0	1 [*3]
Time limit exceeded	Automatic Writing operation		Reverse Data	Toggle	1	0	1
	Automatic Erasing operation		0	Toggle	1	1	Undefined
	Write operation at erase suspension		0	Toggle	1	1	Undefined

*1: Bit 2 toggles when reading continuously from the erase suspension sector.

*2: Bit 6 toggles even when reading continuously from any address.

*3: During writing erasure suspension, bit 2 will be "1" when reading the address that is being written.

However, bit 2 toggles when reading continuously from sectors in which erasure is suspended.

The bits lists in the Table 21.5-1 have the following meaning:

[bit 7]:DPOLL=Data polling

[bit 6]:TOGGLE=Toggle bit

[bit 5]:TLOVER=Time limit exceeded

[bit 3]:SETIMR=Sector erasing timer

[bit 2]:TOGGL2=Toggle bit 2

Each bit is briefly described below:

[bit 7]:DPOLL (Data polling)

- Automatic Writing operation

When reading is carried out while executing the auto write algorithm, the flash memory outputs the reversed data of the data that was last written to bit 7. When read access is carried out once the auto write algorithm is completes, the flash memory outputs bit 7 of the read data at the address indicated by the address signal.

- Automatic Erasing operation

While executing the auto erase algorithm, if reading is carried out, the flash memory outputs "0" regardless of the address indicated by the address signal. In the same way, "1" is output when it ends.

- Sector erasing being suspended

When reading is carried out while sector deletion is suspended. The flash memory outputs "1" if the address indicated by the address signal belongs to the sector during deletion. If it does not belong to the sector during deletion, the read value "bit 7" of the address indicated by the address signal is output. Whether the sector status is erase suspended or not, and which sector is being erased, can be determined by referring to the toggle bit of bit 6 mentioned later.

Note:

When the auto algorithm operation approaches the end, bit 7 (data polling) will be changed asynchronously during reading. This indicates that the flash memory sends information about its operating status to bit 7, and sends the defined data to the next one. Even when the flash memory finishes the auto algorithm, and bit 7 outputs the defined data, other bits are not defined yet. Defined data of other bits are read continuously.

[bit 6]:TOGGLE (Toggle bit)

- Auto writing or erasing operation

While executing the auto writing or erase algorithm, if reading continuously, the flash memory outputs the results while setting "1" and "0" as toggles to bit 6. When the auto writing or erase algorithm ends, toggling of bit 6 is stopped for continuous reading, and valid data is output. Toggle bit is validated after the last writing cycle of each command sequence is executed.

If writing is attempted, but the sector to be written on is protected from being overwritten, after toggling for about 2 µs, toggling ends without rewriting the data. In terms of erasing, if all selected sectors are protected from being overwritten, the toggle bit toggles for about 100 µs, and then returns to read mode without rewriting any data.

- Sector erasing being suspended

When reading is carried out while sector deletion is suspended, the flash memory outputs "1" if the address indicated by the address signal belongs to the sector during deletion. Outputs bit 6 of the read value of the address indicated by the address signal, if it does not belong to the sector being erased.

[bit 5]:TLOVER (Time limit exceeded)

- Auto writing or erasing operation

Bit 5 indicates that execution of the auto algorithm is exceeded the time prescribed within the flash memory (number of internal pulse times). Bit 5 outputs "1" under this status. In other words, when this flag outputs "1" while the auto algorithm is operating, it indicates that writing or deletion has failed.

Attempts to write bit 5 to non-blank areas without prior erasure may cause failure. In this case, defined data cannot be read from bit7 (data polling), and bit6 (toggle bit) will continuously toggle. Under this status, if the time limit is exceeded, "1" will be output to bit 5. Please note that this indicates that the flash memory was not used correctly rather than any defect with the flash memory. If this event occurs, the reset command should be executed.

[bit 3]:SETIMR (Sector erasing timer)

- Sector erase operation

After the first sector deletion command sequence is executed, the waiting period for sector deletion is indicated. Bit 3 outputs "0" during this period, and "1" if the sector erase wait period is exceeded. The data polling and toggle bits are validated after executing the first sector deletion command sequence.

When "1" is set in this flag while the data polling or toggle bit function indicates that the erase algorithm is being executed, an internally controlled erase operation has started. Writing of continuous command is ignored until the data polling or toggle bit indicates the erase termination. (Input of deletion suspension code only is accepted.)

If this flag is "0", the flash memory accepts the additional sector deletion code to be written. For confirmation, using software to check this flag status prior to writing the sector erase code is recommended. If "1" is shown at the 2nd status check, the additional sector deletion code may not have been accepted.

When reading is carried out while sector deletion is suspended. The flash memory outputs "1" if the address indicated by the address signal belongs to the sector during deletion. If it does not belong to the sector during deletion, the read value "bit 3" of the address indicated by the address signal is output.

[bit 2]:TOGGL2 (toggle bit 2)

- Sector erase operation

This toggle bit is used to detect whether the flash memory is in auto deletion or suspension of deletion status in addition to the toggle bit of bit 6. Bit 2 operates the toggle when continuously reading from the sectors that have been deleted during auto deletion. If the flash memory is under deletion suspension reading mode, bit 2 operates the toggle by continuously reading from the sector in which deletion is suspended.

If the flash memory is under deletion suspension reading mode, "1" is read by bit 2 by continuously reading from the sector in which deletion is not suspended. In contrast to bit 2, bit 6 toggles only during normal writing, erasing, or erase suspension writing.

For example, bits 2 and 6 are used together to detect the erase suspension reading mode. (Bit 2 toggles, whereas bit 6 does not.)

Bit 2 is also used to detect the deleted sectors. When the flash memory is being deleted, bit 2 operates the toggle if reading from the deleted sector.

Appendix

Details that could not be described within the body text, such as I/O map, interrupt vector, peripheral circuit measurement speed, restrictions and commands list for use of the MB91191/MB91192 series are described in the appendix.

Appendix A I/O Map

Addresses shown on the Table A-1 are allocated to registers of the peripheral functions built into the MB91191/MB91192 series.

■ I/O Map

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
00000_H	Port 3 data register	R/W	PDR3	$XXXX XXXX_B$
00001_H	Port 2 data register	R/W	PDR2	$XXXX XXXX_B$
00002_H	Port 1 data register	R/W	PDR1	$XXXX XXXX_B$
00003_H	Port 0 data register	R/W	PDR0	$XXXX XXXX_B$
00004_H	Port 7 data register	R/W	PDR7	$---- ---X_B$
00005_H	Port 6 data register	R/W	PDR6	$XXXX XXXX_B$
00006_H	Port5 data register	R/W	PDR5	$XXXX XXXX_B$
00007_H	Port 4 data register	R/W	PDR4	$XXXX XXXX_B$
00008_H	Port 3 data direction register	W	DDR3	$0000 0000_B$
00009_H	Port 2 data direction register	W	DDR2	$0000 0000_B$
$0000A_H$	Port 1 data direction register	W	DDR1	$0000 0000_B$
$0000B_H$	Port 0 data direction register	W	DDR0	$0000 0000_B$
$0000C_H$	Port 7 data direction register	W	DDR7	$---- ---0_B$
$0000D_H$	Port 6 data direction register	W	DDR6	$0000 0000_B$
$0000E_H$	Port 5 data direction register	W	DDR5	$0000 0000_B$
$0000F_H$	Port 4 data direction register	W	DDR4	$0000 0000_B$
00010_H	Port B data register	R/W	PDRB	$XXXX XXXX_B$
00011_H	Port A data register	R/W	PDRA	$XXXX XXXX_B$
00012_H	Port 9 data register	R/W	PDR9	$---X XXXX_B$
00013_H	Port 8 data register	R/W	PDR8	$XXXX XXXX_B$
00014_H	Reserved			
00015_H				
00016_H	Port D data register	R/W	PDRD	$XXXX XXXX_B$
00017_H	Port C data register	R/W	PDRC	$XXXX XXXX_B$
00018_H	Port B data direction register	W	DDRB	$0000 0000_B$
00019_H	Port A data direction register	W	DDRA	$0000 0000_B$
$0001A_H$	Port 9 data direction register	W	DDR9	$---0 0000_B$
$0001B_H$	Port 8 data direction register	W	DDR8	$0000 0000_B$
$0001C_H$				
$0001D_H$				
$0001E_H$	Port D data direction register	W	DDRD	$0000 0000_B$
$0001F_H$	Port C data direction register	W	DDRC	$0000 0000_B$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
00020$_H$	Port B input enable register	W	PIEB	0000 0000$_B$
00021$_H$	Port A input enable register	W	PIEA	0000 0000$_B$
00022$_H$	Port 9 function selection register	W	PFS9	---0 0000$_B$
00023$_H$	Port 8 function selection register	W	PFS8	0000 0000$_B$
00024$_H$	Reserved			
00025$_H$				
00026$_H$	Port D function selection register	W	PFSD	0000 0000$_B$
00027$_H$	Port C function selection register	W	PFSC	0000 0000$_B$
00028$_H$ to 0002F$_H$	Reserved			
00030$_H$	Reserved			
00031$_H$	10 bit General-purpose prescaler control register	R/W	GPRC	--XX 0XX0$_B$
00032$_H$	10 bit General-purpose prescaler data register	W	GPRD	---- --XX$_B$
00033$_H$				XXXX XXXX$_B$
00034$_H$	RTG0 Control Register	R/W	RTG0C	X100 0001$_B$
00035$_H$	RTG0 Output data register	R/W	RTG0D	---X XXXX$_B$
00036$_H$	RTG0 Timing data register	W	RTG0T	XXXX XXXX$_B$
00037$_H$				XXXX XXXX$_B$
00038$_H$	RTG1 Control Register	R/W	RTG1C	X100 0001$_B$
00039$_H$	RTG1 Output data register	R/W	RTG1D	---X XXXX$_B$
0003A$_H$	RTG1 Timing data register	W	RTG1T	XXXX XXXX$_B$
0003B$_H$				XXXX XXXX$_B$
0003C$_H$	RTG2 Control Register	R/W	RTG2C	X100 0001$_B$
0003D$_H$	RTG2 Output data register	R/W	RTG2D	---X XXXX$_B$
0003E$_H$	RTG2 Timing data register	W	RTG2T	XXXX XXXX$_B$
0003F$_H$				XXXX XXXX$_B$
00040$_H$	Reserved			
00041$_H$	PWM0 Control Register	R/W	PWM0C	0--- 00X0$_B$
00042$_H$	PWM00 Data Register	W	PWMD00	---- XXXX$_B$
00043$_H$				XXXX XXXX$_B$
00044$_H$	PWM01 Data Register	W	PWMD01	---- XXXX$_B$
00045$_H$				XXXX XXXX$_B$
00046$_H$	PWM02 Data Register	W	PWMD02	---- XXXX$_B$
00047$_H$				XXXX XXXX$_B$
00048$_H$	Reserved			
00049$_H$	PWM1 Control Register	R/W	PWM1C	0--- 00X0$_B$
0004A$_H$	PWM10 Data Register	W	PWMD10	---- XXXX$_B$
0004B$_H$				XXXX XXXX$_B$
0004C$_H$	PWM11 Data Register	W	PWMD11	---- XXXX$_B$
0004D$_H$				XXXX XXXX$_B$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
0004E$_H$	PWM12 Data Register	W	PWMD12	---- XXXX$_B$
0004F$_H$				XXXX XXXX$_B$
00050$_H$	Capstan input control register	R/W	CAPDVC	XXXX XXXX$_B$
00051$_H$	Capstan mask timer control register	R/W	CAPMTC	XXXX XXXX$_B$
00052$_H$	Capstan control register	R/W	CAPC	X1X0 ---0$_B$
00053$_H$	Dram control register	R/W	DRMC	X1X0 0X10$_B$
00054$_H$	Dram input control register	R/W	DRMDVC	---- XXXX$_B$
00055$_H$	Dram mask timer control register	R/W	DRMMTC	XXXX XXXX$_B$
00056$_H$	Reserved			
00057$_H$	Reel control register	R/W	RLC	X1X0 X1X0$_B$
00058$_H$	Reel 0 input control register	R/W	RL0DVC	XXXX XXXX$_B$
00059$_H$	Reel 0 mask timer control register	R/W	RL0MTC	XXXX XXXX$_B$
0005A$_H$	Reel 1 input control register	R/W	RL1DVC	XXXX XXXX$_B$
0005B$_H$	Reel 1 mask timer control register	R/W	RL1MTC	XXXX XXXX$_B$
0005C$_H$	Reserved			
0005D$_H$				
0005E$_H$				
0005F$_H$				
00060$_H$	Reserved			
00061$_H$	FRC count data register	R	FRCD2	XXXX XXXX$_B$
00062$_H$		R	FRCD1	XXXX XXXX$_B$
00063$_H$		R	FRCD0	XXXX X000$_B$
00064$_H$	Capture input control register	R/W	CIC1	X10- -000$_B$
00065$_H$		R/W	CIC0	0000 0000$_B$
00066$_H$	Reserved			
00067$_H$	Capture control register	R/W	CAPC	---- 0100$_B$
00068$_H$	Capture source register	R	CAPS	XXXX XXXX$_B$
00069$_H$	Capture data register	R	CAPD2	XXXX XXXX$_B$
0006A$_H$			CAPD1	XXXX XXXX$_B$
0006B$_H$			CAPD0	XXXX XXXX$_B$
0006C$_H$	PPG0 Timing data register	W	PPG0T	XXXX XXXX$_B$
0006D$_H$				XXXX XXXX$_B$
0006E$_H$	PPG1 Timing data register	W	PPG1T	XXXX XXXX$_B$
0006F$_H$				XXXX XXXX$_B$
00070$_H$	16 bit Timer 0 count data register	R	T0CD	XXXX XXXX$_B$
00071$_H$				XXXX XXXX$_B$
00072$_H$	16 bit Timer 0 data register	R/W	T0DR	XXXX XXXX$_B$
00073$_H$				XXXX XXXX$_B$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
00074$_H$	16 bit Timer 0 Control Register	R/W	T0CR	0--- -1X0$_B$
00075$_H$				---0 0000$_B$
00076$_H$	16 bit Timer 1 Control Register	R/W	T1CR	0--- -1X0$_B$
00077$_H$				---0 0000$_B$
00078$_H$	16 bit Timer 1 count data register	R	T1CD	XXXX XXXX$_B$
00079$_H$				XXXX XXXX$_B$
0007A$_H$	16 bit Timer 1 data register	R/W	T1DR	XXXX XXXX$_B$
0007B$_H$				XXXX XXXX$_B$
0007C$_H$	16 bit Timer 2 count data register	R	T2CD	XXXX XXXX$_B$
0007D$_H$				XXXX XXXX$_B$
0007E$_H$	16 bit Timer 2 data register	R/W	T2DR	XXXX XXXX$_B$
0007F$_H$				XXXX XXXX$_B$
00080$_H$	16 bit Timer 2 Control Register	R/W	T2CR	0--- -1X0$_B$
00081$_H$				---0 0000$_B$
00082$_H$	16 bit Timer 3 Control Register	R/W	T3CR	0--- -1X0$_B$
00083$_H$				---0 0000$_B$
00084$_H$	16 bit Timer 3 count data register	R	T3CD	XXXX XXXX$_B$
00085$_H$				XXXX XXXX$_B$
00086$_H$	16 bit Timer 3 data register	R/W	T3DR	XXXX XXXX$_B$
00087$_H$				XXXX XXXX$_B$
00088$_H$	16 bit Timer 4 count data register	R	T4CD	XXXX XXXX$_B$
00089$_H$				XXXX XXXX$_B$
0008A$_H$	16 bit Timer 4 data register	R/W	T4DR	XXXX XXXX$_B$
0008B$_H$				XXXX XXXX$_B$
0008C$_H$	16 bit Timer 4 Control Register	R/W	T4CR	0--- -1X0$_B$
0008D$_H$				---0 0000$_B$
0008E$_H$	8/16 bit Timer / Counter control register	R/W	T5CR1	010- 0000$_B$
0008F$_H$			T5CR0	0100 0000$_B$
00090$_H$	8/16 bit Timer / Counter data register	R/W	T5DR1	XXXX XXXX$_B$
00091$_H$			T5DR0	XXXX XXXX$_B$
00092$_H$	8/16 bit Timer / Counter count data register	R	T5CD1	XXXX XXXX$_B$
00093$_H$			T5CD0	XXXX XXXX$_B$
00094$_H$	8 bit PWC control register	R/W	PWCC	0--- 0X00$_B$
00095$_H$	8 bit PWC Data Register	R	PWCD	XXXX XXXX$_B$
00096$_H$	Key input Control Register	R/W	KEYC	0000 0000$_B$
00097$_H$	Key input Status Register	R	KEYS	XXXX XXXX$_B$
00098$_H$	External interrupt enable register	R/W	EIE	0000 111-$_B$
00099$_H$	External Interrupt request flag	R	EIF	---- XXXX$_B$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
0009A$_{H}$	Reserved			
0009B$_{H}$				
0009C$_{H}$				
0009D$_{H}$				
0009E$_{H}$				
0009F$_{H}$				
000A0$_{H}$	A/DC Control Register	R	ADCH	XXX0 0000$_{B}$
000A1$_{H}$		R	ADCL	000- -XXX$_{B}$
000A2$_{H}$	A/DC Soft conversion analog input selection register	R/W	SCIS	0000 0000$_{B}$
000A3$_{H}$		R/W		0000 0000$_{B}$
000A4$_{H}$	Reserved			
000A5$_{H}$	Soft conversion Status Register	R/W	SCSR	X10- -001$_{B}$
000A6$_{H}$	A/DC Soft conversion FIFO data register	R	SCFD	XXXX --XX$_{B}$
000A7$_{H}$				XXXX XXXX$_{B}$
000A8$_{H}$	Reserved			
000A9$_{H}$	A/DC Hard conversion Status Register	R/W	HCSR	X10- -001$_{B}$
000AA$_{H}$	A/DC Hard conversion FIFO data register	R	HCFD	XXXX --XX$_{B}$
000AB$_{H}$				XXXX XXXX$_{B}$
000AC$_{H}$	Reserved			
000AD$_{H}$				
000AE$_{H}$				
000AF$_{H}$				
000B0$_{H}$ to 001FF$_{H}$				
00200$_{H}$ to 002FF$_{H}$	PPG0 Data RAM	R/W	-	XXXX XXXX$_{B}$
00300$_{H}$ to 0037F$_{H}$	Serial 0 Data buffer RAM	R/W	-	XXXX XXXX$_{B}$
00380$_{H}$ to 003BF$_{H}$	PPG1 Data RAM	R/W	-	XXXX XXXX$_{B}$
003C0$_{H}$	PPG0 Control Register	R/W	PPG0C	0000 X100$_{B}$
003C1$_{H}$	PPG0 start address setting register	R/W	PPG0SA	XXXX XXXX$_{B}$
003C2$_{H}$	Reserved			
003C3$_{H}$				
003C4$_{H}$	PPG1 Control Register	R/W	PPG1C	0000 X100$_{B}$
003C5$_{H}$	PPG1 start address setting register	R/W	PPG1SA	XXXX XXXX$_{B}$
003C6$_{H}$	Reserved			
003C7$_{H}$				
003C8$_{H}$	Serial control register	R/W	S0CR	0100 1000$_{B}$
003C9$_{H}$	Clock Mode setting register	R/W	S0MR	--0X XXXX$_{B}$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
003CA$_H$	Address offset register	W	S0AO	-000 0000$_B$
003CB$_H$	Transfer byte count setting register	R/W	S0BR	0XXX XXXX$_B$
003CC$_H$	Serial control register	S1CR	R/W	0100 1000$_B$
003CD$_H$	Clock Mode setting register	S1MR	R/W	--0X XXXX$_B$
003CE$_H$	Address offset register	W	S1AO	-000 0000$_B$
003CF$_H$	Transfer byte count setting register	S1BR	R/W	0XXX XXXX$_B$
003D0$_H$	Serial control register	S2CR	R/W	0100 1000$_B$
003D1$_H$	Clock Mode setting register	S2MR	R/W	--0X XXXX$_B$
003D2$_H$	Address offset register	W	S2AO	-000 0000$_B$
003D3$_H$	Transfer byte count setting register	R/W	S2BR	0XXX XXXX$_B$
003D4$_H$ to 003EF$_H$	Reserved			
003F0$_H$	0 detection data register	W	BSD0	XXX XXXX$_B$
003F1$_H$				XXX XXXX$_B$
003F2$_H$				XXX XXXX$_B$
003F3$_H$				XXX XXXX$_B$
003F4$_H$	1 detection data register	R/W	BSD1	XXX XXXX$_B$
003F5$_H$				XXX XXXX$_B$
003F6$_H$				XXX XXXX$_B$
003F7$_H$				XXX XXXX$_B$
003F8$_H$	Change point detection data register	W	BSDC	XXX XXXX$_B$
003F9$_H$				XXX XXXX$_B$
003FA$_H$				XXX XXXX$_B$
003FB$_H$				XXX XXXX$_B$
003FC$_H$	Detection result register	R	BSRR	XXX XXXX$_B$
003FD$_H$				XXX XXXX$_B$
003FE$_H$				XXX XXXX$_B$
003FF$_H$				XXX XXXX$_B$
00400$_H$	Interrupt control register 0	R/W	ICR00	---1 1111$_B$
00401$_H$	Interrupt control register 1	R/W	ICR01	---1 1111$_B$
00402$_H$	Interrupt control register 2	R/W	ICR02	---1 1111$_B$
00403$_H$	Interrupt control register 3	R/W	ICR03	---1 1111$_B$
00404$_H$ to 0042F$_H$	Interrupt control register 4 to Interrupt control register 47	R/W to R/W	ICR04 to ICR47	---1 1111$_B$ to ---1 1111$_B$
00430$_H$	Delayed interrupt Control Register	R/W	DICR	---- ---0$_B$
00431$_H$	Reserved			
00432$_H$	Reserved			
00433$_H$				
00434$_H$ to 0047F$_H$	Reserved			

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
00480$_H$	Reset factor register / Watchdog cycle control register	R/W	RSRR/WTCR	1XXX XXXX$_B$
00481$_H$	Standby control register	R/W	STCR	0001 11--$_B$
00482$_H$	Reserved			
00483$_H$	Timebase timer clear register	W	CTBR	XXXX XXXX$_B$
00484$_H$	Gear Control Register	R/W	GCR	11-- 11-1$_B$
00485$_H$	Watchdog reset generation delayed register	W	WPR	XXXX XXXX$_B$
00486$_H$	Reserved			
00487$_H$				
00488$_H$ to 005FF$_H$	Reserved			
00600$_H$	Reserved			
00601$_H$				
00602$_H$	Port 1 function selection register	W	PFS1	0000 0000$_B$
00603$_H$	Reserved			
00604$_H$	Port 7 function selection register	W	PFS7	---- --00$_B$
00605$_H$	Port 6 function selection register	W	PFS6	0000 1111$_B$
00606$_H$	Port 5 function selection register	W	PFS5	1111 1111$_B$
00607$_H$	Port 4 function selection register	W	PFS4	0000 0000$_B$
00608$_H$	Reserved			
00609$_H$				
0060A$_H$				
0060B$_H$				
0060C$_H$	Area selection Register 1	W	ASR1	0000 0000$_B$
0060D$_H$				0000 0001$_B$
0060E$_H$	Area mask Register 1	W	AMR1	0000 0000$_B$
0060F$_H$				0000 0000$_B$
00610$_H$	Area selection Register 2	W	ASR2	0000 0000$_B$
00611$_H$				0000 0002$_B$
00612$_H$	Area mask Register 2	W	AMR2	0000 0000$_B$
00613$_H$				0000 0000$_B$
00614$_H$	Area selection Register 3	W	ASR3	0000 0000$_B$
00615$_H$				0000 0003$_B$
00616$_H$	Area mask Register 3	W	AMR3	0000 0000$_B$
00617$_H$				0000 0000$_B$
00618$_H$	Area selection Register 4	W	ASR4	0000 0000$_B$
00619$_H$				0000 0004$_B$
0061A$_H$	Area mask Register 4	W	AMR4	0000 0000$_B$
0061B$_H$				0000 0000$_B$

Table A-1 I/O Map

Address	Register Name	Access	Register	Initial value
0061C$_H$	Area selection Register 5	W	ASR5	0000 0000$_B$
0061D$_H$				0000 0005$_B$
0061E$_H$	Area mask Register 5	W	AMR5	0000 0000$_B$
0061F$_H$				0000 0000$_B$
00620$_H$	Reserved			
00621$_H$	Area mode Register	R/W	AMD1	0--0 0000$_B$
00622$_H$	Reserved			
00623$_H$	Reserved			
00624$_H$ to 0062F$_H$	Reserved			
00630$_H$ to 0077F$_H$	Reserved			
00780$_H$ to 007AF$_H$	Reserved			
007B0$_H$ to 007BF$_H$	Reserved			
007C0$_H$	Flash memory control register (MB91F191A/ MB91F192)	R/W	FSTR	000- ---0$_B$
007C1$_H$	Reserved			
007C2$_H$				
007C3$_H$				
007C4$_H$	Wait Control Register	R/W	WAITC	---- -000$_B$
007C5$_H$	Reserved			
007C6$_H$				
007C7$_H$				
007C8$_H$ to 007FB$_H$	Reserved			
007FC$_H$	Reserved			
007FD$_H$				
007FE$_H$	Little endian register	W	LER	---- -000$_B$
007FF$_H$	Mode register	W	MODR	XXXX XXXX$_B$

Note:

Do not execute RMW commands to registers that have a write-only bit.

RMW system instruction (RMW: Read-modify-write)

AND	Rj,@Ri	OR	Rj,@Ri	EOR	Rj,@Ri
ANDH	Rj,@Ri	ORH	Rj,@Ri	EORH	Rj,@Ri
ANDB	Rj,@Ri	ORB	Rj,@Ri	EORB	Rj,@Ri
BANDL	#u4,@Ri	BORL	#u4,@Ri	BEORL	#u4,@Ri

Appendix B Interrupt vector

Table B-1 shows interrupt vector table.
The interrupt factor of the MB91191/MB91192 series and allocation of the interrupt vector/interrupt control register are described in the interrupt vector table.

■ Interrupt Vector

Table B-1 Interrupt vector table (Continued)

Interrupt cause		Interrupt Vector number		Interrupt level	Offset	TBR default address
		10 decimal	16 decimal			
Reset		0	00$_H$	-	03FC$_H$	000FFFFC$_H$
System reservation		1	01$_H$	-	03F8$_H$	000FFFF8$_H$
System reservation		2	02$_H$	-	03F4$_H$	000FFFF4$_H$
System reservation		3	03$_H$	-	03F0$_H$	000FFFF0$_H$
System reservation		4	04$_H$	-	03EC$_H$	000FFFEC$_H$
System reservation		5	05$_H$	-	03E8$_H$	000FFFE8$_H$
System reservation		6	06$_H$	-	03E4$_H$	000FFFE4$_H$
System reservation		7	07$_H$	-	03E0$_H$	000FFFE0$_H$
System reservation		8	08$_H$	-	03DC$_H$	000FFFDC$_H$
System reservation		9	09$_H$	04$_H$	03D8$_H$	000FFFD8$_H$
System reservation		10	0A$_H$	-	03D4$_H$	000FFFD4$_H$
System reservation		11	0B$_H$	-	03D0$_H$	000FFFD0$_H$
System reservation		12	0C$_H$	04$_H$	03CC$_H$	000FFFCC$_H$
System reservation		13	0D$_H$	-	03C8$_H$	000FFFC8$_H$
Undefined instruction exception		14	0E$_H$	-	03C4$_H$	000FFFC4$_H$
NMI	System reservation	15	0F$_H$	0F$_H$	03C0$_H$	000FFFC0$_H$
IRQ0	System reservation	16	10$_H$	ICR00	03BC$_H$	000FFFBC$_H$
IRQ1	System reservation	17	11$_H$	ICR01	03B8$_H$	000FFFB8$_H$
IRQ2	External interrupt 0 (INT0)	18	12$_H$	ICR02	03B4$_H$	000FFFB4$_H$
IRQ3	External interrupt 1 (INT1)	19	13$_H$	ICR03	03B0$_H$	000FFFB0$_H$
IRQ4	External interrupt 2 (INT2)	20	14$_H$	ICR04	03AC$_H$	000FFFAC$_H$
IRQ5	24 bit FRC capture	21	15$_H$	ICR05	03A8$_H$	000FFFA8$_H$
IRQ6	Programmable Pulse Generator ch0(PPG0)	22	16$_H$	ICR06	03A4$_H$	000FFFA4$_H$
IRQ7	Programmable Pulse Generator ch1(PPG1)	23	17$_H$	ICR07	03A0$_H$	000FFFA0$_H$
IRQ8	Real timing generator ch0(RTG0)	24	18$_H$	ICR08	039C$_H$	000FFF9C$_H$
IRQ9	Real timing generator ch1(RTG1)	25	19$_H$	ICR09	0398$_H$	000FFF98$_H$
IRQ10	Real timing generator ch2(RTG2)	26	1A$_H$	ICR10	0394$_H$	000FFF94$_H$
IRQ11	10 bit A/D converter (HARD)	27	1B$_H$	ICR11	0390$_H$	000FFF90$_H$
IRQ12	8/16 bit Timer / Counter (Timer 5-0)	28	1C$_H$	ICR12	038C$_H$	000FFF8C$_H$
IRQ13	8/16 bit Timer / Counter (Timer5-1)	29	1D$_H$	ICR13	0388$_H$	000FFF88$_H$
IRQ14	16 bit Timer / Counter (Timer4)	30	1E$_H$	ICR14	0384$_H$	000FFF84$_H$

Table B-1 Interrupt vector table (Continued)

Interrupt cause		Interrupt Vector number		Interrupt level	Offset	TBR default address
		10 decimal	16 decimal			
IRQ15	16 bit Timer (Timer0)	31	$1F_H$	ICR15	0380_H	$000FFF80_H$
IRQ16	16 bit Timer (Timer 1)	32	20_H	ICR16	$037C_H$	$000FFF7C_H$
IRQ17	16 bit Timer (Timer2)	33	21_H	ICR17	0378_H	$000FFF78_H$
IRQ18	16 bit Timer (Timer3)	34	22_H	ICR18	0374_H	$000FFF74_H$
IRQ19	Serial ch0(SIO0)	35	23_H	ICR19	0370_H	$000FFF70_H$
IRQ20	Serial ch1(SIO1)	36	24_H	ICR20	$036C_H$	$000FFF6C_H$
IRQ21	Serial ch2(SIO2)	37	25_H	ICR21	0368_H	$000FFF68_H$
IRQ22	10 bit A/D converter (SOFT)	38	26_H	ICR22	0364_H	$000FFF64_H$
IRQ23	Key input Interrupt	39	27_H	ICR23	0360_H	$000FFF60_H$
IRQ24	System reservation	40	28_H	ICR24	$035C_H$	$000FFF5C_H$
IRQ25 to IRQ45	System reservation	41 to 61	29_H to $3D_H$	ICR25 to ICR45	0358_H to 0308_H	$000FFF58_H$ to $000FFF08_H$
IRQ46	FLASH	62	$3E_H$	ICR46	0304_H	$000FFF04_H$
IRQ47	Delayed Interrupt Factor bit	63	$3F_H$	ICR47	0300_H	$000FFF00_H$
System reservation (using for REALOS*)		64	40_H	-	$02FC_H$	$000FFEFC_H$
System reservation (using for REALOS*)		65	41_H	-	$02F8_H$	$000FFEF8_H$
System reservation		66 to 255	42_H to FF_H	-	$02F4_H$ to 0000_H	$000FFEF4_H$ to $000FFD00_H$

*: Use 0x40, 0x41 for the system code when using REALOS/FR.

ICR: This is the register set within the interrupt controller, and it sets the interrupt levels for each interrupt request. ICR is prepared corresponding to each of the interruption demand.

TBR: It is a register which shows the first address of the vector table for EIT.

The address to which the offset value determined per TBR and EIT factor is added will be the vector address.

Note:

From address which TBR shows to vector region for EIT region of 1KB

The size per vector is 4 bytes, and the relationship between the vector number and vector address is expressed as below.

vctadr = TBR + vctofs
= TBR +(3FCH - 4 × vct)

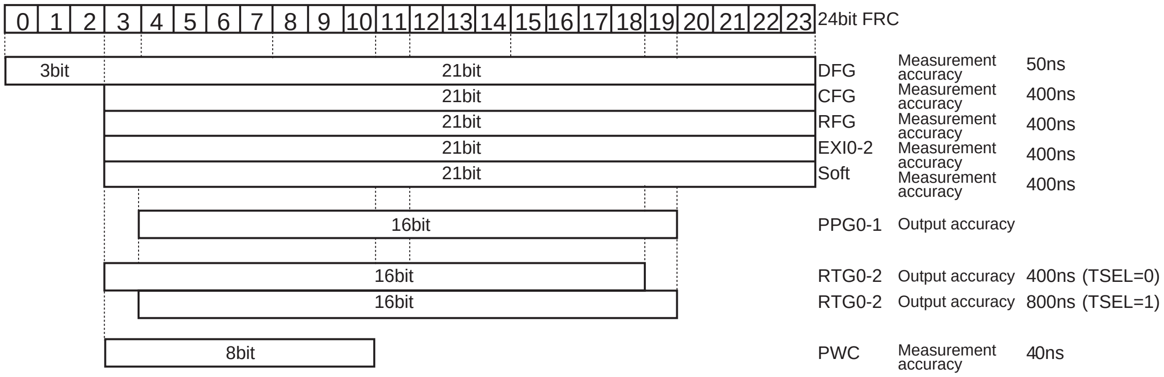
vctadr: Vector Address
vctofs: Vector offset
vct: Vector number

Appendix C Measurement accuracy of peripheral circuit

Figure C-1 shows the measurement accuracy of peripheral circuit relative to FRC and the output timing accuracy.

■ Measurement Accuracy of Peripheral Circuit Relative to FRC and Output Timing Accuracy

Figure C-1 Measurement accuracy of peripheral circuit relative to FRC and output timing accuracy



Appendix D Restrictions for Using MB91191/MB91192 series

This section shows the restrictions for using MB91191/MB91192 series.

■ Restrictions for Using MB91191/MB91192 Series

● Restrictions of using gear function

There is a limit to the gear combination of the CPU system and peripheral system for the MB91191/ MB91192 series in order to ensure correct operation of the peripheral function.

Table D-1 At CHC=0

φ θ		Peripheral			
		1/1	1/2	1/4	1/8
CPU system	1/1	●	×	×	×
	1/2	○	▲	×	×
	1/4	○	▲	▲	×
	1/8	○	▲	▲	▲

●: Selectable. But need to set 1wait.

○: Selectable.

▲: Selectable. However, there is the restrictions of peripheral.

×: Not selectable

Table D-2 At CHC=1

φ θ		Peripheral			
		1/1	1/2	1/4	1/8
CPU system	1/1	●	×	×	×
	1/2	▲	▲	×	×
	1/4	▲	▲	▲	×
	1/8	▲	▲	▲	▲

Appendix E Instruction List

Instruction list of FR series is shown. Beforehand, the following matters are explained for better understanding of the command list.

- **How to Read the Instruction List**
- **Symbol of addressing mode**
- **Instruction format**

■ How to Read the Instruction List

Instruction operand is XXX scr, dest (src--> dest).

Mnemonic			Type	OP	CYCLE	NZVC	Operation		Remark
ADD	Rj,	Rj	A	AG	1	CCCC	Ri+Rj	→Rj	
*ADD	#s5,	Rj	C	A4	1	CCCC	Ri+s5	→Ri	
	,		,	,	,	,	,		
	,		,	,	,	,	,		
↓	↓		↓	↓	↓	↓	↓	↓	
1)	2)		3)	4)	5)	6)	6)	7)	

1. The instruction name is shown.

   *: The indication is the extend command whose command was extended or added by the assembler regardless of the CPU specifications.

2. A specifiable Addressing mode is shown in the operand by the sign.
   - Refer to the "Addressing mode symbols" for the symbol meanings.
3. The instruction format is shown.
4. The hexadecimal number is displayed to the instruction code.
5. The number of machine cycles is shown.
   - a: It is a memory access cycle, and there is a possibility to postpone by the Ready function.
   - b: It is a memory access cycle, and there is a possibility to postpone by the Ready function. If the register that is targeted for LD operation is referred to by the command that immediately follows it, an interlock is activated, and the execution cycle number will be increased by 1.
   - c: When the command immediately following is one that reads or writes R15, SSP, or USP, an interlock is activated, and the execution cycle number will be increased by 1, thus becoming 2.
   - d: When the command immediately following refers to the MDH/MDL, an interlock is activated, and the execution cycle number will be increased by 1, thus becoming 2.
   - a, b, c, d, and the minimum are one cycle.

6. The flag change is shown.

Flag change
C: change -: No change 0: Clear 1: Set

Meaning of flag
N: Negative flag Z: Zero flag V: Over flag C: Carrying flag

7. The instruction operation is written.

■ Symbol of Addressing Mode

Table E-1 Description of symbol of addressing mode

Symbol	Explanation
Ri	Register direct (R0 to R15, AC, FP, SP)
Rj	Register direct (R0 to R15, AC, FP, SP)
R13	Register direct (R13, AC)
Ps	Register direct (direct program status register)
Rs	Register direct (TBR, RP, SSP, USP, MDH, MDL)
CRi	Register direct (CR0 to CR15)
CRj	Register direct (CR0 to CR15)
#i8	Unsigned 8 bit value immediately (-128 to 255) Note: -128 to -1 are used as 128 to 255.
#i20	Unsigned 20 bit value immediately (-0X80000 to 0XFFFFF) Note: -0X7FFFF to -1 are used as 0X7FFFF to 0XFFFFF.
#i32	Unsigned 32 bit value immediately (-0X80000000 to 0XFFFFFFFF) Note: -0X80000000 to -1 are used as 0X80000000 to 0XFFFFFFFF.
#s5	Signed 5 bit value immediately (-16 to 15)
#s10	Signed 10 bit value immediately (Only the multiple of -512 to 5084)
#u4	Unsigned 20 bit value immediately (0 to 15)
#u5	Unsigned 5 bit value immediately (0 to 31)
#u8	Unsigned 8 bit value immediately (0 to 255)
#u10	Unsigned 10 bit value immediately (Only the multiple of 0 - 10204:)
@dir8	Unsigned 8 bit direct address (0 to 0XFF)
@dir9	Unsigned 9 bit direct address (Only the multiple of 0 to 0X1FE2)
@dir10	Unsigned 10 bit direct address (Only the multiple of 0 to 0X3FC4).
label9	Divergence address of signed 9 bits (Only the multiple of -0X100 to 0XFC2)
label12	Divergence address of signed 12 bits (Only the multiple of -0X800 to 0X7FC2)
label20	Divergence address of signed 20 bits (-0X80000 to 0X7FFFF)
label32	Divergence address of signed 32 bits (-0X80000000 to 0X7FFFFFFF)
@Ri	Register indirect (R0 to R15, AC, FP, SP)
@Rj	Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj)	Relativity is register indirect (Rj:R0 to R15, AC, FP, SP)
@(R14,disp10)	Relative indirectly register (Only the multiple of disp10:-0X200 to 0X1FC4)
@(R14,disp9)	Relative indirectly register (Only the multiple of disp9:-0X100 to 0XFE2)
@(R14,disp8)	Relativity is register indirect (disp8:-0X80 to 0X7F)
@(R15,udisp6)	Relative indirectly register (Only the multiple of udisp6:0 to 604)
@Ri+	Register indirectly with post increment (R0 to R15, AC, FP, SP)
@R13+	Register indirectly with post increment (R13, AC)
(reglist)	Register list

■ Instruction Format

Table E-2 Instruction format

<table>
<tr><th>Type</th><th>Instruction format</th></tr>
<tr><td>A</td><td>MSB LSB
16bit<table><tr><td>OP</td><td>Rj</td><td>Ri</td></tr><tr><td>8</td><td>4</td><td>4</td></tr></table></td></tr>
<tr><td>B</td><td><table><tr><td>OP</td><td>i8/o8</td><td>Ri</td></tr><tr><td>4</td><td>8</td><td>4</td></tr></table></td></tr>
<tr><td>C</td><td><table><tr><td>OP</td><td>u4/m4</td><td>Ri</td></tr><tr><td>8</td><td>4</td><td>4</td></tr></table></td></tr>
<tr><td>C’</td><td>Only ADD, ADDN, CMP, LSL, LSR, ASR instructions<table><tr><td>OP</td><td>s5/u5</td><td>Ri</td></tr><tr><td>7</td><td>5</td><td>4</td></tr></table></td></tr>
<tr><td>D</td><td><table><tr><td>OP</td><td>u8/rel8/dir/
reglist</td></tr><tr><td>8</td><td>8</td></tr></table></td></tr>
<tr><td>E</td><td><table><tr><td>OP</td><td>SUB-OP</td><td>Ri</td></tr><tr><td>8</td><td>4</td><td>4</td></tr></table></td></tr>
<tr><td>F</td><td><table><tr><td>OP</td><td>rel11</td></tr><tr><td>5</td><td>11</td></tr></table></td></tr>
</table>

E.1 Instruction list of FR series

Instruction list of FR series is described in order to following instruction.

■ Instruction List of FR Series

■ Addition and Subtraction Instruction

Table E-3 Addition and subtraction instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
ADD Rj, Ri	A	A6	1	CCCC	Ri+Rj →Ri	
*ADD #s5, Ri	C'	A4	1	CCCC	Ri+s5 →Ri	Upper in assembler
						1 bit is regarded as a sign.
ADD #u4, Ri	C	A4	1	CCCC	Ri+extu(i4) →Ri	0 expansions
ADD2 #u4, Ri	C	A5	1	CCCC	Ri+extu(i4) →Ri	Minus expansion
ADDC Rj, Ri	A	A7	1	CCCC	Ri+Rj +c→Ri	Addition with carry
ADDN Rj, Ri	A	A2	1	----	Ri+Rj →Ri	
*ADDN #s5, Ri	C'	A0	1	----	Ri+s5 →Ri	Upper in assembler
						1 bit is regarded as a sign.
ADDN #u4, Ri	C	A0	1	----	Ri+extu(i4) →Ri	0 expansions
ADDN2 #u4, Ri	C	A1	1	----	Ri+extu(i4) →Ri	Minus expansion
SUB Rj, Ri	A	AC	1	CCCC	RiRj →Ri	
SUBC Rj, Ri	A	AD	1	CCCC	RiRj c→Ri	Reduction with carry
SUBN Rj, Ri	A	AE	1	----	Ri-Rj →Ri	

■ Comparison Operation Instruction

Table E-4 Comparison operation instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
CMP Rj, Ri	A	AA	1	CCCC	Ri-Rj	
*CMP #s5, Ri	C'	A8	1	CCCC	Ri-s5	Upper in assembler
						1 bit is regarded as a sign.
CMP #u4, Ri	C	A8	1	CCCC	Ri-extu(i4)	0 expansions
CMP2 #u4, Ri	C	A9	1	CCCC	Ri-extu(i4)	Minus expansion

■ Logical Operation Instruction

Table E-5 Logical operation instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
AND Rj, Ri	A	82	1	CC--	Ri &=Rj	Word
AND Rj, @Ri	A	84	1+2a	CC--	(Ri) &=Rj	Word
ANDH Rj, @Ri	A	85	1+2a	CC--	(Ri) &=Rj	Half word
ANDB Rj, @Ri	A	86	1+2a	CC--	(Ri) &=Rj	Byte
OR Rj, Ri	A	92	1	CC--	Ri | =Rj	Word
OR Rj, @Ri	A	94	1+2a	CC--	(Ri) | =Rj	Word
ORH Rj, @Ri	A	95	1+2a	CC--	(Ri) | =Rj	Half word
ORB Rj, @Ri	A	96	1+2a	CC--	(Ri) | =Rj	Byte
EOR Rj, Ri	A	9A	1	CC--	Ri ^ =Rj	Word
EOR Rj, @Ri	A	9C	1+2a	CC--	(Ri) ^ =Rj	Word
EORH Rj, @Ri	A	9D	1+2a	CC--	(Ri) ^ =Rj	Half word
EORB Rj, @Ri	A	9E	1+2a	CC--	(Ri) ^ =Rj	Byte

■ Bit Manipulation Instructions

Table E-6 Bit manipulation instructions

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
BANDL #u4, @Ri BANDH #u4, @Ri *BAND #u8, @Ri *1	C C	80 81	1+2a 1+2a	---- ---- ----	(Ri) &=(0xF0+u4) (Ri) &=((u4 < <4)+0x0F) (Ri) &=u8	The subordinate position four bits are operated. The high rank four bits are operated.
BORL #u4, @Ri BORH #u4, @Ri *BOR #u8, @Ri *2	C C	90 91	1+2a 1+2a	---- ---- ----	(Ri) | =u4 (Ri) | =(u4 < <4) (Ri) | =u8	The subordinate position four bits are operated. The high rank four bits are operated.
BEORL #u4, @Ri BEORH #u4, @Ri *BEOR #u8, @Ri *3	C C	98 99	1+2a 1+2a	---- ---- ----	(Ri) ^ =u4 (Ri) ^ =(u4 < <4) (Ri) ^ =u8	The subordinate position four bits are operated. The high rank four bits are operated.
BTSTL #u4, @Ri BTSTH #u4, @Ri	C C	88 89	2+a 2+a	0C-- CC--	(Ri) &u4 (Ri) &(u4 < <4)	Lower 4 bit test Upper 4 bit test

*1: The assembler generates BANDL when the u8&0x0F bit is raised, or generates BANDH when the u8&0xF0 bit is raised. There is a case generating both BANDL and BANDH.

*2: The assembler generates BORL when the u8&0x0F bit is raised, or generates BORH when the u8&0xF0 bit is raised. There is a case generating both BORL and BODH.

*3: The assembler generates BEORL when the u8&0x0F bit is raised, or generates BEORH when the u8&0xF0 bit is raised. There is a case generating both BEORL and BEORH.

■ Multiplication and Division Instructions

Table E-7 Multiplication and division instructions

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
MUL Rj,Ri	A	AF	5	CCC-	Ri*Rj→MDH,MDL	32bit*32bit=64bit
MULU Rj,Ri	A	AB	5	CCC-	Ri*Rj→MDH,MDL	None
MULH Rj,Ri	A	BF	3	CC--	Ri*Rj→MDL	16bit*16bit=32bit
MULUH Rj,Ri	A	BB	3	CC--	Ri*Rj→MDL	None
DIV0S Ri	E	97-4	1	----		Step operation
DIV0U Ri	E	97-5	1	----		32bit/32bit=32bit
DIV1 Ri	E	97-6	d	-C-C		
DIV2 Ri	E	97-7	1	-C-C		
DIV3	E	9F-6	1	----		
DIV4S	E	9F-7	1	----		
*DIV Ri *1			36	-C-C	MDL/Ri→MDL, MDL%Ri→MDH	
*DIVU Ri *2			33	-C-C	MDL/Ri→MDL, MDL%Ri→MDH	

*1:Generates DIV0S, DIV1 × 32, DIV2, DIV3, DIV4S. Instruction code length is 72 bytes.

*2:Generates DIV0U, DIV1 × 32. Instruction code length is 66 bytes.

■ Shift Instruction

Table E-8 Shift instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
LSL Rj, Ri	A	B6	1	CC-C	Ri < <Rj→Ri	Logical shift
*LSL #u5, Ri(u5:0 to 31)	C'	B4	1	CC-C	Ri < <u5→Ri	
LSL #u4, Ri	C	B4	1	CC-C	Ri < <u4→Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	Ri < <(u4+16)→Ri	
LSR Rj, Ri	A	B2	1	CC-C	Ri > >Rj→Ri	Logical shift
*LSR #u5, Ri(u5:0 to 31)	C'	B0	1	CC-C	Ri > >u5→Ri	
LSR #u4, Ri	C	B0	1	CC-C	Ri > >u4→Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	Ri > >(u4+16)→Ri	
ASR Rj, Ri	A	BA	1	CC-C	Ri > >Rj→Ri	Arithmetic shift
*ASR #u5, Ri(u5:0 to 31)	C'	B8	1	CC-C	Ri > >u5→Ri	
ASR #u4, Ri	C	B8	1	CC-C	Ri > >u4→Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	Ri > >(u4+16)→Ri	

■ Value Move Operation of Value Sets/16 Bits/32 Bits Immediately

Table E-9 Value move operation of value sets/16 bits/32 bits immediately

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
LDI:32 #i32,Ri	E	9F-8	3	----	i32→Ri	
LDI:20 #i20,Ri	C	9B	2	----	i20→Ri	0 expansions of high rank 12bit.
LDI:8 #i8,Ri	B	C0	1	----	i8→Ri	0 expansions of high rank 24 bit.
*LDI# {i8 | i20 | i32}, Ri *					{i8 | i20 | i32} →Ri	

*: When the immediate value is an absolute value, i8, i20, or i32 is automatically selected by the assembler.
When the immediate value is a relative value, or includes an external reference symbol, i32 is automatically selected.

■ Memory Loading Instruction

Table E-10 Memory loading instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
LD @Rj,Ri	A	04	b	----	(Rj) → Ri	
LD @(R13,Rj),Ri	A	00	b	----	(R13+Rj) → Ri	
LD @(R14,disp10),Ri	B	20	b	----	(R14+disp10) → Ri	
LD @(R15,udisp6),Ri	C	03	b	----	(R15+udisp6) → Ri	
LD @R15+,Ri	E	07-0	b	----	(R15) → Ri,R15+=4	
LD @R15+,Rs	E	07-8	b	----	(R15) → Rs,R15+=4	Rs: special register*
LD @R15+,PS	E	07-9	1+a+b	CCCC	(R15) → PS,R15+=4	
LDUH @Rj,Ri	A	05	b	----	(Rj) →Ri	0 expansions
LDUH @(R13,Rj),Ri	A	01	b	----	(R13+Rj) →Ri	0 expansions
LDUH @(R14,disp9),Ri	B	40	b	----	(R14+disp9) →Ri	0 expansions
LDUB @Rj,Ri	A	06	b	----	(Rj) → Ri	0 expansions
LDUB @(R13,Rj),Ri	A	02	b	----	(R13+Rj) → Ri	0 expansions
LDUB @(R14,disp8),Ri	B	60	b	----	(R14+disp8) → Ri	0 expansions

*:Special register Rs: TBR, RP, USP, SSP, MDH, MDL

Note:

The assembler calculates and sets values as follows in the 08 and 04 fields for hardware specifications.
disp10/4→o8, disp9/2→o8, disp8→o8, disp10, disp9, disp8 are with sign.
udisp6/4→o4udisp6 is without sign.

■ Memory Store Instruction

Table E-11 Memory store instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
ST Ri,@Rj	A	14	a	----	Ri→(Rj)	Word
ST Ri,@(R13,Rj)	A	10	a	----	Ri→(R13+Rj)	Word
ST Ri,@(R14,disp10)	B	30	a	----	Ri→(R14+disp10)	Word
ST Ri,@(R15,udisp6)	C	13	a	----	Ri→(R15+udisp6)	
ST Ri,@-R15	E	17-0	a	----	R15-=4,Ri→(R15)	
ST Rs,@-R15	E	17-8	a	----	R15-=4,Rs→(R15)	Rs: special register *
ST PS,@-R15	E	17-9	a	----	R15-=4,PS→(R15)	
STH Ri,@Rj	A	15	a	----	Ri→(Rj)	Half word
STH Ri,@(R13,Rj)	A	11	a	----	Ri→(R13+Rj)	Half word
STH Ri,@(R14,disp9)	B	50	a	----	Ri→(R14+disp9)	Half word
STB Ri,@Rj	A	16	a	----	Ri→(Rj)	Byte
STB Ri,@(R13,Rj)	A	12	a	----	Ri→(R13+Rj)	Byte
STB Ri,@(R14,disp8)	B	70	a	----	Ri→(R14+disp8)	Byte

*:Special register Rs: TBR, RP, USP, SSP, MDH, MDL

Note:

The assembler calculates and sets values as follows in the 08 and 04 fields for hardware specifications.
disp10/4→o8, disp9/2→o8, disp8→o8, disp10, disp9, disp8 are with sign.
udisp6/4→o4udisp6 is without sign.

■ Transfer Instruction between Registers

Table E-12 Transfer instruction between registers

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
MOV Rj,Ri	A	8B	1	----	Rj→Ri	Transfer between general purpose register
MOV Rs,Ri	A	B7	1	----	Rs→Ri	Rs: special register *
MOV Ri,Rs	A	B3	1	----	Ri→Rs	Rs: special register *
MOV PS,Ri	E	17-1	1	----	PS→Ri	
MOV Ri,PS	E	07-1	c	CCCC	Ri→PS	

*: Special register Rs: TBR, RP, USP, SSP, MDH, MDL

■ Normal Divergence (There is no delay) Instruction

Table E-13 Normal divergence (There is no delay) instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
JMP @Ri	E	97-0	2	----	Ri→PC	
CALL label12	F	D0	2	----	PC+2→RP,	
					PC+2+(label12-PC-2)→PC	
CALL @Ri	E	97-1	2	----	PC+2→RP,Ri→PC	
RET	E	97-2	2	----	RP→PC	Return
INT #u8	D	1F	3+3a	----	SSP-=4,PS→(SSP),	
					SSP-=4,PC+2→(SSP),	
					0: I flag, 0: S flag	
					(TBR+0x3FC-u8 × 4) →PC	
INTE	E	9F-3	3+3a	----	SSP-=4,PS→(SSP),	for emulator
					SSP-=4,PC+2→(SSP),	
					0→S Flag,	
					(TBR+0x3D8) →PC	
RETI	E	97-3	2+2a	CCCC	(R15) →PC,R15-=4,	
					(R15) →PS,R15-=4	
BRA label9	D	E0	2	----	PC+2+(label9-PC-2) →PC	
BNO label9	D	E1	1	----	not diverge	
BEQ label9	D	E2	2/1	----	if(Z==1)then	
					PC+2+(label9-PC-2) →PC	
BNE label9	D	E3	2/1	----	↑ s/Z==0	
BC label9	D	E4	2/1	----	↑ s/C==1	
BNC label9	D	E5	2/1	----	↑ s/C==0	
BN label9	D	E6	2/1	----	↑ s/N==1	
BP label9	D	E7	2/1	----	↑ s/N==0	
BV label9	D	E8	2/1	----	↑ s/V==1	
BNV label9	D	E9	2/1	----	↑ s/V==0	
BLT label9	D	EA	2/1	----	↑ s/VxorN==1	
BGE label9	D	EB	2/1	----	↑ s/VxorN==0	
BLE label9	D	EC	2/1	----	↑ s/(VxorN)orZ==1	
BGT label9	D	ED	2/1	----	↑ s/(VxorN)orZ==0	
BLS label9	D	EE	2/1	----	↑ s/CorZ==1	
BHI label9	D	EF	2/1	----	↑ s/CorZ==0	

Note:

- "2/1" of CYCLE number is following;
  2: branching
  1: not branching
- The assembler calculates and sets values as follows in the rel11 and rel8 fields for hardware specifications. (label12-PC-2)/2→rel11, (label9-PC-2)/2→rel8, label12, label9 are with sign.
- When executing RETI instruction, S flag is required to be "0".

■ Delay Divergence Instruction

Table E-14 Delay divergence Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
JMP:D @Ri	E	9F-0	1	----	Ri→PC	
CALL:D label12	F	D8	1	----	PC+4→RP, PC+2+(label12-PC-2)→PC	
CALL:D @Ri	E	9F-1	1	----	PC+4→RP,Ri→PC	
RET:D	E	9F-2	1	----	RP→PC	Return
BRA:D label9	D	F0	1	----	PC+2+(label9-PC-2) →PC	
BNO:D label9	D	F1	1	----	not diverge	
BEQ:D label9	D	F2	1	----	if(Z==1)then PC+2+(label9-PC-2) →PC	
BNE:D label9	D	F3	1	----	↑ s/Z==0	
BC:D label9	D	F4	1	----	↑ s/C==1	
BNC:D label9	D	F5	1	----	↑ s/C==0	
BN:D label9	D	F6	1	----	↑ s/N==1	
BP:D label9	D	F7	1	----	↑ s/N==0	
BV:D label9	D	F8	1	----	↑ s/V==1	
BNV:D label9	D	F9	1	----	↑ s/V==0	
BLT:D label9	D	FA	1	----	↑ s/VxorN==1	
BGE:D label9	D	FB	1	----	↑ s/VxorN==0	
BLE:D label9	D	FC	1	----	↑ s/(VxorN)orZ==1	
BGT:D label9	D	FD	1	----	↑ s/(VxorN)orZ==0	
BLS:D label9	D	FE	1	----	↑ s/CorZ==1	
BHI:D label9	D	FF	1	----	↑ s/CorZ==0	

Note:

- The assembler calculates and sets values as follows in the rel11 and rel8 fields for hardware specifications. (label12-PC-2)/2->rel11, (label9-PC-2)/2→rel8, label12, label9 are with sign.
- In terms of delayed branches, branching must be implemented after executing the next command (delay slot).
- Commands that can be placed at the delay slot are all 1 cycle, a, b, c, and d cycle commands.
  Two or more-cycle instruction cannot be put.

■ The Other Instruction

Table E-15 The other Instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
NOP	E	9F-A	1	----	Anything does not change either.	
ANDCCR #u8 ORCCR #u8	D D	83 93	c c	CCCC CCCC	CCR and u8 →CCR CCR or u8 →CCR	
STILM #u8	D	87	1	----	i8→ILM	Value set of ILM immediately
ADDSP #s10 *1	D	A3	1	----	R15+=s10	ADDSP Instruction
EXTSB Ri EXTUB Ri EXTSH Ri EXTUH Ri	E E E E	97-8 97-9 97-A 97-B	1 1 1 1	---- ---- ---- ----	sign extension 8→32bit 0 expansions 8→32bit sign extension16→32bit 0 expansions16→32bit	
LDM0 (reglist) LDM1 (reglist) *LDM (reglist) *2	D D	8C 8D		---- ---- ----	(R15) →reglist, R15 increment (R15) →reglist, R15 increment (R15) →reglist, R15 increment	loading multi R0 - R7 loading multi R8 - R15 loading multi R0 - R15
STM0 (reglist) STM1 (reglist) *STM (reglist) *3	D D	8E 8F		---- ---- ----	R15 decrement, reglist→(R15) R15 decrement, reglist→(R15) R15 decrement, reglist→(R15)	store multi R0 - R7 store multi R8 - R15 store multi - R0 - R15
ENTER #u10 *4	D	0F	1+a	----	R14→(R15-4), R15-4→R14, R15-u10→R15	Entrance processing of function
LEAVE	E	9F-9	b	----	R14+4→R15, (R15-4)→R14	Exit processing of function
XCHB @Rj,Ri	A	8A	2a	----	Ri→TEMP (Rj)→Ri TEMP→(Rj)	For semaphore control Byte Data

*1: s10 is set by that assembler calculates s10/4 to s8. s10 has the sign.

*2: Under the reglist, if one of R0-R7 is specified, LDM0 is generated, whereas if one of R8-R15 is specified, LDM1 is generated. In other case, both LDM0 and LDM1 may be able to be generated.

*3: Under the reglist, if one of R0-R7 is specified, STM0 is generated, whereas if one of R8-R15 is specified, STM1 is generated. There is a case generating both STM1 and STM0.

*4: u10 is set by that assembler calculates u10/4 to u8. u10 does not have the sign.

■ 20-bit Normal Divergence Macro Instruction

Table E-16 20-bit Normal divergence macro instruction

Mnemonic	Operation	Remark
*CALL20 label20,Ri	Address of the following instruction-->RP, label20→PC	Ri: Temporary register (Refer to reference 1.)
*BRA20 label20,Ri	label20→PC	Ri: Temporary register (Refer to reference 2.)
*BEQ20 label20,Ri	if(Z==1)thenlabel20→PC	Ri: Temporary register (Refer to reference 3.)
*BNE20 label20,Ri	↑ s/Z==0	↑
*BC20 label20,Ri	↑ s/C==1	↑
*BNC20 label20,Ri	↑ s/C==0	↑
*BN20 label20,Ri	↑ s/N==1	↑
*BP20 label20,Ri	↑ s/N==0	↑
*BV20 label20,Ri	↑ s/V==1	↑
*BNV20 label20,Ri	↑ s/V==0	↑
*BLT20 label20,Ri	↑ s/VxorN==1	↑
*BGE20 label20,Ri	↑ s/VxorN==0	↑
*BLE20 label20,Ri	↑ s/(VxorN)orZ==1	↑
*BGT20 label20,Ri	↑ s/(VxorN)orZ==0	↑
*BLS20 label20,Ri	↑ s/CorZ==1	↑
*BHI20 label20,Ri	↑ s/CorZ==0	↑

Reference 1: CALL20

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
CALL    label12
```

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:20  #label20,Ri
CALL    @Ri
```

Reference 2: BRA20

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
BRA    label9
```

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:20  #label20,Ri
JMP     @Ri
```

Reference 3: Bcc20

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
Bcc    label9
```

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

Bxcc false xcc is a contradiction condition of cc.

```
LDI:20  #label20,Ri
JMP     @Ri
false:
```

■ 20-bit Delayed Divergence Macro Instruction

Table E-17 20-bit Delayed divergence macro instruction

Mnemonic	Operation	Remark
*CALL20:D label20,Ri	Address of the following instruction +2→RP, label20→PC	Ri: Temporary register (Refer to reference 1.)
*BRA20:D label20,Ri	label20→PC	Ri: Temporary register (Refer to reference 2.)
*BEQ20:D label20,Ri	if(Z==1)thenlabel20→PC	Ri: Temporary register (Refer to reference 3.)
*BNE20:D label20,Ri	↑ s/Z==0	↑
*BC20:D label20,Ri	↑ s/C==1	↑
*BNC20:D label20,Ri	↑ s/C==0	↑
*BN20:D label20,Ri	↑ s/N==1	↑
*BP20:D label20,Ri	↑ s/N==0	↑
*BV20:D label20,Ri	↑ s/V==1	↑
*BNV20:D label20,Ri	↑ s/V==0	↑
*BLT20:D label20,Ri	↑ s/VxorN==1	↑
*BGE20:D label20,Ri	↑ s/VxorN==0	↑
*BLE20:D label20,Ri	↑ s/(VxorN)orZ==1	↑
*BGT20:D label20,Ri	↑ s/(VxorN)orZ==0	↑
*BLS20:D label20,Ri	↑ s/CorZ==1	↑
*BHI20:D label20,Ri	↑ s/CorZ==0	↑

Reference1: CALL20:D

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.
CALL:D label12

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.
LDI:20 #label20,Ri
CALL:D @Ri

Reference 2: BRA20:D

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows. x
BRA:D label9

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.
LDI:20 #label20,Ri
JMP:D @Ri

Reference 3: Bcc20:D

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.
Bcc:D label9

2) When label20-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.
Bxcc false xcc is a contradiction condition of cc.
LDI:20 #label20,Ri
JMP:D @Ri
false:

■ 32-bit Normal Divergence Macro Instruction

Table E-18 32-bit Normal divergence macro instruction

Mnemonic	Operation	Remark
*CALL32 label32,Ri	Address of the following instruction→RP, label32→PC	Ri: Temporary register (Refer to reference 1.)
*BRA32 label32,Ri	label32→PC	Ri: Temporary register (Refer to reference 2.)
*BEQ32 label32,Ri	if(Z==1)thenlabel32→PC	Ri: Temporary register (Refer to reference 3.)
*BNE32 label32,Ri	↑ s/Z==0	↑
*BC32 label32,Ri	↑ s/C==1	↑
*BNC32 label32,Ri	↑ s/C==0	↑
*BN32 label32,Ri	↑ s/N==1	↑
*BP32 label32,Ri	↑ s/N==0	↑
*BV32 label32,Ri	↑ s/V==1	↑
*BNV32 label32,Ri	↑ s/V==0	↑
*BLT32 label32,Ri	↑ s/VxorN==1	↑
*BGE32 label32,Ri	↑ s/VxorN==0	↑
*BLE32 label32,Ri	↑ s/(VxorN)orZ==1	↑
*BGT32 label32,Ri	↑ s/(VxorN)orZ==0	↑
*BLS32 label32,Ri	↑ s/CorZ==1	↑
*BHI32 label32,Ri	↑ s/CorZ==0	↑

Reference 1: CALL32

1) When label20-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
CALL    label12
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:32  #label32,Ri
CALL    @Ri
```

Reference 2: BRA32

1) When label32-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
BRA    label9
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:32  #label32,Ri
JMP     @Ri
```

Reference 3: Bcc32

1) When label32-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows. x

```
Bcc    label9
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
Bxcc false xcc is a contradiction condition of cc.
LDI:32  #label32,Ri
JMP     @Ri
false:
```

■ 32-bit Delayed Divergence Macro Instruction

Table E-19 32-bit Delayed divergence macro instruction

Mnemonic	Operation	Remark
*CALL32:D label32,Ri	Address of the following instruction +2→RP, label32→PC	Ri: Temporary register (Refer to reference 1.)
*BRA32:D label32,Ri	label32→PC	Ri: Temporary register (Refer to reference 2.)
*BEQ32:D label32,Ri	if(Z==1)thenlabel32→PC	Ri: Temporary register (Refer to reference 3.)
*BNE32:D label32,Ri	↑ s/Z==0	↑
*BC32:D label32,Ri	↑ s/C==1	↑
*BNC32:D label32,Ri	↑ s/C==0	↑
*BN32:D label32,Ri	↑ s/N==1	↑
*BP32:D label32,Ri	↑ s/N==0	↑
*BV32:D label32,Ri	↑ s/V==1	↑
*BNV32:D label32,Ri	↑ s/V==0	↑
*BLT32:D label32,Ri	↑ s/VxorN==1	↑
*BGE32:D label32,Ri	↑ s/VxorN==0	↑
*BLE32:D label32,Ri	↑ s/(VxorN)orZ==1	↑
*BGT32:D label32,Ri	↑ s/(VxorN)orZ==0	↑
*BLS32:D label32,Ri	↑ s/CorZ==1	↑
*BHI32:D label32,Ri	↑ s/CorZ==0	↑

Reference 1: CALL32:D

1) When label32-PC-2 is -0x800 - +0x7fe, the instruction is generated as follows.

```
CALL:D  label12
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:32  #label32,Ri
CALL:D  @Ri
```

Reference 2: BRA32:D

1) When label32-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
BRA:D   label9
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
LDI:32  #label32,Ri
JMP:D   @Ri
```

Reference 3: Bcc32:D

1) When label32-PC-2 is - 0x100 - + 0xfe, the instruction is generated as follows.

```
Bcc:D   label9
```

2) When label32-PC-2 is outside the area as per 1) and includes an external reference symbol, the command is generated as follows.

```
 Bxcc false xcc is a contradiction condition of cc.
LDI:32  #label32,Ri
JMP:D   @Ri
false:
```

■ Direct Addressing Instruction

Table E-20 Direct addressing instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
DMOV @dir10, R13	D	08	b	----	(dir10) →R13	Word
DMOV R13, @dir10	D	18	a	----	R13→(dir10)	Word
DMOV @dir10, @R13+	D	0C	2a	----	(dir10) →(R13),R13+=4	Word
DMOV @R13+, @dir10 *	D	1C	2a	----	(R13) →(dir10),R13+=4	Word
DMOV @dir10, @-R15	D	0B	2a	----	R15-=4, (R15) →(dir10)	Word
DMOV @R15+, @dir10	D	1B	2a	----	(R15) →(dir10),R15+=4	Word
DMOVH @dir9, R13	D	09	b	----	(dir9) →R13	Half word
DMOVH R13, @dir9	D	19	a	----	R13→(dir9)	Half word
DMOVH @dir9, @R13+	D	0D	2a	----	(dir9) →(R13),R13+=2	Half word
DMOVH @R13+, @dir9 *	D	1D	2a	----	(R13) →(dir9),R13+=2	Half word
DMOVB @dir8, R13	D	0A	b	----	(dir8) →R13	Byte
DMOVB R13, @dir8	D	1A	a	----	R13→(dir8)	Byte
DMOVB @dir8, @R13+	D	0E	2a	----	(dir8) →(R13),R13++	Byte
DMOVB @R13+, @dir8 *	D	1E	2a	----	(R13) →(dir8),R13++	Byte

Note:

The assembler calculates and sets values as follows in the dir8, dir9, and dir10 fields.

Dir8→dir, dir9/2→dir, dir10/4→dirdir8, dir9, dir10 are without sign.

■ Resource Instruction

Table E-21 resource instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
LDRES @Ri+, #u4	C	BC	a	----	(Ri) → Resource of u4 Ri+=4	u4: Channel number
STRES #u4, @Ri+	C	BD	a	----	(Ri) → Resource of u4 Ri+=4	u4: Channel number

■ Coprocessor Control Instruction

Table E-22 Coprocessor control instruction

Mnemonic	Type	OP	CYCLE	NZVC	Operation	Remark
COPOP #u4, #u8, CRj, CRi	E	9F-C	2+a	----	Operation instruction	
COPLD #u4, #u8, Rj, CRi	E	9F-D	1+2a	----	Rj→CRi	
COPST #u4, #u8, CRj, Ri	E	9F-E	1+2a	----	CRj→Ri	
COPSV #u4, #u8, CRj, Ri	E	9F-F	1+2a	----	CRj→Ri	Error trap none

Note:

- {CRi | CRj}:=CR0 | CR1 | CR2 | CR3 | CR4 | CR5 | CR6 | CR7 | CR8 | CR9 | CR10 | CR11 | CR12 | CR13 | CR14 | CR15

u4:= channel specification
u8:= command specification

INDEX

CM71-10113-1E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

FR20

32 BIT MICRO CONTROLLER

MB91191/MB91192 SERIES

HARDWARE MANUAL

March 2004 the first edition

Published **FUJITSU LIMITED** Electronic Devices
Edited Business Promotion Dept.